

И. В. МАКСИМЕЙ

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ НА ЭВМ



«РАДИО И СВЯЗЬ»



И. В. МАКСИМЕЙ

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ НА ЭВМ



МОСКВА «РАДИО И СВЯЗЬ»
1988

Максимей И. В. Имитационное моделирование на ЭВМ. — М.: Радио и связь, 1988. — 232 с.: ил. — ISBN 5-256-00001-2.

Для основных способов формализации объектов (имитация активностями, событийное моделирование, транзактный, агрегатный и процессный способы имитации) рассматриваются принципы организации квазипараллелизма и анализируются 19 систем автоматизации моделирования.

Обсуждается методология построения и испытания имитационных моделей сложных систем для выбранных способов имитации. Рассматриваются типовые этапы создания и использования моделей: составление содержательного описания объекта, разработка его концептуальной модели, формализация и преобразование в описание модели, программирование и отладка программы модели, испытание и исследование свойств модели, планирование имитационных экспериментов и эксплуатация модели. Методики иллюстрируются для каждого способа имитации.

Предлагаются способы и средства повышения технологичности постановки экспериментов, в том числе обсуждаются пути повышения уровня технологии моделирования в двух моделирующих комплексах: МК PLSIM и МК АСИМ.

Приведен пример построения имитационной модели вычислительного процесса в ОС ЕС для режима пакетной обработки заданий на основе реальной статистики по структуре решаемых задач.

Для научных работников, специализирующихся в области проектирования сложных систем; может быть полезна инженерам.

Табл. 16. Ил. 39. Библиогр. 83 назв.

Рецензенты: доктор физ.-мат. наук профессор В. В. Калашников, доктор техн. наук Г. И. Пранявичюс

Редакция литературы по вычислительной технике

М $\frac{1502000000-089}{046(01)-88}$ 1-88

ISBN 5-256-00001-2

© Издательство «Радио и связь», 1988

ПРЕДИСЛОВИЕ

Создание народнохозяйственных комплексов привело к широкому использованию математического моделирования, в частности имитационного моделирования. При проведении имитационного эксперимента (ИЭ) моделируются поведение частей указанных комплексов и их взаимодействие с учетом влияющих факторов. На ЭВМ вычисляются любые характеристики поведения объекта, предусмотренные программой проектирования. Широкое применение имитационного моделирования в ходе проектирования и эксплуатации сложных систем делает актуальной задачу создания методик построения и испытания имитационных моделей (ИМ) сложных систем (СС).

В книге основное внимание уделяется методологическим вопросам построения и испытания ИМ СС, а также вопросам повышения уровня технологии ИЭ. В отличие от большинства книг по моделированию, содержащих многочисленные примеры простейших ИМ, автор ограничился иллюстрацией рассматриваемых методологических вопросов на одной достаточно сложной модели вычислительной системы, руководствуясь тем, что при такой форме изложения легче показать способы построения ИМ СС и отразить возникающие при этом проблемы.

Предполагается, что читатель знаком с математическими основами построения ИМ (теорией вероятностей, математической статистикой, планированием экспериментов, методами статистических испытаний, программированием на ЭВМ), поэтому обсуждаются лишь специфические вопросы планирования ИЭ, представления и анализа результатов моделирования. Таким образом, в книге предпринята попытка помочь специалистам в области имитационного моделирования приобрести необходимый минимум навыков в организации ИЭ на ЭВМ. Практически по всем рассматриваемым вопросам даются ссылки на первоисточники, из которых можно почерпнуть дополнительные сведения.

Автор дает собственную трактовку понятия модельного времени, на ее основе выделяет «молекулы», из которых обычно составляется ИМ, и рассматривает наиболее употребительные способы формализации реальных объектов и системные вопросы организации имитации для каждого из пяти способов формализации: имитации активностями, событийного моделирования, имитации транзактами, агрегатами и процессами. С помощью схем

взаимодействия управляющей программы моделирования с моделью для каждого из способов формализации обсуждаются вопросы организации квазипараллелизма в системе дискретных событий. В ряде случаев начинающие исследователи совершают ошибки технологического характера либо из-за незнания, либо из-за игнорирования некоторых этапов создания и использования ИМ. В отличие от общепринятого подхода, автор выделяет восемь технологических этапов создания, испытания и использования ИМ, в соответствии с которыми излагается весь материал книги.

Для каждого из способов формализации СС поэтапно обсуждаются основные аспекты создания ИМ. Для иллюстрации и сравнения между собой особенностей каждого из способов формализации СС используется пример моделирования работы одного и того же производственного участка.

Для анализа технологических возможностей средств автоматизации моделирования выделены наиболее существенные технологические признаки систем автоматизации моделирования. По этим признакам в табличной форме проведен анализ наиболее употребительных отечественных и зарубежных систем моделирования (СМ).

При рассмотрении путей повышения уровня технологии постановки ИЭ на ЭВМ предметом обсуждения является то, каким способом, за счет каких решений достигается повышение уровня технологии создания и испытания ИМ СС в нескольких системах моделирования. При этом описываются основные технологические возможности разработанных под руководством автора двух моделирующих комплексов: МК АСИМ, организующего имитацию транзактным способом, и МК PLSIM, реализующего процессный способ имитации.

Все замечания по содержанию книги будут приняты автором с благодарностью.

ВВЕДЕНИЕ

Реальные сложные системы можно исследовать с помощью двух типов математических моделей: аналитических и имитационных. В аналитических моделях поведение СС записывается в виде некоторых функциональных соотношений или логических условий. Наиболее полное исследование удастся провести в том случае, когда получены явные зависимости, связывающие искомые величины с параметрами СС и начальными условиями ее изучения. Однако это удается выполнить только для сравнительно простых систем. Для СС исследователю зачастую приходится идти на упрощения представления реальных явлений, дающие возможность описать их поведение и представить взаимодействия между компонентами СС. Это позволяет изучить хотя бы некоторые общие свойства СС, например оценить устойчивость системы и сходимость реального переходного процесса к некоторому значению. Для построения аналитических моделей имеется мощный математический аппарат (алгебра, функциональный анализ, разностные уравнения, теория вероятностей, математическая статистика, теория массового обслуживания и т. д.). Наличие математического аппарата и относительная быстрота и легкость получения информации о поведении СС способствовали повсеместному и успешному распространению аналитических моделей в различных областях науки и техники.

Когда явления в СС настолько сложны и многообразны, что аналитическая модель становится слишком грубым приближением к действительности, то исследователь вынужден использовать имитационное моделирование. В имитационной модели поведение компонент СС описывается набором алгоритмов, которые затем реализуют ситуации, возникающие в реальной системе. Моделирующие алгоритмы позволяют по исходным данным, содержащим сведения о начальном состоянии СС, и фактическим значениям параметров системы отобразить реальные явления в системе и получить сведения о возможном поведении СС для данной конкретной ситуации. На основании этой информации исследователь может принять соответствующие решения. Отметим при этом, что предсказательные возможности имитационного моделирования значительно меньше, чем у аналитических моделей.

На основании опыта имитационного моделирования и анализа литературных источников [43, 67, 75, 81] можно рекомендовать исследователю использовать имитационную модель сложной системы при решении своих задач в следующих случаях:

1. Если не существует законченной постановки задачи исследования и идет процесс познания объекта моделирования. Имитационная модель служит средством изучения явления.

2. Если аналитические методы имеются, но математические процедуры столь сложны и трудоемки, что имитационное моделирование дает более простой способ решения задачи.

3. Когда кроме оценки влияния параметров СС желательно осуществить наблюдение за поведением компонент СС в течение определенного периода.

4. Когда имитационное моделирование оказывается единственным способом исследования сложной системы из-за невозможности наблюдения явлений в реальных условиях.

5. Когда необходимо контролировать протекание процессов в СС путем замедления или ускорения явлений в ходе имитации.

6. При подготовке специалистов и освоении новой техники, когда на имитационной модели обеспечивается возможность приобретения необходимых навыков в эксплуатации новой техники.

7. Когда изучаются новые ситуации в СС, о которых мало что известно или неизвестно ничего. В этом случае имитация служит для предварительной проверки новых стратегий и правил принятия решений перед проведением экспериментов на реальной системе.

8. Когда особое значение имеет последовательность событий в проектируемой СС и модель используется для предсказания узких мест в функционировании системы и других трудностей, появляющихся в поведении СС при введении в нее новых компонент.

Однако ИМ наряду с характерными для них достоинствами имеют ряд существенных недостатков. Разработка хорошей ИМ часто обходится дороже создания аналитической модели и требует больших временных затрат. Иногда может показаться, что ИМ точно отражает реальное положение дел в СС, а в действительности это не так. Р. Шеннон показал в [81], что ИМ в принципе не точна и мы не в состоянии измерить степень этой неточности. Тем не менее имитационное моделирование является одним из наиболее широко используемых методов при решении задач синтеза и анализа СС. Из достоинств имитационного метода выделим: возможность описания поведения компонент СС на высоком уровне детализации, отсутствие ограничений на вид зависимостей между параметрами ИМ и состоянием внешней среды СС, возможность исследования динамики взаимодействия компонент во времени и пространстве параметров системы. Указанные достоинства обеспечивают имитационному методу широкое распространение.

Зачастую исследователи обращаются к имитационному методу, не представляя при этом тех трудностей, с которыми они встретятся, и совершают ряд ошибок методологического характера. Затем наступает разочарование, иногда преждевременное. В данной книге обращается внимание на трудности, возникающие перед исследователем при использовании имитации, и предлагаются методики, позволяющие их преодолеть.

Глава 1

ОБЩИЕ ВОПРОСЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

1.1. СУЩНОСТЬ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Прежде всего о том, почему необходим двойной термин «имитационное моделирование». Имитация и моделирование являются почти синонимами. Фактически все расчетные методы на ЭВМ во всех областях науки и техники являются моделями реальных процессов. Чтобы отличить математические модели друг от друга, исследователи стали давать им дополнительные названия. Термин «имитационное моделирование» означает, что мы имеем дело с такими математическими моделями, с помощью которых результат нельзя заранее вычислить или предсказать, поэтому для предсказания поведения реальной СС необходим эксперимент (имитация) на модели при заданных исходных данных.

Имитация представляет собой численный метод проведения на ЭВМ экспериментов с математическими моделями, описывающими поведение СС в течение заданного или формируемого периода времени. Поведение компонент СС и их взаимодействие в ИМ чаще всего описываются набором алгоритмов, реализуемых на некотором языке моделирования. Все эти описания представляют собой программную ИМ, которую необходимо вначале отладить и испытать, а затем использовать для постановки ИЭ на ЭВМ. Поэтому под процессом имитации на ЭВМ понимаются и конструирование модели, и ее испытание, и применение модели для изучения некоторого явления или проблемы.

При построении ИМ исследователя интересует прежде всего возможность вычисления некоторого функционала, заданного на множестве реализаций процесса функционирования изучаемой СС и характеризующего поведение объекта имитации. Наиболее важным для исследователя функционалом является показатель эффективности системы. Имитируя различные реальные ситуации на ИМ, исследователь получает возможность решения таких задач, как оценка эффективности различных принципов управления системой, сравнение вариантов структуры системы, определение степени влияния изменений параметров системы и начальных условий имитации ее поведения на показатель эффективности системы.

Существует множество толкований основных определений таких понятий, как имитационная модель, компоненты и параметры модели, функциональные

зависимости, ограничения, целевые функции моделирования. В дальнейшем мы будем пользоваться определениями, приведенными Р. Шенноном [81, § 1.7].

1. Каждая модель представляет собой некоторую комбинацию таких составляющих, как компоненты, переменные, параметры, функциональные зависимости, ограничения, целевые функции.

2. Под компонентами понимают составные части, которые при соответствующем объединении образуют систему. Иногда компонентами считают также элементы системы или ее подсистемы. Система определяется как группа или совокупность объектов, объединенных некоторой формой регулярного взаимодействия или взаимозависимости для выполнения заданной функции. Изучаемая система состоит из компонент.

3. Параметрами являются величины, которые исследователь может выбирать произвольно, в отличие от переменных модели, которые могут принимать только значения, определяемые видом данной функции. В модели системы будем различать переменные двух видов — экзогенные и эндогенные. Экзогенные переменные называются также входными. Это означает, что они порождаются вне системы или являются результатом взаимодействия внешних причин. Эндогенными переменными называются переменные, возникающие в системе в результате воздействия внутренних причин. В тех случаях, когда эндогенные переменные характеризуют состояние или условия, имеющие место в системе, назовем их переменными состояния. Когда же необходимо описать входы и выходы системы, мы имеем дело с входными и выходными переменными.

4. Функциональные зависимости описывают поведение переменных и параметров в пределах компоненты или же выражают соотношения между компонентами системы. Эти соотношения по своей природе являются либо детерминистскими, либо стохастическими. Оба типа соотношений обычно выражаются в виде алгоритмов, которые устанавливают зависимость между переменными состояниями и экзогенными переменными.

5. Ограничения представляют собой устанавливаемые пределы изменения значений переменных или ограничивающие условия их изменений. Они могут вводиться либо разработчиком, либо устанавливаться самой системой вследствие присущих ей свойств.

6. Целевая функция (функция критерия) представляет собой точное отображение целей или задач системы и необходимых правил оценки их выполнения. Выражение для целевой функции должно быть однозначным определением целей и задач, с которыми должны соизмеряться принимаемые решения.

1.2. 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_{i1} . Коррекция движения и сброс первой ступени i -й ракеты на стадии 2 определяют ΦD_{i2} . Вторая коррек-

ция движения и сброс второй ступени i -й ракеты означают выполнение ΦD_{ij} . 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_{ij} . 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_{ij} . Будем говорить, что в результате выполнения ΦD_{ij} в СС происходит событие C_{ij} . Каждое из событий в реальной СС связано, как правило, с соответствующей компонентой K_i . При этом любое ΦD_{ij} выполняется на некотором временном интервале τ_{ij} . Для каждой K_i введем понятие локального времени t_i . В СС все t_i изменяются одновременно, однако характер этих изменений различен и определяется последовательностью временных интервалов $\{\tau_{ij}\}$.

При построении ИМ СС ΦD_{ij} аппроксимируются некоторыми упрощенными функциональными действиями $\Phi D'_{ij}$. Степень этого упрощения определяет уровень детализации ИМ. Отличия $\Phi D'_{ij}$ от ΦD_{ij} порождают ошибки имитации реальной СС. В ИМ ΦD_{ij} представляется парой $(\Phi D'_{ij}, \tau_{ij})$, которая выполняется следующим образом. Вначале реализуется $\Phi D'_{ij}$ при неизменном значении t_i , а затем уже отображается изменение t_i на величину τ_{ij} , пницируя таким образом появление события C_{ij} .

На рис. 1.1 представлен пример развития действий K_i в системе координат $(\Phi D'_{ij}, t_i)$. Конечно, подобная система координат носит условный характер, поскольку по оси ординат нельзя отложить «значение» функционального действия $\Phi D'_{ij}$. Подобное представление используется только для изображения того факта, что с изменением «локального времени» t_i некоторая K_i выполняет несколько различных ΦD_{ij} . Так, согласно рис. 1.1 K_i последовательно выполняет ΦD_{i1} , ΦD_{i2} , ΦD_{i3} , а в СС соответственно происходят события C_{i1} , C_{i2} , C_{i3} . Причем с появлением каждого нового C_{ij} происходят изменения ΦD_{ij} и увеличение его временной координаты t_i соответственно на величину τ_{i1} , τ_{i2} , τ_{i3} . Условно на рис. 1.1 появление событий в K_i при выполнении ΦD_{ij} показано штрихпунктирной линией. В ИМ появление событий реализуется ступенчатой линией $(0, a, C_{i1}, b, C_{i2}, d, C_{i3})$. Это означает, что вначале выполняется $\Phi D'_{i1}$ при неизменном t_i , а затем

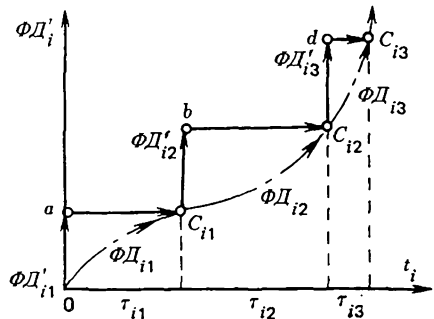


Рис. 1.1. Пример аппроксимации функциональных действий i -й компоненты системы K_i в имитационной модели

уже отображается изменение t_i на величину τ_{i1} , инициируя появление события C_{i1} . Затем реализуется $\Phi D'_{i2}$ при неизменном времени t_i и отображается изменение t_i на величину τ_{i2} , инициируя в K_i свершение события C_{i2} . Далее в два приема (реализация $\Phi D'_{i3}$ и последующее изменение t_i на величину τ_{i3}) аналогичным образом инициируется событие C_{i3} . Отметим при этом, что порядок появления события может быть и обратным: сначала изменяется t_i , а затем уже выполняется $\Phi D'_{ij}$.

В ИМ каждое $\Phi D'_{ij}$ описывается в общем случае некоторым алгоритмом AL_{ij} . В ходе имитации происходят реализация $\Phi D'_{ij}$ по соответствующим алгоритмам AL_{ij} и последующее изменение t_i на величину τ_{ij} . Таким образом, любая ИМ описывается набором некоторых «молекул», каждая из которых содержит в себе описание алгоритма выполнения AL_{ij} соответствующего $\Phi D'_{ij}$ и оператора Mt_{ij} , осуществляющего изменение временной координаты t_i на величину τ_{ij} . Пару (AL_{ij}, τ_{ij}) обычно называют ij -й активностью ИМ и обозначают AK_{ij} . Иногда активности называют «работами» [81]. 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 C_{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 AL_{ij} . 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 реализоваться алгоритм только одной 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 C_{ij} в модели и выполнение алгоритмов AL_{ij} компонент K_i модели системы.

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

С помощью модельного времени t_0 реализуется квазипараллельная работа компонент ИМ. Приставка «квази» в данном случае отражает последовательный 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 значения t_i при выполнении $АЛ_{ij}$ нескольких K_i совпадают (это означает, что в реальной системе происходит одновременно несколько событий C_{ij}), то последовательно обслуживаются $АЛ_{ij}$, совпадающие по времени выполнения, т. е. имеющие одинаковые значения t_{ij} . Здесь и далее под t_{ij} будем понимать конкретное значение t_i , при котором происходит событие C_{ij} . При этом модельное время t_0 не меняется до окончания выполнения всех совпавших по времени реализации алгоритмов $АЛ_{ij}$. Таким способом последовательно выполняются соответствующие $ФД'_{ij}$ при неизменном значении t_0 . После каждой реализации $АЛ_{ij}$, обеспечивающей выполнение в ИМ $ФД'_{ij}$, выполняется оператор корректировки временной координаты $Мt_{ij}$. Чаше всего эта корректировка сводится к вычислению нового значения t_{ij} по формуле

$$t_{ij} = t_0 + \tau_{ij}.$$

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 $АЛ_{ij}$ и оператора корректировки временной координаты t_i).

Когда имитация одновременно появившихся событий C_{ij} завершена, выполнены соответствующие алгоритмы активностей $АЛ_{ij}$ 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. 1.2). Пусть в системе функционируют три компоненты K_i ($i = \overline{1, 3}$). При функционировании K_1 последовательно происходят четыре события ($C_{11}, C_{12}, C_{13}, C_{14}$) соответственно четырем моментам изменения t_1 ($t_{11}, t_{12}, t_{13}, t_{14}$). Между этими моментами K_1 выполняет четыре различных функциональных действия ($ФД_{11}, ФД_{12}, ФД_{13}, ФД_{14}$). Каждое из указанных функциональных действий выполняется в течение соответствующих интервалов времени ($\tau_{11}, \tau_{12}, \tau_{13}, \tau_{14}$). Аппроксимация $ФД_{ij}$ осуществляется последовательностью $\{ФД'_{1j}\}$, $j = \overline{1, 4}$. Аналогичным образом K_2 последовательно выполняет три функциональных действия ($ФД_{2j}$, $j = \overline{1, 3}$) соответственно в течение трех интервалов времени (τ_{2j}), которые в ИМ аппроксимируются соответственно $ФД'_{2j}$. Наконец, K_3 также последовательно выполняет три функциональных действия ($ФД_{3j}$, $j = \overline{1, 3}$) соответственно в течение трех интервалов времени (τ_{3j}), аппроксимируемых в ИМ соответственно тремя $ФД'_{3j}$. Отметим, что каждое $ФД'_{ij}$ описывается соответствующим

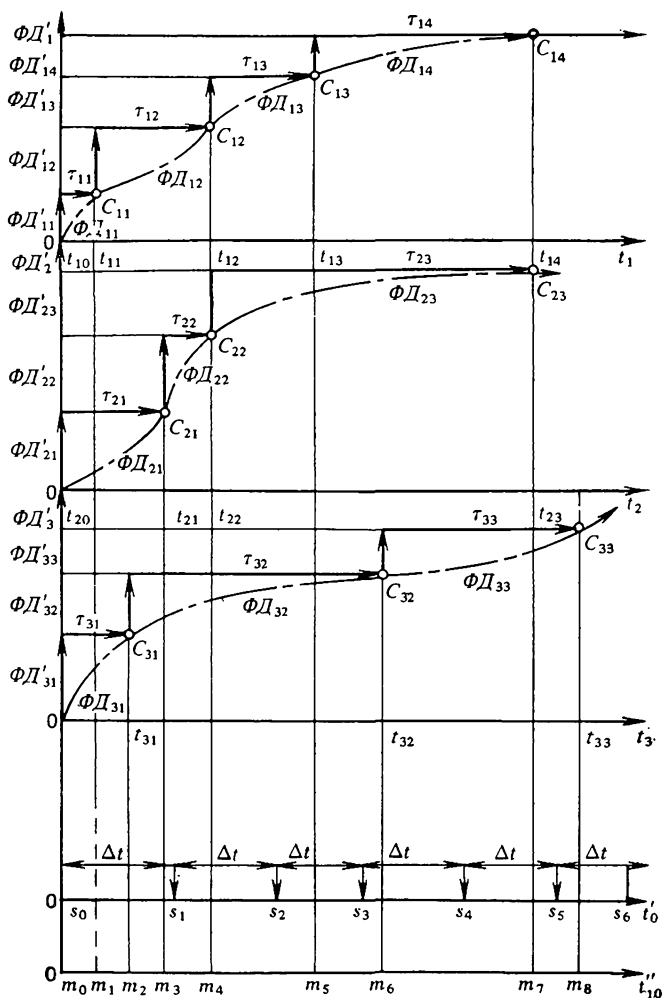


Рис. 1.2. Временная диаграмма моделирования событий в реальной системе из трех компонент

алгоритмом AL_{ij} и реализуется соответствующей активностью, представляющей собой выполнение этого алгоритма при неизменном значении t_i и модификацию временной координаты t_i на величину τ_{ij} . Алгоритм функционирования K_i в координатах $(\Phi D'_i, t_i)$ представлен на рис. 1.2. В ходе выполнения активностей (AL_{ij}, τ_{ij}) в модели K_i происходят последовательно события C_{ij} .

Рассмотрим имитацию событий C_{ij} в системе при каждом из способов изменения t_0 . При имитации по способу фиксированного шага модельное время, обозначенное на рис. 1.2 как t'_0 , меняется каждый раз на величину шага Δt . Тогда в моменты времени 0,

$\Delta t_1, 2\Delta t_1, 3\Delta t_1$ и т. д. в ИМ происходят соответственно события s_0, s_1, s_2, s_3 и т. д., означающие переход на обслуживание управляющей программой моделирования тех событий C_{ij} , которые попадают внутрь очередного интервала Δt . Как видно из рис. 1.2, в действительности события C_{11}, C_{31}, C_{21} происходят на интервале (s_0, s_1) . Считается, что они происходят одновременно в момент $t'_0 = \Delta t$. 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 $t'_0 = 2\Delta t$ 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_i не связаны с моментами t_{ij} ; появления событий в реальной системе. Таким образом, точность моделирования событий C_{ij} 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 m_i , которые соответствуют моментам t_{ij} 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 τ_{12} , а затем выполняется алгоритм AL_{22} и модифицируется координата t_2 на величину τ_{22} . Таким образом, t''_0 каждый раз сдвигается на величину, равную минимальному значению приращения модельного времени τ_{ij} , 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_{ij} в модели системы, являющееся результатом выполнения активности AK_{ij} , обслуживается УПМ и требует для реализации некоторого

ресурса времени работы ЭВМ. Поэтому чем чаще обслуживаются события УПМ, тем больше возрастает машинное время t_3 . С этой точки зрения способ фиксированного шага изменения t_0 является более экономичным, поскольку имеет место групповое обслуживание всех событий C_{ij} , которые попали внутрь очередного шага изменения Δt модельного времени.

На практике предпочтение способу фиксированного шага отдается в двух случаях. Во-первых, когда события C_{ij} распределены равномерно на всем интервале моделирования и исследователь может подобрать интервал изменения временной координаты Δt , обеспечивающий минимальную погрешность имитации. Во-вторых, когда событий очень много и они появляются группами. Во всех остальных случаях способ задания шага до следующего события более предпочтителен. В тех случаях, когда события C_{ij} распределены неравномерно и появляются они через значительные временные интервалы τ_{ij} , способ задания шага до следующего события экономичнее и точнее способа фиксированного изменения t_0 . С учетом достоинств и недостатков каждого из способов изменения t_0 наметилась специализация их применения при моделировании СС.

На практике наибольшее распространение получил способ шага до следующего события. Поэтому в дальнейшем будем рассматривать все методологические вопросы, связанные только со вторым способом изменения модельного времени.

1.3. СПОСОБЫ ОРГАНИЗАЦИИ КВАЗИПАРАЛЛЕЛИЗМА В ИМИТАЦИОННЫХ МОДЕЛЯХ

1.3.1. СТРУКТУРНАЯ СХЕМА ИМИТАЦИОННЫХ МОДЕЛЕЙ

Как уже установлено, в ИМ происходит аппроксимация каждого ΦD_{ij} парой действий: упрощенным функциональным действием $\Phi D'_{ij}$ и модификацией 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 Mt_{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 Mt_{ij} в функции УПМ входят: запуск на выполнение алгоритмов AL_{ij} , организация взаимодействия AL_{ij} друг с другом в ходе имитации и проверка условий окончания имитации. Таким образом, любая ИМ представляет собой совокупность набора «молекул», отражающих поведение объекта имитации, и УПМ, организующей взаимодействие 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 изу-

чения на основе статистики, характеризующей поведение компонент K_i , реальной системы в различных режимах ее функционирования. Для сбора статистики моделирования в ИМ вводится соответствующая подпрограмма, 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, Mt_{ij}), подпрограмм запуска и завершения имитации, УПМ и подпрограммы сбора статистики моделирования.

На рис. 1.3 представлена схема перевода объекта моделирования в его имитационную модель. Из рисунка видно, что любое ΦD_{ij} в ИМ описывается соответствующей активностью AK_{ij} .

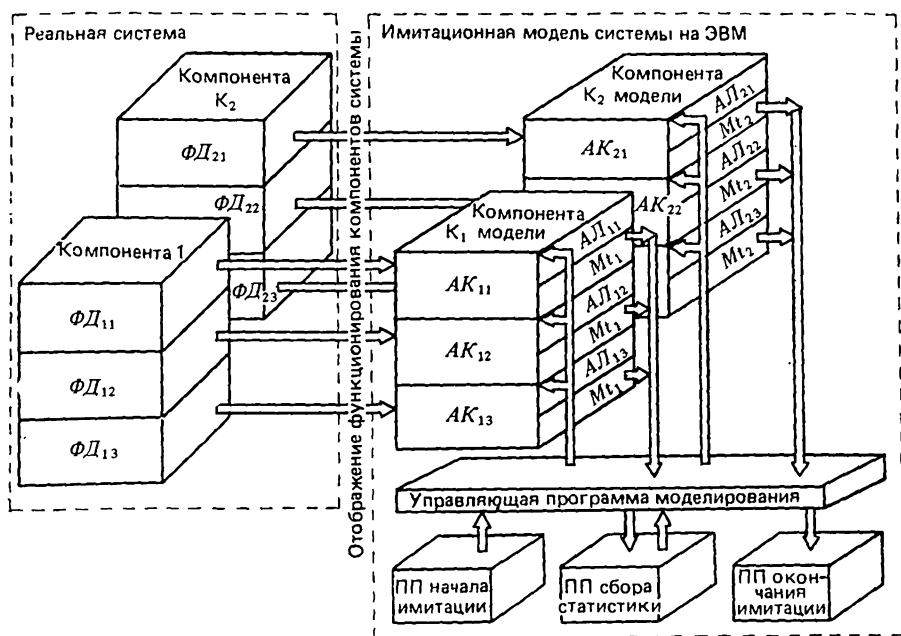


Рис. 1.3. Различие между объектом моделирования и его имитационной моделью

Каждая AK_{ij} представляет собой попарное сочетание описания алгоритма AL_{ij} и оператора модификации временной координаты компоненты модели Mt_{ij} . УПМ передает управление на начало программы, реализующей алгоритмы AL_{ij} . Возврат на УПМ имеет место при выполнении операторов Mt_{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 сложных систем состав активностей в разных компонентах и характер их взаимодействия друг с другом могут быть различными. В одних системах все FD_{ij} различны и для реализации каждого алгоритма AL_{ij} активности в модели требуются выполнение своих условий. В других системах некоторые FD_{ij} в разных компонентах аналогичны, совпадают по времени и приводят к одному и тому же событию C_{ij} . Для реализации в ИМ каждой такой группы FD_{ij} одним алгоритмом AL_{ij} требуется выполнение своих определенных условий. Для некоторых систем существует сильное взаимодействие между FD_{ij} , которые можно аппроксимировать явно задаваемыми математическими зависимостями при создании ИМ. В зависимости от состава алгоритмов AL_{ij} , 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 описания 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 организации квазипараллелизма в ИМ: просмотр активностей, составление расписания событий, управление обслуживанием транзактов, управление агрегатами, синхронизация процессов.

Одну и ту же СС принципиально можно представить любым из указанных способов формализации, однако ИМ, построенные на их основе, будут отличаться размерами и количеством ресурсов, затрачиваемых на их создание, испытание и использование. Поэтому вопросам формализации ИМ с помощью перечисленных пяти способов в гл. 2 будет уделено особое внимание. В данной же главе мы остановимся на способах организации квазипараллельного обслуживания активностей со стороны УПМ, что представляет интерес не только для специалистов по разработке средств автомати-

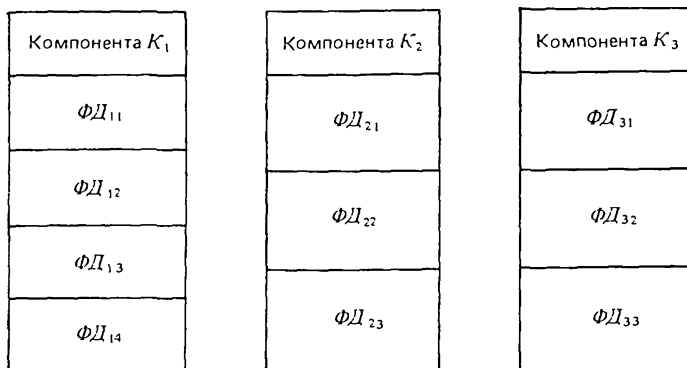


Рис. 1.4. Структура объекта, используемого для иллюстрации способов формализации и моделирования

зации моделирования, но и для пользователей этих средств при создании своих ИМ.

Отличительные 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. 1.4). В соответствии с временной диаграммой перехода K_i в состояния C_{ij} в системе координат $\Phi Д'_{ij}$ время его выполнения (см. рис. 1.2) установлено следующее. Компонента K_1 выполняет четыре функциональных действия ($\Phi Д_{11}$, $\Phi Д_{12}$, $\Phi Д_{13}$, $\Phi Д_{14}$) во времени t_1 , которые в ИМ описываются соответственно четырьмя активностями. Аналогичным образом компоненты K_2 и K_3 выполняют по три функциональных действия ($\Phi Д_{2j}$, $\Phi Д_{3j}$, $j = \overline{1, 3}$) во временах соответственно t_2 и t_3 . Опуская для простоты некоторые детали, рассмотрим основные функции УПМ и принципы ее взаимодействия с компонентами модели системы K_i при каждом из способов организации квазипараллелизма.

1.3.2. ОРГАНИЗАЦИЯ КВАЗИПАРАЛЛЕЛИЗМА СПОСОБОМ ПРОСМОТРА АКТИВНОСТЕЙ

Данный способ организации квазипараллелизма используется при моделировании реальных систем, характеризующихся следующим. Все $\Phi Д_{ij}$ компонент K_i реальной системы различны, причем для выполнения каждой из них требуется выполнение своих условий. Эти условия конкретны, известны заранее исследователю и могут быть представлены алгоритмически. В результате выполнения $\Phi Д_{ij}$ в системе происходят различные события C_{ij} . Связи между $\Phi Д_{ij}$ отсутствуют, все $\Phi Д_{ij}$ функционируют независимо друг от друга. В таких случаях исследователь описывает ИМ в виде двух частей: множества активностей $\{AK_{ij}\}$ и набора процедур проверки выполнимости условий инициализации актив-

ностей. Под инициализацией AK_{ij} понимают передачу управления от УПМ на выполнение AL_{ij} данной активности. В процедурах проверки выполнимости условий инициализации AK_{ij} реализуется зависимость выполнения ΦD_{ij} от конкретной ситуации, имеющей место в реальной системе на каждый момент t_{ij} . Выполнение на ЭВМ алгоритма AL_{ij} данной активности назовем обслуживанием AK_{ij} . Завершается обслуживание AK_{ij} выполнением оператора модификации временной координаты компоненты Mt_{ij} . По этому оператору управление возвращается УПМ. Таким образом, имитация на ЭВМ представляет собой чередование выполнения алгоритмов активностей, операторов модификации временной координаты t_i и алгоритма УПМ. Вся ИМ представляет собой набор активностей AK_{ij} , каждая из которых после выполнения на ЭВМ возвращает управление УПМ.

Схема взаимодействия активностей с УПМ представлена на рис. 1.5. Перед началом имитации соответствующая подпрограмма устанавливает значения начальных состояний компонент модели K_i и параметров, определяющих условия инициализации активностей AK_{ij} . Обычно проверка выполнимости условия инициализации AK_{ij} состоит либо в определении значений параметров модели, либо в вычислении моментов t_{ij} , в которые должно начаться выполнение соответствующего ΦD_{ij} , либо в проверке значений переменных модели. С помощью алгоритма реализуется своя последовательность проверок выполнимости условия инициализации AK_{ij} . Для простоты предполагаем, что каждая AK_{ij} имеет только одну проверку условия ее инициализации (см. рис. 1.5). Отметим, что пользователь описывает не только алгоритмы активностей AL_{ij} и операторы модификации временных координат Mt_{ij} , но и алгоритмы проверки условий инициализации AK_{ij} .

Алгоритм инициализации УПМ AK_{ij} состоит в следующем. По первому циклу (параметр цикла равен максимальному числу AK_{ij} в ИМ) управление последовательно передается на проверку выполнимости условий активизации AK_{ij} . Имена тех активностей, для которых эти условия выполнены, заносятся в список инициализируемых AK_{ij} . По завершении этого цикла начинается второй цикл (параметр цикла равен числу инициализируемых AK_{ij}), в котором осуществляется последовательная передача управления на выполнение AL_{ij} (обслуживание AK_{ij}). Каждая AK_{ij} завершается оператором Mt_{ij} , который вычисляет новый момент t_{ij} инициализации AK_{ij} и возвращает управление на продолжение просмотра второго цикла. Может оказаться, что при выполнении некоторых AK_{ij} , инициализированных на втором цикле, вновь устанавливаются соответствующие значения параметров и переменных, указывающие на то, что выполнены условия инициализации некоторых AK_{ij} . Поэтому управление передается на начало первого цикла для повторного просмотра выполнимости условий инициализации AK_{ij} . Только в случае, когда после первого цикла список инициализируемых AK_{ij} пуст, УПМ корректирует t_0 . Корректировка сводится к замене старого значения t_0 минимальным

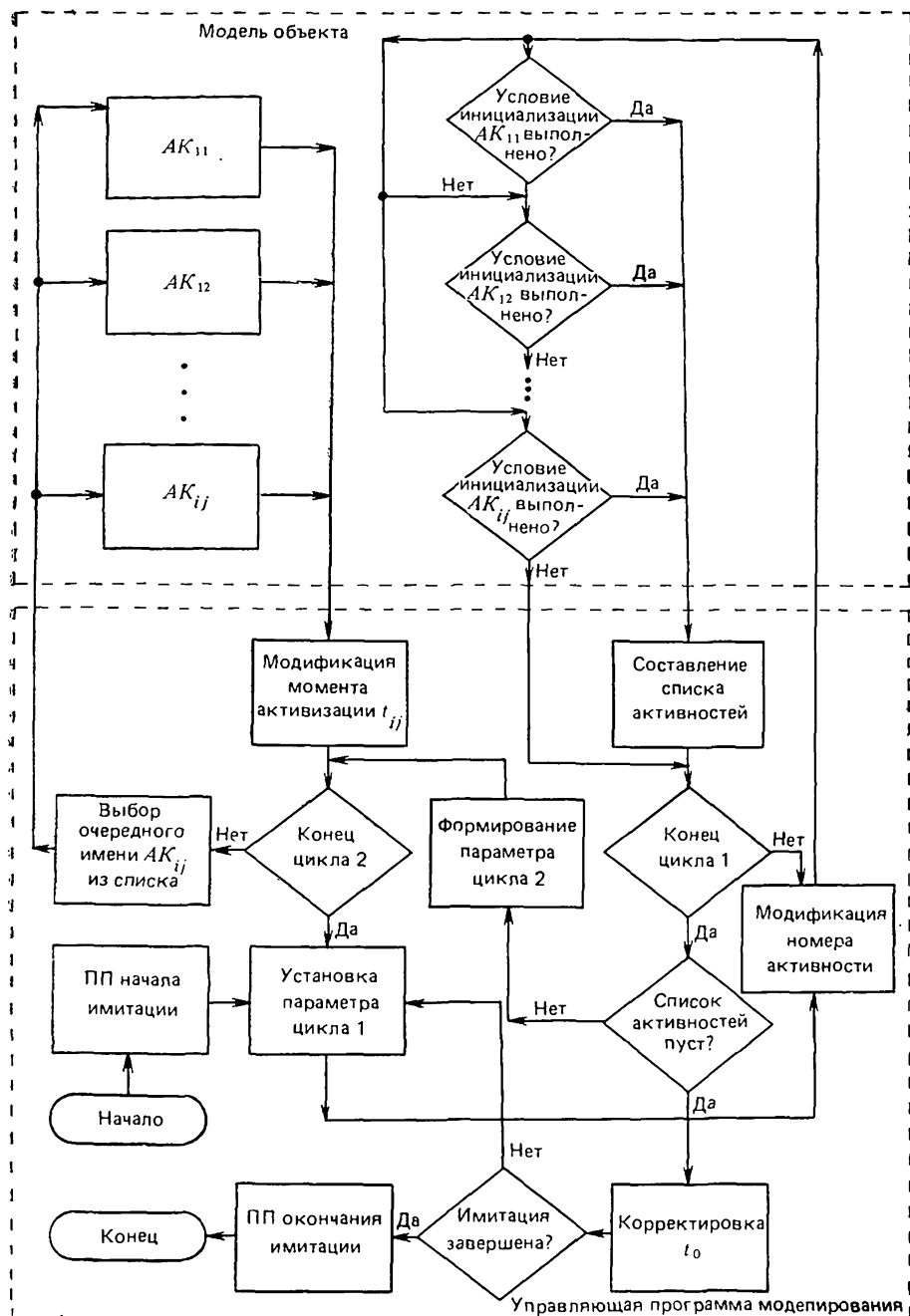


Рис. 1.5. Схема модели объекта, состоящей из активностей

значением момента инициализации активности t_{ij} . Все t_{ij} заносятся в упорядоченный по их значениям список моментов инициализации AK_{ij} . Когда завершается обслуживание некоторой AK_{ij} , в этот список заносятся значение t_{ij} и имя соответствующей AK_{ij} . После корректировки t_0 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 AL_{ij} может привести к инициализации других AK_{ij} , то в УПМ возможны повторные циклы проверки выполнимости условий инициализации AK_{ij} . При этом события C_{ij} в ИМ не регламентированы, а лишь указываются условия, при которых они могут произойти. Отметим, что инициализация AK_{ij} и сдвиг ее временной координаты оператором Mt_{ij} разрешаются только тогда, когда выполняются все условия ее инициализации. Если хотя бы одно из этих условий остается невыполненным, то соответствующая AK_{ij} не попадает в список инициализируемых активностей.

Иногда удельный вес безуспешных проверок очень велик в работе УПМ. Данный способ организации квазипараллелизма AK_{ij} выгоден только при наличии достаточно простых алгоритмов проверки выполнимости условий инициализации активностей.

1.3.3. ОРГАНИЗАЦИЯ КВАЗИПАРАЛЛЕЛИЗМА СПОСОБОМ СОСТАВЛЕНИЯ РАСПИСАНИЯ СОБЫТИЙ

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

Пусть в нашем примере (см. рис. 1.4) у компонент системы совпадают следующие ΦD_{ij} , приводя к одним и тем же событиям:

$\Phi D_{11}, \Phi D_{12}, \Phi D_{13}$ — к событию C_{11} ;

$\Phi D_{21}, \Phi D_{22}, \Phi D_{23}$ — к событию C_{12} ;

$\Phi D_{31}, \Phi D_{32}, \Phi D_{33}$ — к событию C_{13} ;

ΦD_{14} — к событию C_{14} .

В таких случаях исследователь также описывает ИМ в виде двух частей: множества активностей $\{AK_{ij}\}$ и набора процедур

проверки появления событий и инициализации соответствующих активностей. При этом каждая AK_{ij} имитирует выполнение группы совпавших функциональных действий ΦD_{ij} у различных компонент K_i системы. В нашем случае AK_{11} инициализируется при появлении события C_{11} и имитирует выполнение ΦD_{11} , ΦD_{12} , ΦD_{13} . Аналогичным образом инициализируются: AK_{12} при появлении события C_{12} , AK_{13} при появлении события C_{13} и AK_{14} при появлении события C_{14} . Инициализация AK_{ij} имеет тот же смысл, что и для предыдущего способа организации квазипараллелизма, и выполняется УПМ.

В процедурах проверки появления событий C_{ij} реализуется зависимость выполнения ΦD_{ij} от конкретной ситуации, имеющей место в реальной системе. Выполнение алгоритма AL_{ij} также называется обслуживанием AK_{ij} , которое завершается оператором модификации временной координаты Mt_{ij} активности, обслуживающей группу одинаковых ΦD_{ij} , 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} в модели системы и обслуживания событий C_{ij} .

Выполнение этих процедур синхронизируется в модельном времени списковым механизмом планирования УПМ. Каждый элемент этого списка определяет момент t_{ij} 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. 1.6. Перед началом имитации соответствующая подпрограмма устанавливает начальные состояния компонент модели системы и задает исходные значения параметров, определяющих условия инициализации групповых активностей AK_{ij} . Принципы проверки выполнимости условий появления в ИМ событий C_{ij} те же, что и при имитации активностями. Здесь также с помощью соответствующих алгоритмов пользователь составляет описание процедур проверки условий появления событий C_{ij} .

Алгоритм инициализации УПМ процедур обслуживания событий состоит в следующем. По первому циклу (параметр цикла равен максимальному числу различных событий C_{ij} 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 C_{ij}), в котором осуществляется последовательная передача управления на выполнение процедур обслуживания событий. Каждая такая процедура реализу-

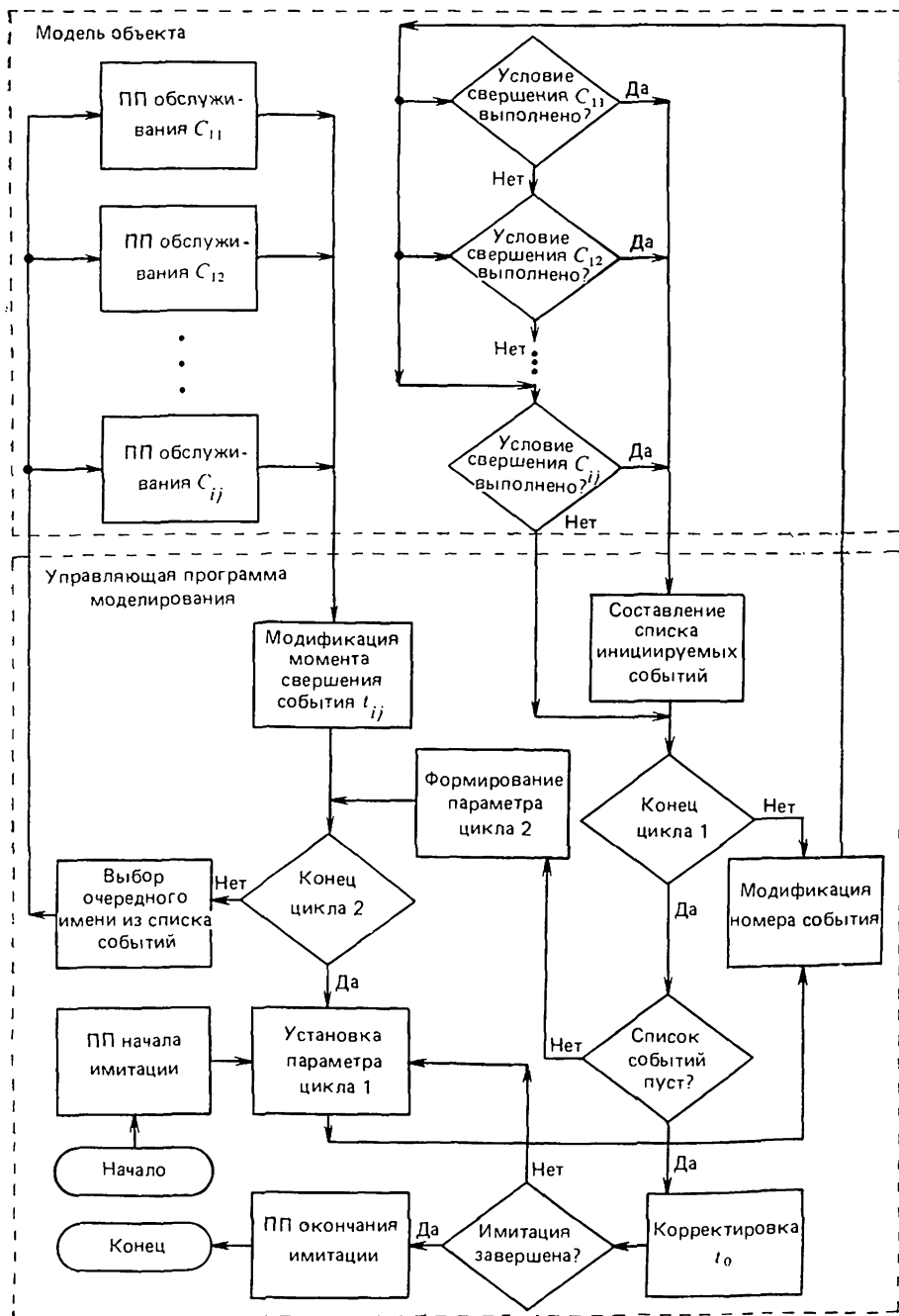


Рис. 1.6. Схема модели объекта, состоящей из процедур обслуживания событий

ет алгоритм соответствующей групповой активности и завершается оператором модификации временной координаты Mt_{ij} , который модифицирует значение момента свершения данного события в будущем и возвращает управление на продолжение второго цикла.

Для того чтобы учесть 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 t_{ij} , которое становится новым значением модельного времени t_0 . После корректировки t_0 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) данный способ имитации более экономичен, чем способ имитации активностями. Однако у рассматриваемого способа имеется существенный недостаток, который заключается в следующем. 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 одним и тем же событиям. Это обстоятельство может затруднить анализ результатов моделирования и модификацию ИМ.

1.3.4. ТРАНЗАКТНЫЙ СПОСОБ ОРГАНИЗАЦИИ КВАЗИПАРАЛЛЕЛИЗМА В ИМИТАЦИОННЫХ МОДЕЛЯХ

В ряде случаев функциональные действия ΦD_{ij} компонент реальной системы одинаковы. Общее число этих действий ограничено. Каждое ΦD_{ij} 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 ΦD_{ij} друг от друга, которую удобно представить в виде схемы. Взаимодействие такого вида активностей

аналогично работе систем массового обслуживания (СМО) [42]. Для имитации поведения реальной системы в таких случаях используется транзактный способ организации квазипараллелизма в ИМ. Однотипные активности пользователем объединяются и называются приборами массового обслуживания [42]. Инициаторами появления событий C_{ij} 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 ΦD_{12} , ΦD_{22} , ΦD_{23} совпадают. Кроме того, в любой данный момент выполняется только одно из них и исследователя интересует влияние этих ΦD_{ij} на поведение всей системы. Взаимосвязь между ΦD_{ij} исследуемой системы 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. 1.7 изображена схема взаимодействия УПМ с моделью объекта при транзактном способе имитации реальной системы, представленной на рис. 1.4. В источниках ($ИСТ_j$) 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_{ij} и до наступления этих моментов закрепляет за транзактами соответствующие блоки обслуживания ($БЛ_{ij}$). Алгоритм работы УПМ состоит в следующем. ПП сканирования источников просматривает условия создания новых транзактов каждым из источников и передает управление ПП

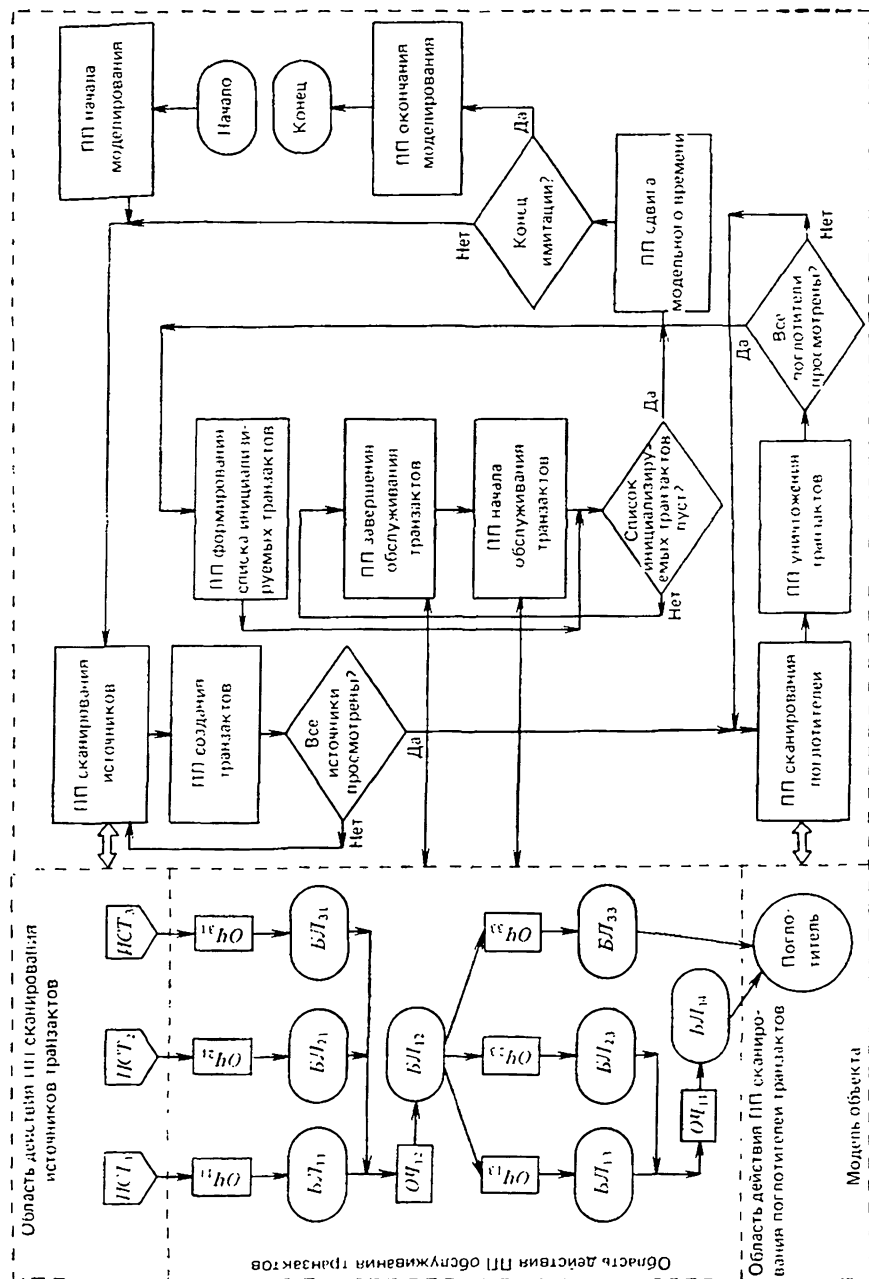


Рис. 1.7. Схема модели объекта при транзактном способе имитации

создания транзактов при совпадении этих условий. ПП создания транзактов формирует новые транзакты и помещает их в очереди к блокам, в нашем примере к $БЛ_{11}$, $БЛ_{21}$ 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 ($ОЧ_{ij}$) к блоку $БЛ_{ij}$, завершившему обслуживание, выбирает из очереди $ОЧ_{ij}$ новый транзакт и закрепляет за ним освободившийся транзакт. При этом формируется момент будущей инициализации транзакта $t_{ij} = t_0 + \tau_{ij}$ (здесь τ_{ij} означает длительность выполнения $ФД_{ij}$, которое имитируется обслуживанием транзактов блоком $БЛ_{ij}$, 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 $БЛ_{ij}$. При этом выполняются те же действия по формированию момента t_{ij} 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 $\{t_{ij}\}$ минимальное значение, которое становится новым модельным временем t_0 . Просматриваются 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 усло-

вий 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 внешнюю простоту описания ИМ приходится платить достаточно большими накладными расходами на организацию квазипараллелизма. Эти дополнительные затраты машинного времени t_3 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.

1.3.5. АГРЕГАТНЫЙ СПОСОБ ОРГАНИЗАЦИИ КВАЗИПАРАЛЛЕЛИЗМА В ИМИТАЦИОННОЙ МОДЕЛИ

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 K_i является входным сигналом для другой компоненты системы. Если при этом сами функциональные действия ΦD_{ij} аппроксимируются явно задаваемыми математическими зависимостями, позволяющими определять момент появления выходных сигналов компонент K_i 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 [10]. В любой момент t_{ij} агрегат может находиться в одном из возможных состояний. Состояния агрегата z являются функциями времени. Зависимости $z(t)$ 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 $X_l(t)$ от других агрегатов, и выходы, на которых формируются выходные сигналы $Y_r(t)$. Здесь l и r означают номера соответственно входных и выходных сигналов. Кроме того, у агрегата имеются дополнительные входы, на которые поступают управляющие сигналы $g(t)$. Выходные сигналы $Y_r(t)$ формируются из входных $X_l(t)$ и

управляющих $g(t)$ сигналов оператором выхода G в результате его взаимодействия с оператором H . Значения операторов H и G задаются исследователем при аппроксимации агрегатами выполнения ΦD_{ij} реальной системы.

Заметим, что квазипараллельная работа агрегативных систем может быть реализована различными способами (активностями, планированием событий, взаимодействием транзактов, процессами). Поэтому выделение агрегативного способа организации квазипараллелизма в ИМ является условным и обуславливается наличием ряда специфических функций, которые должна 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 от одного агрегата к другому. Моделирование поведения агрегата представляет собой последовательную цепь переходов из одного особого состояния в другое. Когда законы поступления управляющих и входных сигналов заданы, то в функции УПМ входит проверка условий перехода агрегата в особые состояния и возможности 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 . Кроме того, через интервалы 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 имеется N входных и m выходных полюсов. На входные полюса поступают элементарные сигналы $X_l (l=\overline{1, N})$. 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 выходным для одного агрегата и входным для другого. Подробности формализации реальных объектов с помощью агрегативного представления рассмотрены в гл. 3. Ниже рассматриваются особенности организации квазипараллельного обслуживания агрегатов УПМ для случая, когда каждое функциональное действие компонент реальной системы ΦD_{ij} аппроксимируется в ИМ отдельным КЛА. Конечно, привязка отдельных ΦD_{ij} к различным агрегатам не всегда целесообразна и рассматривается нами только для простоты изложения специфики организации УПМ квазипараллельной работы агрегатов.

На рис. 1.8 представлена схема взаимодействия УПМ с моделью объекта, состоящей из КЛА. Каждый КЛА аппроксимирует соответствующее ΦD_{ij} реальной системы, представленной на рис. 1.4. Агрегативная система является замкнутой, и возникнове-

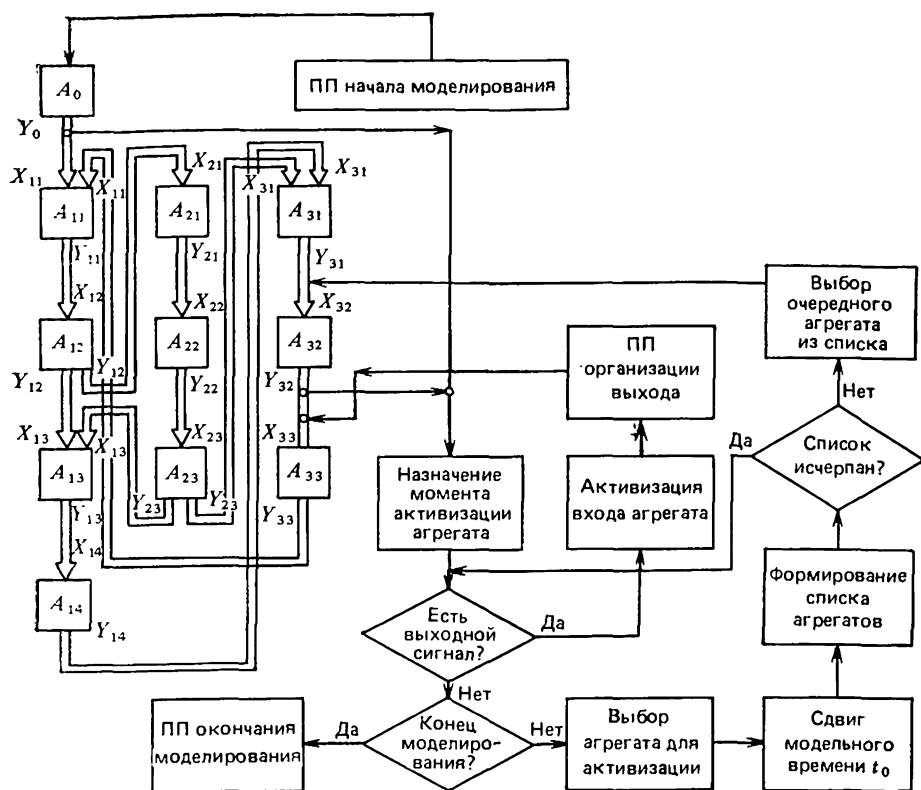


Рис. 1.8. Схема модели объекта, состоящей из агрегатов

ние выходных сигналов в агрегате A_0 возможно лишь вследствие возникновения внутренних событий. Поэтому A_0 представляет собой модель поведения внешней среды, которая воздействует на исследуемую систему. Каждому ΦD_{ij} в ИМ соответствует свой агрегат A_{ij} . Процесс эволюции состояний в агрегативной ИМ и появление событий C_{ij} обеспечивает УПМ. Допустим, на выходе A_{32} появляется выходной сигнал Y_{32} . Произойдет пассивизация агрегата A_{32} , назначение момента будущей активизации агрегата A_{32} и формирование списка выходных сигналов. Под пассивизацией агрегата будем понимать занесение его в список агрегатов, ожидающих выполнения либо временных, либо логических условий, согласно которым УПМ возвращается к обслуживанию 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 формализации реальной системы составляет матрицу коммутации, указывающую коммутацию агрегатов соответствующим

щими каналами. Более подробно структура матрицы коммутации и правила ее составления приведены в гл. 3. При наличии выходного сигнала Y_{32} по матрице коммутации агрегатов УПМ определяет имя входного сигнала (в данном случае это будет X_{33}) и организует активизацию агрегата A_{33} .

Когда все выходные сигналы Y_{ij} агрегатов A_{ij} обслужены, 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} .

Для простоты предполагаем, что все A_{ij} обладают только одной координатой изменения состояний и оператор переходов соответствует простейшему алгоритму вычисления момента $t_{ij} = t_0 + \tau_{ij}$ появления следующего события C_{ij} . При выборе минимального значения t_{ij} определяется точка входа (адрес) или имя того агрегата, который в данный момент должен быть активизирован. 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 последними 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, согласно [68] 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 свершения 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 [10, 11, 68]. При этом исследователь указывает способ выдачи и характеристики опе-

раторов перехода H и операторов выдачи выходных сигналов G , составляет матрицы 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_3 . За удобство описания и математическую строгость модели приходится расплачиваться дополнительным расходом t_3 , и это обстоятельство является одним из факторов, сдерживающих использование агрегативного способа организации квазипараллелизма при моделировании сложных систем.

1.3.6. ПРОЦЕССНЫЙ СПОСОБ ОРГАНИЗАЦИИ КВАЗИПАРАЛЛЕЛИЗМА В ИМИТАЦИОННЫХ МОДЕЛЯХ

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

Процессным способом можно организовать имитацию любых сложных систем. Однако процессный способ имитации особенно эффективен в тех случаях, когда требуется высокий уровень детализации выполнения ΦD_{ij} при их аппроксимации с помощью AL_{ij} и сама ИМ используется для поиска узких мест в системе. При такой постановке задачи на моделирование очень важно соблюдение сходства структуры модели и объекта исследования, что обеспечивается процессным способом имитации. В нашем примере (см. рис. 1.4) реальную систему удобнее всего имитировать процессным способом, когда в один и тот же момент времени не может реализоваться более одного ΦD_{ij} данной i -й компоненты реальной системы. Тогда исследователю удобнее рассматривать функционирование компонент модели K_i как единое це-

лое. Всю имитационную модель можно представить в виде набора описаний процессов, каждое из которых описывает один класс процессов, например компоненты K_i для нашего случая. При этом могут иметь место информационные и управляющие связи не только между компонентами K_i , но даже и между отдельными алгоритмами их функционирования.

На рис. 1.9 представлена схема взаимодействия УПМ с моделью объекта, состоящей из описаний процессов. Алгоритм функционирования ИМ представляется последовательным взаимодействием процессов и УПМ. Причем в процессы объединяются связанные между собой активности, которые определяют функционирование одной и той же компоненты модели K_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_{ij} в K_i . Следовательно, изменения состояний реальной системы соответствуют изменениям соответствующих состояний процессов и появлению событий C_{ij} .

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_{ij} 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 [51]. 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 $WAIT$ (τ_{ij}). Каждый такой оператор означает, что данному i -му процессу по окончании выполнения алгоритма AL_{ij} активности AK_{ij} назначается момент следующей активизации t_{ij} по окончании ожидания процесса в модельном времени длительностью τ_{ij} . Началу выполнения AL_{ij} процессов соответствуют адреса a_{ij} в подпрограммах, реализующих выполнение активностей, объединенных в процессы. УПМ работает с массивом состояний

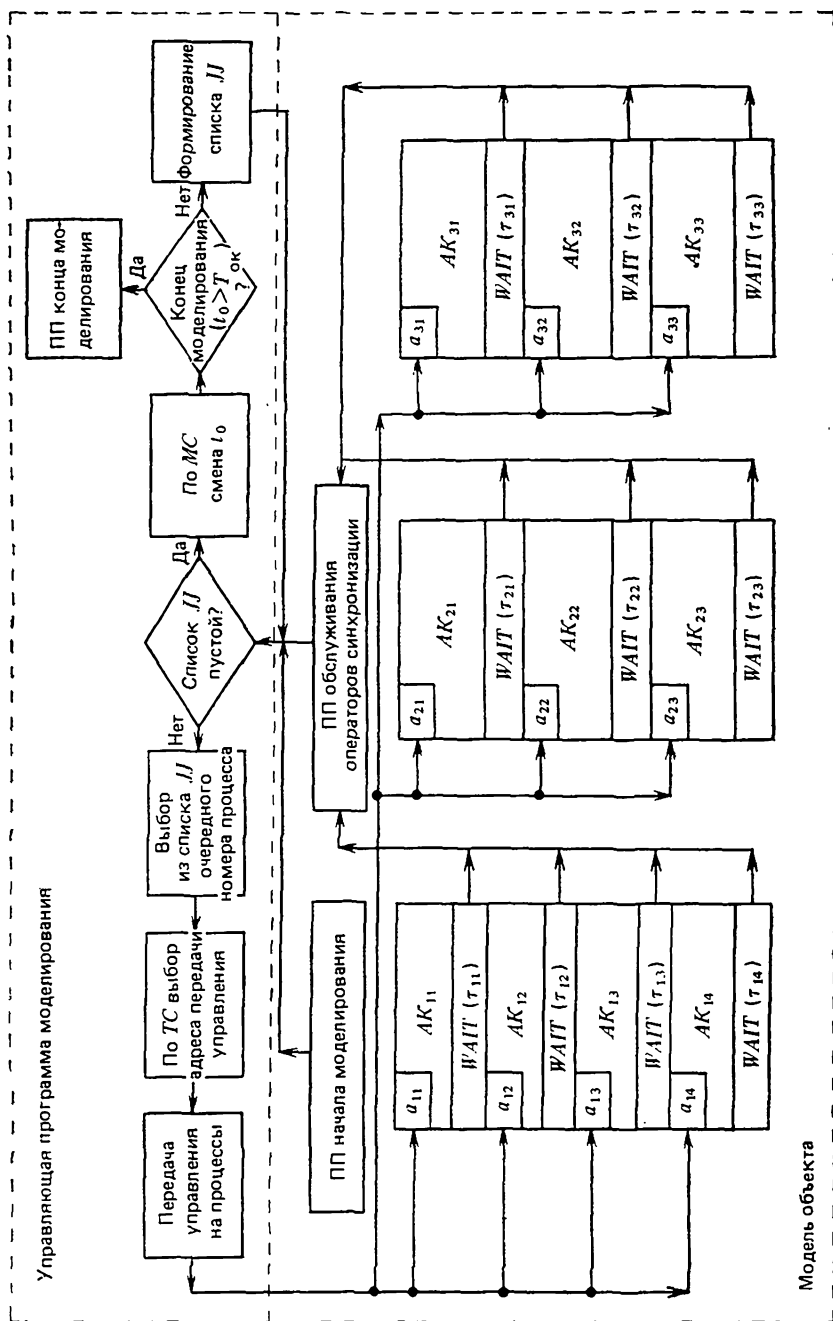


Рис. 1.9. Схема модели объекта, состоящей из процессов

процессов (МС) и таблицей состояний процессов (ТС). В МС каждый элемент представляет собой пару значений (i, t_{ij}) , где i — номер процесса, t_{ij} — момент j -й активизации процесса в будущем и появления события C_{ij} . 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_{ij} — адрес передачи управления на выполнение j -й активности в i -м процессе; $t_{нд i}$ — время, которое осталось i -му процессу находиться в состоянии ожидания, когда он останавливается другим процессом; π_i — приоритет i -го процесса, согласно которому осуществляется последовательное обслуживание УПМ двух одновременно активизируемых процессов.

Все элементы МС упорядочены по возрастанию значения t_{ij} . 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 ($t_{ij} = \infty$) и соответствующий ему элемент находится в конце этого массива.

Определено понятие «конфликтная ситуация в системе дискретных событий»: несколько событий C_{ij} 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 одно и то же модельное время t_0 . Запись номеров процессов i в список JJ и выбор их из этого списка УПМ осуществляет согласно приоритетам процессов π_i . Обслуживание УПМ очередного элемента списка JJ состоит в передаче управления на выполнение алгоритма $АЛ_{ij}$ i -го процесса. Выполнение этого алгоритма продолжается до появления в нем очередного оператора синхронизации процесса $WAIT$ (τ_{ij}). С помощью операторов ожидания типа $WAIT$ (τ_{ij}) имитируется изменение временных координат активностей, входящих в состав i -го процесса.

В нашем примере (см. рис. 1.4) реальный объект состоит из трех компонент, и поэтому его ИМ состоит также из трех процессов ($i=1, 2, 3$). Пусть задано время окончания моделирования $T_{ок}$, а условием окончания моделирования является достижение модельным временем t_0 значения $T_{ок}$. УПМ обслуживает процессы согласно алгоритму, представленному на рис. 1.9. Пусть в начальный момент моделирования $t_0 = m_0$ (см. рис. 1.2) список активизации процессов JJ пуст и процессам назначены соответственно моменты активизации t_{11}, t_{21}, t_{31} в МС.

Шаг 1. Поскольку список JJ 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 $t_{ij} \leq t_0$, выбираются из МС и в список JJ занос-

сятся согласно приоритетам процессов π_i . Приоритеты процессов определяются по таблице ТС.

Шаг 3. Выбирается первый элемент списка JJ , который необходимо в момент $t_{ij}=t_0$ 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 $WAIT$ (τ_{ij}). В ходе реализации алгоритма активности AK_{ij} происходит вычисление или задание значения τ_{ij} для операторов синхронизации процессов. Появления в алгоритме i -го процесса операторов синхронизации $WAIT$ (τ_{ij}) возвращают управление УПМ, которая выполняет в дальнейшем следующие действия:

формирование нового элемента МС и занесение его в этот массив согласно значению $t_{ij}=t_0+\tau_{ij}$;

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

Продолжаем рассмотрение приведенного примера, используя временную диаграмму перехода компонент реальной системы (процессов модели системы) из состояния в состояние (см. рис. 1.2) и схему взаимодействия УПМ с процессами (см. рис. 1.9). 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 $T_{ок}$ равно моменту m_8 на оси изменения модельного времени t_0 (см. рис. 1.2).

Управляющая программа моделирования вначале передает управление по адресу a_{11} . Выполнение алгоритма активности AK_{11} завершается появлением оператора $WAIT$ (τ_{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 ($t_{11} = m_0 + \tau_{11}$);

формирует новую строку МС (1, t_{11}) и заносит ее в этот массив.

Далее по алгоритму УПМ (поскольку JJ не пуст) происходит выбор процесса $i=2$, и управление передается по адресу a_{21} для выполнения алгоритма активности 21 до появления оператора $WAIT$ (τ_{21}) и возврата управления на ПП обслуживания опера-

торов синхронизации. Формируется новый момент активизации процесса 2 ($t_{21} = m_0 + \tau_{22}$). 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 обслуживается процесс $i=3$ и управление передается по адресу a_{31} для выполнения алгоритма активности 31 до появления оператора WAIT (τ_{31}) и возврата на ПП обслуживания операторов синхронизации. Формируется новый момент активизации процесса 3 ($t_{31} = m_0 + \tau_{31}$). В ТС модифицируется адрес продолжения выполнения процесса (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. 1.2).

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

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

При этом $i=1$ заносится в список JJ , и поскольку $t_{11} < m_8$, то управление передается на формирование списка JJ . Так как в 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 (τ_{12}). Процессу 1 назначается новый момент активизации $t_{12} = t_0 + \tau_{12} = m_0 + \tau_{12}$, 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 продолжают обслуживать УПМ процессов и смена модельного времени согласно временной диаграмме рис. 1.2 до момента, когда модельное время $t_0 = m_8$. В этот момент по окончании обслуживания процессов 1 и 2 выполняется условие $t_0 \geq T_{\text{ок}}$ и процесс моделирования завершается, а управление передается на программу окончания моделирования.

При процессном 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 в модель приходится вносить частые изменения. Процессный способ имитации обладает хорошими изобразительными возможностями при осуществлении многоуровневого и модульного подхода к моделированию систем. Эти преимущества возрастают по мере роста размеров модели.

1.4. ТЕХНОЛОГИЯ МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ

1.4.1. ИСПОЛЬЗОВАНИЕ ИМ НА ЭТАПАХ ПРОЕКТИРОВАНИЯ СЛОЖНЫХ СИСТЕМ

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 затратах, то актуальной является задача создания современной технологии производства таких программ. Имитационные модели являются специфичным и достаточно сложным программным 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 увязана 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 проектирования 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 все-

сторонних 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 пересматривается формулировка 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 основных 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 необходимо знакомить 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, ищутся пути ее реализации, составляется план выполнения проекта системы, который должен:

содержать график распределения времени по этапам работ с указанием необходимых 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 системы 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; полное 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 и производственных условий и приводятся результаты анализа различий между реальным образцом системы и ее моделью.

1.4.2. ТЕХНОЛОГИЧЕСКИЕ ЭТАПЫ СОЗДАНИЯ И ИСПОЛЬЗОВАНИЯ ИМИТАЦИОННЫХ МОДЕЛЕЙ

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 алгоритмизация 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. На рис. 1.10 представлена схема взаимосвязи технологических этапов моделирования.

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 имитация функционирования объекта моделирования.

Перед разработчиками **ИМ** ставятся вполне конкретные цели моделирования и формулируются основные критерии эффек-

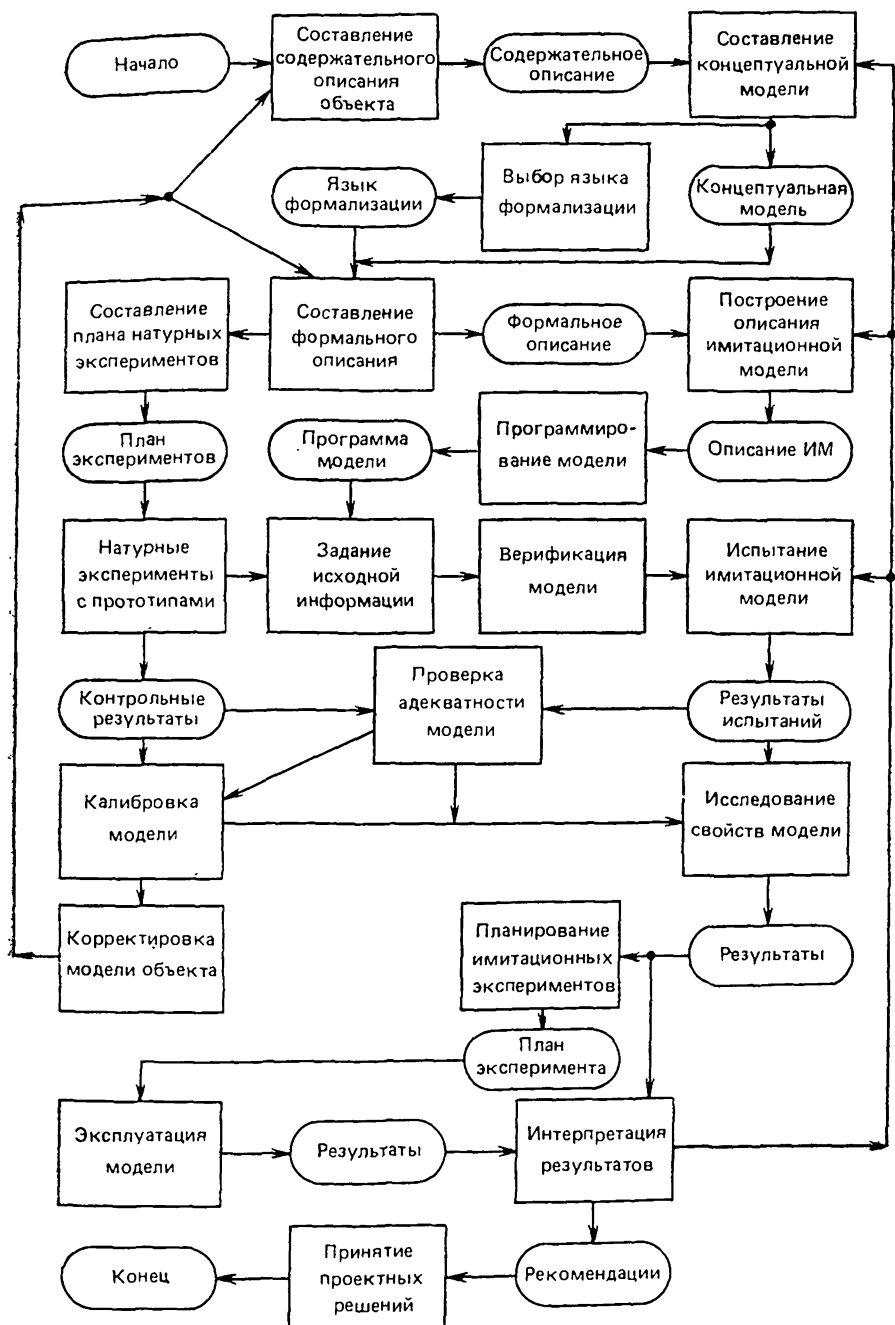


Рис. 1.10. Схема взаимосвязи технологических этапов моделирования

тивности, 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 оно представляет собой техническое описание объекта моделирования, описание внешней среды, с которой он взаимодействует, и временную диаграмму этого взаимодействия.

Составление концептуальной модели. Производится в следующей последовательности.

На основе анализа поставленной задачи определяется общий замысел модели [66]. Выдвигаются гипотезы и фиксируются все допущения, необходимые для построения ИМ.

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

Выполняются такие работы, как выбор параметров и переменных системы, представляющих интерес для моделирования; уточнение критериев 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 моделирования. Изучаются возможности применения известных методов обработки и анализа результатов, выбираются способы представления ожидаемых результатов моделирования. Одновременно с этим исследователь должен выбрать язык будущей формализации процессов в объекте моделирования.

Результатом выполнения работ являются концептуальная модель, выбранный язык формализации и способ организации имитации, реализуемый языком формализации. В состав концептуальной модели входят: уточненное 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 об-

работки 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 формул и алгоритмических записей.

В зависимости от выбранного способа организации квазипараллелизма используются свой язык формализации и своя методика составления формального описания объекта имитации. Ранее были рассмотрены пять методов организации имитации на ЭВМ: имитация активностями, событийное представление, транзактная имитация, агрегатное и процессное представление квазипараллелизма в имитационной модели. Каждый из способов имеет свою специфику реализации. Описания одной и той же имитационной модели для каждого способа имитации существенно отличаются друг от друга. Вопросам составления формального описания при использовании каждого из методов имитации посвящен § 2.3.

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 предшествующая 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 исходной 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 обеспечили бы исследователю легкость 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. Вопросы анализа средств автоматизации моделирования и выбора системы моделирования при каждом способе формализации рассматриваются в гл. 3.

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 имитационную 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 в модельном 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; средство 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 программной 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.

Далее переходят ко второй проверке достоверности программы модели системы. В процессе этой проверки устанавливается соответствие операций в программе и описании 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 компо-

нент 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 [67]. 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.

Как видно из схемы на рис. 1.10, 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 (провести 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 калибровку 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 упростил физические явления, исключил из рассмотрения ряд важных сторон функционирования системы, что привело к неадекватности модели объекту. В этом случае исследователь должен вернуться к этапу формализации системы (см. рис. 1.10). 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 критериев 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 излагает 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.

Стационарность режима моделирования характеризует собой некоторое установившееся равновесие процессов в модели системы, когда дальнейшая имитация бессмысленна, 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 моделирования, при последовательном варьировании параметров моделирования на всем диапазоне их изменений.

Эксплуатация имитационной модели. Этот этап начинается 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 проведению рабочих расчетов на модели. Это весьма трудоемкий процесс, требующий больших затрат ресурса ЭВМ и обилия канцелярской работы. Отметим, что уже на ранних этапах создания ИМ необходимо тщательно продумывать 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 модифицируется на языке моделирования, и поэтому требуются 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 составление концептуальной модели системы соответственно.

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 системы моделирования. Рекомендации по интер-

претации результатов моделирования на имитационных моделях в общем случае давать довольно трудно. Исследователю можно порекомендовать 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.

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.

Глава 2

ПОСТРОЕНИЕ МОДЕЛЕЙ СЛОЖНЫХ СИСТЕМ

2.1. СОСТАВЛЕНИЕ СОДЕРЖАТЕЛЬНОГО ОПИСАНИЯ ОБЪЕКТА МОДЕЛИРОВАНИЯ

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 ограничений, 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. Совокупность 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 среде. Как правило, информации, представленной в исходной документации, бывает недостаточно для описания поведения сложной системы и изучения различных сторон ее поведения. Зачастую это 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 не-

достающую информацию. Такой процесс назовем составлением содержательного описания сложной системы.

Рекомендуется следующая последовательность действий при составлении содержательного описания сложной системы: выбор показателей качества, отражающих цели моделирования; определение управляющих переменных, выбор состава контролируемых характеристик объекта моделирования; детализация описания режимов функционирования системы; дополнение описания информации о внешней среде. На этом этапе исследователь широко применяет опрос инженерно-технического персонала или специалистов, имеющих опыт работы с прототипом проектируемой или реально существующей системы, поведение которой необходимо исследовать.

Выбор показателей качества моделируемой системы. Этот достаточно важный этап 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 тривиальное большинство. Ничего действительно важного не происходит, пока не затронуто жизненно важное меньшинство» [81]. На первом этапе важно отделить главное от второстепенного. Например, если исследователя интересуют вопросы 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 компонента 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. Перерабатывается и дополняется имеющаяся информация для возможного выделения алгоритмов функционирования в каждом из режимов работы системы. Любые неясности устраняются. Со-

ставляются 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 этапе 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 модификация 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 приступать к математической постановке задачи моделирования. Общего рецепта составления содержательного описания не существует. Успех зависит от интуиции и профессиональной подготовки разработчиков имитационной модели.

Для иллюстрации основных этапов разработки имитационной модели рассмотрим следующий пример.

Имеется некоторый участок цеха обработки деталей двух типов четырьмя станками (рис. 2.1). Деталь 1-го типа обрабатывается строго последовательно станками OCT_1 , OCT_2 , OCT_3 . Выбор типа детали при ее обработке на станке OCT_1 случайный. Запрещена одновременная обработка деталей разных типов станками OCT_1 и OCT_2 . Обработка деталей 1-го и 2-го типов соответственно станками OCT_3 и OCT_4 осуществляется только в том случае, когда эти станки свободны. На каждом из станков OCT_j ($j=1, 4$) обработка детали производится в порядке поступления деталей. Приведенный текст описания цеха обработ-

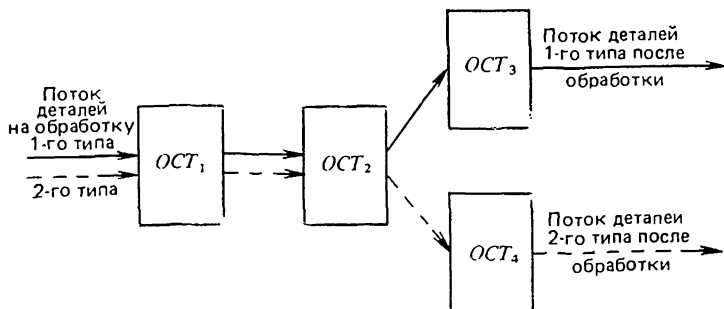


Рис. 2.1. Схема обработки деталей

ки деталей является простейшей документацией, из которой для простоты изложения исключены технологические аспекты, относящиеся к процессу обработки деталей каждым станком.

Выбор показателя качества моделируемого участка обработки деталей определяется назначением моделирования. Целью моделирования является изучение влияния интенсивности поступления деталей в цех на загрузку станков, пропускную способность цеха, среднее время полной обработки одной детали. При этом с помощью модели исследователь надеется выявить узкие места в технологической цепочке обработки деталей. Итак, в качестве показателей качества моделируемой системы будут выступать перечисленные выше характеристики обработки деталей.

Управляемой переменной моделирования будет интенсивность поступления деталей в цех на обработку. Контролируемыми характеристиками объекта моделирования для данного случая являются статистики моделирования, позволяющие определить загрузку станков (суммарное время обработки деталей каждым станком), пропускную способность цеха (количество деталей каждого типа, обработанных цехом), среднее время полной обработки одной детали (интервал от момента поступления детали в цех на обработку до момента завершения ее обработки в цехе).

Детализация описания режимов функционирования сводится к следующим действиям. С помощью хронометража или на основании нормативных данных определяются выборки значений времен обработки деталей станками $\{\tau_{ij}\}$. Здесь индексы i и j означают принадлежность выборки значений τ соответственно к i -му типу деталей, обрабатываемых j -м станком. Затем известными статистическими методами по выборкам $\{\tau_{ij}\}$ формируются соответствующие функции распределения времени обработки i -й детали j -м станком $F_{ij}(\tau)$. Заметим, что на данном шаге для нашего примера не рассматриваются контроль качества деталей, обеспечение станков обслуживающим персоналом, порядок и технология обработки деталей каждым станком и т. д. Эти вопросы в соответствии с рассматриваемой целью моделирования являются лишними.

Описание внешней среды сводится к указанию механизма выбора типа деталей i . Детали 1-го типа поступают на обработку с вероятностью p_1 , а детали 2-го типа — с вероятностью $p_2 = 1 - p_1$. Интенсивность поступления деталей извне является общей для деталей обоих типов. Поступление деталей в цех не зависит от ситуации в цехе и от времени и продолжительности работы цеха.

Приведенный выше текст является содержательным описанием объекта моделирования — в нашем примере участка цеха обработки деталей двух типов четырьмя станками. Содержательное описание объекта моделирования для нашего примера включает следующую документацию:

схему, приведенную на рис. 2.1;

текст, описывающий порядок обработки деталей станками;

функции распределения $F_{ij}(\tau)$, представленные либо аналитически, либо в табличном виде.

2.2. РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ ОБЪЕКТА МОДЕЛИРОВАНИЯ

Имея содержательное описание объекта моделирования, исследователь приступает к разработке его концептуальной модели.

При построении простых имитационных моделей содержательное описание практически представляет собой концептуальную модель. Для построения более сложных ИМ появляется необходимость в выборе способа формализации объекта моделирования. Конечно, когда исследователь располагает средствами автоматизации моделирования, ориентированными на вполне определенный способ формализации, привык к ним (иногда разработал сам эти средства), то проблемы выбора у него не существует и он пытается строить ИМ только этим способом. Однако с появлением различных средств моделирования и усложнением ИМ эта проблема становится весьма актуальной. Отсюда меняются цель и состав работ на этапе разработки концептуальной модели объекта моделирования. На основе анализа содержательного описания определяется общий замысел модели, выдвигаются основные гипотезы, фиксируются сделанные допущения. Уточняется задача моделирования, и если содержательное описание не дает необходимой информации для последующих этапов разработки ИМ, то имеет место возврат на этап составления содержательного описания. Обычно концептуальная модель сложной системы представляет собой упрощенное алгоритмическое отображение реальной системы. С учетом рекомендаций Н. П. Бусленко [10] сложная система расчленяется на конечное число частей (декомпозиция системы), сохраняя при этом связи, обеспечивающие их взаимодействие. Полученные части при необходимости вновь расчленяются до тех пор, пока не получаются элементы, удобные для математического или алгоритмического описания. В результате этого сложная система представляется в виде многоуровневой конструкции взаимосвязанных элементов, объединяемых в подсистемы (подмодели) различных уровней. При этом стремятся к тому, чтобы получаемые подмодели отвечали реально существующим фрагментам системы.

В зависимости от целей моделирования используются различные виды концептуальных моделей систем. В ИМ, предназначенной для отработки вопросов оптимизации архитектуры сложной системы, основной удельный вес в выполняемых функциях модели занимают стохастические элементы при достаточно простом алгоритме функционирования компонент системы. Основным математическим методом, используемым в ней, является метод статистических испытаний. Если же при проектировании сложной системы на моделях осуществляется поиск узких мест в будущей системе, то основным является представление алгоритмических аспектов ее функционирования. Под узким местом в системе будем понимать такую ситуацию, когда скорость поступления запросов одних компонент системы на некоторый общий ресурс системы превышает скорость их удовлетворения со стороны управляющей части системы. В результате к этим ресурсам в системе выстраиваются длинные очереди заказов от компонент системы; в то же время другие ресурсы могут оставаться неиспользованными. Под ресурсом понимают в одних случаях наличие механизмов

и приспособлений общего пользования, а в других случаях ресурсом является место, где выполняются некоторые типовые операции многими компонентами системы. Иногда в качестве ресурсов выступают персонал или ремонтные бригады, которые последовательно обслуживают различные компоненты сложной технической системы. По нашему мнению, основным содержанием этапа составления концептуальной модели является выбор способа формализации объекта моделирования. С этой точки зрения для всех применяемых способов формализации исследователь должен выполнить следующую последовательность работ: провести декомпозицию системы; выбрать параметры и переменные; уточнить критерии эффективности функционирования системы; аппроксимировать реальные процессы математическими выражениями; продумать вопросы задания исходной информации; учесть выдвигаемые гипотезы и предположения; установить структуру модели. Сравнивая получаемые при каждом способе формализации результаты, исследователь делает окончательный выбор. Результаты работ на этапе фиксируются в документации. Как видим, этот этап представляет собой как бы «черновую» (прикидочную) формализацию объекта моделирования. Ниже рассматриваются перечисленные виды работ исследователя при различных способах формализации. Для простоты изложения будем считать, что способы формализации эквивалентны соответствующему способу организации квазипараллелизма.

Декомпозиция сложной системы. Для выполнения этой работы можно воспользоваться рекомендациями, изложенными в [10]. Предлагая читателю ознакомиться с этими рекомендациями, рассмотрим особенности декомпозиции, связанные с различными способами имитации.

При организации квазипараллелизма с помощью активностей сложную систему принципиально можно представить множеством разнотипных активностей AK_{ij} и подпрограмм проверки выполнения условий инициализации AK_{ij} . Декомпозиция системы на AK_{ij} должна быть такова, чтобы за заданные интервалы времени τ_{ij} ошибка аппроксимации ими функциональных действий ΦD_{ij} в системе координат «функциональные действия в модели $\Phi D'_i$ — время функционирования компонент модели t_i » (см. рис. 1.1) была бы незначительна.

Для событийного способа имитации возможна группировка ΦD_{ij} по видам событий C_{ij} , к которым они приводят систему. Поэтому необходимо стремиться к тому, чтобы активности AK_{ij} , аппроксимирующие эти группы ΦD_{ij} , имели одинаковую точность аппроксимации, что означает одинаковые значения интервалов изменения времени выполнения этих активностей τ_{ij} . Для такой декомпозиции объекта имитации на множество программ обслуживания событий C_{ij} весьма существенна слабая взаимосвязь между событиями. В этом случае свершение каждого из событий определяется своей подпрограммой проверки условий их выполнения.

Для транзактного способа имитации функциональные действия ΦD_{ij} аппроксимируются приборами массового обслуживания. Каждому типу ΦD_{ij} соответствует свой тип обслуживающего прибора, реализующего простейший алгоритм $АЛ_{ij}$. При этом наблюдается тесная связь между $АЛ_{ij}$, которую представляют в виде схемы. Существенным для декомпозиции системы при транзактном представлении ИМ является то, что все ΦD_{ij} аппроксимируются ограниченным стандартизованным набором приборов массового обслуживания.

При агрегатном способе имитации все ΦD_{ij} представляются универсальными математическими агрегативными схемами. В качестве определяющей характеристики декомпозиции системы агрегативными схемами также является величина ошибки аппроксимации ΦD_{ij} соответствующими активностями, реализуемыми этими агрегативными схемами. Существенна при этом возможность описания связей между компонентами системы K_i в ИМ с помощью сигналов (входных, управляющих, выходных).

При процессном способе организации квазипараллелизма декомпозиция системы определяется двумя соображениями. Во-первых, каждой компоненте K_i соответствует свой процесс. Таким способом достигается соответствие структуры ИМ и реальной системы. Во-вторых, все ΦD_{ij} аппроксимируются соответствующими активностями исходя из величины ошибки аппроксимации. Если ошибка аппроксимации некоторых ΦD_{ij} на интервале τ_{ij} велика, то этот интервал разбивается на несколько более мелких интервалов, на каждом из которых часть ΦD_{ij} аппроксимируется своей активностью. Таким образом, компоненты модели K_i делят на последовательность активностей AK_{ij} , для которых ошибки аппроксимации ΦD_{ij} находятся в допустимой для данного исследования интервале. Важным для процессного способа имитации является то, что условия свершения событий C_{ij} индивидуальны для каждой компоненты реальной системы K_i и активности AK_{ij} тесно взаимосвязаны между собой.

Просматривая на данном этапе трудности декомпозиции реальной системы при каждом из возможных способов имитации, исследователь ранжирует их по предпочтительности с точки зрения удобства декомпозиции объекта моделирования.

Выбор параметров и переменных. Параметрами модели будем считать те величины, значения которых исследователь может выбирать произвольно и изучение влияния изменения значений которых на исследуемую систему представляет цель имитации. Остальные характеристики модели могут принимать только вполне определенные значения, задаваемые перед началом имитации. Они являются переменными модели. Параметры входят в состав показателей качества моделируемой системы, выбранных на этапе составления ее содержательного описания, и являются варьируемыми характеристиками в функциях показателей качества. Второй группой характеристик, входящих в состав функций показателей качества, являются статистики моделирования. Они

измеряются в ходе моделирования, и по окончании очередной имитации варианта системы их значения подставляются в функцию показателя качества моделируемой системы.

В общем случае описание параметра должно содержать в себе следующую информацию: определение и идентификатор параметра, единицы измерения, диапазон изменения, качественные характеристики (однозначный — многозначный, регулируемый — нерегулируемый), место применения в модели, источник получения значений параметра. Для каждого способа имитации исследователь оценивает легкость определения значений квантов изменения временной координаты τ_{ij} для активностей, имитирующих соответствующие функциональные действия ΦD_{ij} . При имитации активностями и процессами, а также при событийном способе организации квазипараллелизма выделяют общие для всех алгоритмов активностей AL_{ij} характеристики, которые называются глобальными. Поэтому подпрограммы проверки условий активизации событий S_{ij} обычно используют именно эти характеристики модели. Для агрегатного способа имитации значения параметров и переменных модели определяют результат вычислений операторов перехода агрегата в различные состояния, а также условия выдачи выходных сигналов другим агрегатам. В случае транзактного способа организации квазипараллелизма параметры модели обычно выступают в роли характеристик специальных блоков, обслуживающих транзакты. Например, это могут быть характеристики выделения ресурсов памяти транзактам в одних блоках и освобождения этих ресурсов в других блоках программы модели. Прорабатывая на данном шаге вопросы параметризации объекта моделирования, исследователь определяет для себя степень удобства каждого способа имитации с точки зрения задания параметров моделирования.

Уточнение критериев эффективности функционирования системы. Исследователю приходится оценивать, при каком способе формализации выражение эффективности системы как функции от параметров и переменных системы окажется наиболее удобным и легко интерпретируемым. Оценивается удобство сбора статистики моделирования. Так, при имитации активностями, процессами, а также при событийном способе организации квазипараллелизма в качестве статистик моделирования могут выступать частоты использования или удельный вес времени выполнения в общем времени имитации соответственно активностей, процессов, подпрограмм обслуживания событий. Для транзактного способа имитации в качестве статистик моделирования обычно используются: типовые характеристики обслуживания транзактов; средние времена жизни транзактов в системе (от момента создания в соответствующих источниках до момента их уничтожения в поглотителях); коэффициенты использования блоков транзактами (отношение времени обслуживания блоками транзактов к общему времени имитации); характеристики обслуживания очередей транзактов к блокам (средняя длина очереди, среднее время на-

хсждения транзакта в очереди, количество транзактов, побывавших в очереди); вероятности завершения обслуживания транзактов к заданному сроку. Для агрегатного способа имитации кука-занным ранее статистикам моделирования зачастую добавляются частоты поступления управляющих или выходных сигналов от одного агрегата к другому, которые могут иметь самую различную интерпретацию в зависимости от объекта и целей моделирования.

В результате работы на данном шаге исследователь уточняет зависимость критериев эффективности моделируемой системы от накапливаемой статистики моделирования и определяет место и моменты включения процедур для вычисления статистик моделирования.

Аппроксимация реальных процессов математическими выражениями. Возможны два случая: исследователь знает вид функциональной зависимости поведения процессов от параметров и переменных моделирования, когда реализация процессов зависит от сложившейся ситуации в системе. На данном шаге оценивается удобство аппроксимации этих процессов при каждом способе имитации.

При аппроксимации реальных процессов событийным способом имитации необходимо обратить внимание на идентичность алгоритмов, обслуживающих однотипные события в совпавших активностях. Аппроксимация реальных явлений обслуживанием транзактов требует представления взаимодействия компонент системы в виде схемы. Для агрегатного способа имитации аппроксимацию реальных процессов облегчает наличие хорошо изученных математических методов аппроксимации агрегатами. Для способа имитации активностями или процессами точность аппроксимации функциональных действий ΦD_{ij} активностями определяется значениями интервалов изменения модельного времени τ_{ij} . Иногда для уменьшения ошибки такой аппроксимации приходится разбивать интервалы τ_{ij} на несколько более мелких, создавая при этом новые активности.

На данном шаге исследователь сопоставляет величину ошибки от аппроксимации функциональных действий в реальных процессах при каждом из способов имитации и сложность выполнения такой аппроксимации. Под сложностью выполнения аппроксимации функциональных действий ΦD_{ij} обычно понимают количество активностей, число взаимосвязей между ними, трудность написания и наглядность представления алгоритмов активностей $АЛ_{ij}$. Весьма существенна для выбора способа формализации также возможность реализации алгоритмов $АЛ_{ij}$ на базовом языке программирования, имеющемся в распоряжении исследователя.

Задание исходной информации. Часто для задания исходной информации необходимо провести натурные эксперименты на моделируемой системе или на прототипах (при проектировании системы). Сложность проведения натурных экспериментов может существенно повлиять на выбор способа имитации. Может ока-

заться, что для обеспечения необходимой точности при аппроксимации некоторых явлений одним способом требуется проведение множества дорогостоящих экспериментов, тогда как при другом способе имитации эти эксперименты можно существенно сократить. Оцениваются объем работ по сбору статистики и степень доступности прототипов реальных компонент системы для организации соответствующих измерений.

Независимо от способа имитации зачастую определение функций распределения значений τ_{ij} квантов изменения временных координат $F_{ij}(\tau)$ является составной частью процедуры задания исходной информации моделирования. Индексы i и j означают, что, возможно, для каждой активности необходимо находить свой закон распределения значений квантов изменения модельного времени.

Порядок проведения имитационных экспериментов чаще всего излагается в виде сценариев проведения эксперимента. При этом необходимо оценить трудозатраты на реализацию выбираемых сценариев. Может оказаться, что концептуальная модель объекта выбрана неверно, так как требует сложных сценариев для проведения имитационного эксперимента с большим расходом материальных и людских ресурсов. Как показано в [68], в ходе работ на данном этапе разработчик модели должен получить ответ на следующие вопросы: какую исходную информацию следует считать необходимой для построения модели; где можно получить эту информацию и какими методами следует производить обработку исходной информации. Весьма важно предусмотреть либо возможность замены труднодоступной информации более доступной, либо другие варианты решения построения модели. В ряде случаев для получения недостающей исходной информации полезно использовать аналитические модели, чаще всего регрессионные или дисперсионные.

Р. Шеннон [81], например, рекомендует провести следующую последовательность работ: «...просмотреть литературу, составить библиографические справки, проанализировать документы и отчеты, провести консультации со специалистами и экспертами».

Выдвижение гипотез и предположений. В каждом конкретном случае при выдвижении гипотез для описания функциональных действий ΦD_{ij} необходимо учитывать следующие факторы: объем информации, имеющейся для решения задачи; состав задач, для решения которых этой информации недостаточно; наличные ресурсы времени и памяти ЭВМ, на которой будет осуществляться имитация; ожидаемые результаты моделирования. На данном шаге исследователю приходится проводить серию натурных и имитационных экспериментов, в ходе которых проверяются выдвинутые гипотезы об алгоритмическом представлении функциональных действий. Уточняются алгоритмы активностей AL_{ij} и определяются формулы для расчета интервалов изменения временных координат компонент модели τ_{ij} . Так, при имитации активностями, событиями и процессами зачастую выдвигаются гипоте-

зы о том или ином виде регрессионной зависимости при оценке интервалов изменения модельного времени τ_{ij} . Принятие подобных гипотез облегчает расчеты интервалов изменения модельного времени в ходе имитационного эксперимента. Для транзактного способа имитации чаще всего выдвигаются предположения об экспоненциальном характере функций распределения параметров τ_{ij} и допущение о пуассоновском характере входного потока транзактов. Для агрегатного способа имитации иногда приходится принимать гипотезы о конкретном виде операторов H — перемещения агрегата из состояния в состояние — и операторов G — формирования выходных сигналов. Подтверждение экспериментом выдвинутых предположений и гипотез может повлиять на выбор способа формализации реальных явлений.

Исследователь также использует гипотезы и предположения при составлении самих алгоритмов $АЛ_{ij}$, аппроксимирующих функциональные действия $\Phi Д_{ij}$. Может оказаться, что некоторые из этих предположений чрезвычайно трудно реализовать одним из способов имитации и очень просто учесть при другом способе организации квазипараллелизма. Это обстоятельство иногда оказывается решающим для выбора способа имитации.

Установление основной структуры модели. При решении этого вопроса необходимо учитывать сложившуюся обстановку к моменту начала проектирования сложной системы: состояние среды, в которой будет функционировать система; состав функций, для выполнения которых предназначена система; наличие взаимосвязей между человеком и системой. Прежде всего нужно дать ответы на следующие вопросы: какие функции выполняет система и каковы способы их реализации в модели, каков состав детерминированных и недетерминированных функций системы и какова аппроксимация этих функций в модели, каково влияние факторов внешней среды на работу системы, какие используются способы взаимосвязи человека с системой и каким образом они аппроксимируются в модели.

Характер ответов на эти вопросы является определяющим при выборе способа формализации. Так, при имитации активностями исследователь проводит группировку активностей по условиям их выполнения, составляет списки взаимосвязанных друг с другом активностей и выбирает окончательный вариант схемы взаимодействия активностей с УПМ. Для событийного способа имитации события $С_{ij}$ группируются по условиям их выполнения. Затем составляются списки взаимосвязей между событиями и уточняется схема связей с УПМ. При имитации процессами исследователь, наоборот, объединяет активности по принадлежности к одной реальной компоненте системы и составляет списки взаимосвязанных функциональных действий $\Phi Д_{ij}$ внутри отдельных компонент сложной системы. Для агрегатного или транзактного способа имитации по схеме уточняется соответствие входов и выходов агрегатов или блоков, на которые поступают сигналы или транзакты.

Таким образом, при установлении основной структуры модели исследователь должен сделать окончательный выбор способа формализации. При этом необходимо провести следующую последовательность действий: оценить целесообразность построения имитационной модели и удобство аппроксимации результатов при каждом из способов имитации; проанализировать точность аппроксимации функциональных действий в системе и соответствие принятых гипотез и предположений; рассмотреть возможные режимы функционирования компонент системы; исследовать влияние возмущающих факторов на поведение системы; проверить достоверность исходной информации для моделирования и источников ее получения; заново пересмотреть постановку задачи на моделирование и оценить удобство способов имитации для ее решения. Указанная последовательность действий представляет собой проверку соответствия концепций имитационной модели реальной системе.

Завершается этап составления концептуальной модели проведением оценок будущих точностных характеристик имитации реальной системы при каждом из способов имитации. Возможно, что некоторые из способов имитации исследователь забраковал на предыдущих шагах данного этапа. Для случая, когда некоторые способы имитации равнозначны, оценка точностных характеристик имитации позволяет сделать окончательный выбор способа имитации (а значит, и способа формализации) объекта моделирования.

Документирование результатов. Последним действием при формулировке концептуальной модели является документирование результатов. Составляется итоговый документ, который должен содержать следующую информацию: подробную постановку задачи, список параметров, переменных и критериев эффективности системы, перечисление гипотез и предположений, обоснование целесообразности построения модели, описание концептуальной модели согласно принятому способу формализации.

Итак, рассмотренные выше действия привели исследователя к обоснованному выбору способа формализации. Можно утверждать, что к данному моменту разработки имитационной модели исследователь построил концептуальную модель, ориентированную на вполне определенный способ организации имитации.

Пример. Составим концептуальную модель производственного участка обработки деталей двух типов четырьмя обрабатывающими станками, содержательное описание которого было выполнено ранее.

Декомпозиция системы осуществляется просто. Отдельными элементами данной системы (производственного участка) будут обрабатывающие станки (см. рис. 2.1). В качестве параметров системы выступают интенсивности поступления деталей первого и второго типов (λ_1 и λ_2). Для простоты допустим, что поступление деталей обоих типов подчиняется пуассоновскому закону. Переменными модели системы являются функции распределения длительностей обслуживания i -й детали j -м обрабатывающим станком $F_{ij}(\tau)$. В качестве статистик

моделирования будут выступать: коэффициенты загрузки станков OCT_j (η_j), количество обработанных j -м станком деталей v_j , размеры очередей к каждому из станков l_j , общее время обработки i -й детали на производственном участке T_i . Поскольку моделируется достаточно простая структура, то для данного случая в качестве критериев эффективности могут выступать коэффициенты загрузки станков η_j и средние значения времен обработки деталей T_i .

В результате имитационного моделирования необходимо найти следующие функциональные зависимости: $\eta_j = \varphi_{ij}(\lambda_i)$; $T_i = \psi_{ij}(\lambda_i)$. При задании функций распределения $F_{ij}(\tau)$ длительностей обслуживания i -й детали на j -м станке достаточно аппроксимация ступенчатыми функциями. Выдвигаем гипотезу, что φ_{ij} и ψ_{ij} имеют вид полиномов, порядок и значения коэффициентов которых необходимо определить в ходе имитационных экспериментов. С помощью таких эмпирических зависимостей можно предсказывать характеристики загрузки оборудования (η_j) и времена обслуживания деталей на производственном участке (T_i) в зависимости от изменяющихся характеристик входного потока деталей λ_i при заданных $F_{ij}(\tau)$.

Структура модели достаточно проста и имеет блочный характер. Проверка достоверности концептуальной модели в силу ее простоты и близости к реальной системе не производится. Документацией концептуальной модели является:

- схема, приведенная на рис. 2.1;
- текст содержательного описания;
- список параметров и переменных (λ_i ; $F_{ij}(\tau)$);
- состав статистики моделирования (η_j , v_i , l_i , T_i);
- соотношения для критериев эффективности φ_{ij} ; ψ_{ij} ;
- аппроксимирующие выражения для задаваемых переменных $F_{ij}(\tau)$.

2.3. ФОРМАЛИЗАЦИЯ ОБЪЕКТА МОДЕЛИРОВАНИЯ

Процесс формализации сложной системы включает в себя два этапа (см. рис. 1.10): выбор языка формализации и составление формального описания системы. Выбору языка формализации каждого способа имитации посвящена гл. 3. При составлении формального описания системы исследователю рекомендуется следующая последовательность действий: уточнение декомпозиции системы; алгоритмизация компонент модели; информационная стыковка алгоритмов модели; установление управляющих связей между компонентами модели; уточнение взаимодействия с УПМ; документация этапа.

Уточнение декомпозиции системы. В соответствии с показателями качества функционирования системы на предыдущем этапе был установлен уровень детализации моделирования. Исследователь также выполнил декомпозицию системы на составные компоненты, выделил места взаимодействия компонент системы друг с другом. Поэтому основные усилия исследователя на данном этапе направляются на уточнение схемы взаимодействия компонент системы между собой и с УПМ.

При имитации активностями уточняется состав активностей, имитирующих функциональные действия ΦD_{ij} , и подпрограмм проверки условий инициализации активностей AK_{ij} . Обычно уточ-

нение сводится к делению некоторых активностей на составляющие, соответствующие меньшим значениям τ_{ij} их временной координаты, если при оценке ошибки аппроксимации ΦD_{ij} активностями AK_{ij} ее значение оказалось слишком большим. Кроме того, возможно введение ряда служебных активностей, в функции которых входит, например, сбор статистики моделирования или анализ причин появления конфликтных ситуаций в модели системы.

Для событийного способа имитации весьма существенна проверка совместимости выполнения условий появления событий реальной системы и обслуживания их одной и той же подпрограммой модели.

В случае транзактного способа имитации важно, например, определить условия запуска блоков на обслуживание транзактов, имитирующих реальные функциональные действия ΦD_{ij} , а также достаточность состава обслуживающих блоков и дисциплин обслуживания транзактов блоками для имитации всего многообразия явлений в реальной системе. Под дисциплиной обслуживания транзактов понимают порядок поступления и выбора транзактов из очереди к данному блоку.

Для агрегатного способа формализации исследователь обязан определить состав операторов H и G , а также проверить возможность и достаточность описания с их помощью алгоритма AL_{ij} активности AK_{ij} .

В случае процессного способа имитации важно проверить состав и связи активностей, объединяемых в процессы. Иногда может оказаться, что некоторые функциональные действия ΦD_{ij} одной и той же компоненты модели могут выполняться одновременно в силу ряда допущений при построении модели. Поэтому имитацию таких функциональных действий необходимо выполнять двумя процессами. Это означает, что одна реальная компонента системы представлена двумя процессами. Причем эти процессы будут состоять в основном из идентичных активностей и отличаться по составу друг от друга одной или двумя активностями.

Алгоритмизация компонент модели. Уточнив декомпозицию системы в соответствии с выбранным способом имитации, исследователь переходит к уточнению алгоритмов, аппроксимирующих функциональные действия ΦD_{ij} компонент реальной системы. Для этой цели составляются временные диаграммы (ВД) функционирования этих компонент. При имитации активностями (или процессами) с помощью ВД изучаются места взаимодействия активностей (или процессов) друг с другом. Для устранения конфликтных ситуаций при обслуживании активностей (или процессов) УПМ исследователь устанавливает их приоритеты.

При событийном способе имитации с помощью ВД определяются общие по времени и условиям появления события в системе и выделяются места взаимодействия компонент при появлении типовых событий.

В случае транзактного способа имитации ВД составляются для блоков обслуживания транзактов. На основании анализа по ВД возможных конфликтных ситуаций в обслуживании транзактов им (транзактам) определяются приоритеты обслуживания и устанавливаются дисциплины обслуживания транзактов этими блоками.

При агрегатном способе имитации также иногда возможно применение ВД. С их помощью исследуются некоторые случаи изменения координат агрегатов и возможность одновременного поступления на вход агрегата нескольких сигналов. ВД при агрегатном способе имитации позволяет установить приоритеты обслуживания сигналов и наметить последовательность действий УПМ при возникновении конфликтных ситуаций в модели системы.

Далее исследователь обращает внимание на особенности алгоритмического описания функциональных действий при выбранном способе имитации. Уточняется состав глобальных характеристик модели, доступных всем элементам модели, и локальных переменных, используемых и располагаемых внутри алгоритма элемента модели. Под элементами модели для каждого способа имитации будем понимать соответственно активности, подпрограммы обслуживания событий, блоки, обслуживающие транзакты, агрегаты, процессы. Уточняется алгоритмическая часть элементов модели, особенно в тех случаях, когда для более точной аппроксимации функциональных действий используется несколько близких по функциям алгоритмов активностей. Для транзактного способа имитации такого уточнения не требуется, поскольку функциональные возможности блоков обслуживания транзактов, а следовательно, и их алгоритмы фиксированы. При агрегатном способе имитации уточняются состав и назначение координатных переменных. Проверяется правильность описания операторов перехода агрегатов из состояния в состояние H и операторов выхода G .

Информационная стыковка алгоритмов модели. Следующим шагом формализации является определение мест использования и переработки информации компонентами модели. Уточняется состав входной и выходной информации модели. Для каждой компоненты модели уточняется состав управляющих параметров, влияющих на выполнение алгоритмов $АЛ_{ij}$. При различных способах имитации различен и порядок действий исследователя по информационной стыковке $АЛ_{ij}$.

Так, при имитации активностями выделяются активности, связанные информационно друг с другом, и составляется информационная схема взаимодействия активностей. Весьма существенна увязка всех мест взаимодействия активностей по информации с приоритетами активностей и возможным составом операций взаимодействия активностей. Увязка обычно состоит в согласовании t_{ij} времени появления информации для активностей, имеющих различные приоритеты обслуживания УПМ. Необходимо предусмотреть способы задержки передачи информации от активностей,

имеющих больший приоритет, к активностям с меньшим приоритетом обслуживания УПМ.

Для событийного способа имитации наличие информационной взаимосвязи событий друг с другом может существенно повлиять на принятие решения об объединении или разнесении событий и применении общих или различных подпрограмм обслуживания событий в модели. Если при одних и тех же событиях в разные моменты поведения реальной системы используется различная информация для выработки управляющих воздействий между компонентами системы, то в модели они должны быть представлены различными программами обслуживания событий.

При транзактном способе имитации не моделируются информационные связи между компонентами модели. Поэтому наличие таких связей ставит под сомнение возможность транзактного представления модели системы. Иногда помогают выйти из этого затруднения выделение двух типов транзактов (информационных и управляющих) и использование блоков с управляющими входами [52]. Подобное ограничение существует и для агрегатного способа имитации.

Легче всего организуется имитация информационного взаимодействия при процессном способе формализации. В каждом процессе выделяются активности, информационно связанные с активностями других процессов. Составляется схема информационного взаимодействия между процессами и определяются приоритеты процессов при информационной увязке процессов друг с другом.

Уточняется схема информационных связей между компонентами модели. Для способов имитации активностями, процессами и событиями устанавливаются места рождения и потребления информации соответствующими элементами модели и сравниваются между собой сроки создания и потребления информации. В ряде случаев приходится строить модели массивов информации. Особенно важна эта процедура для процессного способа имитации, поскольку процессы могут обмениваться информацией большого объема, а в силу ограничений на объем модели трудно организовать информационный интерфейс между процессами через глобальные переменные. Для транзактного способа, например при использовании в качестве средства формализации МК АСИМ [55], можно организовать специальные логические схемы по обслуживанию информационных транзактов отдельно от управляющих. При агрегатном способе имитации некоторые сигналы могут нести часть информационной нагрузки, в то время как другие сигналы определяют только управляющие воздействия между агрегатами.

Установление управляющих связей между компонентами модели. Следующим шагом переработки концептуальной модели и содержательного описания в формальное описание является составление схемы управляющих связей между компонентами модели. Такая схема обычно представляет собой ориентированный граф, в котором вершинами являются элементы модели, а связи показывают направление воздействия элементов друг на друга и

условия появления этого воздействия. Объем работы по составлению схемы зависит от принятого способа имитации.

Например, для транзактного способа имитации схему управляющих связей вообще составлять не требуется, поскольку развязка блоков обслуживания транзактов осуществляется через очереди к блокам. Для агрегатного способа достаточно составить матрицу коммутации агрегатов друг с другом, и исследователю гарантируется реализация управляющих связей между агрегатами.

В случае имитации активностями, событиями и процессами исследователю приходится уточнять схемы управляющих связей между компонентами модели, которые им были ранее составлены на этапе создания концептуальной модели. Как правило, необходимость в уточнении схемы управляющих связей возникает из-за того, что на этапе формализации выясняется необходимость разложения компонент или разделения отдельных функциональных действий ΦD_{ij} на несколько более мелких. Как следствие, появляются новые компоненты модели. В тех местах, где имеет место взаимодействие одного процесса с другим, на схеме управляющих связей устанавливаются «сигнальные линии» (ребра ориентированного графа). Начало такой «сигнальной линии» находится в конце той активности, которая формирует воздействие на другую активность. Конец «сигнальной линии», обозначаемый стрелкой, входит в начало алгоритма выполнения соответствующей компоненты модели (либо активности, либо подпрограммы обслуживания событий, либо процесса).

Уточнение взаимодействия с УПМ. Далее исследователь обязан уточнить вопросы взаимодействия компонент модели с УПМ. В соответствии с выбранным способом имитации устанавливается свой набор операторов взаимодействия компонент модели с УПМ. Уточняется структура операторов, с помощью которых организуются возврат на УПМ и модификация временной координаты компонент модели системы. Для агрегатного и транзактного способов имитации этот шаг формализации необязателен, поскольку структура взаимодействий элементов модели с УПМ унифицирована.

Документация этапа. В результате уточнения концептуальной модели исследователь получает строгое описание алгоритмов AL_{ij} выбранного способа формализации и операторов взаимодействия активностей как с УПМ, так и друг с другом. По завершении этапа формализации представляется следующая документация:

временная диаграмма функционирования элементов модели;

схема функционального взаимодействия компонент друг с другом и с УПМ;

схема информационного взаимодействия между компонентами; описание текста алгоритмов AL_{ij} на языке формализации.

Для агрегатного способа имитации требуются еще матрица коммутаций агрегатов друг с другом и описание начального со-

стояния моделируемой системы. При транзактном способе необходимо также описание источников и поглотителей транзактов.

Результатом работ является формальное описание сложной системы, свободное от второстепенной информации (имеющейся в содержательном описании) и устанавливающее структуру алгоритмического представления объекта моделирования. Может оказаться, что информации, имеющейся в содержательном описании, недостаточно для формализации объекта моделирования. В этом случае следует вернуться к этапу составления содержательного описания и дополнить его данными, необходимость в которых обнаруживается при формализации объектов моделирования. На практике таких возвратов может быть несколько.

Пример. Продемонстрируем этап формализации на нашем примере с производственным участком обработки деталей двух типов четырьмя станками.

Декомпозиция системы осуществляется просто: каждый из станков OCT_j представляет собой активную компоненту объекта моделирования. Составляется временная диаграмма взаимодействия станков (рис. 2.2). На временную диаграмму заносятся моменты начала и конца обработки i -й детали j -м OCT ($t_{н\ ijh}$, $t_{к\ ijh}$) и длительности обработки деталей i -го типа j -м OCT (τ_{ij}); h — порядковый номер, присваиваемый детали любого типа при поступлении ее на обработку к OCT_j . На первой временной оси t_1 зафиксированы периоды обработки деталей 1-го и 2-го типов первым станком, на временной оси t_2 — периоды обработки деталей 1-го и 2-го типов вторым станком, на осях t_3 и t_4 — соответственно периоды обработки деталей 1-го и 2-го типов третьим и четвертым станками. На нижней временной оси t_0 зафиксированы моменты q , соответствующие началу или концу обработки любой детали любым станком. На временной диаграмме также указаны моменты изменения значения признака π_j , где π_j — признак нахождения j -го станка в состоянии обработки i -й детали ($\pi_j=1$ — OCT_j занят, $\pi_j=0$ — OCT_j простаивает).

На временной диаграмме интервалы обработки деталей 1-го типа заштрихованы линиями «накрест», а интервалы обработки станками деталей 2-го типа — параллельными косыми линиями.

В качестве входной информации используются функции распределения длительностей квантов обслуживания деталей i -го типа j -м станком $F_{ij}(\tau)$ и вероятность p_i поступления на первый станок детали первого типа. Эта вероятность вычисляется по интенсивностям поступления деталей i -го типа (λ_i) следующим образом:

$$p_1 = \lambda_1 / (\lambda_1 + \lambda_2).$$

Ранее были выбраны в качестве критериев эффективности системы коэффициенты загрузки станков η_j и средние значения времен обработки деталей T_i . Вычисления значений этих откликов модели производим по следующим формулам:

$$\eta_j = \sum_{i=1}^2 \sum_{h=1}^{\omega_{ij}} \frac{t_{к\ ijh} - t_{н\ ijh}}{T_{\pi}} \quad (2.1)$$

где η_j — коэффициенты загрузки OCT_j ; ω_{ij} — количество обработанных деталей i -го типа j -м станком (в расчет берутся лишь детали, прошедшие обработ-

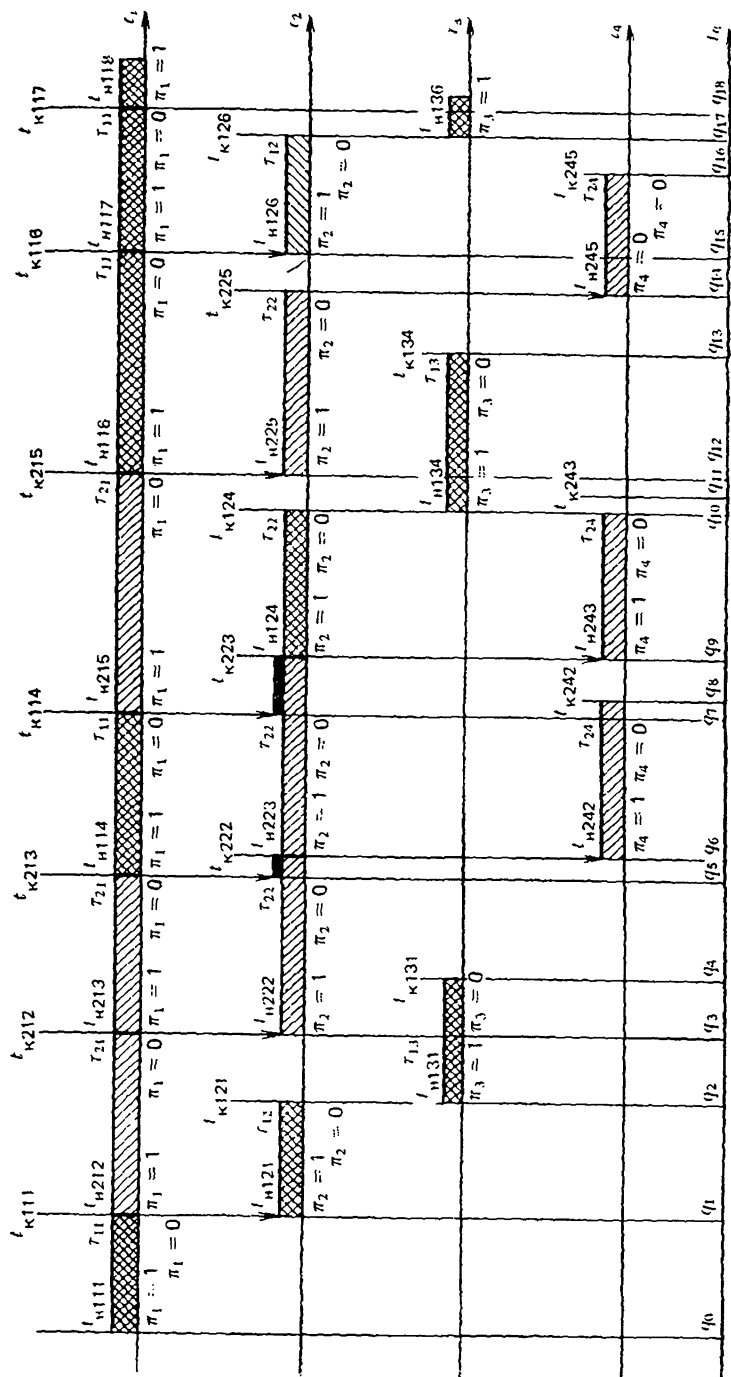


Рис. 2.2. Временная диаграмма функционирования производственного участка

ку на j -м станке); $T_{\text{н}}$ — календарный период исследования производственного участка;

$$T_1 = \frac{1}{\omega_{13}} \sum_{h=1}^{\omega_{13}} (t_{\text{н } 13 h} - t_{\text{н } 11 h}); \quad T_2 = \frac{1}{\omega_{24}} \sum_{h=1}^{\omega_{24}} (t_{\text{н } 24 h} - t_{\text{н } 21 h}), \quad (2.2)$$

где ω_{13} и ω_{24} — число деталей, обработанных соответственно третьим и четвертым станками (в расчет берутся лишь детали, обработанные соответственно третьим и четвертым станками).

В ходе имитационных экспериментов необходимо найти функциональные зависимости

$$\eta_j = \varphi_{ij}(\lambda_i); \quad T_i = \psi_{ij}(\lambda_i),$$

которые согласно нашей гипотезе имеют вид полиномов. Порядок этих полиномов и значения коэффициентов необходимо определить в процессе моделирования. Для начала предполагаем, что целевые функции имеют вид линейных регрессионных зависимостей откликов имитационной модели от параметров моделирования:

$$\begin{aligned} \eta_j &= \beta_{0j} + \beta_{1j} \lambda_1 + \beta_{2j} \lambda_2; \\ T_i &= \gamma_{0i} + \gamma_{1i} \lambda_1 + \gamma_{2i} \lambda_2, \end{aligned} \quad (2.3)$$

где β_{0j} и γ_{2i} — коэффициенты регрессионных зависимостей, значения которых необходимо определить при имитационном моделировании.

При составлении схем информационных связей между *ОСТ*, воспользуемся временной диаграммой (см. рис. 2.2). Для этой цели выделим глобальные переменные, характеризующие связи между станками:

π_j — принимают значение 1, когда j -й станок занят обработкой детали, и равны 0 при простоях станков;

α_j — счетчики числа деталей, находящихся в очереди на обработку к j -му станку;

$ОЧ_j$ — очереди, содержащие номера типов деталей i на обработку к j -му станку.

Дальнейшие действия по формализации производственного участка обработки деталей зависят от принятого способа имитации в модели.

Формализация производственного участка активностями. Переход станка от простоя к занятости и обратно будем отображать в модели соответствующей активностью AK_{ij} . Схема формализации объекта моделирования в этом случае выглядит так, как показано на рис. 2.3. Индекс i у обозначения подпрограмм, реализующих алгоритмы активностей, соответствует стадии обработки деталей ($i=1$ — начало обработки, $i=2$ — конец обработки), а индекс j — номер *ОСТ*. Через $УЗ_{ij}$ обозначим условия инициализации (запуска) данной активности. И хотя проверка условий инициализации активностей выполняется соответствующими блоками, описываемыми отдельно, на схеме формализации объекта моделирования для удобства представления механизма инициализации активностей изобразим эти блоки рассосредоточенными по соответствующим алгоритмам активностей (верхние участки подпрограмм). Если $УЗ_{ij}$ выполняются, то УПМ включает соответствующую активность, AK_{ij} в список инициализируемых активностей. При невыполнении $УЗ_{ij}$ УПМ данную активность пропускает и меняет номер активности, подлежащей рассмотрению.

Активности AK_{i0} моделируют процесс поступления деталей i -го типа извне

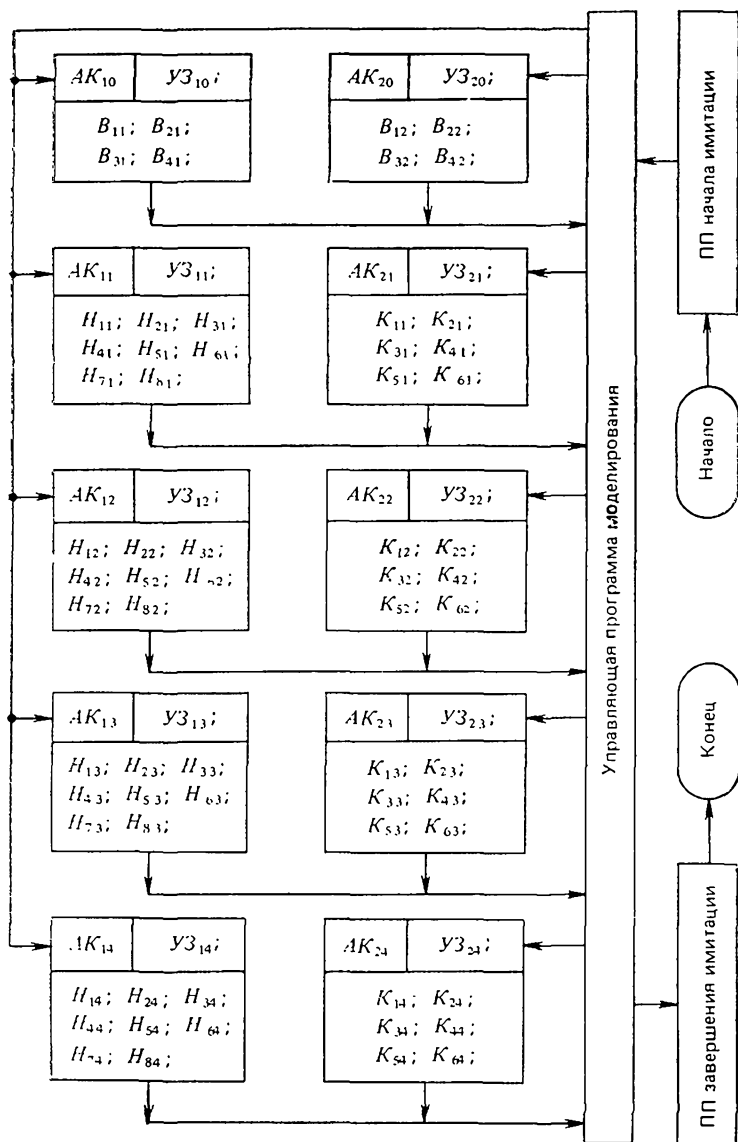


Рис. 2.3. Схема моделирования производственного участка с помощью активностей

на обработку к первому станку. УПМ включает в работу алгоритм AK_{i0} при выполнении условия UZ_{i0} , которое состоит в наступлении момента поступления детали i -го типа ($t_i \leq t_0$) на обработку OCH_1 на обработку ($t_i \leq t_0$). Операторы B_{1i} записывают в OCH_1 к OCT_1 очередной номер детали типа i . Операторы B_{2i} увеличивают значения счетчика числа деталей в очереди OCH_1 к OCT_1 ($\alpha_1 = \alpha_1 + 1$). Операторы B_{3i} формируют моменты t_{ia} очередной активизации AK_{i0} , представ-

ляющие собой сумму текущего значения модельного времени t_0 и интервала временн, после которого новая деталь i -го типа поступает в систему:

$$t_{ia} = t_0 + 1/\lambda_i,$$

где λ_i — значение интенсивностей поступления деталей i -го типа для данного варианта моделирования.

Операторы B_{4i} возвращают управление УПМ, которая заносит очередные значения t_{ia} в упорядоченный по величине список моментов инициализации активностей.

Алгоритмы выполнения активностей AK_{1j} и AK_{2j} состоят из операторов соответственно H_{aj} и K_{bj} ($a = \overline{1, 8}$; $b = \overline{1, 6}$). Активности AK_{1j} инициализируются УПМ при совместном наступлении двух условий: j -й станок свободен и имеются в $OЧ_j$ к нему детали ($УЗ_{1j} : \pi_j = 0 \wedge \alpha_j > 0$). Активности AK_{2j} инициализируются УПМ при наступлении момента окончания обработки детали j -м станком ($УЗ_{2j} : t_{ja} \leq t_0$).

Операторы H_{1j} ($STATIST_{1j}$) и K_{1j} ($STATIST_{2j}$) обеспечивают фиксацию моментов соответственно начала ($t_{n\ ijh}$) и конца ($t_{к\ ijh}$) обработки деталей. Более подробно выполняемые ими функции будут рассмотрены ниже при иллюстрации этапа преобразования формального описания в описание имитационной модели нашего производственного участка.

Алгоритм выполнения активностей AK_{1j} ($j = \overline{1, 4}$) состоит в следующем. С помощью операторов H_{2j} устанавливаются признаки занятости j -го станка обработкой детали ($\pi_j = 1$). Операторы H_{3j} организуют выбор из $OЧ_j$ номера детали i . Номер очередной детали i , обрабатываемой j -м станком в данный момент модельного времени, содержится в глобальной переменной модели δ_j . Запоминание номера детали i в δ_j обеспечивается оператором H_{4j} ($\delta_j = 1$). Уменьшение на единицу числа деталей, находящихся в очереди к j -му станку, выполняется оператором H_{5j} ($\alpha_j = \alpha_j - 1$). Для формирования моментов очередной инициализации активностей используются функции распределения длительностей обслуживания деталей $F_{ij}(\tau)$. Оператор H_{6j} по номеру детали i и $F_{ij}(\tau)$ формирует очередное значение интервала обработки деталей i j -м станком τ_{ij} . С помощью операторов H_{7j} формируются запланированные моменты очередной инициализации активностей AK_{2j} , представляющие собой сумму текущего значения модельного времени и τ_{ij} :

$$t_{ja} = t_0 + \tau_{ij}.$$

Наконец, операторы H_{8j} возвращают управление УПМ.

При выполнении алгоритмов активностей AK_{2j} реализуется следующая последовательность операторов. С помощью операторов K_{2j} сбрасывается признак занятости j -го станка обработкой детали ($\pi_j = 0$). Операторы K_{3j} обеспечивают извлечение номера детали i из δ_j и формирование по нему номера очереди ν , куда поступает деталь i -го типа после завершения обработки на j -м станке. Операторы K_{4j} записывают номер детали типа i в очередь $OЧ_\nu$. Увеличение на единицу значения счетчиков числа деталей в очередях осуществляется операторами K_{5j} ($\alpha_\nu = \alpha_\nu + 1$). Возврат на УПМ обеспечивается операторами K_{6j} .

Формализация производственного участка при событийном способе имитации. Анализ алгоритмов выполнения активностей, аппроксимирующих процесс обработки деталей станками, показывает, что их можно объединить по группам. Все запуски станков на обработку деталей объединим в один тип собы-

тий C_3 . Аналогично объединим все окончания обработки деталей на станках в другой тип событий C_4 . Обозначим алгоритмы активностей AK_{1j} в виде подпрограммы обслуживания событий C_3 ($ПОС_3$), а алгоритмы активностей AK_{2j} в виде подпрограммы обслуживания событий C_4 ($ПОС_4$). Сохраняется состав алгоритмов активностей AK_{i0} , моделирующих процесс поступления извне на обработку деталей i -го типа к первому станку и условия их запуска УПМ ($УЗ_{i0}$). Будем считать, что поступление на обработку деталей i -го типа означает появление событий C_i ($i=1, 2$). Алгоритмы активностей AK_{i0} также назовем подпрограммами обслуживания событий C_1 и C_2 ($ПОС_1$ и $ПОС_2$). Блоки проверки условий инициализации любой из активностей назовем подпрограммами проверки выполнимости событий. Количество блоков проверки выполнимости условий инициализации активностей ($УЗ_{ij}$) с восьми сократится до двух. Однако сами алгоритмы проверки условий увеличатся в размере.

Подпрограмма ($УЗ_{1j}$) обслуживания события C_3 инициализируется УПМ, когда j -й станок свободен и к нему в $ОЧ_j$ имеются детали: $(\pi_1=0 \wedge \alpha_1 > 0) \vee (\pi_2=0 \wedge \alpha_2 > 0) \vee (\pi_3=0 \wedge \alpha_3 > 0) \vee (\pi_4=0 \wedge \alpha_4 > 0)$. Аналогично подпрограмма ($УЗ_{2j}$) обслуживания события C_4 иницируется УПМ при наступлении момента окончания обработки детали любым j -м станком: $(t_{1a} \leq t_0) \vee (t_{2a} \leq t_0) \vee (t_{3a} \leq t_0) \vee (t_{4a} \leq t_0)$. Алгоритмы активностей AK_{1j} и AK_{2j} увеличатся на один оператор Φ_0 . С помощью этого оператора определяется номер станка j , начало или завершение работы которого обслуживает в данный момент подпрограмма обслуживания события $ПОС_3$ или $ПОС_4$.

На рис. 2.4 представлена схема формализации объекта при событийном способе имитации. Обозначения операторов, реализующих алгоритмы подпрограмм обслуживания событий, имеют тот же смысл, что и при формализации активностями. Как видим, число блоков проверки выполнимости условий, просматриваемых УПМ, сократилось вдвое. Зато увеличились сами алгоритмы проверки выполнимости условий. Практически проверка $УЗ_{1j}$ и $УЗ_{2j}$ требует одного и того же ресурса ЭВМ при обоих рассмотренных способах формализации. Увеличение алгоритмов AK_{1j} и AK_{2j} на один оператор при общем уменьшении их количества вдвое дает экономию в памяти ЭВМ, отводимой под реализацию модели. Можно рекомендовать при простом алгоритме обслуживания событий отдавать предпочтение событийному способу имитации, поскольку возможна экономия памяти, отводимой под описание модели. Кроме того, при большом числе активностей, имеющих одинаковые алгоритмы определения условий инициализации и обслуживания активностей, возможна экономия машинного времени моделирования за счет сокращения общего времени работы УПМ при уменьшении размеров списков событий.

Формализация производственного участка процессным способом. Объединим активности AK_{1j} и AK_{2j} по станкам. Каждый из станков как компоненту объекта моделирования представим в виде процедуры-процесса. В результате временная диаграмма, приведенная на рис. 2.2, в точности представит взаимодействие процессов $ОСТ_j$. Любой j -й процесс состоит из двух активностей (AK_{1j} и AK_{2j}). По окончании выполнения алгоритма некоторой активности процесс переходит в другое состояние. Каждому состоянию S процессов $ОСТ_j$ соответствует свой адрес продолжения алгоритма As_j процесса. На рис. 2.5 представлена схема формализации объекта моделирования на основе процессного способа имитации. Обозначения алгоритмов активностей и состав операторов обработки имеют тот же смысл, что и при имитации активностями. Однако отсутствуют

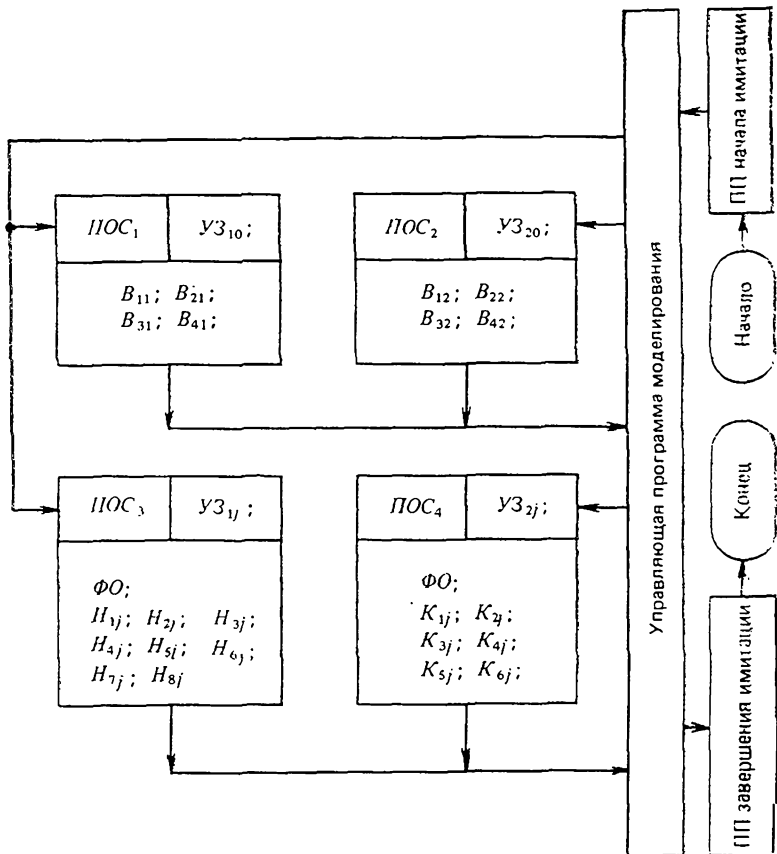


Рис. 2.4. Схема моделирования производственного участка с помощью событий

программы проверки выполнимости условий инициализации активностей, поскольку выбор инициализируемого процесса формализован либо по оператору изменения временной координаты процесса, либо по оператору условной инициализации процесса. Функции оператора изменения временной координаты t_j процесса выполняют операторы возврата на УПМ (H_{8j} и K_{6j}). Первыми операторами в алгоритмах выполнения активностей AK_{1j} являются операторы D_j , выполняющие функции операторов условной инициализации j -го процесса.

По этим операторам процессы OCT_j ожидают выполнимости условий инициализации $УЗ_{1j}$ активностей AK_{1j} . Для нашего примера УПМ выполняет те же функции, что и в случае имитации активностями, однако пользователю нет необходимости повторять описание блоков проверки $УЗ_{1j}$ для каждой активности AK_{1j} . Таким образом, алгоритмы проверки выполнимости $УЗ_{1j}$ от пользователя скрыты, что придает описанию модели большее сходство с моделируемой системой.

Активности AK_{i0} , моделирующие процесс поступления извне на обработку деталей i -го типа к первому станку, представляются отдельными процессами.

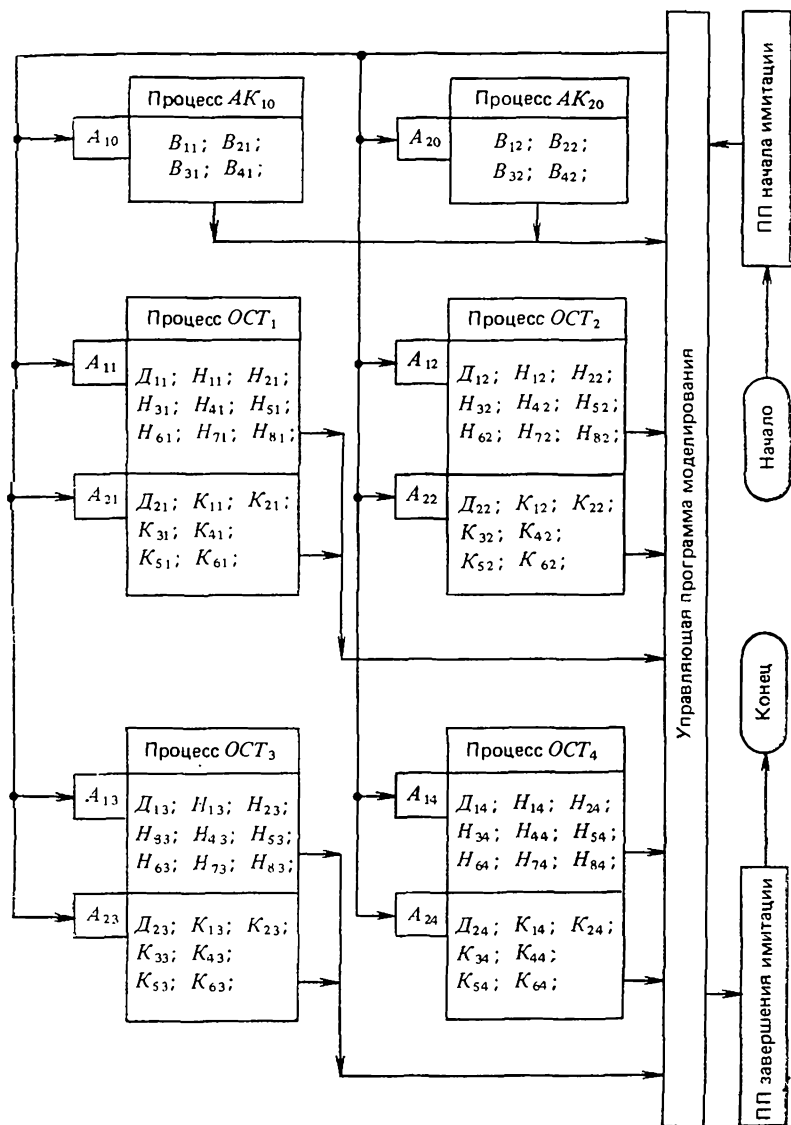


Рис. 2.5. Схема моделирования производственного участка процессами

При этом сохраняются обозначения и выполняемые функции операторов $B_{1i}—B_{4i}$ ($i=1, 2$). В алгоритмах активностей AK_{1j} и AK_{2j} , объединенных в процессы, используются операторы D_{1j} и D_{2j} , модифицирующие адреса передачи управления A_{Sj} ($S=1, 2$) при очередной инициализации УПМ j -го процесса. Так, если процесс OCT_1 находится в состоянии $S=1$, УПМ при инициализации OCT_1 передает управление по адресу A_{11} . Если же OCT_1 находится в состоянии $S=2$, УПМ инициализирует процесс OCT_1 с адреса A_{21} . По оператору D_{11} j -й процесс

переводится в состояние $S=2$, а по оператору D_{2j} j -му процессу устанавливается состояние $S=1$.

Можно заметить, что алгоритмы выполнения процессов OCT_j совпадают. Некоторые средства автоматизации моделирования [51] позволяют экономить текст программы за счет механизмов размножения процессов. Суть механизма размножения процессов состоит в следующем. Каждому j -му процессу отводятся свои рабочие поля, процессы с одинаковыми алгоритмами выполняются одной и той же программой, применяющей программную индексацию при обслуживании рабочих полей различных процессов. Поэтому для нашего случая системы моделирования МК PLSIM [51] пользователь описывает алгоритмы только двух типов процессов: внешней среды (процессы AK_{i0}) и обработки деталей j -м станком (процессы OCT_j). Таким образом, достигается экономия памяти ЭВМ при описании модели производственного участка.

Транзактный способ формализации производственного участка. При этом способе имитации для формализации производственного участка используем два типа элементов, с помощью которых можно представить динамику обработки деталей станками: блоки и очереди. В блоках могут происходить события трех типов: создание или уничтожение транзактов; изменение числового атрибута транзакта; задержка транзакта на некоторый интервал времени. Схема формализации производственного участка при транзактном способе имитации представлена на рис. 2.6. Блоки имеют двойную индексацию. Первый индекс означает номер уровня в схеме обслуживания транзактов, второй индекс соответствует номеру маршрута транзакта и определяется типом обрабатываемой детали (i).

Каждой поступающей извне на обработку детали i -го типа соответствует свой транзакт. В блоках 11 и 12 создаются транзакты с атрибутами i , обозначающими тип детали. Эти блоки выполняют функции источников транзактов и обеспечивают поступление транзактов в очередь 1 с интенсивностями соответственно λ_1 и λ_2 . УПМ просматривает блоки 11 и 12 и в моменты создания помещает транзакты в очередь 1 для последующего их обслуживания блоком 21. В соответствии со значением атрибута (типом детали) блок 21 направляет транзакты либо на блок 31 (для $i=1$), либо на блок 32 (для $i=2$). Блоки 31 и 32 имитируют работу станка OCT_j путем задержки транзакта на времена τ , формируемые по функциям распределения длительностей интервалов обработки деталей $F_{i1}(\tau)$.

Далее транзакты поступают в очередь 2, из которой они выбираются согласно принятому правилу обслуживания деталей блоком 41. Блок 41 изменяет маршрут следования транзактов по значению атрибута i . Блоки 51 и 52 имитируют работу станка OCT_2 путем задержки транзактов на время τ , формируемые соответственно по функциям распределения длительности интервалов обработки деталей $F_{i2}(\tau)$. По завершении обслуживания блоками 51 и 52 транзакты поступают соответственно в очереди 3 и 4. При освобождении блоков 51 и 52 УПМ выбирает транзакты из очередей 3 и 4 и подает их на входы соответственно блоков 61 и 62.

В блоках 61 и 62 имитируется работа соответственно станков OCT_3 и OCT_4 путем задержки транзакта на времена τ , формируемые по соответствующим функциям распределения длительностей интервалов обработки деталей $F_{i3}(\tau)$ и $F_{i4}(\tau)$. По завершении обслуживания блоками 61 и 62 транзакты поступают соответственно на блоки 71 и 72, являющиеся поглотителями транзактов. Эти

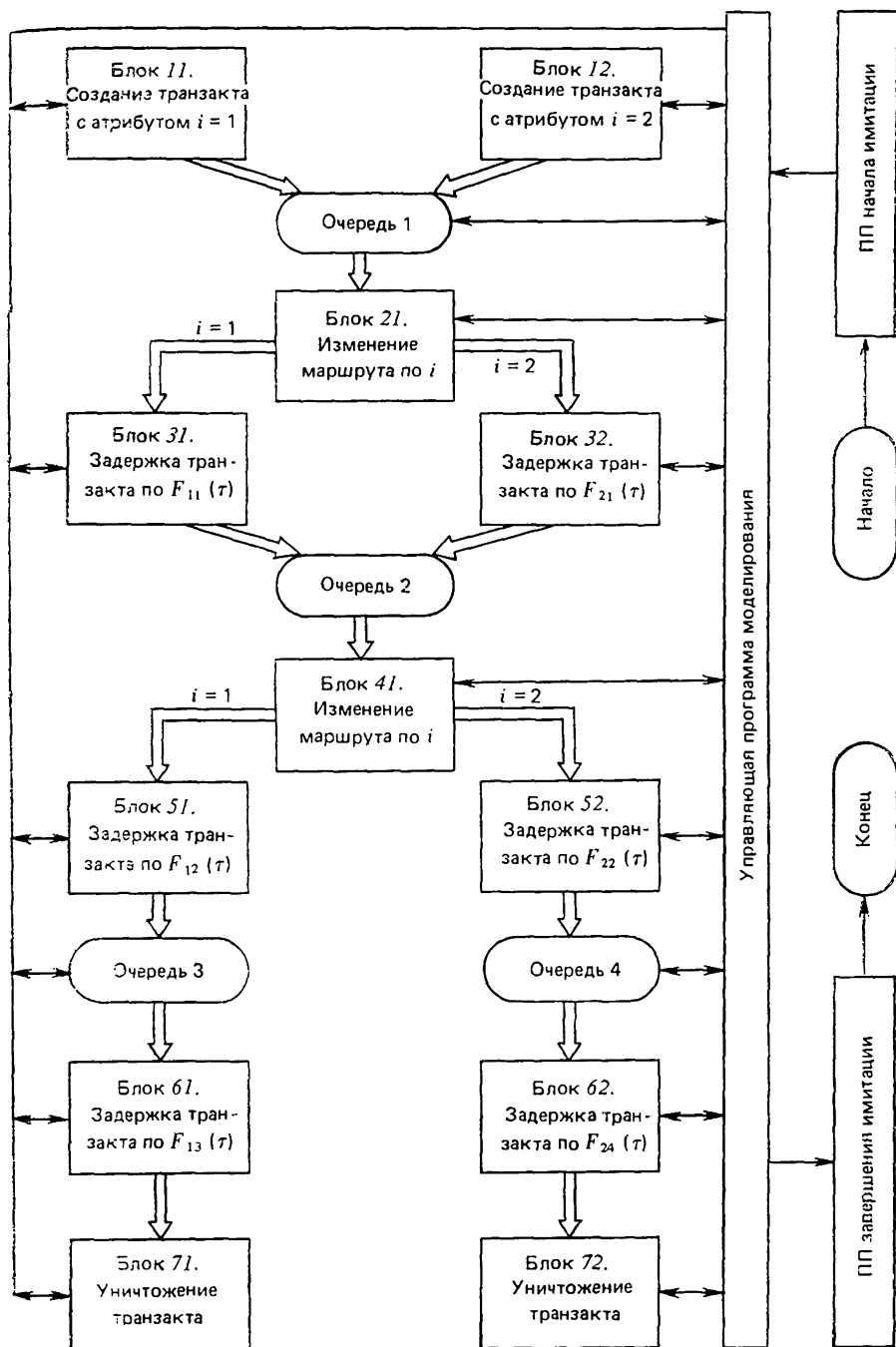


Рис. 2.6. Схема транзактного способа имитации производственного участка

полотители транзактов постоянно просматриваются УПМ и в моменты уничтожения транзактов формируют множество статистик моделирования.

Агрегатный способ формализации производственного участка. Будем использовать при формализации производственного участка алгоритм обработки деталей станками, представленный в виде кусочно-линейных агрегатов, и методику формализованного описания КЛИА с помощью управляющих последовательностей [68]. Схема связей станков обработки деталей при агрегатном способе имитации производственного участка представлена на рис. 2.7.

Агрегат A_0 моделирует внешнюю среду и формирует выходные сигналы в моменты поступления в цех обработки очередной детали i -го типа ($i=1, 2$). Интервалы между поступлениями i -й детали определяются соответствующими функциями распределения $F_{i0}(\tau)$. Предполагаем, что входные потоки деталей каждого типа являются простейшими и интервалы τ определяются интенсивностями поступления деталей извне на обработку λ_i (среднее значение $\bar{\tau}_i = 1/\lambda_i$). Таким образом, с вероятностью $p_1 = \lambda_1/(\lambda_1 + \lambda_2)$ оператором $\rho(p_1)$ вырабатываются номера заявок $i=1$, а с вероятностью $p_2 = 1 - p_1$ оператор $\rho(p_1)$ формирует номера заявок $i=2$.

Агрегаты A_j ($j=\overline{1, 4}$) имитируют обработку деталей соответственно на станках OCT_j . Длительность обработки деталей i -го типа j -м станком определяется соответствующей функцией распределения $F_{ij}(\tau)$. Агрегат A_1 имеет один входной (X_1) и один выходной (Y_1) полюс. Агрегат A_2 имеет один входной X_1 и два выходных полюса Y_1 и Y_2 . Появление сигналов на полюсах Y_1 и Y_2 имитирует передачу деталей соответственно первого и второго типов после их обработки либо на OCT_3 , либо на OCT_4 . Агрегаты A_3 и A_4 выходных полюсов не имеют, что имитирует завершение обработки деталей. Каждый из агрегатов A_j ($j=\overline{1, 4}$) имеет свою очередь заявок, имитирующих поступление деталей к OCT_j на обработку в момент обработки OCT_j предыдущей детали. Выбор заявок из очередей осуществляется по правилу: первым пришел — первым обслуживается. Связь между полюсами агрегатов осуществляется по схеме, представленной на рис. 2.7.

Согласно [10] в матрице R коммутации входов X_1 агрегатов на пересечении строк с номерами агрегатов j и столбцов с номерами входных контактов d располагаются пары чисел (a, b) , указывающих номер агрегата a и номер вы-

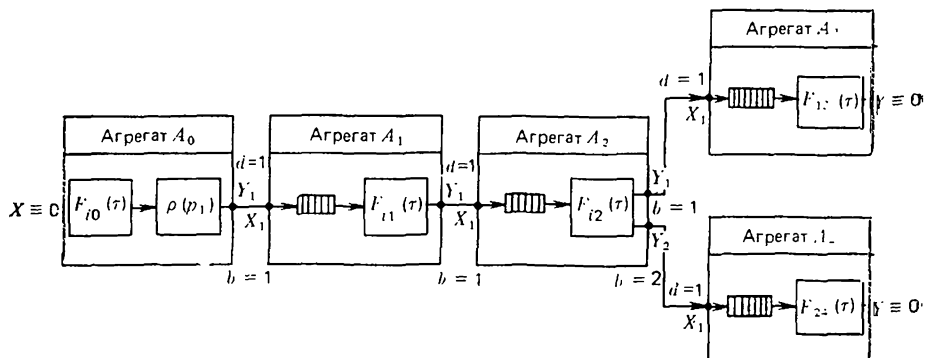


Рис. 2.7. Схема производственного участка при агрегатном способе имитации

ходного контакта b , с которым соединен d -й входной контакт. Коммутация выходов Y_b агрегатов задается матрицей V , в которой на пересечении строк с номерами агрегатов a и столбцов с номерами выходных контактов b располагаются пары чисел (j, d) , указывающих номер агрегата j и номер входного контакта d , на который поступает данный сигнал. Для нашего примера матрицы коммутации полюсов агрегатов R и V будут иметь вид:

$$R = \begin{pmatrix} 0,0 \\ 0,1 \\ 1,1 \\ 2,1 \\ 2,2 \end{pmatrix}; \quad V = \begin{pmatrix} 1,1 & 0,0 \\ 2,1 & 0,0 \\ 3,1 & 4,1 \\ 0,0 & 0,0 \\ 0,0 & 0,0 \end{pmatrix}.$$

Первая строка матрицы R имеет нулевое значение, поскольку на входной полюс агрегата A_0 не поступают сигналы. Вторая строка матрицы R означает, что на вход агрегата A_1 поступает входной сигнал с выходного полюса ($b=1$) агрегата A_0 . Поскольку все агрегаты имеют только по одному входному полюсу, то матрица R состоит только из одного столбца, элементами которого являются пары чисел (a, b) . Максимальное число выходных полюсов у агрегатов модели равно двум, поэтому матрица V состоит из двух столбцов, элементами которых являются пары чисел (j, d) . Первая строка матрицы V означает, что агрегат A_0 с выходного полюса 1 посылает сигналы, которые поступают на входной полюс ($d=1$) агрегата A_1 . Третья строка матрицы V указывает на то, что с выходного полюса 1 сигналы поступают на входной полюс ($d=1$) агрегата A_3 , а с выходного полюса 2 сигналы поступают на входной полюс ($d=1$) агрегата A_4 .

Приступим к описанию алгоритмов функционирования каждого из типов агрегатов. К первому типу агрегатов отнесем A_0 , не имеющий входных сигналов и имитирующий внешнюю среду производственного участка. Ко второму типу отнесем A_j ($j=\overline{1,4}$), имитирующие обработку деталей станками OCT_j .

1. Алгоритм агрегата A_0 состоит из следующих операторов описания:

O_1 . Входные сигналы: $X \equiv 0$.

O_2 . Выходные сигналы: $Y_i = i$, где i — номер типа детали.

O_3 . Множество событий: $E = E'''$, где $E''' = \{e'''\}$; e''' — поступление детали извне в систему (например, со склада).

O_4 . Параметры агрегата:

$$\beta_0 = (F_{10}(\tau); F_{20}(\tau)),$$

где $F_{i0}(\tau)$ — функция распределения интервалов времени между поступлениями деталей i -го типа в систему.

O_5 . Описание состояний агрегата:

$$z_0(t_m) = (W_0(e''', t_m)),$$

где t_m — текущий момент времени; $W_0(e''', t_m)$ — оператор определения момента изменения временной координаты агрегата.

O_6 . Установление начального состояния агрегата:

$$z_0(0) = W_0(e''', 0) = \tau_i,$$

где τ_i формируется по функциям распределения $F_{i0}(\tau)$.

O_7 . Описание операторов переходов и выходов:

$H(e''')$ — оператор перехода агрегата.

Момент очередной активизации агрегата вычисляется оператором

$$W(e'', t_{m+1}) = +t_m \tau_i,$$

где

$$i = \begin{cases} 1 & \text{с вероятностью } p_1, \\ 2 & \text{с вероятностью } 1-p_1. \end{cases}$$

Розыгрыш номера детали i ведется оператором $\rho(p_1)$; $G(e''')$ — оператор выхода $Y_1(t_{m+1}) = i$.

2. Алгоритм агрегата A_j ($j = \overline{1, 4}$) состоит из следующих операторов описания:

О₁. Входные сигналы: $X_1 = i$, где i — тип детали, $i \in \{1, 2\}$.

О₂. Выходные сигналы: $Y_b = \{i\}$, где i — тип детали, $b \in \{0, 1, 2\}$; $b=0$ означает отсутствие выходных сигналов (например, у агрегатов A_3 и A_4).

О₃. Множество событий: $E = E' \cup E''$; $E' = \{e'\}$; $E'' = \{e''\}$, где e' — поступление входного сигнала; e'' — окончание обработки детали.

О₄. Параметры агрегата: $\beta_j = (F_{ij}(\tau))$, где $F_{ij}(\tau)$ — функция распределения длительностей обслуживания деталей i -го типа станком OCT_j .

О₅. Описание состояний агрегата:

$$z_j(t_m) = (n_j(t_m), q_j(t_m), Q_j(t_m), W_j(e'', t_m)),$$

где $n_j(t_m)$ — состояние OCT_j в текущий момент времени t_m ;

$$n_j(t_m) = \begin{cases} 0, & OCT_j \text{ свободен,} \\ 1, & OCT_j \text{ обслуживает деталь 1-го типа,} \\ 2, & OCT_j \text{ обслуживает деталь 2-го типа;} \end{cases}$$

$q_j(t_m)$ — длина очереди на обработку к OCT_j в момент t_m ; $Q_j(t_m)$ — состояние очереди, представляющее собой последовательность номеров типов детали в момент t_m ; $W_j(e'', t_m)$ — оператор изменения временной координаты агрегата.

О₆. Установление начального состояния агрегата:

$$n_j(0) = 0; \quad q_j(0) = 0; \quad Q_j(0) = \emptyset; \quad W_j(e'', 0) = \infty, \quad j = \overline{1, 4},$$

где \emptyset и ∞ означают соответственно пустую очередь и факт того, что момент будущей инициализации агрегата УПМ равен бесконечно большому числу.

О₇. Описание операторов переходов и выходов включает в себя четыре оператора.

Оператор $H(e')$ описывает фазу начала обработки детали (поступление входного сигнала $X_1 = (i)$):

$$n_j(t_{m+1}) = \begin{cases} i, & \text{если } n_j(t_m) = 0, \\ n_j(t_m) & \text{иначе;} \end{cases}$$

$$q_j(t_{m+1}) = \begin{cases} 1 + q_j(t_m), & \text{если станок занят } (n_j(t_m) \neq 0), \\ q_j(t_m) & \text{иначе и состояние очереди не меняется;} \end{cases}$$

$$Q_j(t_{m+1}) = \begin{cases} \text{запись в очередь } (Q_j(t_m), i), & \text{если } n_j(t_m) \neq 0 \text{ по правилу FIFO;} \\ Q_j(t_m), & \text{если } OCT_j \text{ свободен;} \end{cases}$$

$$W_j(e'', t_{m+1}) = \begin{cases} t_m + \tau_{ij}, & \text{если } n_j(t_m) \neq 0, \\ W_j(e'', t_m) & \text{иначе.} \end{cases}$$

Для модификации момента будущей инициализации агрегата j очередное значение τ_{ij} определяется по функции распределения $F_{ij}(\tau)$.

Оператор выходов $G(e'')$ является пустым множеством, поскольку деталь поступает либо на обработку, либо ожидает в очереди.

Оператор $H(e'')$ описывает фазу окончания обработки детали:

$$n_j(t_{m+1}) = \begin{cases} \text{выбор из очереди } (Q_j(t_m), i), & \text{если } q_j(t_m) \neq 0, \\ 0 & \text{иначе;} \end{cases}$$

$$q_j(t_{m+1}) = \begin{cases} q_j(t_m) - 1, & \text{если } q_j(t_m) \neq 0, \\ 0 & \text{иначе;} \end{cases}$$

$$Q_j(t_{m+1}) = \begin{cases} \text{выбор из очереди } (Q_j(t_m), i), & \text{если } q_j(t_m) \neq 0, \\ 0 & \text{иначе;} \end{cases}$$

$$W_j(e'', t_{m+1}) = \begin{cases} t_m + \tau_{ij}, & \text{если } q_j(t_m) \neq 0, \\ \infty & \text{иначе.} \end{cases}$$

Момент будущей инициализации агрегата становится равным бесконечно большой величине при отсутствии деталей в очереди на обработку, что означает моделирование останова станка.

Операторы выходов $G(e'')$ имеют смысл только для агрегатов A_1 и A_2 :

$$Y_1(t_{m+1}) = \begin{cases} n_j(t_m), & \text{если } (j=1 \wedge n_j(t_m) \neq 0) \vee (j=2 \wedge n_j(t_m)=1), \\ 0 & \text{иначе;} \end{cases}$$

$$Y_2(t_{m+1}) = \begin{cases} 2, & \text{если } (n_j(t_m)=2 \wedge j=2), \\ 0 & \text{иначе.} \end{cases}$$

Описание алгоритмов каждого агрегата унифицировано. Группа операторов описания $O_1—O_5$ используется системой моделирования, реализующей агрегатный способ имитации [68], для составления справочной информации о структуре агрегатов для УПМ при последующей имитации обработки деталей станками. Программа задания начальных условий моделирования устанавливает агрегаты в начальные состояния с помощью операторов O_6 . Описание матриц R и V коммутации полюсов агрегатов и группы операторов O_7 позволяет УПМ организовать квазипараллельное функционирование агрегатов в модельном времени t_0 .

Схема взаимодействия УПМ с агрегатами при имитации процессов обработки деталей станками представлена на рис. 2.8. Унифицировано также обращение агрегатов к УПМ. Управление от агрегатов на УПМ переходит при каждом выполнении операторов $W_j(e'', t_{m+1})$, $j=1, 4$; $Y_1(t_{m+1})$ и $Y(t_{m+1})$. Причем при выполнении операторов $W_j(e'', t_{m+1})$ формируется переменная τ , означающая приращение моделируемого времени нахождения агрегата в состоянии e'' до момента t_{m+1} . Передачу выходных сигналов между агрегатами и выбор очередного агрегата для активизации осуществляет УПМ. Для этой цели она использует матрицы коммутации агрегатов V и R , подготовленные исследователем до начала имитации.

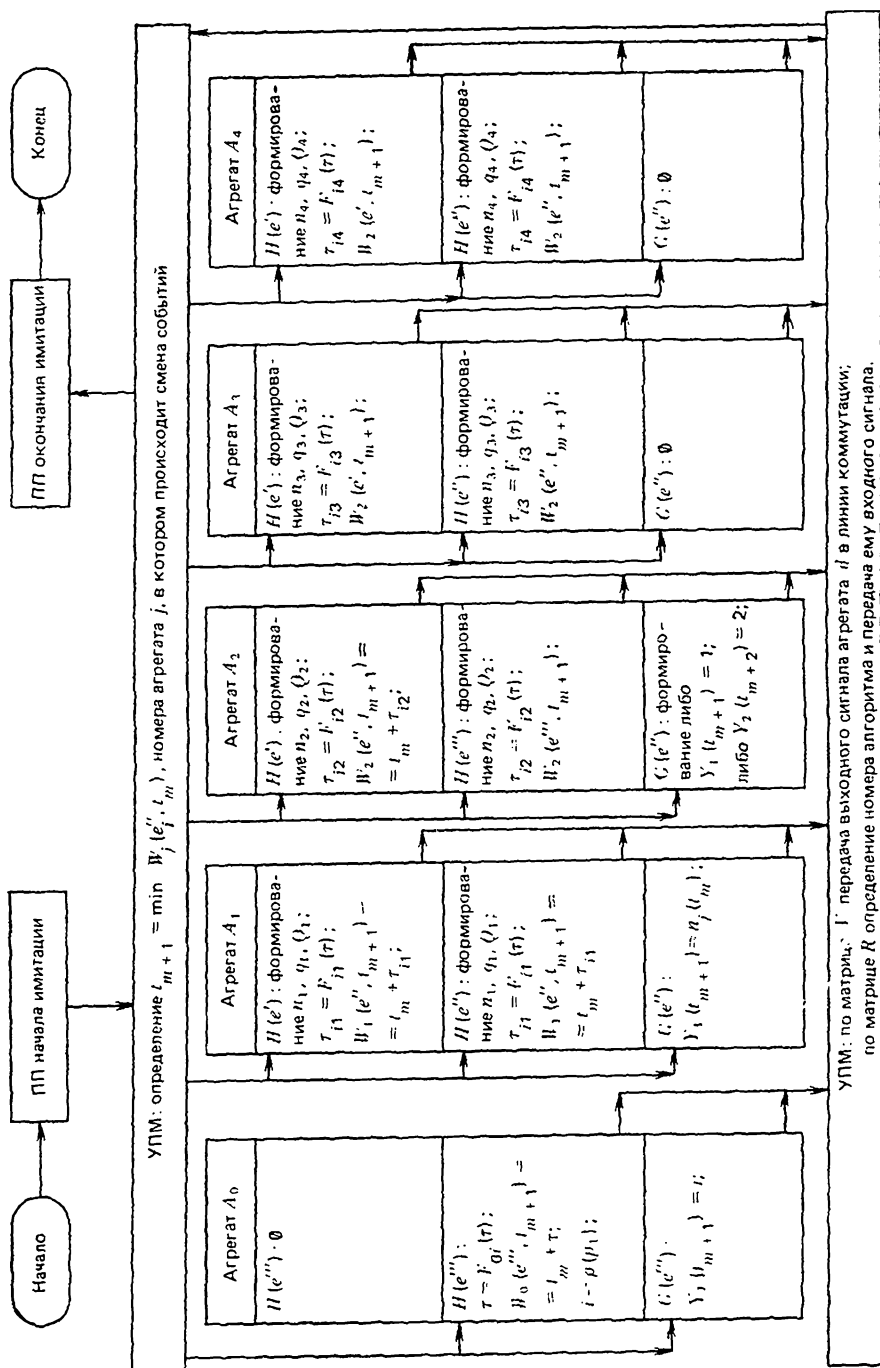


Рис. 2.8. Схема моделирования производственного участка агрегатами

2.4. ПРЕОБРАЗОВАНИЕ ФОРМАЛЬНОГО ЯЗЫКА В ОПИСАНИЕ ИМИТАЦИОННОЙ МОДЕЛИ

На этом этапе исследователь обычно выполняет последовательность действий: обеспечивает внутреннюю и внешнюю синхронизацию компонент модели; синхронизирует моменты появления информации от одной компоненты модели к другой; описывает поведение компонент при появлении конфликтных ситуаций в модели; уточняет состав исходной информации для имитации; составляет описание программы окончания имитации; устанавливает контроль за ходом имитации; уточняет состав документации на моделирование. Объем и содержание работ при выполнении указанной последовательности действий зависят от способа имитации и сложности объекта моделирования. Тем не менее можно говорить об установившейся методике реализации указанных действий, к изложению которой мы переходим.

Внутренняя синхронизация компонент модели. Переход от формального описания к описанию имитационной модели системы также осуществляется на основе декомпозиции сложной системы на составные части. Для каждой компоненты сложной системы кроме указания ее алгоритмических функций необходимо установить временную координату функционирования. Добавление операторов преобразования временной координаты компоненты модели является первым шагом превращения формального описания объекта моделирования в его имитационную модель.

В ряде случаев из-за больших ошибок аппроксимации явлений отдельными активностями для процессного и событийного способов имитации, а также и при имитации активностями может появиться необходимость дополнительного разделения компонент (активностей, подпрограмм обслуживания событий, процессов) на несколько более мелких, которые выполняются с меньшим интервалом τ_{ij} изменения временной координаты. Очевидно, что при этом алгоритмы соответствующих активностей тоже как бы распадаются на несколько частей. Подобного рода работы и составляют основное содержание процедуры установления временных координат функционирования компонент модели.

Для транзактного и агрегатного способов имитации такая процедура является лишней, поскольку она выполнена уже на этапе формализации объекта моделирования, и изменение временных координат агрегатов или блоков модели унифицировано. При этом весьма важно обеспечить внутреннюю и внешнюю синхронизацию компонент модели в модельном времени.

Внутренняя синхронизация компонент обеспечивается самими компонентами с помощью семафоров. Под семафором обычно понимают некоторую глобальную переменную, значение которой либо разрешает, либо запрещает активизацию имитатора компоненты в модельном времени. В случае запрета активизации все компоненты модели выстраиваются в очередь на активизацию согласно их приоритетам. Сброс семафора выполняется одной или не-

сколькими компонентами и приводит к немедленной активизации всех ждущих этого семафора компонент модели. Установка же семафора может немедленно порождать очередь компонент на активизацию.

Семафоры могут устанавливаться с различной целью. Например, при использовании общего ресурса сложной системы какой-либо компонентой выставляется семафор для всех конкурирующих компонент, а после освобождения этого ресурса семафор сбрасывается.

Заметим, что организация внутренней синхронизации компонент модели, например, с помощью семафоров обычно не используется при транзактном и агрегатном способах. Для транзактного способа имитации сами очереди к блокам являются хорошим средством внутренней синхронизации и семафоры не нужны. В случае агрегатного способа имитации внутренняя синхронизация выполнена уже на этапе составления формального описания, когда исследователь либо организует взаимодействие координат агрегата, либо использует управляющие и входные сигналы для имитации взаимодействия агрегатов. Зато при процессном способе имитации для обеспечения внутренней синхронизации требуются зачастую кроме семафоров установка операторов взаимодействия процессов друг с другом и выбор дисциплин обслуживания очередей к процессам для их информационного обмена друг с другом. Заметим, что на данном этапе разработки имитационной модели поведение исследователя определяется выбранным ранее способом формализации объекта моделирования.

Внешняя синхронизация компонент модели. Такая синхронизация чаще всего реализуется системой моделирования, которую выбрал исследователь. Для этой цели используются операторы синхронизации компонент в системе дискретных событий. Например, в случае процессного представления различаем два типа операторов синхронизации: немедленного перевода процесса в ждущее состояние (типа *WAIT* (τ), где τ — длительность задержки процесса в ждущем состоянии) и перевода процесса в ждущее состояние до выполнения некоторого условия *A* (типа *WAIT WHILE(A)*). Состав, синтаксис и семантика операторов синхронизации зависят от используемой системы моделирования. Однако одно существенное свойство этих операторов является общим для любых моделей, построенных на основе процессного представления объекта моделирования. Речь идет о делении этими операторами алгоритма функционирования компонент ИМ на кванты.

Выбор мест установки операторов синхронизации процессов модели является искусством. В основу деления алгоритмов процессов на кванты необходимо положить точностные аспекты имитации явлений и процессов. Подобным же образом исследователь поступает и при имитации активностями или событийным способом имитации. Для транзактного способа имитации проводить внешнюю синхронизацию компонент модели не требуется, поскольку она уже выполнена на этапе формализации, когда осуществля-

лась стыковка источников и поглотителей транзактов с алгоритмом функционирования объекта по схеме модели объекта. Агрегатный способ имитации также не требует организации внешней синхронизации, поскольку эти работы выполнены на этапе формализации и, кроме того, все взаимодействия агрегатов осуществляются через УМП и весьма унифицированно.

Синхронизация моментов появления информации для компонент модели. Важным моментом при построении модели сложной системы является синхронизация появления информации, доступной для ее активных компонент. Иногда своевременное обновление информации, общей для компонент, имеет решающее значение. Например, важно, чтобы результаты выполнения алгоритма функционирования n -го кванта процесса m появились в общем поле информации, доступной для компоненты, а именно в момент t_1 . Появление этих результатов раньше или позже момента t_1 может существенно исказить динамику имитации взаимодействия компонент сложной системы. Чтобы обеспечить такую синхронизацию моментов доступности информации к компонентам модели (например, процессам), часто прибегают к следующему приему. Используют рабочий массив, который доступен только для данного процесса, и оператор условного ожидания момента времени t_1 (`WAIT WHILE (t_1)`). При наступлении момента t_1 происходит перепись рабочего массива в общее поле информации процессов модели.

Аналогично может осуществляться синхронизация моментов доступности информации компонентам модели при имитации активностями или событиями. В более простых случаях для синхронизации моментов доступности информации компонентам модели достаточно использовать глобальные переменные модели. Отметим, что агрегатный и транзактный способы имитации не обеспечивают подобной синхронизации взаимодействия между компонентами модели.

Таким образом, после установки операторов внутренней и внешней синхронизаций алгоритмы компонент модели оказываются разделенными на кванты. Именно в этих местах в ходе имитации и будет происходить обращение к управляющей программе моделирования выбранной системы моделирования.

Отражение конфликтных ситуаций в модели. Следующим шагом при преобразовании формального описания сложной системы в описание имитационной модели является отражение в ней возможных конфликтных ситуаций, с которыми может столкнуться УПМ в ходе имитации. Мы уже рассматривали случай общего использования ресурсов системы различными компонентами модели и применения для этой цели семафоров. Не менее важным моментом при процессном подходе является синхронизация процессов друг с другом с помощью операторов запуска и останова процессов. Почти во всех системах моделирования алгоритмического типа, построенных на принципах процессного представления объекта моделирования, присутствуют операторы запуска и оста-

нова процессов. Очевидно, что в таких случаях в модели могут происходить одновременные запуски процессов. Как правило, отражение конфликтных ситуаций при этом берут на себя системы моделирования и чаще всего они используют системы приоритетов самих процессов. Поэтому на шаге преобразования формального описания объекта в его имитационную модель исследователь уточняет приоритеты компонент модели (активностей, событий, процессов, транзактов, управляющих и входных сигналов агрегатов). При транзактном и процессном способах имитации в случае использования очередей транзактов к блокам или процессов к общим ресурсам системы исследователю приходится изменять дисциплины обслуживания этих очередей.

В ряде моделей сложных систем весьма существенно отражение информационных связей между компонентами в информационной базе модели. В тех случаях, когда обмен информацией между компонентами системы является средством подачи исходных данных для выполнения алгоритмов функционирования компонент модели, составляются модели массивов. Модель массива представляет собой набор характеристик размера массива и параметров, на которые реагируют алгоритмы компонент модели. Если необходима имитация передачи массивов между компонентами модели, то они оформляются в виде «подкрашенных» требований. Подкраской требований является набор характеристик модели массива, хранящийся в информационной базе. По модели в этом случае циркулирует только идентификатор (адрес) модели массива, который выбирается из очереди очередной компонентой модели. По этому идентификатору компоненты модели определяют местоположение информации в модели массива. Затем уже данные, находящиеся в модели по указанному адресу, компоненты модели используют при отражении алгоритма их функционирования.

Уточнение исходной информации для моделирования. Следующим шагом преобразования формального описания в описание имитационной модели является уточнение исходной информации для моделирования. Чаще всего набор действий по заданию исходной информации для компонент модели оформляется в виде блока задания начальных условий имитации. Алгоритм блока задания начальных условий реализуется один раз в начале имитации мгновенно в модельном времени. В ходе задания начальных условий устанавливаются режимы моделирования. Состав действий зависит от способа имитации.

Например, при процессном подходе к имитации для этой цели используется специальный процесс, именуемый инициатором модели. Инициатор модели устанавливает каждый из процессов модели в начальное состояние перед очередным прогоном имитационного эксперимента на ЭВМ. В функции инициатора модели входит также выдача исследователю документальной информации о начальных условиях процесса имитации. В ряде случаев в функции инициатора входят также задание и модификация параметров мо-

делирования. Обычно при реализации инициатора осуществляется замена формальных параметров модели фактическими значениями для данного варианта имитации функционирования большой системы. Поэтому исследователь в алгоритме инициатора должен предусмотреть вопросы подстановки фактических значений на место формальных параметров.

При имитации активностями также выделяются активности-лидеры, т. е. такие активности, в функции которых входит установка остальных активностей в начальное состояние. Для запуска таких служебных активностей обычно выделяется некоторая глобальная переменная модели, указывающая на режим «разгона» модели и изменяемая активностью-лидером по окончании установки остальных активностей в начальное состояние.

Для событийного способа имитации выделяется специальное начальное состояние модели, которое обслуживается соответствующей подпрограммой обслуживания событий. При реализации подпрограммы обслуживания начального события устанавливаются начальные состояния всех компонент модели и назначаются соответствующие условия их свершения.

При транзактном способе имитации устанавливаются начальные значения параметров источников транзактов, состояние очередей, а также занятость блоков обслуживания транзактов, с которых необходимо начать имитацию.

Такой режим работы модели до появления начального состояния и называют «разгоном» имитационной модели. В случае агрегатного способа имитации дело упрощается, поскольку для каждого агрегата описано начальное состояние. Система автоматизации моделирования, реализующая агрегатный способ имитации, реализует операторы задания начальных условий. Причем от исследователя никакая дополнительная работа на данном шаге перехода к имитационной модели не требуется, поскольку он выполнял эту работу уже на этапе формального описания системы.

Организация окончания имитации. Следующим моментом в разработке модели является задание условий окончания имитации очередного варианта моделирования сложной системы в виде специального блока модели, контролирующего ход имитации и называемого финальным. Его алгоритм функционирования отражает последовательность действий по контролю за моментами окончания имитации и начала обработки результатов моделирования. В некоторых случаях в функции финального блока входит также планирование очередного имитационного эксперимента. Иногда контроль за моментом окончания имитации может вестись с заданным шагом модельного времени. В эти моменты финальный блок активизируется и проверяет выполнение условий окончания имитации. При выполнении условий окончания моделирования процесс имитации завершается, модельное время фиксируется и в дальнейшем алгоритм финального блока или процесса реализуется мгновенно в модельном времени.

Контроль за ходом имитации. В некоторых случаях исследо-

вателю необходимо следить за ходом имитации, иметь возможность принять решение о прекращении имитации. При процессном способе имитации для этой цели необходимо предусмотреть блок или процесс контроля за ходом имитации. Алгоритм блока или процесса контроля зависит от состава и структуры модели. В нем предусматривается его активизация с заданным шагом. В ходе активизации формируется осведомительная информация о состоянии модели, которая выдается либо на печать, либо на экран терминального устройства.

В тех случаях, когда предполагается длительная работа имитационной модели на ЭВМ, в функции блока контроля может входить копирование во внешней памяти ЭВМ состояния модели. Это обеспечивает возможность возобновления работы модели с прерванного места при появлении отказов в ЭВМ. Для этой цели используются либо стандартные средства математического обеспечения ЭВМ, на которой реализуется модель, либо специальные средства возобновления имитации, предоставляемые системой моделирования. Подобные же действия должен предусмотреть исследователь при имитации активностями или подпрограммами обслуживания событий. В этом случае разрабатываются процедуры проверки выполнимости условий наступления контролируемой ситуации в модели и фиксации экспресс-информации о ходе имитации.

Для транзактного способа имитации оперативный контроль за ходом имитации организовать довольно трудно, да он в большинстве случаев и не нужен. При агрегатном способе имитации исследователь должен заранее побеспокоиться о контроле за поведением агрегатов с помощью контрольных координат изменения состояний агрегатов. Сама организация квазипараллельной работы агрегатов стандартизована.

Организация сбора статистики. Завершается преобразование формального описания в описание имитационной модели выбором мест установки операторов сбора статистики. Выбор места фиксации поведения объекта моделирования также является искусством исследователя. Существует множество способов фиксации статистики моделирования. Прежде всего это применение так называемых «наблюдателей» модели. Трассировка алгоритмов компонент модели с помощью наблюдателей позволяет отслеживать как развитие их функций во времени, так и их взаимодействие друг с другом. Каждый такой оператор-наблюдатель изменяет значения некоторого счетчика моментов своей активизации и записывает их во внешнюю память. Эта информация используется затем финальным блоком при обработке статистики моделирования. Иногда важно фиксировать времена жизни процесса в модели. Например, при процессном представлении для этой цели фиксируются момент создания (или первой активизации) процесса и момент его уничтожения (или последней пассивизации процесса). Разница между этими моментами и представляет собой время жизни объекта.

В некоторых системах моделирования, например в МК PLSIM [51], в распоряжение исследователя предоставляется набор операторов сбора статистики. Эти операторы исследователь помещает в выбранные места в алгоритмах процессов, разрывая дополнительно временные интервалы функционирования процессов, обозначенные операторами синхронизации.

Ряд систем моделирования [56, 59] не требует от исследователя установки операторов сбора статистики, обеспечивая ему сбор стандартной статистики о функционировании компонент модели. Чаще всего в состав такой статистики входят следующие характеристики: число взаимодействий компонент модели, общее время и коэффициент использования компоненты модели, число отказов функционирования компоненты системы и время, в течение которого она находилась в аварийном состоянии.

В ряде случаев стандартной статистики функционирования процессов исследователю недостаточно. Мало информации дает также и трассировка модели с помощью наблюдателей. Тогда при процессном и событийном способах формализации, а также при имитации активностями исследователю приходится заводить специальные процессы, события, активности, фиксирующие статистическую информацию о моделируемом объекте. Такие специальные процессы (события или активности) могут активизироваться либо с заданным шагом модельного времени, либо при выполнении логических условий в ходе имитационного эксперимента на ЭВМ. В обоих случаях исследователь должен предусмотреть в алгоритме финального процесса набор процедур для вычисления статистик моделирования и на их основе определить отклики имитационной модели согласно выбранным критериям качества вариантов моделирования большой системы. Для агрегатного и транзактного способов сбор статистики стандартизован и данный шаг перехода от формального описания к описанию имитационной модели отсутствует.

Документирование. Завершается этап перехода к описанию имитационной модели уточнением состава документации на модель. Результатом работы исследователя на данном этапе является описание имитационной модели, готовое к программированию на выбранном языке моделирования. Для событийного и процессного способов имитации, а также при использовании активностей размер имитационной модели существенно увеличивается по сравнению с формальным описанием за счет введения дополнительных блоков и операторов моделирования. При агрегатном и транзактном способах имитации этот этап практически вырождается и имитационная модель мало чем отличается от формального описания. В этом одно из достоинств этих методов и причина их большой оперативности в разработке имитационной модели.

Пример. В качестве иллюстрации изложенной выше методики произведем переход от формального описания рассматриваемого нами производствен-

ного участка к описанию имитационной модели при каждом из способов имитации.

Чтобы преобразовать формальное описание производственного участка в имитационную модель *аппаратом активностей*, в качестве временной координаты активностей используем значения моментов окончания обслуживания деталей станками t_{ik} и модельное время t_0 (см. рис. 2.3). Когда выполняется условие $t_{ik} \leq t_0$, то происходит инициализация активностей. Внутренняя синхронизация активностей для нашего примера не требуется. В качестве внешних операторов синхронизации используются операторы возврата на УПМ.

Ряд систем моделирования позволяет объединить оператор формирования значения t_{ik} с оператором возврата на УПМ для создания одного оператора синхронизации активностей. Выбирать место установки операторов синхронизации не приходится, поскольку любая программа, реализующая активности, заканчивается этими операторами синхронизации. Вид же операторов синхронизации зависит от используемого языка моделирования. Модели массивов в данном примере не используются.

Конфликтные ситуации в модели разрешаются УПМ при проверке условий инициализации активностей. Поскольку активности взаимодействуют друг с другом посредством глобальных переменных, то отражением конфликтных ситуаций (инициализации активностей в одно и то же модельное время) занимается УПМ. Информационное взаимодействие между активностями не требуется. Для указания момента или условия окончания моделирования используется специальный оператор в составе алгоритма начальной активности. Перед каждым изменением модельного времени t_0 УПМ проверяет выполнимость этого условия и в случае его удовлетворения передает управление на выполнение финальной активности. В алгоритме финальной активности осуществляется обработка статистики моделирования. Использование активностей-контролеров в данной имитационной модели излишне.

Завершается перевод формального описания в модель выбором мест установки операторов сбора статистики $STATIST_{ij}$ (i, j — индексация операторов соответственно по порядку выполнения активностей и по номерам OCT_j). В данном примере алгоритм активностей позволяет установить операторы $STATIST_{ij}$ в начале его выполнения (операторы B_1 , H_1 и K_1 на рис. 2.3). При этом операторы $STATIST_{1j}$ фиксируют моменты начала работы $OCT_j(t_{n_{ijh}})$ и длительности простоя $OCT_{\tau_{ij}}$, а операторы $STATIST_{2j}$ определяют моменты окончания работы $OCT_j(t_{k_{ijh}})$ и длительность работы $OCT_{\tau_{ij}}$. Таким образом, за каждой активностью закреплены две глобальные статистики; $t_{n_{ijh}}$ и $t_{k_{ijh}}$. По этим статистикам операторы сбора статистики формируют множества значений $\{\tau_{ij}\}$ и $\{\tau_{ij}\}$. Накоплением этой статистики обычно занимается УПМ. При выполнении условия окончания имитации с помощью этих статистик финальная активность вычисляет по формулам (2.1) и (2.2) значения целевых функций для данного варианта моделирования.

Преобразование формального описания производственного участка в имитационную модель *аппаратом событий* осуществляется аналогично. В качестве временной координаты событий УПМ служат значения t_{jk} и модельное время t_0 (см. рис. 2.4). В моменты, когда $t_{jk} \leq t_0$, происходит инициализация программ, обслуживающих эти события. Внутренняя синхронизация событий отсутствует. Внешними операторами синхронизации являются операторы возврата на УПМ, которые могут быть объединены с операторами формирования

значений t_{jk} . Устанавливаются операторы синхронизации в конце подпрограмм, обслуживающих события. Вид операторов синхронизации зависит от используемого языка моделирования.

Модели массивов не нужны, поскольку программы обслуживания событий взаимодействуют друг с другом посредством глобальных переменных. По этой же причине отражением конфликтных ситуаций в системе занимается УПМ при анализе условий запуска подпрограмм обслуживания событий. Информационное взаимодействие между программами, обслуживающими события, не требуется.

Программа, реализующая исходное событие t_0 (см. рис. 2.2), запускает в работу OCT_1 . Эта же программа с помощью специального оператора устанавливает в начальное состояние указатель выполнения условий окончания или задает интервал времени имитации T_k . Перед каждым изменением модельного времени t_0 УПМ проверяет выполнимость этого условия и в случае его удовлетворения передает управление на выполнение программы, обслуживающей финальное событие. В ней осуществляется обработка статистики моделирования. Использование программ обслуживания контрольных событий в данной имитационной модели излишне.

Установка операторов сбора статистики $STATIST_{ij}$ производится аналогично предыдущему случаю. В программах, обслуживающих события с нечетным номером, определяется длительность простоев OCT_j (τ_{pij}), а программы с четными номерами событий вычисляют длительность работы OCT_j (τ_{pij}). Вычисление текущих значений τ_{pij} и τ_{pij} также осуществляется на основе глобальных статистик t_{pijh} и t_{kijh} , которые закреплены за каждой программой обслуживания событий. Накоплением статистики в виде множеств $\{\tau_{pij}\}$ и $\{\tau_{pij}\}$ занимается УПМ. При выполнении условий окончания имитации с помощью этих статистик программа обработки финального события по формулам (2.1) и (2.2) вычисляет значения целевых функций для данного варианта моделирования.

При процессном представлении формальное описание производственного участка преобразуется в имитационную модель следующим образом. Изменением значений временной координаты процессов t_{jk} занимается УПМ. Внутренняя синхронизация взаимодействия процессов осуществляется с помощью атрибута i в заявках на обслуживание, посылаемых процессом OCT_1 в очередь $OЧ_1$ к процессу OCT_2 , и выбранной дисциплиной обслуживания очереди $OЧ_1$ процессом OCT_2 . Для внешней синхронизации процессов служат операторы возврата на УПМ. Причем вместо последних используем операторы ожидания типа $WAIT$ (τ_{ik}), чем объединим два оператора: формирования t_{jk} и возврата на УПМ. Для информационного взаимодействия между процессами используются заявка и очередь к процессу OCT_2 . Заявки «подкрашиваются» и состоят из атрибута, означающего номер детали i .

Алгоритм задания начальных условий указывает начальные значения в формате заявки у процессов, имитирующих поступление деталей извне, а также запускает в работу процесс OCT_1 . Кроме того, в процессе задания начальных условий с помощью специального оператора, структура которого зависит от выбранного языка моделирования, указывается время окончания моделирования, которое запоминается УПМ. Перед каждым изменением модельного времени t_0 УПМ проверяет выполнимость этого условия и при его удовлетворе-

нии передает управление на процесс окончания моделирования. В алгоритме этого процесса осуществляется обработка статистики моделирования.

Использование процессов-контроллеров в данной имитационной модели излишне. Завершается перевод формального описания в модель установкой операторов сбора статистики $STATIST_{ij}$ (j — номер процесса-имитатора OCT_i — номер активности процесса). С помощью операторов $STATIST_{1j}$ определяется длительность простоев $OCT_j \tau_{pij}$, а операторы, $STATIST_{2j}$ вычисляю длительности работы процессов $OCT_j \tau_{pij}$. Вычисление текущих значений τ_{pij} и τ_{pij} также осуществляется на основе глобальных статистик t_{kijh} и t_{kijh} процессов OCT_j . Накоплением статистики в виде множеств $\{\tau_{pij}\}$ и $\{\tau_{pij}\}$ занимается УПМ. При выполнении условий окончания имитации с помощью этих статистик по формулам (2.1) и (2.2) вычисляются значения целевых функций для данного варианта моделирования. В заключение отметим, что в данном примере из-за простоты модели не используются ни процесс-контроллер, ни наблюдатели.

При транзактном подходе временная координата моделирования t_{ik} устанавливается для каждой заявки, УПМ передвигает транзакты от блока к блоку (см. рис. 2.6) и при их освобождении блоком меняет у транзактов значения t_{ik} . Внутренняя синхронизация транзактов осуществляется с помощью очередей и установки дисциплин обслуживания этих очередей. Внешняя синхронизация при транзактном подходе отсутствует, поскольку вся модель представляет собой единую программу, взаимодействующую с УПМ при выполнении каждого блока модели. Конфликтные ситуации возникают при необходимости одновременного обслуживания разными блоками своих транзактов. Эти ситуации разрешаются УПМ на основе реализованного в ней алгоритма обслуживания одновременно активизируемых транзактов.

Информационные связи между блоками осуществляются посредством транзактов. Транзакты состоят из одного атрибута (номера детали i). Блоки изменения маршрута используют значения этих атрибутов в алгоритме своей работы. Атрибут i формируется блоками 11 и 12 в момент создания транзактов. Задание начальных условий (длительность моделирования T_n) осуществляется специальным оператором, выполняющимся в начале моделирования. По этому оператору УПМ запоминает значение T_n . При каждом изменении модельного времени проверяется условие $t_0 \geq T_n$. При его выполнении процесс имитации завершается и управление передается по адресу, указанному оператором задания времени моделирования. Вид этих операторов зависит от языка моделирования, реализующего транзактный способ имитации. В конце описания модели следует набор операторов обработки статистики.

От установки специальных обслуживающих блоков, выполняющих функции сбора статистики, мы в данном примере отказались, поскольку УПМ, как правило, фиксирует следующую стандартную статистику: время жизни транзакта (от момента рождения до момента уничтожения транзакта), коэффициенты использования транзактами блоков, характеристики очередей (средняя и максимальная длина очереди, число транзактов, побывавших в очереди, среднее время пребывания транзакта в очереди). Именно простота освоения обусловила широкое применение транзактного способа имитации, несмотря на ряд его очевидных недостатков (затраты ресурсов машинного времени и памяти ЭВМ на организацию имитации; множество дополнительных блоков, не имеющих прямого отношения к исследуемому объекту).

Преобразование формального описания в имитационную модель при *агрегатном представлении* не требуется, поскольку формальное описание объекта в виде совокупности взаимосвязанных КЛА и есть имитационная модель.

2.5. ПРОГРАММИРОВАНИЕ И ОТЛАДКА ИМИТАЦИОННОЙ МОДЕЛИ

После того как построено описание имитационной модели сложной системы, можно приступать к его программной реализации. В процессе этой работы необходимо: выбрать вычислительные средства автоматизации моделирования, проверить достоверность схемы программы модели, осуществить кодировку программы модели, выполнить автономную отладку компонент программы модели, провести комплексную отладку программы модели, составить окончательный вариант документации на модель.

Выбор вычислительных средств. В основу выбора можно положить требуемые затраты ресурсов ЭВМ на построение программы модели и организацию модельного эксперимента на ЭВМ. Следует учитывать как размеры оперативной памяти ЭВМ, требующейся для реализации модели, так и машинное время, необходимое для трансляции и моделирования очередного варианта сложной системы. Немаловажную роль играют простота доступа исследователя к данной ЭВМ и стоимость проведения на ней экспериментов. Однако определяющими факторами при выборе ЭВМ для реализации программы модели являются наличие на ней средств автоматизации постановки модельного эксперимента и их качество. Учет возможностей средств автоматизации моделирования весьма существен, поскольку в общем объеме работ по созданию программы модели сложной системы ее отладка составляет значительную долю.

Выбор средств автоматизации моделирования. В принципе модель можно реализовать на любом универсальном алгоритмическом языке. Однако для облегчения программирования и отладки ИМ больших систем используют специальные языки моделирования. Языки моделирования различаются: методами организации моделирования, наименованиями и структурой блоков модели, способами проверки операций и условий взаимодействия элементов моделирования, видами статистических испытаний, которым можно подвергнуть имеющиеся данные, легкостью изменения структуры модели. Однако их объединяет общее свойство — близость к языку специалистов-аналитиков. Использование языков моделирования облегчает написание программы модели, обеспечивает более строгое следование выбранной концепции построения модели, помогает четко классифицировать элементы системы, обеспечивает гибкость программы (возможности внесения в нее изменений), позволяет различать элементы одного класса по их характеристикам и свойствам, описывает связи между элементами системы и внешней средой, позволяет корректировать число эле-

ментов модели в соответствии с изменениями внутренних условий модели.

Эти возможности обуславливаются специфическими свойствами языков моделирования, отличающими их от универсальных алгоритмических языков программирования на ЭВМ. Среди таких свойств необходимо отметить способности: генерировать случайные числа, продвигать модельное время, накапливать выходные данные моделирования, проводить статистический анализ накапливаемых данных, распределять выходные данные по заранее заданным форматам, выявлять и регистрировать логические несоответствия и другие ситуации, связанные с ошибками в модели.

Кроме того, для организации квазипараллельного выполнения процессов при моделировании языки моделирования позволяют производить: определение типа события после извлечения его из списка событий; вызов стандартных программ для приведения переменных состояний в соответствие с определенным событием; идентификацию конкретных состояний; хранение и извлечение данных из списков, таблиц или массивов, включая список событий и списки состояний процессов модели. На языке моделирования исследователь может четко и ясно описывать модельные понятия.

Существенным достоинством некоторых языков моделирования является наличие у них возможностей использования стандартных блоков организации моделирования. В ряде средств автоматизации моделирования [51] имеются библиотеки специальных процессов, с помощью которых можно формировать блоки задания начальных условий. Они представляют собой программные заготовки на базовом языке моделирования, которые настройкой параметров можно преобразовать к требованиям модели сложной системы. Использование таких языков моделирования значительно облегчает работу исследователя по реализации в составе модели стандартных блоков моделирования.

Проверка достоверности схемы модели. Методика проверки в общем виде рассматривается в п. 1.5.2. Остановимся на некоторых моментах, не освещенных выше.

При имитации активностями необходимо проверить, правильно ли схема модели отражает взаимодействие активностей друг с другом. Весьма существенно проверить также места модификации глобальных переменных модели, влияющих на свершение условий инициализации соответствующих активностей.

Для событийного способа имитации исследователь должен проверить, не потерял ли физический смысл объекта имитации о раздробления алгоритмов, имитирующих функциональные действия, на множество подпрограмм обслуживания событий.

Для транзактного способа проверка достоверности схемы особенно существенна. Может оказаться, что языки моделирования реализующие транзактный способ имитации, существенно искажают динамику имитации, и схема программы модели отличается от схемы концептуальной модели объекта моделирования.

При процессном способе имитации исследователь имеет дело с множеством схем соответственно числу типов процессов. Возможны два вида реализации программ процессов: с циклической и древовидной структурами. В первом случае алгоритм процесса замкнут с конца на начало и представляет собой чередование алгоритмических операторов с операторами синхронизации. Во втором случае по окончании выполнения всей цепочки операторов синхронизации процесс переходит в пассивное состояние.

В зависимости от структуры процессов проверка достоверности схемы программы модели также существенно различается. Для процессов с циклическим алгоритмом выполнения положительный исход проверки достоверности схемы каждого отдельного процесса не гарантирует достоверности всей модели. Задача усложняется, если процессы взаимодействуют друг с другом с помощью операторов запуска и останова процессов. Во втором случае, просмотрев процессы поочередно, исследователь может быть уверен в том, что программа модели соответствует замыслу. В обоих случаях при проверке схемы программы модели зачастую используются диаграммы развития процессов в модельном времени.

Отметим, что проверке по схеме не поддаются вопросы информационной стыковки модели. И совсем уж плохо обстоит дело с «прокруткой» программы модели по схеме для случая, когда алгоритм поведения зависит от процессов, с которыми он находится в информационном взаимодействии. В этом случае решение проблем переносится на этап комплексной отладки имитационной модели.

Кодировка программы модели. Следующим шагом этапа программирования имитационных моделей является собственно кодировка программы модели на выбранном языке моделирования. Наиболее четко влияние кодировки на создание модели проявляется при агрегатном и транзактном способах имитации. В обоих случаях дело сводится к использованию унифицированных операторов при реализации текстов формального описания объекта имитации. Для остальных способов имитации кодировка алгоритмов осуществляется средствами базового языка моделирования, а взаимодействия и синхронизация компонент имитационной модели объекта моделирования кодируются соответствующими операторами выбранного языка моделирования.

Автономная отладка программы имитационной модели. В принципе автономная отладка программы имитационной модели сложной системы распадается на множество отладок отдельных подпрограмм. Отладка каждой компоненты модели может вестись по методикам отладки отдельных процедур. Однако необходимо отметить ряд трудностей при автономной отладке компонент программы имитационных моделей сложных систем. Это прежде всего необходимость моделирования входа для каждой компоненты во времени в случае, когда управляющая информация поступает распределенно по времени. В тех случаях, когда в компонентах модели используются стохастические элемен-

ты, исследователю при отладке приходится затрачивать дополнительные усилия на запись во внешнюю память последовательности начальных значений генераторов псевдослучайных чисел с тем, чтобы обеспечить повторяемость стохастических характеристик при отладке различных ветвей алгоритма компоненты модели.

При автономной отладке подпрограмм обслуживания активностей или событий целесообразно составлять цепочки взаимодействующих друг с другом активностей или событий, что позволит согласовать их функционирование в модельном времени. Вторым, не менее важным аспектом автономной отладки подпрограмм обслуживания активностей или событий является согласование информационного обмена между ними через глобальные переменные или массивы.

В процессе автономной отладки процессов возникают трудности, обусловленные привлечением к отладке большого числа программистов. Все эти программисты в целях ускорения работы отладку «своих» процессов производят в параллель, используя мультипрограммный режим работы ЭВМ. При отладке возникает необходимость в изменении «своих» процессов, их рабочих массивов. Это зачастую приводит к дополнительной работе по приведению в соответствие отдельных «кусков» модели сложной системы. Эта трудность часто усугубляется неопытностью отдельных программистов.

Для ускорения отладки процессов, которые должны взаимодействовать с другими процессами (а последние к данному времени также еще не отлажены), создаются простейшие макрокоманды на языке моделирования, моделирующие это взаимодействие.

Следует также заметить, что некоторые ошибки в процессах модели проявляются непостоянно по месту и времени, что затрудняет их поиск и устранение.

При автономной отладке модели, реализованной транзактным способом имитации, организуются пробные запуски всей модели и начинается отладка синтаксиса программы модели. Затем приступают к проверке правильности алгоритмов обслуживания транзактов по ветвям общей схемы модели.

Для агрегатного способа имитации основная трудность автономной отладки состоит в проверке правильности алгоритмов перевода агрегата из состояния в состояние и взаимодействия с УПМ. Обычно при отладке используется принцип наращивания выполняемых агрегатом функций.

Важным моментом в организации автономной отладки компонент программы имитационной модели является возможность использования дополнительных средств отладки. Для отладки подпрограмм обслуживания активностей или событий зачастую применяется набор служебных подпрограмм обслуживания активностей или событий, предназначенных для имитации внешнего окружения отлаживаемого алгоритма активности или подпрограммы обслуживания событий. При агрегатном и транзактном способах

имитации дополнительных средств отладки не требуется из-за высокой степени стандартизации взаимодействия блоков или агрегатов с УПМ.

Для организации отладки процессов обычно служит стандартный набор средств отладки, представляемый выбранной ЭВМ. В него входят: различные загрузчики рабочих программ, библиотечные и поддерживающие программы, генераторы данных, программы управления отладкой, средства автономного редактирования данных и программ. Различают два вида загрузчиков: в оперативную память и во внешнюю память. Основное их назначение состоит в помещении тела модели и рабочих массивов соответственно либо в оперативную, либо во внешнюю память ЭВМ. Библиотеки и поддерживающие программы имеют средства пополнения и исключения любого процесса, а также средства поддержания массивов в заданном состоянии после окончания имитации очередной ветви процесса.

Весьма существенно для разработчика программы модели иметь в используемом языке моделирования средства обнаружения ошибок. Сюда входят: программы распечатки состояний процессов, операторы трассировки алгоритмов процессов, средства мгновенной фотографии развития процесса во внешней памяти ЭВМ.

Программы управления отладкой обеспечивают исследователю три вида услуг: введение вставок в алгоритм процесса; управление вводом и выводом информации для данного процесса; включение отладочных средств, разработанных разработчиком модели.

Средства автономного редактирования данных и программ позволяют пользователю: выводить на печать информацию с магнитной ленты, приводить к необходимому формату и сортировать данные; распечатывать последовательности макрокоманд; осуществлять автономную сортировку исходной информации; редактировать на магнитной ленте массивы с сортировкой входных и выходных данных модели.

Зачастую эффективным средством при отладке программы имитационной модели с древовидной структурой является трассировка алгоритма компонент модели. Возможны три вида организации трассировки компонент модели: каждой команды, фиксации переходов в алгоритмах компонент, регистрации выполнения только макрокоманд.

Вообще автономная отладка компонент модели требует двойного объема работ программистов по сравнению с отладкой процедур обычных рабочих программ. Увеличение объема работ обусловливается необходимостью обеспечить воздействие «внешней среды» компонент модели на их поведение. В состав «внешней среды» компоненты модели входят имитаторы поведения других компонент и, возможно, блоки, имитирующие внешнее окружение моделируемой сложной системы.

Комплексная отладка программ модели. Отладив каждую компоненту модели в отдельности, исследователь объединяет ком-

поненты в единую модель. При этом возникают ошибки из-за несоответствия в динамике функционирования отдельных компонент и наличия неточностей в информации взаимодействия между компонентами.

При имитации процессами, активностями и событиями исследователь вначале упрощает взаимосвязи между двумя процессами (активностями, программами обслуживания событий) по информации (ставит соответствующие «заглушки» в алгоритмы процессов) и лишь после этого переходит к отладке их взаимодействия. Начинается эта отладка по операторам пуска и останова компонент модели. Затем отлаживается информационное взаимодействие между двумя процессами (активностями, программами обслуживания событий). К двум отлаженным компонентам добавляется третья с упрощенными связями по информации (заглушками в алгоритме третьей компоненты). Отлаживают сначала взаимодействие между тремя компонентами по операторам пуска и останова этих компонент, а затем переходят к отладке информационных взаимодействий между тремя компонентами (устраняются заглушки в их алгоритмах). При этом широко используются временные диаграммы функционирования компонент модели. В компоненты модели обычно вставляются операторы выдачи содержимого таблиц процессов (активностей, подпрограмм обслуживания событий) и результатов отладки, представляемые системой моделирования.

По окончании проверки соответствия информации алгоритму поведения компонент модели эти операторы отладки исключаются из тела компонент модели. Существенно помогает при этом наличие библиотеки процедур у системы моделирования и средств оперативной компоновки программы модели из библиотеки процедур. Развитые средства автоматизации моделирования, как правило, предоставляют в распоряжение разработчика программы модели сложной системы широкий спектр диагностических сообщений об ошибках динамики имитации. Это значительно облегчает исследователю локализацию динамических ошибок в компонентах модели. Таким способом модель наращивается по методу «этап за этапом». Начиная с отладки взаимодействия двух компонент, исследователь затем наращивает программу модели до полного состава. При каждом добавлении новой компоненты модели необходимо проверить взаимодействие ее с внешней средой. Таким образом, разработчики фактически создают последовательность вложенных программных моделей и опыт отладки одной модели используется для улучшения последующей модели.

При транзактном способе имитации комплексная отладка модели сводится к отладке семантики обслуживания транзактов по схеме имитационной модели. Например, зачастую для успешной отладки имитационной модели достаточно отладить синтаксис операторов языка, реализующего транзактный способ. Поэтому-то и язык моделирования GPSS [82], реализующий транзактный

способ имитации, получил широкое распространение. Простота использования и высокие технологические возможности (прежде всего в программировании и отладке) для многих исследователей делают GPSS (а с ним и способ имитации) предпочтительным перед другими языками моделирования, несмотря на ограниченные его описательные возможности и слабые средства обработки статистики и планирования экспериментов.

В случае агрегатного способа имитации комплексная отладка модели ограничивается согласованием взаимодействия агрегатов по результатам пробного моделирования.

Вообще отладка программы модели имеет много общего с отладкой больших программ реального времени. Поэтому при комплексной отладке программ ИМ можно использовать методики отладки программ реального времени.

Документация программы модели. Завершается создание программы модели сложной системы составлением программной документации на модель. Традиционные методы составления программной документации не подходят для случая проектирования моделей сложных систем. Необходимы более точные и подробные спецификации и описания модели. Обычно при создании рабочих программ моделей сложных систем документация создается после написания и отладки программ компонент модели. При разработке программ моделей больших систем документация не должна рассматриваться как нечто дополнительное, что можно сделать после завершения основной работы по составлению программ. В данном случае документация представляет собой инструмент, без которого разработка комплекса взаимосвязанных программ станет чрезвычайно затруднительной.

Одной из трудностей при разработке компонент модели является частое изменение самого логического построения процедуры, реализующей реальный процесс. При отладке компонент модели исследователь обнаруживает непредвиденные трудности. Он постоянно изменяет процедуры. Однако, учитывая ограничения, налагаемые из-за того, что его работа должна сопрягаться с работой разработчиков других компонент модели, он должен подходить к внесению изменений очень осторожно. Согласно Р. Шеннону [81] разработчик не должен изменять структуру информации, используемой многими разработчиками без консультации с ними. Поэтому все вносимые изменения должны тщательно документироваться. В противном случае разработка общей системы выйдет из-под контроля, и одни разработчики не будут знать, что делают другие.

В условиях постоянных изменений надо установить жесткий контроль за состоянием и ведением документации, иначе нет никакой надежды на то, что в момент объединения компонент в модель они будут соответствовать друг другу. Именно поэтому необходимо так проработать документацию, чтобы были обеспечены возможность введения существенных изменений и обязательное ознакомление с ними всех программистов, которые должны о

них знать. Как следует из [81], при создании программы модели большой системы поток документации может стать таким большим, что возникает необходимость в принятии специальных усилий к сокращению изменений до минимума после начала отладки компонент модели.

При комплексной отладке осуществляется объединение множества компонент модели. При этом, естественно, могут возникнуть ошибки, которые часто трудно проследить и исправить. Наличие упорядоченной документации может существенно помочь в этом. Уровень детализации документации зависит от сложности разрабатываемой программы модели. Для модели сложной системы принято составлять подробные и тщательно проработанные спецификации компонент модели. Согласно [81] каждой спецификации компоненты должен присваиваться кодовый номер, который точно отражает тип спецификации. Этот номер, в частности, содержит номера всех других спецификаций, связанных с данной компонентой, чтобы можно было проследить внесение изменений в других компонентах. Важно в составе разработчиков программы модели иметь «библиотекаря», который будет контролировать состояние документации у остальных разработчиков и регистрировать все вносимые изменения, появляющиеся по мере развития модели системы. Для регистрации этих постоянных изменений необходимо предусмотреть средства ведения соответствующих архивов. Напомним, что нужно стремиться к тому, чтобы поток информации об изменениях в модели был минимальным, так как иначе разработчики моделей отдельных компонент системы могут потонуть в бумагах и некогда станет отлаживать компоненты модели.

Не будем приводить описание программы модели для нашего примера моделирования участка производства. Для специалиста, владеющего языком моделирования, реализующим соответствующий способ имитации, программирование и отладка модели не представляют трудностей. Заметим лишь, что в этой работе многое зависит от степени работоспособности версии системы автоматизации моделирования, имеющейся в распоряжении исследователя. В данном случае (как, впрочем, и в любом другом) можно рекомендовать использовать для программирования наиболее проверенные и технологичные средства моделирования. Иначе исследователю придется тратить больше времени на отладку самой системы моделирования, чем на отладку программы модели.

Глава 3

АНАЛИЗ СУЩЕСТВУЮЩИХ СРЕДСТВ АВТОМАТИЗАЦИИ МОДЕЛИРОВАНИЯ

3.1. НАЗНАЧЕНИЕ И ВИДЫ ЯЗЫКОВ МОДЕЛИРОВАНИЯ

3.1.1. ПРЕИМУЩЕСТВА ЯЗЫКОВ МОДЕЛИРОВАНИЯ

Для того чтобы реализовать на ЭВМ модель сложной системы, нужен аппарат моделирования, который в большинстве существующих универсальных алгоритмических средств программирования отсутствует или недостаточно развит. В нем должны быть предусмотрены:

- способы организации данных, обеспечивающие простое и эффективное моделирование;

- удобные средства формализации и воспроизведения динамических свойств моделируемой системы;

- возможности имитации стохастических систем, т. е. процедуры генерирования и анализа случайных величин и временных рядов.

Такой аппарат должен иметь специализированные средства автоматизации моделирования, в состав которых обычно входят: язык описания объекта моделирования, средства обработки языковых конструкций (компилятор, транслятор или интерпретатор); УПМ, осуществляющая имитацию во времени, и набор стандартных программных средств, реализующих дополнительные возможности по организации модельных экспериментов на ЭВМ.

Применение универсальных языков программирования в имитационном моделировании позволяет исследователю достигнуть гибкости при разработке, отладке и испытании модели. Однако при этом затрачиваются большие усилия на программирование модели. Поэтому целесообразнее применять специализированные средства моделирования, которые обладают следующими преимуществами перед универсальными: меньшими затратами времени на программирование; более эффективными методами выявления ошибок имитации; краткостью, точностью выражения понятий, характеризующих имитационные процессы; возможностью для некоторой предметной области исследований заранее построить стандартные компоненты, которые могут применяться пользователями при построении любых имитационных моделей; автоматическим формированием типов данных, соответствующих принятому способу имитации и необходимым в процессе имитационного моделирования; удобством накопления и представления выходных данных; возможностью распределения памяти ЭВМ в процессе имитационного эксперимента.

Специализированные языки имитационного моделирования делятся на две самостоятельные группы, соответствующие двум видам имитации: для непрерывных и дискретных процессов. В пос-

ледние годы наметилась тенденция моделирования непрерывно-дискретных систем. Для обеспечения средствами автоматизации указанного класса систем разработаны языки моделирования непрерывно-дискретных процессов.

3.1.2. ЯЗЫКИ НЕПРЕРЫВНОГО ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Группа языков непрерывного имитационного моделирования делится, в свою очередь, на языки аналогового моделирования и языки, которые применяются для решения систем дифференциальных уравнений, описывающих детерминированные замкнутые непрерывные системы. В языках аналогового моделирования используется компонентная эмуляция аналоговых и гибридных ЭВМ. Например, сумматор заменяется кодом операции суммирования, интегратор — кодом операции интегрирования и т. д. Взаимодействия между этими дополнениями функциональных компонент описываются с помощью блочно-ориентированного языка точно так же, как коммутационная панель аналоговой ЭВМ связывает компоненты аналоговой вычислительной программы. Эмулируются и структура, и элементы аналоговой или гибридной ЭВМ.

Для моделирования явлений, представляемых переменными, непрерывными по диапазону своих значений, но дискретными по времени свращения событий, удобен язык ДИНАМО [15]. На языке ДИНАМО непрерывные процессы описываются соответствующими операторами, позволяющими представить зависимости в системе в виде дифференциальных уравнений первого порядка. Вводятся два типа переменных: состояний и скорости. Переменные состояний используются для описания состояний объекта моделирования и условий перехода моделируемой системы в эти состояния. Переменные скорости позволяют исследователю описывать, как меняются эти состояния за некоторый отрезок времени. Язык ДИНАМО нашел применение в тех случаях, когда исследователя интересуют информационные системы с обратной связью и он апробирует на модели методы решения задач в области управления.

В ряде случаев исследователь изучает причинно-следственные связи в сложной системе. Ему нужно использовать такие средства, как интегрирование, параметрический анализ или графическая оптимизация. Для этой цели он строит имитатор непрерывной системы на цифровой ЭВМ. Модель представляется состоящей из взаимосвязанных функциональных элементов и эмулируется на цифровой ЭВМ. Наиболее удобным для этих целей является язык 360/Sistem CSMP [40]. С помощью этого языка специалисты, которые ранее применяли аналоговую технику, легко могут перейти к цифровой ЭВМ. Описание модели на этом языке представляет собой совокупность фраз, задающих правила коммутации общепринятых элементов аналогового моделирующего устройства: интеграторов, сумматоров, делителей, инверторов и т. д.

Поведение этих элементов реализуется соответствующими стандартными функциями. Для моделирования типовых элементов аналогового устройства у исследователя имеется возможность самому строить набор специальных функций применительно к его конкретной задаче. Допускаются также вставки в текст модели на языке ФОРТРАН для вычисления начальных условий, анализа результатов, выбора нового варианта значений параметров модели.

3.1.3. ЯЗЫКИ МОДЕЛИРОВАНИЯ ДИСКРЕТНЫХ СИСТЕМ

В зависимости от подхода к описанию рассматриваемых объектов имитации и метода организации квазипараллелизма языки моделирования дискретных систем можно разделить на: ориентированные на просмотр активностей; составляющие расписание событий; управляющие процессами; организующие взаимодействие транзактов; ориентированные на взаимодействие агрегатов, из которых скомпонована сложная система.

Языки, ориентированные на просмотр активностей. В этом случае алгоритм функционирования системы представляется последовательным выполнением активностей. В УПМ предусмотрен механизм планирования, который обрабатывает упорядоченный во времени список моментов завершения активностей. Когда начинается выполнение некоторой активности, в этот список заносится соответствующее время завершения и точка возврата в управляющую программу. После выполнения моделирующей программой всех операций, которые соответствуют некоторому моменту времени моделирования, возбуждается механизм временного планирования. Он вычеркивает из списка элемент, соответствующий первой по времени завершения активности, устанавливает значение переменной, отображающей время моделирования, равным времени завершения соответствующей активности и передает управление в указанную точку управляющей программы. Завершение активностей может привести к инициализации новых активностей.

Некоторые языки, например SMPL [2], позволяют не регламентировать обслуживание активностей, а лишь указывать условия, при которых они могут произойти. УПМ просматривает набор всех условий перед сдвигом модельного времени на очередной интервал, что позволяет определить, могут ли произойти какие-либо события. Программа-модель составляется из двух частей: проверки условий инициализации активностей и обслуживания активностей.

Языки, составляющие расписание событий. В языках, ориентированных на реализацию событийного способа имитации, таких как GASP [40], СИМСКРИПТ [40], SMPL [40], ДИСМ [59], МОДИС-ВЕС [22], программа модели организована в виде совокупности так называемых процедур обслуживания событий. Выполнение этих процедур синхронизируется во времени списковым механизмом планирования. Каждый элемент списка определяет

время события и имя процедуры события. Процедура события определяет, существуют ли условия запуска данной активности, в случае их наличия выполняет необходимые для запуска действия и планирует время ее завершения. После выполнения всех действий, соответствующих текущему времени моделирования, активизируется механизм планирования, который выбирает новую процедуру события.

Программа, написанная на языке, реализующем событийный способ имитации, зачастую теряет всякое сходство со структурой реальной системы. Это связано прежде всего с тем, что одним и тем же событиям могут соответствовать активности, аппроксимирующие поведение различных компонент системы. И при интерпретации результатов моделирования исследователю иногда трудно изучать взаимодействие компонент системы по множеству таких объединенных событий в системе. Это весьма существенный недостаток событийного способа имитации, особенно в тех случаях, когда имитационная модель используется как средство проектирования математического обеспечения систем управления реальными системами.

Языки, управляющие процессами. Процессный способ имитации получил наибольшее распространение из-за того, что в нем сочетаются достоинства имитации активностями и событиями.

В языках, ориентированных на процессы, таких как SOL [40], ASPOL [2], SIMULA [40], ДИС [64], СЛЕНГ-Б [20], АЛСИМ [50], СИМУЛА-ИПМ [3], СИМУЛА-67 [4], МПЛ/1 [69], СКИФ-Ф [39], МК PLSIM [51], моделирующая программа организуется в виде набора описаний процессов, каждое из которых соответствует одному классу процессов. Описание процесса объединяет активности всех процессов данного класса и имеет форму процедуры, которая выполняется одновременно для всех представителей этого класса, существующих в системе в текущее время. Фактически она является реентерабельной программой, выполняемой в квазипараллельном режиме, когда управление передается в различные точки программы по мере того, как происходят инициализации, выполнение и задержка выполнения различных процессов. Синхронизация выполнения процессов реализуется с помощью механизма планирования, рассмотренного в п. 1.3.6.

Программа имитационной модели, написанная на языке, реализующем процессный способ имитации, имеет ту же структуру, что и объект моделирования. Каждой компоненте системы соответствует свое описание процесса, что облегчает исследователю интерпретацию результатов моделирования, особенно в тех случаях, когда модель используется как средство изучения узких мест в системе.

Языки, организующие взаимодействие транзактов. В языках моделирования, реализующих транзактный способ имитации, в качестве динамических объектов при описании программы модели используются транзакты (заявки на обслуживание). Каждому

транзакту может принадлежать определенный набор параметров, меняемых соответствующими блоками. Типичными представителями таких языков являются GPSS [40, 82], МОДЕЛЬ-6 [65], ИМСС [66], CSS [83], АСИМ [57]. Программа-модель на таком языке представляется в виде схемы, отображающей пространственное перемещение транзактов в системе. Затем осуществляется символическое кодирование схемы, при котором каждому блоку ставится в соответствие строка символов, отражающих текст программы. Управляющая программа моделирования создает транзакты, передвигает их по определенным блокам и производит действия, связанные с обслуживанием транзакта каждым блоком. По мере прохода транзактов через блоки происходит сдвиг модельного времени. Синхронизация обслуживания транзактов была рассмотрена в п. 1.3.4.

Языки, ориентированные на взаимодействие агрегатов. В таких языках моделирования в качестве динамических объектов выступают агрегаты. В функции управляющей программы моделирования входит синхронизация взаимодействия агрегатов. Она распознает, какое событие произошло в агрегате, управляет выдачей агрегатом выходных сигналов, синхронизирует работу в агрегатной системе, организует передачу выходных сигналов агрегата на входные полюсы других агрегатов. Сами же агрегаты имитируют поведение компонент сложной системы. Описания агрегатов также являются реентерабельными программами, выполняемыми в квазипараллельном режиме, когда управление передается различным программам, реализующим различные алгоритмы функционирования агрегата по мере того, как происходят инициализация и задержка выполнения алгоритмов агрегатов. Агрегатный метод имитации используется в системах моделирования АИС [11, 36, 37] и САПАС [68].

3.1.4. ЯЗЫКИ, ОРИЕНТИРОВАННЫЕ НА МОДЕЛИРОВАНИЕ НЕПРЕРЫВНО-ДИСКРЕТНЫХ ПРОЦЕССОВ

В последнее время широкое распространение получили языки, обеспечивающие имитацию непрерывно-дискретных систем. Все компоненты сложных систем, у которых состояния изменяются дискретным образом, описываются аналогично процессному способу имитации. Для описания остальных компонент системы используются такие же уравнения, как для языков непрерывного моделирования с применением механизмов приращения модельного времени с фиксированным шагом изменения. Внешнюю синхронизацию процессов осуществляет управляющая программа моделирования способом, аналогичным способу, используемому при имитации с помощью процессов. Кроме того, в функции управляющей программы моделирования входят проверка условий инициализации компонент модели, имитирующих поведение системы с непрерывным представлением модельного времени, и передача

управления этим компонентам. Примерами таких языков моделирования могут служить НЭДИС [40], DISLIN [45], НЕДИС-ЕС [60].

3.1.5. КЛАССИФИКАЦИЯ ЯЗЫКОВ МОДЕЛИРОВАНИЯ

К классификации языков моделирования можно подходить по-разному. Иногда к языкам предъявляют требования, определяемые предметной областью их использования. В работе [40] содержится достаточно подробная классификация языков моделирования по описательным возможностям. В нашем случае мы прежде всего выделяем соответствие языков моделирования рассмотренным ранее способам формализации реальных объектов. Поскольку универсальные языки моделирования определяют изобразительные возможности представления алгоритмов, то естественна классификация по способам реализации систем моделирования. При этом классификация на такой основе фактически представляет собой сравнение алгоритмических возможностей языков моделирования.

В табл. 3.1 приведена классификация систем моделирования по способу имитации процессов, протекающих в реальных системах. Отметим, что из всего множества языков моделирования рассматриваются только наиболее распространенные средства имитации, реализованные как на отечественных, так и на современных зарубежных типах ЭВМ, используемых в нашей стране.

Таблица 3.1

Классификация систем моделирования по способам имитации

Виды имитации	Способ организации имитации	Наименование системы моделирования
Непрерывных процессов	Аппроксимацией процессов Системами уравнений	ДИНАМО [15] 360/Sistem CSMP [40]
Непрерывно-дискретных процессов	Событиями Процессами	НЕДИС [40], НЕДИС-ЕС [60] DISLIN [45]
Дискретных процессов	Активностями Событиями	SMPL [2] СИМСКРИПТ [40], ДИСК [59], МОДИС-ВЕС [22], GASP [40], CMPL [40]
	Процессами	ДИС [64], СЛЕНГ-Б [20], АЛСИМ [50], СКИФ [39], СИМУЛА-ИПМ [3], СИМУЛА-67 [4], МПЛ/1 [69], МК PLSIM [51], SOL [40], ASPOL [2], SIMULA [40]
	Транзактами	МОДЕЛЬ-6 [65], ИМСС [66], GPSS [82], GPSSV [40], АСИМ [57], CSS [83]
	Агрегатами	АИС [11, 36, 37], САПАС [68]

По способу реализации различаем системы моделирования, использующие универсальную базовую систему программирования, и специально построенные системы, ориентированные на моделирование определенного класса сложных систем. В первом случае система моделирования представляет собой расширение возможностей универсальной системы программирования.

Обычно в качестве базового языка программирования используется АЛГОЛ, ФОРТРАН или PL/I. В табл. 3.2 приведена классификация систем моделирования по использованию базового языка программирования и типам ЭВМ, на которых они реализованы. Так, алголоподобную базу реализации имеют языки в таких системах моделирования, как ДИС, СЛЕНГ-Б, СИМУЛА-ИПМ, СИМУЛА-67, АЛСИМ, ASPOL, SIMULA, SOL. На базе ФОРТРАНА реализованы языки в системах моделирования DISLIN, ДИНАМО, ИМСС, СКИФ-Ф, СИМСКРИПТ, GASP. Язык PL/I в качестве базового используется в системах моделирования ДИСМ, СКИФ-П, SMPL, CMPL, МПЛ/1, МК, PLSIM. В ряде случаев для моделирования разработаны системы моделирования с собственными языками моделирования. К таким системам можно отнести: НЕДИС, МОДИС-ВЕС, CSS, GPSS, GPSSV, АИС, АСИМ. Практически у всех перечисленных ранее систем моделирования название языка моделирования совпадает с названием системы моделирования.

На ранних этапах разработки систем моделирования считалось более эффективным использовать базовый язык, расширив

Таблица 3.2

Классификация систем моделирования по использованию базового языка программирования и типам ЭВМ

Реализация на ЭВМ типа	Системы моделирования			
	Реализация на базовом языке программирования			Собственные язы- ковые средства моделирования
	АЛГОЛ	ФОРТРАН	PL/I	
М-220	ДИС СЛЕНГ-Б	—	—	—
БЭСМ-6	ДИС СИМУЛА-ИПМ СИМУЛА-67 АЛСИМ	ДИНАМО	—	МОДЕЛЬ-6 НЕДИС
ЕС ЭВМ	СКИФ-Б СИМУЛА-67 АЛСИМ	СКИФ-Ф ИМСС	СКИФ-П ДИСМ МПЛ/1 МКPLSIM	НЕДИС-ЕС МОДИС-ВЕС АСИМ GPSS CSS АИС САПАС CPSSV 360/Sistem CSMP ДИНАМО
Зарубежные типы ЭВМ	ASPOL SIMULA SOL	CASP СИМСКРИПТ	SMPL CMPL	

его до уровня языка моделирования. Однако с развитием моделирования в ряде предметных областей исследования оказалось, что эффективнее разрабатывать языки моделирования со своими системами компилирования. Этот способ более перспективен из-за отсутствия ограничений, налагаемых на описание модели с помощью универсальных языков программирования, и благодаря более высокому уровню технологии их использования в определенной предметной области исследований.

3.2. ИНСТРУМЕНТАЛЬНЫЕ И ТЕХНОЛОГИЧЕСКИЕ ВОЗМОЖНОСТИ СРЕДСТВ АВТОМАТИЗАЦИИ МОДЕЛИРОВАНИЯ

При выборе средств автоматизации моделирования исследователя обычно интересуют две группы их свойств: изобразительные возможности представления алгоритма и удобство работы при использовании программного продукта. Автор на основе собственного опыта и контактов со специалистами, применяющими различные системы автоматизации моделирования, пришел к следующим выводам. Во-первых, из изобразительных возможностей языка моделирования исследователя интересуют три аспекта: состав операторов моделирования, наличие доступа к модельным характеристикам, возможности универсального базового языка программирования для реализации алгоритмов. Во-вторых, удобство работы с системой моделирования определяется технологией ее использования, которая поддерживается данной СМ.

Обычно исследователь пытается получить ответы на вопросы в следующей последовательности: реализована ли СМ на ЭВМ, имеющейся у пользователя; имеет ли она средства автоматизации представления и интерпретации результатов моделирования; существует ли возможность управления имитационным экспериментом; каков состав средств планирования экспериментов и обработки результатов моделирования; возможно ли управление процессом сбора статистики в ходе моделирования; имеются ли средства отладки и диагностики; удобен ли ввод-вывод информации в СМ; каков способ сборки компонент ИМ сложной системы. Конечно, этот список может быть дополнен. Однако анализ описаний существующих средств автоматизации моделирования убеждает, что вопросам технологии имитационного моделирования уделяется недостаточно внимания. Поэтому читателю предлагается систематизированная в виде таблиц информация об инструментальных и технологических возможностях наиболее распространенных СМ.

Будем считать, что инструментальные возможности средств автоматизации моделирования обусловлены:

- 1) описательными возможностями языков, реализуемых системами моделирования, в части отражения способа имитации;
- 2) возможностью (или отсутствием таковой) доступа пользователя к модельным характеристикам;

- 3) возможностью рекурсии в описании модели;
- 4) возможностью использования универсальных средств программирования, особенно при реализации арифметических операций.

Под описательными возможностями системы моделирования будем понимать богатство типов элементов модели. Каждый тип элементов модели, позволяющий исследователю формализовать явления в реальной системе, назовем фундаментальным элементом. От состава и числа типов фундаментальных элементов, используемых для построения моделей сложных систем, зависит удобство использования и легкость освоения исследователем системы моделирования. Читателю, интересующемуся описательными возможностями различных языков моделирования, можно рекомендовать работу [40].

Технологические возможности средств автоматизации моделирования определяются:

- 1) способом компилирования программ модели;
- 2) способом организации ввода-вывода информации в программе модели;
- 3) возможностью выдачи сообщения пользователю в ходе имитации;
- 4) наличием средств организации отладки и диагностики программы имитационной модели;
- 5) наличием средств организации сбора статистики и возможностью управления ими;
- 6) составом средств обработки результатов моделирования;
- 7) наличием специальных средств испытания имитационных моделей;
- 8) способностью управления имитационным экспериментом;
- 9) наличием средств автоматизации представления и интерпретации результатов имитации;
- 10) типом ЭВМ, на которой реализована система моделирования, и характеристиками организации вычислительного процесса на этой ЭВМ.

Ниже будут рассмотрены инструментальные и технологические возможности 19 наиболее распространенных в нашей стране систем моделирования. Выбор для сравнения указанных систем моделирования определялся двумя обстоятельствами: наличием доступной технической документации или первоисточников, по которым можно составить представление о возможностях системы моделирования, и степенью знакомства автора с системой моделирования.

Результаты сравнения инструментальных и технологических возможностей системы моделирования представлены в табл. 3.3—3.6. Таблицы построены в форме ответов по каждому из 14 пунктов, характеризующих возможности СМ. Содержание ответов базируется на информации из первоисточников, ссылки на которые даны в п. 14 таблиц. Таблицы позволяют исследователю провести сравнение интересующих его систем моделирования по лю-

Таблица 3.3

Возможности систем моделирования, реализующих имитацию различными

Способ имитации	Система моделирования	1. Описание реальных объектов	2. Доступ пользователя к модельным характеристикам	3. Рекурсия в описании модели
Активными	SMPL	Элементами	Запись и изъятие события из расписания	Имеется
Транзактами	GPSS	Транзактами, средствами, памятью, очередями	Не допускается	Не допускается
	МОДЕЛЬ-6	Транзактами, службами, пакетами, наборами, переключателями	Не допускается	Не допускается
Агрегатами	АИС	Кусочно-линейными агрегатами	Не допускается	Не допускается
	САПАС	Кусочно-линейными агрегатами	Не допускается	Не допускается
Способ имитации	Система моделирования	8. Отладка и диагностика программы — модели	9. Организация сбора статистики	10. Обработка результатов моделирования
Активными	SMPL	Средствами языка ФОРТРАН	Командами сбора стандартной статистики	Средствами языка ФОРТРАН
Транзактами	GPSS	Указателями динамических ошибок	Стандартной статистикой об объекте	Операторами обработки многопрогонных режимов имитации
	МОДЕЛЬ-6	Указателями выдачи информации о состояниях в системе	С помощью наблюдателей и таблиц использования средств транзактами	Средствами автокода БЭМШ
Агрегатами	АИС	С помощью диагностической системы АЛИСА	Стандартной статистикой агрегатов	Набором задач обработки в составе внешнего программного обеспечения
	САПАС	Транслятором, обеспечивающим диагностику вводимых операторов	Типовыми операторами сбора статистики	Операторами включения любой программы обработки на PL/I

тособами

4. Реализация арифметических операций	5. Способ компилирования программы модели	6. Обеспечение ввода-вывода информации	7. Формирование сообщений в ходе имитации
Операторами языка ФОРТРАН	Запись программы на ФОРТРАНЕ, затем компилирование и выполнение	Средствами языка ФОРТРАН	Операторами языка ФОРТРАН
Не предусмотрена	Интерпретация текста модели	Встроенным вводом-выводом информации в модель	Стандартными средствами вывода сообщений
Операторами автокода БЭМШ	Фрагменты на автокоде вставляются в текст, а затем интерпретация	Специальным набором операторов ввода-вывода	Операторами запроса об обстановке
Средствами базового языка КОМПАС	Перевод на ФОРТРАН, трансляция, редактирование и выполнение	С помощью диалоговой системы АЛИСА	Стандартным набором операторов выдачи сообщений
Средствами базового языка САПАС	Однопроходная трансляция интерпретирующего типа	Набором специальных операторов ввода-вывода	Стандартной статистикой о работе агрегатов
11. Обеспечение испытания модели	12. Реализация модельных экспериментов	13. Автоматизация представления результатов имитации	14. Реализация на ЭВМ
Отсутствует	Возможностью раздельного инициирования программ	Командами сбора стандартной статистики	IBM-7090 [2]
Отсутствует	Многопрогонным режимом имитации	Операторами сбора стандартной статистики	IBM CORP [40] ЕС ЭВМ [40, 82]
Отсутствует	Средствами автокода БЭМШ	Операторами сбора стандартной статистики	БЭСМ-6 [65]
Задачами качественного анализа агрегатных моделей	Диалоговой системой АЛИСА и набором задач внешнего программного обеспечения	Диалоговой системой АЛИСА и набором задач внешнего программного обеспечения	ЕС ЭВМ [11, 36, 37]
Отсутствует	Возможностью изменения параметров агрегата с пульта оператора	Стандартной статистикой агрегата	ЕС ЭВМ [68]

Таблица 3.4

Возможности систем моделирования, реализующих имитацию событийным

Система моделирования	1. Описание реальных объектов	2. Доступ пользователя к модельным характеристикам	3. Рекурсия в описании модели
СИМ-СКРИПТ	Элементами	Запись и изъятие события из расписания	Не допускается
SMPL	Средствами, транзактами, очередями	Запись в список событий новых элементов	»
GASP	Элементами	Прямой доступ к системному времени и к параметрам объектов	»
ДИСМ	Объектами	Возможность записи и выбора из очереди запланированных событий	»
МОДИС-ВЕС	Объектами, автоматами, сообщениями, параметрами	Прямой доступ к системному времени объектов	»
Система моделирования	8. Отладка и диагностика программы модели	9. Организация сбора статистики	10. Обработка результатов моделирования
СИМ-СКРИПТ	Средствами языка ФОРТРАН и моментальными снимками	Стандартной статистикой	Средствами языка ФОРТРАН
SMPL	Средствами языка ФОРТРАН	То же	То же
GASP	Средствами языка ФОРТРАН и слежением за свершением событий	Командами языка Collect Histog	»
ДИСМ	Средствами языка PL/I	Стандартной статистикой	Средствами языка PL/I
МОДИС-ВЕС	Банком описаний моделей и набором сообщений об ошибках в модели	То же	То же

способом

4. Реализация арифметических операций	5. Способ компилирования программы модели	6. Обеспечение ввода-вывода информации	7. Формирование сообщений в ходе имитации
Операторами языка ACCUMULATE	Стандартный для языка ФОРТРАН	Набором входных и выходных команд	Использованием генератора сообщений
Операторами языка ФОРТРАН	То же	Средствами языка ФОРТРАН	Операциями над средствами и очередями
То же	»	Набором программ ввода-вывода информации	Сообщениями о структуре очередей, статистике их использования, списке заплазированных событий
Операторами языков ФОРТРАН и PL/I	Стандартный для PL/I	Набором входных и выходных макрокоманд	Системными процедурами выдачи системного времени и задержек
Операторами языков ФОРТРАН и Ассемблер	Трансляция с PL/I, а затем интерпретация	Специальным набором процедур ввода-вывода	Специальными средствами выдачи сообщений
11. Обеспечение испытания модели	12. Реализация модельных экспериментов	13. Автоматизация представления результатов имитации	14. Реализация на ЭВМ
Отсутствует	Картами инициализации, разделным компилированием секций и их объединением	Командами сбора стандартной статистики	RAND CORP [40] IBM-7090/94 [40] IBM-7040/44 [40]
То же	Средствами языка ФОРТРАН	Средствами языка ФОРТРАН	IBM-7090/94 [40]
»	Картами инициализации, многопрограммным режимом имитации	То же	RAND CORP [40] IBM-7040/44 [40]
»	Раздельным компилированием секций модели на PL/I и объединением редактором	Командами сбора стандартной статистики	ЕС ЭВМ [59]
»	Прерыванием и возобновлением имитации, объединением элементов по переменным	То же	ЕС ЭВМ [22]

Таблица 3.5

Возможности систем моделирования, реализующих имитацию процессным

Система моделирования	1. Описание реальных объектов	2. Доступ пользователя к модельным характеристикам	3. Рекурсия в описании модели
СИМУ-ЛА-67	Объектами с заданной структурой, переменными моделирования	Отсутствует	Не допускается
СКИФ	Процессами, сообщениями, устройствами, памятью, списками, таблицами	То же	То же
АЛСИМ	Процессами, переменными, стандартными функциями, устройствами, памятью	Прямой доступ к локальным переменным процессов	»
ДИС	Схемами, процессами, блоками, линиями связи	Отсутствует	»
МПЛ/1	Автоматами, сообщениями, регистрами, устройствами, хост-ЭВМ, задачами, пакетами	То же	»
Система моделирования	8. Отладка и диагностика программы модели	9. Организация сбора статистики	10. Обработка результатов моделирования
СИМУ-ЛА-67	Специальным режимом генерации модели с помощью процедур отладки	Отсутствует	Средствами языка АЛГОЛ
СКИФ	Системой трассировки объектов	Набором специальных операторов сбора статистики	Средствами языков АЛГОЛ, PL/1, ФОРТРАН
АЛСИМ	Набором операторов прерывания и возобновления имитации, диалогом с пользователем	То же	Средствами языка АЛГОЛ
ДИС	Средствами языка АЛГОЛ	Отсутствует	То же
МПЛ/1	Средствами языка PL/1	Набором специальных операторов сбора статистики	Средствами языка PL/1

способом

4. Реализация арифметических операций	5. Способ компилирования программы модели	6. Обеспечение ввода-вывода информации	7. Формирование сообщений в ходе имитации
Операторами языка АЛГОЛ	В рамках мониторинг системы ДУБНА	Набором операторов ввода-вывода	Операторами выдачи
Операторами языков ФОРТРАН, АЛГОЛ, PL/I	Компоновка из библиотечных процессов на этапе редактирования	Операторами интерпретирующего типа	Набором сервисных программ
Операторами языка АЛГОЛ	Компилирование на языке АЛГОЛ, затем интерпретация операторов	Набором операторов ввода-вывода	Операторами выдачи
То же	Стандартный для языка АЛГОЛ	То же	Не предусмотрена
Операторами языка PL/I	Препроцессор на PL/I, компилятор PL/I формирует объектную программу	Средствами языка PL/I	Специальными операторами
11. Обеспечение испытания модели	12. Реализация модельных экспериментов	13. Автоматизация представления результатов имитации	14. Реализация на ЭВМ
Отсутствует	Средствами языка АЛГОЛ	Отсутствует	БЭСМ-6 [4] ЕС ЭВМ [4]
То же	Средствами языков АЛГОЛ, PL/I, ФОРТРАН	Операторами сбора статистики	М-220 [39] ЕС ЭВМ [39]
»	Набором операторов коммутации входов модели при указании ее состава	То же	БЭСМ-6 [50] ЕС ЭВМ [50]
»	Параметризуемой конструкцией ОПИСАНИЕ ЭКСПЕРИМЕНТА	Отсутствует	М-220 [64] БЭСМ-6 [40]
»	Средствами языка PL/I	Операторами сбора статистики	ЕС ЭВМ [69]

Таблица 3.6

Возможности систем моделирования, реализующих имитацию процессным

Система моделирования	1. Описание реальных объектов	2. Доступ пользователя к модельным характеристикам	3. Рекурсия в описании модели
<i>Дискретные</i>			
МК PLSIM	Процессами, очередями, устройствами, памятью, флагами, сигналами, трассами	Отсутствует	Не допускается
SOL	Процессами, переменными, средствами, памятью, транзактами	То же	Допускается
ASPOL	Процессами, событиями, средствами, памятью	»	То же
<i>Непрерывно-дискретные</i>			
DISLIN	Процессами, граф-схемами алгоритмов	Отсутствует	Не допускается
НЕДИС	Объектами, переменными, памятью, транзактами, средствами описания дифференциальных уравнений	То же	Допускается
Система моделирования	8. Отладка и диагностика программы модели	9. Организация сбора статистики	10. Обработка результатов моделирования
<i>Дискретные</i>			
МК PLSIM	Специальной системой отладки и диагностики для локализации ошибок	Специальной системой сбора статистики с управлением сбором	Специальной системой с набором операторов статистической обработки на PL/I
SOL	Выборочным слежением за временем прогона модели	Операторами TABU-LATE с параметрическим заданием	Средствами языка АЛГОЛ
ASPOL	Специальной системой отладки и возможностью контроля ошибок	Стандартной статистикой	То же
<i>Непрерывно-дискретные</i>			
DISLIN	Средствами языка ФОРТРАН	Стандартной статистикой	Средствами языка ФОРТРАН
НЕДИС	Специальными операторами отладки	То же	С помощью автоматизированной системы обработки данных [60]

способом

4. Реализация арифметических операций	5. Способ компилирования программы модели	6. Обеспечение ввода-вывода информации	7. Формирование сообщений в ходе имитации
<i>системы</i>			
Операторами языка PL/I	Отсутствие препроцессора, расширяющаяся система процедур на PL/I	Набором специальных операторов задания состава модели и операторов ввода-вывода	Набором специализированных операторов, выдачей временной диаграммы
Операторами языка АЛГОЛ	Компилирование текста в псевдокод, затем интерпретация в ходе имитации	Специальными макрокомандами READ и WRITE	Операторами выдачи
То же	Стандартный для языка АЛГОЛ	Специальными операторами ввода-вывода информации	То же
<i>системы</i>			
Операторами языка ФОРТРАН	Препроцессором в операторы ФОРТРАНА, затем компилирование с языка ФОРТРАН	Набором специальных операторов ввода-вывода	Операторами информирования пользователя о состоянии процессов
Операторами языков ФОРТРАН и Ассемблер	Перевод на Ассемблер, затем генерация программы модели и редактирование	Специальными модулями ввода заданий и вывода информации	Системой трассировки и выходными операторами
11. Обеспечение испытания модели	12. Реализация модельных экспериментов	13. Автоматизация представления результатов имитации	14. Реализация на ЭВМ
<i>системы</i>			
Набором стандартных процедур организации испытания моделей	Специальной системой с набором операторов планирования экспериментов на PL/I	Набором операторов документирования	ЕС ЭВМ [51]
Отсутствует	Средствами языка АЛГОЛ	Оператором TABULATE	UNIVAC-1107 [40]
То же	То же	Командами сбора стандартной статистики	СДС-6000 [2] СДС-7000 [40]
<i>системы</i>			
Отсутствует	Специальным набором операторов постановки эксперимента	Набором специальных операторов	БЭСМ-6 [45]
То же	Средствами управления экспериментом	Командами сбора стандартной статистики и построения графиков	БЭСМ-6 [40] ЕС ЭВМ [60]

бому из 14 пунктов независимо от способа имитации дискретных процессов.

Отметим, что сравнивать между собой по вышеперечисленным пунктам можно только системы, реализующие один и тот же способ имитации. В противном случае достоинства способа формализации, проявляющиеся при построении данной конкретной модели, можно отнести к достоинствам языка моделирования. Зачастую оказывается, что при изменении модели или даже условий проведения моделирования удобнее было бы использовать другой способ имитации и, как следствие, другой язык моделирования. Кроме того, каждая из приведенных систем имеет свою область применения, в пределах которой она обладает преимуществами либо инструментального, либо технологического характера перед другими системами. Поэтому предлагаемое сравнение можно рассматривать как первое приближение в тех случаях, когда у исследователя нет опыта применения какой-либо из систем моделирования или же когда этот опыт негативный и исследователь стоит перед выбором нового средства моделирования для решения своих задач. В тех случаях, когда исследователь уже привык к определенной системе моделирования, данное сравнение поможет уяснить ему истинные технологические возможности применяемой им системы моделирования и может быть даст рекомендации на переход к более технологичной системе моделирования.

Как видно из табл. 3.3—3.6, у каждой из приведенных систем моделирования имеются свои сильные стороны, которые делают ее конкурентоспособной по сравнению с другими системами моделирования при построении некоторых моделей. Отметим некоторые из них, которые, по мнению автора, являются наиболее интересными.

Весьма перспективным способом компилирования программы модели обладает СМ АИС. Перевод базового языка на ФОРТРАН позволяет распространить СМ АИС на все типы ЭВМ, где используется ФОРТРАН.

Интерпретация текста модели, реализованная в СМ GPSS, является наиболее экономичным способом реализации описания модели по сравнению с другими СМ.

Применение препроцессора, реализованного в СМ МПЛ-1, позволяет легко наращивать возможности системы и расширять предметные области ее использования.

Процедурный способ расширения системы моделирования, реализованный в СМ PLSIM, более экономичен по сравнению с СМ МПЛ-1 за счет исключения этапа препроцессирования, но более труден при переходе из одной предметной области в другую.

Наличие специальных модулей ввода заданий и вывода информации в СМ НЕДИС делает ее удобной и привычной для широкого круга специалистов, имеющих опыт программирования на ЕС ЭВМ.

В ряде случаев исследователь может предпочесть набор специальных операторов для задания состава модели, ввода и вывода информации, реализованных в МК PLSIM.

Больше всего преимуществ при организации ввода-вывода информации исследователь получает при использовании диалоговой системы АЛИСА, реализованной в СМ АИС, именно из-за возможности оперативного изменения информации в интерактивном режиме взаимодействия с системой.

Весьма существенно для исследователя наличие специального набора операторов выдачи сообщений о ходе имитации, реализованного в СМ: DISLIN, МК PLSIM, МПЛ/1. В ряде случаев пользователь может применить систему трассировки, реализованную в СМ НЕДИС, либо генератор сообщений, реализованный в СМ СИМСКРИПТ.

Решающим фактором, определяющим технологичность системы моделирования, может оказаться наличие системы отладки имитационных моделей. Учитывая это обстоятельство, в СМ МОДИС-ВЕС введены специальный банк описаний модели и система выдачи сообщений об ошибках в модели, возникающих в ходе имитации дискретных процессов. В СМ МК PLSIM реализована специальная система отладки и диагностики для локализации ошибок. Использование диалоговых возможностей системы АЛИСА существенно облегчает этап отладки имитационных моделей, реализованных с помощью СМ АИС в виде агрегативных систем.

По организации сбора статистики наибольшие возможности пользователю представляют СМ МК PLSIM, СКИФ, МПЛ/1. Важным для исследователя при этом является возможность управления сбором статистики, планирования его возобновления, исключения или выделения переходного периода имитации при сборе статистики моделирования.

Наличие специальных средств обработки результатов моделирования в составе СМ существенно повышает ее технологические возможности. С этой точки зрения, по мнению автора, больше всего возможностей предоставляет СМ МК PLSIM. Весьма перспективным является включение либо специальной системы обработки данных АСОД (как это делается в СМ НЕДИС), либо набора задач обработки результатов моделирования в составе внешнего программного обеспечения (как это реализуется в СМ АИС).

Как видно из табл. 3.3—3.6, вопросам автоматизации испытаний имитационных моделей уделяется мало внимания. Только две из рассмотренных СМ имеют средства испытания имитационных моделей. В СМ МК PLSIM для этой цели в распоряжение исследователя предоставляется набор стандартных процедур организации испытания моделей. СМ АИС предлагает воспользоваться специальными программами, имеющимися в составе ее внешнего программного обеспечения.

Вопросам организации и планирования модельных экспериментов на ЭВМ также до недавнего времени уделялось мало внима-

ния при разработке СМ. В основном дело ограничивалось введением либо специальных карт инициализации подмоделей с разделным их компилированием и последующим автоматическим их объединением в одну модель, либо средств организации многопрогноного режима имитации. Использование диалоговой системы АЛИСА существенно облегчает организацию имитационных экспериментов в рамках СМ АИС. В СМ МК PLSIM и DISLIN для планирования экспериментов реализована система операторов, запрограммированных в виде PL-процедур. Интересными являются идеи применения в СМ ДИС параметризуемой конструкции «описание эксперимента» и набора операторов коммутации входов модели или указание ее состава в СМ АЛСИМ.

Средствами автоматизации представления результатов имитации в большинстве СМ являются операторы сбора статистики. В СМ МК PLSIM для этой цели пользователю предоставляется набор операторов документирования результатов моделирования. СМ НЕДИС позволяет организовать графическое представление результатов моделирования.

Наконец, анализ табл. 3.3—3.6 показывает, что основная доля отечественных СМ реализована на ЕС ЭВМ. Правда, такие СМ, как НЕДИС, АЛСИМ, СИМУЛА-67, имеют также реализации на широко распространенной отечественной ЭВМ БЭСМ-6. Однако технологические возможности реализации указанных СМ на ЕС ЭВМ и БЭСМ-6 различны и определяются возможностями организации вычислительного процесса на этих ЭВМ.

3.3. О ВЫБОРЕ СИСТЕМЫ МОДЕЛИРОВАНИЯ

Анализ табл. 3.2 показывает, что в нашей стране наибольшее распространение получили более 30 СМ. При построении гибридных имитационных моделей для имитации непрерывных процессов можно рекомендовать два языка моделирования. Так, аппроксимацию непрерывных процессов можно реализовать с помощью языка ДИНАМО [15], который использует в качестве базового язык программирования ФОРТРАН и реализован на ЭВМ БЭСМ-6 и некоторых зарубежных ЭВМ. Для решения систем уравнений при имитации непрерывных процессов можно применить СМ 360/Sistem CSMP [40], который реализован на ряде зарубежных ЭВМ фирмы IBM.

Для имитации сложных систем с помощью активностей можно предложить язык SMPL [2], который реализован на ЭВМ IBM-7090.

Имитация событийным способом реализуется пятью системами моделирования. При наличии соответствующих зарубежных ЭВМ исследователю можно рекомендовать следующие языки моделирования: СИМСКРИПТ [40], SMPL [40] и GASP [40]. Все эти языки используют ФОРТРАН в качестве базового языка программирования. Перечисление языков велось выше в порядке предпочтительности технологических возможностей, предоставля-

емых системами моделирования. Из отечественных систем моделирования можно рекомендовать ДИСМ [59] и МОДИС-ВЕС [22]. Первая из них базируется на языке PL/1, вторая имеет собственный язык моделирования и специализированный транслятор. С нашей точки зрения в большинстве случаев предпочтение следует отдать ДИСМ благодаря широким возможностям базового языка программирования.

Для реализации транзактного способа имитации возможно использование шести систем моделирования. При наличии у исследователя зарубежных ЭВМ больше всего подходит GPSS [40]. Для ЭВМ БЭСМ-6 пока существует единственный язык имитации МОДЕЛЬ-6 [65], базирующийся на автокоде БЭМШ. Если в распоряжении исследователя имеется ЕС ЭВМ, то здесь к его услугам четыре хорошие системы моделирования. Для исследователя, слабо владеющего программированием на ЭВМ, можно рекомендовать декларативный язык моделирования АСИМ [57]. При умении программировать на ФОРТРАНЕ можно остановиться на системе моделирования ИМСС [66]. Если предметом исследования является проектирование и использование систем на базе ЕС ЭВМ, то технологичным средством является система моделирования CSS [83]. Широкое применение благодаря простоте освоения в самых различных областях науки и промышленности получил язык моделирования GPSS [82].

Для моделирования агрегатным способом имитации сложных систем на ЕС ЭВМ разработаны три системы моделирования. К сожалению, на других типах ЭВМ нет хорошо апробированных реализаций систем моделирования с агрегатным способом имитации. Наибольший интерес представляет язык КОМПАС [37], реализованный в составе второй версии системы моделирования АИС [36]. Поскольку язык КОМПАС использует в качестве базового языка программирования ФОРТРАН, то возможно его распространение и на ЭВМ БЭСМ-6 с применением мониторной системы ДУБНА [44]. Учитывая это обстоятельство, а также возможность формирования заказов на модель из агрегатов, содержащихся в библиотеке системы АИС, с помощью диалоговой системы АЛИСА на естественном языке, можно рекомендовать вторую версию агрегативной системы АИС [37]. Для пользователя, владеющего языком программирования PL/1, весьма привлекательна система моделирования на базе языка САПАС [68]. И только в тех случаях, когда указанных систем моделирования в распоряжении исследователя не имеется, следует использовать первую версию АИС [11], имеющую в своем составе язык и специальный транслятор для ЕС ЭВМ.

Самый большой выбор имеется у исследователя, выбравшего процессный способ имитации объектов моделирования. Для имитации чисто дискретных процессов наибольшее распространение получили 16 систем моделирования. Все используемые в нашей стране типы ЭВМ имеют реализации систем моделирования на основе процессного способа имитации сложных систем. Для ЭВМ

БЭСМ-6 реализованы системы моделирования АЛСИМ [50], СИМУЛА-ИПМ [3], СИМУЛА-67 [4] и ДИС [40]. Все эти системы имеют в качестве базового языка программирования АЛГОЛ-60.

При наличии ЕС ЭВМ у исследователя имеется выбор из семи систем моделирования. Наиболее технологичными системами моделирования, базирующимися на языке программирования PL/1, являются МК PLSIM [51] и МПЛ/1 [69]. Пользователю, имеющему опыт моделирования с использованием языка СЛЕНГ-Б, можно рекомендовать одну из трех версий системы моделирования СКИФ. Конкретная версия определяется выбором базового языка программирования. При программировании на языках PL/1 и ФОРТРАН рекомендуются соответственно версии СКИФ-П [39], СКИФ-Ф [39]. В тех случаях, когда исследователя привлекает богатство описательных возможностей, предоставляемых языком АЛГОЛ, можно обратиться к системам моделирования СИМУЛА-67 [4] и АЛСИМ [50]. Исследователям, имеющим зарубежные ЭВМ указанных в табл. 3.3—3.6 типов, можно предложить четыре типа языков моделирования, реализующих процессный способ имитации. Все эти языки в качестве базового языка программирования применяют АЛГОЛ-60. Наиболее богат по своим изобразительным возможностям язык моделирования SIMULA [40]. Языки моделирования SOL [40] и ASPOL [2] имеют примерно равные технологические возможности. Поэтому выбор между ними будет зависеть от конкретной модели, которая реализуется при процессном подходе к имитации.

В тех случаях, когда исследуются непрерывно-дискретные системы, в распоряжение исследователя можно предоставить два языка моделирования. Для ЭВМ БЭСМ-6 исследователю, программирующему на ФОРТРАНЕ в рамках мониторной системы ДУБНА, можно рекомендовать систему моделирования DISLIN [45]. Шире распространен и имеет большие возможности имитации непрерывных процессов язык моделирования НЕДИС [60]. Кроме того, НЕДИС является едва ли ни единственной технологичной системой моделирования непрерывно-дискретных процессов на ЕС ЭВМ.

4.1. ТЕХНОЛОГИЧЕСКИЕ ЭТАПЫ ИСПЫТАНИЯ
И ЭКСПЛУАТАЦИИ ИМИТАЦИОННЫХ МОДЕЛЕЙ

После того как ИМ реализована на ЭВМ, исследователь должен выполнить последовательность технологических этапов: испытание ИМ, исследование свойств модели, составление плана имитационных экспериментов на ЭВМ, эксплуатацию модели.

Как уже отмечалось, в ИМ можно различать такие составляющие, как компоненты, переменные, параметры, функциональные зависимости, ограничения, целевые функции. Под компонентами понимают составные части, которые при соответствующем объединении образуют систему. На реальную систему воздействуют переменные G^* , которые можно измерять, но нельзя ими управлять, и параметры X^* , которые исследователь может изменять в ходе натурных экспериментов. На выходе системы возможно измерение выходных характеристик Y^* . При этом существует некоторая неизвестная исследователю функциональная зависимость между выходными характеристиками, переменными и параметрами системы: $Y^* = \varphi^*(X^*, G^*)$. Модель системы, в свою очередь, определяется как совокупность компонент, объединенных для выполнения заданной функции $Y = \varphi(X, G)$. Здесь Y, X, G — векторы соответственно результата действия модели системы, параметров моделирования, переменных модели. Параметры модели X исследователь в ходе имитационного эксперимента может выбирать произвольно. Переменные G модели должны принимать только те значения, которые характерны для данного объекта моделирования или условий его функционирования. Поэтому в ходе имитации реальных процессов вектор G формируется на основе реальной информации. Для этой цели обычно замеряются фактические значения переменных G^* (внешней нагрузки) и результатов поведения системы Y^*_Q при заданных значениях параметров системы Q^* (Q^* является выборкой измеренных значений вектора X^*).

Испытания ИМ начинаются с задания исходной информации моделирования. Перед исследователем чаще всего возникают следующие проблемы: где взять исходную информацию, имеется ли возможность использования прототипов объекта моделирования для измерения переменных G^* , применимы ли статистические гипотезы при стохастическом характере переменных G^* , возможна ли аппроксимация сложных процессов при определении G . Решению указанных проблем посвящен п. 4.2.1.

Затем необходима верификация модели, которая состоит в проверке соответствия ИМ тому замыслу, который был заложен при ее разработке. Исследователю должен получить гарантию того,

что созданная ИМ во всех ситуациях будет правильно отражать поведение реальной системы. С помощью верификации устанавливается верность логической структуры модели. Обычно верификация выполняется в ходе комплексной отладки программы модели на реальном потоке данных (при заданных значениях G и X). Обсуждению методики выполнения процедуры верификации посвящен п. 4.2.2.

Будем различать модели существующих и проектируемых систем. При моделировании существующих систем исследователя прежде всего интересует, насколько хорошо модель представляет моделируемую систему. Модель, поведение которой слишком отличается от поведения моделируемой системы, практически бесполезна. Поэтому для моделей существующих систем исследователь должен выполнить проверку адекватности ИМ объекту моделирования. Необходимо проверить соответствие между поведением реальной системы (Y^*_q) и поведением ИМ (Y). Для построения модели проектируемой системы производится упрощение намеченной проектировщиком структуры до такой степени, когда становится возможной имитация проектируемых процессов системы на ЭВМ. Возможности и способы упрощения структуры проектируемых систем для целей построения моделей рассматриваются в [67, 80]. Для моделей проектируемых систем проверку адекватности выполнить очень трудно, поскольку нет реального объекта для сравнения с ИМ. Поэтому в этом случае этап проверки адекватности ИМ объекту моделирования зачастую отсутствует. Методике выполнения проверки адекватности модели реальной системы посвящен п. 4.2.3.

Для обеспечения адекватности ИМ реальному объекту исследования осуществляется калибровка модели. Цель калибровки состоит в модификации вида функции $\varphi(X, G)$ модели таким образом, чтобы модель стала адекватно отображать поведение реальной системы. Обсуждению методики выполнения калибровки ИМ посвящен п. 4.2.4.

Исследование свойств ИМ предполагает последовательность выполнения следующих действий: оценки погрешностей имитации из-за неидеальности генераторов псевдослучайных чисел, используемых в ИМ; определения времени моделирования, необходимого для достижения заданной точности моделирования; оценки устойчивости результатов моделирования Y при различных изменениях параметров X и заданных значениях переменных G ; оценки чувствительности результатов моделирования к изменению параметров модели X и исходных данных моделирования G . Обсуждению методики исследования свойств ИМ посвящен § 4.3.

Весьма существенным технологическим этапом организации имитационных экспериментов на ЭВМ является планирование экспериментов. Роль планирования имитационных экспериментов достаточно высока. Поскольку «прогоны» вариантов модели обычно дороги, нужно стремиться к тому, чтобы количество параметров моделирования (размерность X) и число уровней каждого па-

раметра были минимальными и соответствовали целям эксперимента. Важными, с точки зрения автора, являются следующие вопросы: определение требуемого размера выборки статистик моделирования, используемых при вычислении значений \bar{Y} ; определение интервалов изменения параметров моделирования X ; планирование поиска ошибок в имитационном эксперименте. Существо и методики выполнения указанных процедур на этапе планирования имитационных экспериментов рассматриваются в § 4.4.

При наличии модели, которая успешно прошла испытания и точностные характеристики которой известны исследователю, и плана эксперимента, минимизирующего затраты средств и времени для решения задачи исследования с помощью ИМ, исследователь может приступить к эксплуатации ИМ. Важную роль при эксплуатации ИМ играет выбор методики обработки результатов эксперимента. В арсенале исследователя должны быть не только общеизвестные способы обработки результатов моделирования, но и набор специальных приемов по обработке результатов имитации. Мы остановимся на трех приемах, позволяющих, на наш взгляд, получить наиболее удачное представление результатов модельного эксперимента и существенно облегчить решение задачи моделирования (см. пп. 4.5.2—4.5.4).

Тщательное и полное документирование результатов имитационного эксперимента в целях дальнейшего использования этих результатов может значительно увеличить срок жизни модели и вероятность успешной ее реализации. Рассмотрению способов документирования результатов моделирования посвящен п. 4.5.5.

4.2. ИСПЫТАНИЕ ИМИТАЦИОННОЙ МОДЕЛИ

4.2.1. ЗАДАНИЕ ИСХОДНОЙ ИНФОРМАЦИИ ДЛЯ МОДЕЛИРОВАНИЯ

Процедура задания исходной информации для моделирования зависит от типа модели. Если мы имеем дело с функционирующей системой, то зачастую организуется измерение характеристик поведения системы \bar{Y}^*_q , рабочей нагрузки системы G^* и параметров функционирования системы Q^* . В ходе этих измерений определяются точностные характеристики каждой компоненты векторов G^* , Q^* , \bar{Y}^*_q . Важную роль при этом играют качество измерительных средств, их точностные характеристики. Одним из основных требований, выдвигаемых к измерителям параметров рабочей нагрузки и поведения системы, являются их малое влияние на алгоритмы функционирования реальной системы и соответствие их возможностей диапазону измеряемых значений характеристик.

При моделировании проектируемых систем, а также при моделировании функционирующих систем, когда нельзя организовать измерения, для формирования исходной информации проводят измерения на прототипах систем. Если таких прототипов не суще-

ствуется, то исследователю при построении модели приходится использовать экспертные оценки внешней нагрузки G^* , параметров функционирования Q^* и характеристик поведения системы Y^*_{Q} .

Итак, в ходе натурного эксперимента по результатам измерений параметров рабочей нагрузки G^* формируется модель рабочей нагрузки G . Степень соответствия модели рабочей нагрузки G параметрам рабочей нагрузки G^* определяется вектором точностей представления ее компонент δG и уровнем детализации алгоритма системы, т. е. точностью аппроксимации зависимости $\varphi^*(X^*, G^*)$ функцией $\varphi(X, G)$. Весьма существенно соблюдать при этом правило баланса точностей [67]. Согласно этому правилу необходимо соизмерять отклонения модели от описания системы с погрешностью в задании параметров описания модели. Должно быть достигнуто соответствие точностей представления отдельных элементов модели и модели входной нагрузки на систему.

Когда нет возможности определить значения компонент вектора G^* на основании измерений, приходится полагаться на субъективные (экспертные) оценки. Чаще всего при этом используется метод Дельфы [81]. Это итерационная процедура, которая позволяет подвергать мнение каждого эксперта критике со стороны всех остальных специалистов, не заставляя их фактически сталкиваться лицом к лицу. Координатор опроса регулирует процедуру анализа мнений и сохраняет их анонимность. Полученная групповая оценка усредняется и доводится до каждого члена группы экспертов. Затем экспертов просят пересмотреть свою оценку или кратко обосновать свое мнение, если оно выходит за рамки общепринятого. Членам группы сообщаются результаты второго тура и письменные объяснения предельных значений. При этом сохраняется их анонимность. После нескольких итераций координатор подводит итог и завершает процедуру экспертных оценок векторов G^* , Q^* , Y^*_{Q} .

При построении модели рабочей нагрузки G по значениям измеренных или установленных экспертным путем компонент вектора G^* широко применяются статистические методы проверки гипотез о близости [80]. В качестве задаваемых характеристик качества модели рабочей нагрузки на систему G используются уровень доверия β и вектор точностей моделирования переменных δG . В связи с этим перед исследователем возникает множество проблем, связанных с конкретным применением измерительных средств на реальных системах. Одна из проблем — это необходимость удовлетворения требованиям, чтобы измерители значений компонент вектора G не искажали их и по возможности не влияли на те физические явления и процессы, которые являются предметом моделирования. Дать общие рекомендации по этому вопросу вряд ли возможно. В каждом конкретном случае приходится вновь решать задачу измерения параметров в ходе натурных экспериментов на объекте исследования. Например, для моделирования вычисли-

тельного процесса на ЕС ЭВМ автору пришлось разработать семейство программ, обеспечивающих измерение инвариантных характеристик решаемых задач. Эти программы организованы в виде пакета задач исследования OSSTAN [5].

Пример. Для иллюстрации методики задания исходной информации ИМ по результатам натурного эксперимента используем пример моделирования производственного участка, изложенный в § 2.3. Допустим, что ИМ производственного участка реализована с помощью процессного способа имитации. Работа каждого из станков OCT_j представляет собой процесс. Компонентами вектора X являются интенсивности поступления деталей на обработку (λ_1, λ_2) . Под переменными модели G будем понимать вероятность поступления на обработку детали первого типа (p_1) и распределения вероятностей длительностей обслуживания OCT_j деталей i -го типа $F_{ij}(\tau)$. ИМ реализована таким образом, что позволяет фиксировать следующие статистики моделирования:

$\eta_j, j=\overline{1, 4}$, — коэффициенты загрузки деталями j -го станка;

l_j — усредненные длины очередей детали к OCT_j ;

$v_i, i=\overline{1, 2}$, — количество обслуженных деталей i -го типа;

T_i — усредненное значение общего времени обработки детали i -го типа производственным участком.

В качестве вектора откликов модели Y выступает вектор $(\eta_j, T_i), i=\overline{1, 2}; j=\overline{1, 4}$. Предположим, что $\varphi(X, G)$ имеет вид линейной зависимости. Необходимо с помощью имитационного эксперимента найти коэффициенты линейной регрессионной зависимости:

$$\eta_j = a_{0j} + a_{1j} + a_{1j} \lambda_1 \lambda_2 + a_{3j} \lambda_1 \lambda_2, \quad (4.1)$$

$$T_i = b_{0i} + b_{1i} \lambda_1 + b_{2i} \lambda_2 + b_{3i} \lambda_1 \lambda_2,$$

которую можно будет использовать затем для предсказания работы производственного участка.

Согласно изложенной методике проводим натурные эксперименты, в ходе которых фиксируем за интервал времени T^*_p : количество деталей каждого типа, поступивших на обработку (v^*_1, v^*_2) ; множество длительностей времен обслуживания деталей станками $\{\tau^*_{ijk}\}$, где k — порядковый номер детали i -го типа; множество длительностей времен обработки деталей производственным участком $\{T^*_{ik}\}, k=\overline{1, (v^*_1+v^*_2)}$. Определяем фактические интенсивности поступления деталей на обработку: $\lambda^*_1 = v^*_1/T^*_p$; $\lambda^*_2 = v^*_2/T^*_p$. Усредняя значения T^*_{ik} , находим T^*_1 и T^*_2 . С помощью $\{\tau^*_{ijk}\}$ вычисляются реальные значения коэффициентов загрузки OCT_j :

$$\eta_j^* = \sum_{i=1}^2 \sum_{k=1}^{v^*_1+v^*_2} \tau^*_{ijk}/T^*_p. \quad (4.2)$$

Множество $\{\tau^*_{ijk}\}$ используем для построения эмпирической функции распределения $F^*_{ij}(\tau)$. Итак, за один натурный эксперимент находим:

$$Y^*_Q = (T^*_i, \eta^*_j), \quad i=\overline{1, 2}; \quad j=\overline{1, 4}; \quad Q^* = (\lambda^*_1, \lambda^*_2); \quad G^* \sim E^*(\tau).$$

Для оценки ошибок измерения такие натурные эксперименты проводим не менее пяти раз. Затем определяем средние значения характеристик и дисперсии компонент Y^*_Q и Q^* . Эмпирические функции распределения $F^*_{ij}(\tau)$ при этом объединяются известными способами [33].

Для построения модели внешней нагрузки G аппроксимируем $F_{ij}^*(\tau)$ с уровнем доверия β , например, методом моментов [33] и находим параметры распределения $F_{ij}(\tau)$, что эквивалентно заданию G . Различие между $F_{ij}^*(\tau)$ и $F_{ij}(\tau)$ и определит точность задания переменных модели.

4.2.2. ВЕРИФИКАЦИЯ ИМИТАЦИОННОЙ МОДЕЛИ

Верификация имитационной модели состоит в доказательстве утверждений соответствия алгоритма ее функционирования замыслу моделирования путем формальных и неформальных исследований реализованной программы модели.

Неформальные исследования программы модели имеют целью убедить исследователя и заказчика ИМ в соответствии построенной модели замыслу, заложенному конструктором при ее разработке. Они представляют ряд процедур проверки правильности алгоритма функционирования модели объекта исследования. Проведение этих процедур входит в состав комплексной отладки ИМ. Вся последовательность действий комплексной отладки, рекомендованная в гл. 2, направлена на достижение у разработчика уверенности в том, что поведение реализованной программы модели соответствует замыслу и своему назначению. Важно при этом тщательно отладить и проверить поведение программы той части модели, которая прогнозирует работу системы, т. е. имитирует режимы работы, в которых исследователь не мог наблюдать поведение реальной системы.

Формальные методы верификации программы модели включают в себя: использование специальных препроцессоров и постпроцессоров в качестве «читателей» программ; замену некоторых стохастических элементов модели детерминированными и проверку на «ожидаемость» результатов моделирования; использование теста на «непрерывность» моделирования. Препроцессоры и постпроцессоры анализируют текст программы модели путем доказательства логических утверждений и выдают исследователю сообщения о возможном наличии логических ошибок в тексте модели. Если при замене стохастических элементов модели детерминированными исследователь получает очень далекие от ожидаемых ответы программ, то это означает, что программная реализация модели неудачна. В ходе тестирования на «непрерывность» моделирования проверяется соответствие выходных характеристик воздействиям на входе модели по всему диапазону значений параметров модели X .

4.2.3. ПРОВЕРКА АДЕКВАТНОСТИ МОДЕЛИ

Оценка адекватности модели объекту исследования проводится чаще всего для случая, когда можно определить значение отклика системы в ходе натуральных испытаний. Пусть известен отклик реальной системы Y^*_Q при нагрузке G^* , параметрах Q^* и неизвестной из-за сложности и малой изученности протекающих в

объекте процессов функции $Y^*_{Qk} = \varphi^*(Q^*, G^*)$. Модель представляет собой аппроксимирующую зависимость $Y_k = \varphi(Q_k, G)$, $k = \overline{1, N}$ (k и N — соответственно номер и число опытов на модели и реальной системе), найденную в ходе эксплуатации объекта по результатам наблюдений входных воздействий Q^*_{kh} , $k = \overline{1, N}$, и выхода Y^*_{Qkh} при заданных значениях G^* .

Проверку адекватности модели реальному объекту исследования можно проводить различными способами. Мы остановимся на трех наиболее употребительных способах: по средним значениям откликов модели и системы; по дисперсиям отклонений откликов модели от среднего значения откликов системы; по максимальному значению абсолютных отклонений откликов модели от откликов системы.

При первом способе проверяется гипотеза о близости средних значений каждой n -й компоненты откликов модели \bar{Y}_n известным средним значениям n -й компоненты откликов реальной системы \bar{Y}^*_{Qn} . Проводят N_1 опытов на реальной системе и измеряют по каждой n -й компоненте откликов системы выборки значений $\{Y^*_{Qnk}\}$, $k = \overline{1, N_1}$. Выполняют N_2 опытов на модели системы и получают по тем же n -м компонентам откликов модели выборки значений $\{Y_{nh}\}$, $k = \overline{1, N_2}$. Обычно стараются, чтобы объемы выборок были одинаковы ($N_1 = N_2$), но в ряде случаев натурные эксперименты весьма дороги и поэтому $N_2 > N_1$. По выборкам вычисляются оценки математического ожидания и дисперсии откликов модели и системы с помощью следующих соотношений:

$$\bar{Y}_n = \frac{1}{N_2} \sum_{k=1}^{N_2} Y_{nh}; \quad D_n = \frac{1}{N_2 - 1} \sum_{k=1}^{N_2} (Y_{nh} - \bar{Y}_n)^2; \quad (4.3)$$

$$\bar{Y}^*_{Qn} = \frac{1}{N_1} \sum_{k=1}^{N_1} Y^*_{Qnk}; \quad D^*_n = \frac{1}{N_1 - 1} \sum_{k=1}^{N_1} (Y^*_{Qnk} - \bar{Y}^*_{Qn})^2.$$

Основой проверки гипотезы является разность $E_{1n} = (\bar{Y}_n - \bar{Y}^*_{Qn})$, оценкой дисперсии которой будет

$$D_{pn} = \frac{(N_1 - 1) D_n + (N_2 - 1) D^*_n}{N_1 + N_2 - 2}. \quad (4.4)$$

Величины $(\bar{Y}_n - \bar{Y}^*_{Qn})$ и D_{pn} являются статистически независимыми, поэтому можно использовать t -статистику:

$$t_n = (\bar{Y}_n - \bar{Y}^*_{Qn}) \sqrt{\frac{N_1 N_2}{D_{pn} (N_1 + N_2)}}. \quad (4.5)$$

Таким образом, можно воспользоваться таблицей распределения t -статистики [13], взяв число степеней свободы равным $\gamma = N_1 + N_2 - 2$. Обычно задаются уровнем зависимости $\alpha = 0,05$ и при конкретном значении числа степеней свободы γ по таблицам находят критическое значение t -статистики ($t_{кр}$). Если выполняется неравенство $t_n \leq t_{кр}$, то гипотеза о близости средних значений n -й компоненты откликов модели и системы принимается. Только при

близости откликов по всем компонентам векторов \mathbf{Y} и \mathbf{Y}^*_q можно говорить об адекватности модели объекту по первому способу.

При втором способе для каждой n -й компоненты откликов проверяется гипотеза о значимости различий оценок двух дисперсий D^*_{on} и D_{on} ; D^*_{on} определяется из соотношения (4.3), а D_{on} представляет собой оценку дисперсии отклонения откликов модели от среднего значения, полученного при проведении натурных экспериментов:

$$D_{on} = \sum_{k=1}^{N_2} (Y_{nk} - \bar{Y}_{qn}^*)^2 / (N_2 - 1). \quad (4.6)$$

Для сравнения дисперсий составляется F -статистика [13]: $F = D_{on} / D^*_{on}$. Отметим, что поскольку F всегда должна быть больше единицы, то в числителе берется большая из оценок дисперсий. Чаще всего выполняется неравенство $D_{on} \geq D^*_{on}$. Обычно задаются уровнем значимости $\alpha = 0,05$ и при конкретных значениях степеней свободы $\gamma_1 = \gamma_2 = N_2$ (при данном способе проверки адекватности стремятся к $N_2 = N_1$) по таблицам для F -распределения Фишера [13] находят критическое значение F -статистики ($F_{кр}$). Выполнение неравенства $F > F_{кр}$ указывает на то, что гипотеза о значимости различий двух оценок дисперсий принимается и отсутствует адекватность между n -ми компонентами откликов модели и реальной системы. Если отсутствует адекватность модели хотя бы по одной из компонент, то считают, что модель неадекватно отображает реальную систему.

При третьем способе проверяется соответствие отклонений откликов модели и реальной системы по каждой компоненте. Отклонения этих откликов должны быть не более заданной величины. Этот способ применяется при проверке адекватности моделей систем управления. Аналогично первому способу для каждой n -й компоненты откликов формируют две выборки: $\{Y_{nk}\}$ и $\{Y^*_{qnk}\}$, $k = \overline{1, N_1 = N_2}$. По выборкам определяют вектор отклонений откликов модели от откликов реальной системы в процентах. Компонентами этого вектора являются

$$\delta Y_n = \max_k \frac{|Y_{nk} - Y^*_{qnk}|}{\bar{Y}_{qn}^*} 100\%; \quad (4.7)$$

где \bar{Y}_{qn}^* — среднее значение n -й компоненты отклика системы, определяемое по формуле (4.3). Таким образом, δY_n представляет собой максимальное значение отклонений n -й компоненты откликов модели в N_2 модельных экспериментах от соответствующих откликов реальной системы, нормированное средним значением n -й компоненты отклика реальной системы, принимаемым в качестве истинного значения. Задавшись процентным значением допустимого отклонения каждой компоненты откликов модели от откликов системы δY_d , проверяют выполнение неравенства $\delta Y_n \leq \delta Y_d$. Невыполнение данного неравенства хотя бы по одной

Таблица 4.1

Пример проверки адекватности модели производственного участка

Обозначение отклика	Значения составляющих выборки при					Среднее значение отклика	Оценка дисперсии отклика $D_n \sqrt{D^* n}$	Дисперсия разности D_{pn}	Значения t -статистики t_n
	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$				
T_1 , мин	18	19	18	17	17	17	1,50		
T^*_{11} , мин	19	20	17	16	18	18	1,50	1,50	1,29
T_2 , мин	19	23	19	22	22	21	4,50		
T^*_{22} , мин	20	24	23	22	21	22	2,50	3,50	0,84
η_1	0,90	0,88	0,94	0,97	0,91	0,92	0,00150		
η^*_{11}	0,88	0,86	0,92	0,95	0,89	0,90	0,00125	0,00127	0,88
η_2	0,90	0,88	0,84	0,83	0,90	0,87	0,00110		
η^*_{22}	0,88	0,85	0,82	0,84	0,86	0,85	0,00035	0,00072	1,10
η_3	0,63	0,67	0,69	0,64	0,67	0,66	0,00060		
η^*_{33}	0,62	0,62	0,68	0,67	0,66	0,65	0,00080	0,00070	0,58
η_4	0,67	0,70	0,73	0,72	0,68	0,70	0,00065		
η^*_{44}	0,69	0,73	0,75	0,68	0,70	0,71	0,00085	0,00075	0,55

компоненте вектора отклика модели ставит под сомнение адекватность модели реальной системе.

Пример. Проиллюстрируем первый из указанных способов проверки адекватности на нашем примере моделирования производственного участка обработки деталей. Всего было проведено пять натурных экспериментов ($k=\overline{1,5}$). В ходе k -го эксперимента на производственном участке были измерены значения компонент векторов

$$Y_{Qk} = (T_{1k}^*, T_{2k}^*, \eta_{1k}^*, \eta_{2k}^*, \eta_{3k}^*, \eta_{4k}^*).$$

На ИМ при тех же значениях параметров системы $Q_k = Q^*$ измеряем значения компонент отклика модели

$$Y_k = (T_{1k}, T_{2k}, \eta_{1k}, \eta_{2k}, \eta_{3k}, \eta_{4k}).$$

Результаты натурных и модельных экспериментов представлены в табл. 4.1. Используя выражения (4.3), вычислим значения оценок математического ожидания и дисперсий и запишем в табл. 4.1. По формуле (4.4) определим оценку D_{pn} дисперсии разности ($Y^*_{Qn} - Y_n$) для каждой n -й компоненты откликов модели и производственного участка и запишем в табл. 4.1. Используя выражение (4.5), найдем шесть значений t -статистики и занесем в табл. 4.1. Число степеней свободы t -статистики равно $\gamma = 5 + 5 - 2 = 8$. Для уровня значимости $\alpha = 0,05$ при восьми степенях свободы находим критическое значение t -статистики ($t_{кр} = 1,85$). Сравнивая каждое из значений t -статистики в табл. 4.1. с $t_{кр}$, видим, что можно принять гипотезу о близости средних значений по каждой компоненте откликов модели и реальной системы.

4.2.4. КАЛИБРОВКА ИМИТАЦИОННОЙ МОДЕЛИ

Когда модель оказывается неадекватной реальной системе, исследователь переходит к калибровке ИМ. Цель калибровки состоит в уменьшении неточностей формулировки ИМ, обусловлен-

ных ошибочной или недостаточно подробной моделью $\varphi(X, G)$. Исследователь может корректировать либо модель внешней нагрузки G , либо алгоритм функционирования модели $\varphi(X, G)$. В ходе калибровки модели сложной системы можно проводить изменения трех типов: глобальные структурные изменения (например, добавление программ процессов, изменения событий в процессах); локальные изменения в модели (например, замена компонент модели внешне эквивалентными, но более точными); изменение некоторых параметров, используемых в качестве калибровочных.

Можно предположить, что структурные изменения как более трудно осуществимые и дорогие должны рассматриваться в последнюю очередь, когда все попытки откалибровать модель путем изменения параметров и локальных модификаций окажутся безуспешными. На самом деле при проведении калибровки модели существует опасность появления структурных несоответствий или недостаточной степени детальности описания реальных явлений системы в модели, которые могут вызвать сокращение области пригодности модели. Поэтому исследователи стараются «подправить» само описание модели и таким способом откалибровать модель.

При глобальной модификации модели исследователю можно рекомендовать методику, изложенную в [80]. Калибровка модели согласно этой методике состоит из трех шагов: сравнения между собой распределений Y^*_Q и Y ; балансировки модели; оптимизации модели. Каждый из этих шагов является по своей природе итеративным. Он включает ряд изменений модели, за каждым из которых следует тест, который определяет, завершен этот шаг или его следует продолжить.

Шаг 1. Сравнение распределений. Определяется погрешность моделирования, которая должна быть минимизирована в процессе калибровки. В качестве такой погрешности может выступать ранее определенная величина E_{1n} — разность между средними значениями компонент векторов Y^*_Q и Y . В ходе калибровки пытаются изменять алгоритмы модели таким образом, чтобы минимизировать погрешность E_{1n} . Однако более удовлетворительное решение вопроса имеет место при использовании в качестве сравниваемых показателей функций распределения вероятностей значений компонент $F(Y^*_{Qn})$ и $F(Y_n)$. При этом сравнение производится локально (поточечно). Для сравнения выбираются абсолютные значения разностей между функциями распределений $|F(Y^*_{Qn}) - F(Y_n)|$, чтобы избежать частичной компенсации погрешностей из-за различия знаков. Мету расхождения функций распределения вероятностей значений компонент векторов Y^*_Q и Y обозначим E_{2n} . Для одних и тех же выборок Y^*_Q и Y она зависит от статистического критерия сравнения выборок. Если используется критерий Пирсона, то в качестве E_{2n} выступает статистика хи-квадрат [13], если критерий Колмогорова, то в качестве E_{2n} применяется D -статистика Колмогорова [13].

При использовании критерия Пирсона исследователь задается уровнем значимости α и при числе степеней свободы, определяемом по числу интервалов группирования L функций распределения $F(Y^*_{qn})$, находит критическое значение величины $\chi^2(\alpha, L-1)$. Если выполняется неравенство $E_{zn} \leq \chi^2(\alpha, L-1)$, то считают, что различие между функциями распределения незначимо и достигнута близость между n -ми компонентами откликов модели и системы. Проверая значимость различия между функциями распределения по всем компонентам, исследователь определяет, **улучшает ли проведенная модификация компонент модель или нет**. Отметим, что данный шаг предполагает макроскопические изменения модели, например модификацию структуры или замену главных составляющих модели.

Шаг. 2. Балансировка модели. В ходе натурального эксперимента была зафиксирована выборка значений вектора Y^*_q при некотором изменении вектора параметров системы Q^* . Поэтому перед этапом балансировки модели исследователь располагает выборками объема V, N значений каждой n -й компоненты отклика реальной системы $\{Y^*_{qnk}\}$, $k = \overline{1, N}$, при последовательном изменении вектора параметров системы Q^* . На каждой итерации процедуры балансировки модели исследователь меняет значения вектора параметров Q аналогично натурному эксперименту и фиксирует выборки объема N откликов модели системы $\{Y_{nh}\}$, $k = \overline{1, N}$.

Определяются средние значения F_{zn} квадратов разностей компонент откликов Y^*_{qnk} и Y_{nh} соответственно в натурном и модельном экспериментах и по возможности минимизируются

$$E_{zn} = \frac{1}{N} \sum_{k=1}^n (Y^*_{qnk} - Y_{nh})^2 \rightarrow \min,$$

где k — номер эксперимента; Y^*_{qnk} и Y_{nh} — значения откликов n -й компоненты соответственно в k -м натурном и модельном экспериментах.

Составляется уравнение регрессии

$$E_{zn} = \sum_{q=0}^h \beta_q Q_q.$$

Оцениваются коэффициенты β_q этой зависимости, где h — размерность вектора параметров. Затем проверяется значимость коэффициентов уравнения регрессии β_q при выбранном уровне значимости. Если коэффициенты уравнения окажутся значимыми, то на следующей итерации меняются те параметры модели, которые позволяют уменьшить эту значимость. Целью этой процедуры, которая называется методом хорошего баланса, является получение незначимой регрессии E_{zn} .

Шаг. 3. Оптимизация модели. Некоторые параметры системы могут оказаться наиболее существенными для изменения откликов реальной системы Y^*_q . Поэтому не обязательно стремиться к то-

му, чтобы по всем параметрам регрессия E_{3n} была незначима. Достаточно достичь незначимости регрессии E_{3n} только для наиболее существенных параметров. Такие параметры называют калибровочными. На третьем шаге калибровки ИМ рассматриваются только изменения модели, связанные с калибровочными параметрами. Пусть \mathbf{H} — вектор калибровочных параметров, а \mathbf{H}_n — наилучшее значение \mathbf{H} среди тех, которые использовались при различных прогонах модели системы $\Phi(\mathbf{X}, \mathbf{G})$, так как оно дает минимальное значение погрешности E_{3n} .

Строится линейная регрессия между калибровочными параметрами и E_{3n} , основанная только на данных, собранных во время прогонов модели при соответствующих значениях \mathbf{H} . Задав уровень значимости α , проверяют значимость этой линейной регрессии. Если регрессия значима, уравнения регрессии, содержащие только те члены, коэффициенты которых значимы, могут рассматриваться как локальная аппроксимация функции $E_{3n}(\mathbf{H})$ после исключения шумовой компоненты. Направление вектора параметров \mathbf{V} этой регрессии близко к направлению вектора наискорейшего спуска этой функции. Таким образом, следующим «хорошим» значением \mathbf{H} окажется $\mathbf{H}_{\text{нм}} = \mathbf{H}_n - c\mathbf{V}$ ($c > 0$ — константа). Следующий прогон модели выполняется уже с новым значением $\mathbf{H}_{\text{нм}}$. Определяются новые значения погрешностей E_{3n} , которые сравнивают с ранее полученными их значениями при \mathbf{H}_n . Так как погрешности E_{3n} являются случайными величинами, то сравнения любых двух значений еще недостаточно для определения, является ли модифицированная модель значимо лучшей. Поэтому сравниваются последовательности пар наблюдений E_{3n} , представляющие собой квадраты разностей компонент \mathbf{Y}^*_q и \mathbf{Y} . Для каждой пары наблюдений проверяется, является ли модифицированная модель системы лучшей, чем предыдущая. Если улучшение характеристик модели достигнуто, то составляется новая регрессионная зависимость уже вблизи $\mathbf{H}_{\text{нм}}$ и вся процедура проверок повторяется. В противном случае этот шаг и все процедуры калибровки модели завершаются.

Все эти сравнения поведения модели и системы не могут быть исчерпывающими, так как модель нельзя откалибровать для всех экспериментальных условий, в которых она должна работать. Это связано с большим количеством условий, возникающих зачастую непредсказуемо. В свою очередь, если бы была выполнена исчерпывающая калибровка с данными измерений, модель оказалась бы бесполезной, поскольку не требовалось бы предсказывать поведение системы. Модель обычно используется в условиях, в которых ее точность не всегда может быть проверена, поэтому калибровка обычно ограничена несколькими комбинациями параметров системы.

Итак, при заданных определенных требованиях к точности в терминах допустимых погрешностей мы говорим, что модель пригодна для некоторого входного условия, если она удовлетворяет этим требованиям, когда ее переменные соответствуют этому ус-

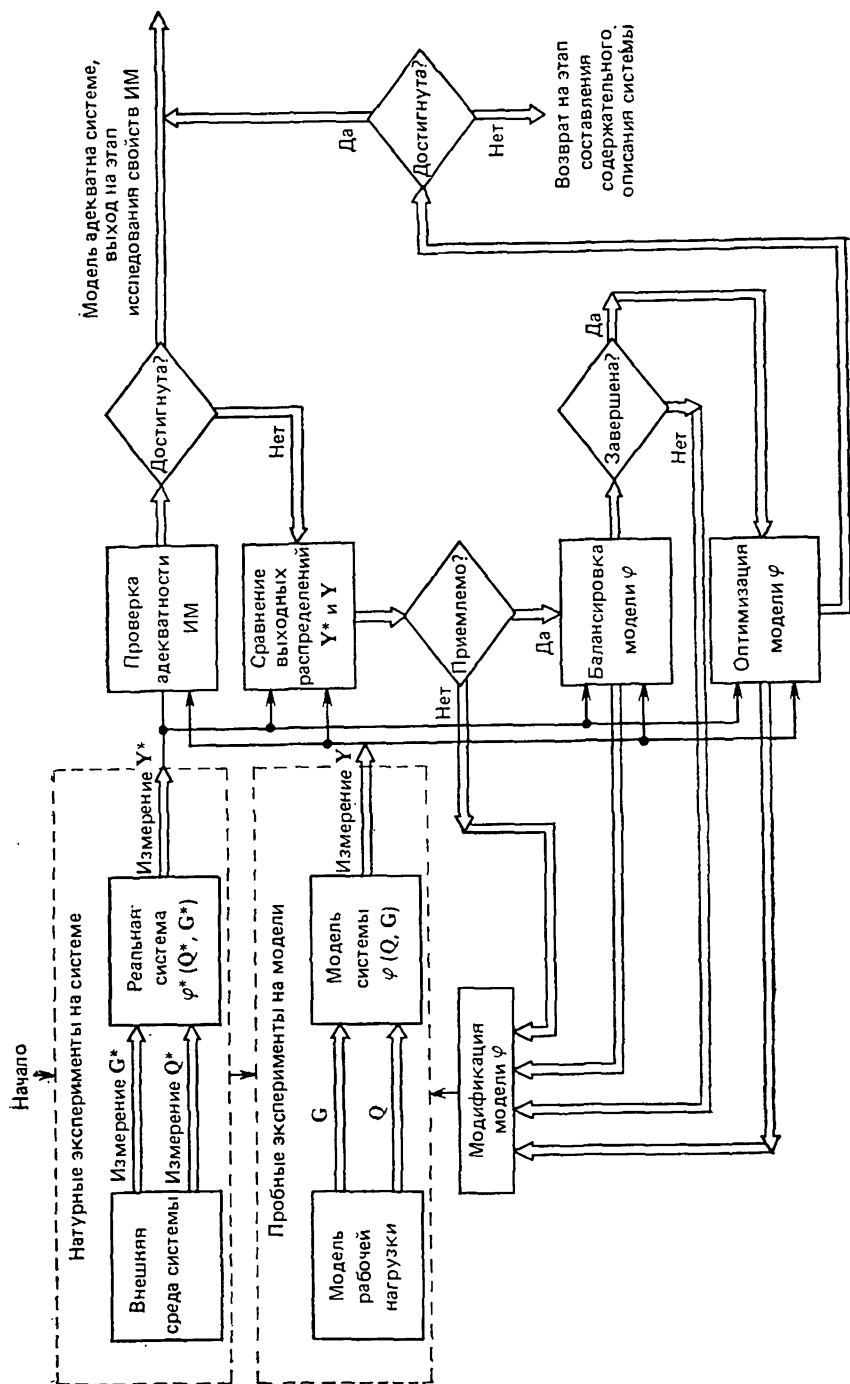


Рис. 4.1. Схема итеративной процедуры проверки адекватности и калибровки имитационной модели

ловию. Область пригодности модели есть множество входных условий, для которых пригодна модель. При калибровке модели мы добиваемся ее пригодности для входных условий моделирования.

Схема изложенной выше трехшаговой итеративной процедуры калибровки модели Φ реальной системы представлена на рис. 4.1. Если в результате процедуры оптимизации модели адекватность ее объекту исследования не достигнута, то необходимо вернуться на этап составления содержательного описания системы, добавить недостающую информацию о системе и провести весь процесс построения модели заново. Для более углубленного изучения данной проблемы можно рекомендовать работы [41, 80].

4.3. ИССЛЕДОВАНИЕ СВОЙСТВ ИМИТАЦИОННОЙ МОДЕЛИ

После того как завершена верификация ИМ и достигнута адекватность модели реальному объекту или если для проектируемых систем исследователь получил некоторую уверенность в том, что модель соответствует своему назначению, необходимо исследовать свойства ИМ. Исследователь должен ясно представлять возможности построенной ИМ. Наиболее существенными, с точки зрения автора, являются следующие процедуры исследования модели: оценка погрешности имитации, обусловленной наличием в ИМ неидеальных генераторов псевдослучайных чисел; определение длительности переходного режима в работе ИМ; оценка устойчивости результатов имитации исследуемых процессов; исследование чувствительности ИМ, т. е. зависимости изменения Y от изменения X . Рассмотрим порядок действий исследователя при выполнении каждой из перечисленных процедур. Для иллюстрации используем модель производственного участка, построение которой освещалось в гл. 3.

4.3.1. ОЦЕНКА ПОГРЕШНОСТИ ИМИТАЦИИ, ОБУСЛОВЛЕННОЙ НАЛИЧИЕМ В ИМ ГЕНЕРАТОРОВ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

В любой вероятностной ИМ сложной системы применяются генераторы псевдослучайных чисел. Как правило, система автоматизации моделирования предоставляет в распоряжение исследователя несколько различных генераторов псевдослучайных чисел. Каждый из них использует базовый генератор, числа с выхода которого затем преобразуются для получения псевдослучайных величин с заданными законами распределения. Оказывается, что такой генератор является источником погрешности имитации. В силу особенностей алгоритма работы генераторов псевдослучайных чисел результаты нескольких прогонов ИМ при одних и тех же значениях параметров X и переменных G будут отличаться друг от друга.

Некоторые исследователи по неопытности применяют единый генератор, обращаясь к нему из различных мест ИМ в ходе ими-

тации реальных функциональных действий. В этом случае процесс моделирования может быстро выродиться из-за выхода псевдослучайной последовательности за пределы отрезка аперiodичности [67]. Зачастую исследователи не проверяют базовые генераторы не только на «длину отрезка аперiodичности», но даже на «истинную равномерность».

Для проверки качества генераторов псевдослучайных чисел имеется множество тестов. Для начального знакомства с методами тестирования можно рекомендовать работы [28, 67]. Существует проблема компромисса между точностью воспроизведения псевдослучайных величин согласно выбранному закону их распределения и сложностью реализации таких генераторов. Еще А. А. Колмогоров выявил тесную связь между «степенью случайности» последовательности и сложностью ее представления. Поэтому широко распространенные датчики, использующие простые алгоритмы имитации псевдослучайных чисел, в принципе не могут быть хорошего качества. Но применение датчиков со сложными алгоритмами имитации зачастую не представляется возможным из-за ограничений ресурса времени ЭВМ на моделирование вариантов сложной системы. Что же делать исследователю в подобных ситуациях? Оценить эту погрешность. В ряде случаев после определения погрешности, связанной с использованием генераторов псевдослучайных чисел, может оказаться, что точность имитации приемлема и проблема отсутствует. Чем и как оценить эту погрешность? Как показала практика исследования вычислительных систем на ИМ, зачастую достаточно определить оценки математического ожидания и дисперсий отклонений компонент вектора откликов Y по следующей процедуре, реализованной в МК PLSIM [51].

В серединной точке области изменения параметров модели X организуется несколько (автор рекомендует $N=10$) прогонов модели с одними и теми же значениями X и G , но с разными начальными значениями для базовых генераторов псевдослучайных чисел ξ_{0k} [67]. Для каждого k -го прогона модели вычисляются значения n -й компоненты вектора отклика Y_{nk} . В результате получают выборки значений отклика $\{Y_{nk}\}$, $k=\overline{1, 10}$. По выборкам определяют оценки математического ожидания и дисперсии. Такими оценками являются соответственно \bar{Y}_n и \bar{D}_n , вычисляемые по формулам (4.3). Определяем доверительный интервал нахождения истинного значения математического ожидания n -й компоненты отклика Y_{ni} . Мы имеем дело с погрешностями, порождаемыми генераторами псевдослучайных чисел, поэтому можно допустить нормальность распределения отклонения Y_{nk} от истинного значения Y_{ni} . Поскольку объемы выборок малы ($k < 30$), то для нахождения доверительных интервалов используется t -статистика

$$t = (\bar{Y}_n - Y_{ni}) \sqrt{(N-1)/D_n},$$

имеющая распределение Стьюдента. Задавшись уровнем значимости $\alpha=0,05$, мы с вероятностью 0,95 можем утверждать, что истинное значение Y_{ni} лежит в пределах

$$\bar{Y}_n - t_{0,05} \sqrt{D_n/(N-1)} \leq Y_{ni} \leq \bar{Y}_n + t_{0,05} \sqrt{D_n/(N-1)},$$

где $t_{0,05}$ — значение t -статистики, определяемое при $(N-1)$ степенях свободы и уровне значимости $\alpha=0,05$.

Доверительный интервал для среднего значения n -й компоненты вектора отклика (при $N=10$ и $\alpha=0,05$) можно записать в виде

$$\bar{Y}_n \pm d_n = \bar{Y}_n \pm 0,753 S_{1n}. \quad (4.8)$$

Значение d_n и определяет погрешность n -й компоненты отклика модели. Затем из всех d_n находят максимальное значение, которое в данном случае и будет верхней границей погрешности, связанной с использованием в ИМ генераторов псевдослучайных величин.

После того как погрешность оценена и она оказалась достаточно высокой, исследователь должен пересмотреть состав генераторов псевдослучайных чисел и процедуру использования самих генераторов в ходе имитации. В таких случаях можно рекомендовать выбрать большее число базовых генераторов псевдослучайных чисел с обязательным хранением своего очередного начального значения ξ_{0ij} для каждой ij -й активности модели. Для увеличения периода апериодичности генераторов псевдослучайных чисел рекомендуются комбинированные алгоритмы [67].

Пример. Применение предложенной процедуры для модели производственного участка сводится к следующему. При одних и тех же значениях G и X (соответственно $F_{ij}(\tau)$; λ_1, λ_2) проводим, например, $N=10$ экспериментов. Эксперименты отличаются друг от друга начальными значениями пяти базовых генераторов ξ_{0q} ($q \leq 5$). Причем в ходе имитации к этим генераторам возможно обращение из пяти мест алгоритма моделирования процесса обработки деталей при определении длительности обработки детали j -м станком ($q=j$) по функциям распределения $F_{ij}(\tau)$ и при розыгрыше типа детали (i) по вероятности p_i . Таким образом, варианты моделирования производственного участка отличаются друг от друга значениями вектора (ξ_{0q}) , $q=\overline{1, 5}$. Для каждого k -го варианта ($k=\overline{1, N}$, $N=10$) фиксируется вектор откликов модели

$$Y_k = (\eta_{jk}, T_{ik}), \quad i=\overline{1, 2}; \quad j=\overline{1, 4}.$$

В результате получим шесть выборок значений компонент откликов $\{\eta_{jk}\}$ и $\{T_{ik}\}$. По формулам (4.3) определяем оценки $\bar{\eta}_j$, T_i , $S^2_{1T_i}$ и $S^2_{1\eta_j}$.

По формуле (4.8) находим значения доверительных интервалов d_n , $n=\overline{1, 6}$. Верхней границей погрешности, связанной с использованием пяти генераторов псевдослучайных чисел в ИМ производственного участка, будет максимальное значение из d_n .

4.3.2. ОПРЕДЕЛЕНИЕ ДЛИТЕЛЬНОСТИ ПЕРЕХОДНОГО РЕЖИМА

Одно из основных отличий имитационных экспериментов от натурных испытаний объекта моделирования заключается в простоте повторения и воспроизведения условий эксперимента, а также в простоте запуска, прерывания и возобновления эксперимента. Это позволяет экспериментатору полностью контролировать модельный эксперимент. Однако необходимо заботиться о том, как задать начальные условия и когда приступить к сбору данных. Обычно имитационные модели применяются для изучения системы в типичных для нее и повторяющихся изо дня в день условиях. К сожалению, в большинстве стохастических моделей требуется некоторое время T_0 для достижения моделью необходимого установившегося состояния. Поэтому исследователь должен позаботиться об уменьшении влияния начального периода моделирования или его исключении из результатов моделирования.

Существуют три способа уменьшения влияния начального периода на динамику моделирования сложной системы [81]: использование длинных прогонов модели; исключение из рассмотрения начального периода прогона; выбор такого начального условия, которое ближе всего к типичному, и тем самым существенное уменьшение длительности переходного режима.

Первый подход можно применять только в случае, если прогон модели не требует много машинного времени. При втором подходе часть машинного времени тратится бесполезно и из-за сокращения объема выборки увеличивается дисперсия отклика модели Y . При этом весьма трудно установить, когда закончился переходный режим. Не существует полностью надежных методов для решения вопроса о том, достигнуто ли установившееся состояние или нет. Установившееся состояние вовсе не означает, что переменная отклика Y достигла некоторого постоянного уровня. Переменная отклика Y продолжает флуктуировать присущим ей образом. Под статистическим равновесием или установившимся состоянием мы понимаем такое состояние, в котором противодействующие влияния сбалансированы и компенсируют друг друга. Предполагается, что для каждой стохастической модели существует распределение вероятностей значений отклика, являющееся характеристикой системы. Модель находится в равновесии, когда отклик не выходит за предельные значения.

Для отделения переходного режима от стационарного у исследователя должна быть возможность наблюдения за моментом входа контролируемого параметра моделирования в стационарный режим. Это можно осуществить, например, путем наблюдения за длиной очереди заказов к наиболее загруженной компоненте модели. Такой контроль может быть выполнен на любой компоненте модели, если исследователь предполагает, что она позже всех входит в стационарный режим. Необходимо задать временной интервал, по окончании которого происходит вычисление оценки средней длины очереди. Проверку можно выполнить

по критерию Вилкоксона [13], который требует задания уровня значимости. С интервалов времени Δt проводится сравнение соседних значений средних длин очереди l_k и l'_k ($k=1, m$), где m — объем выборки; k — номера замеров длины очереди. С помощью критерия Вилкоксона проверяется гипотеза H_0 об однородности двух выборок: $\{l_k\}$ и $\{l'_k\}$. Предполагается, что обе выборки взаимно независимы и извлечены из одной и той же генеральной совокупности и, следовательно, функции распределения случайных величин l_k и l'_k одинаковы. Эту гипотезу можно выразить тождеством

$$H_0: p\{l \leq x\} \equiv p\{l' \leq x\}, \quad x < \infty,$$

и воспользоваться для ее проверки ранговым критерием. На первом этапе реализации процедуры проверки гипотезы об однородности производится объединение случайных величин l_k и l'_k в один вариационный ряд в порядке возрастания их значений. В результате формируется последовательность типа

$$\begin{array}{cccccccccccccccc} A & A & B & A & A & A & B & B & B & B & \dots & A & & B & & B \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & \dots & N-2 & N-2 & N \end{array}$$

где буквами A и B обозначены члены вариационного ряда, принадлежащие выборкам $\{l_k\}$ и $\{l'_k\}$ соответственно. Каждому члену объединенной выборки приписываются ранги r_q , обозначающие порядковые номера в объединенной выборке, $q=1, N$, $N=2m$. На основании установленных рангов вычисляется значение статистики Вилкоксона

$$W_l = \sum_{q=1}^N r_q.$$

Проверка гипотезы об однородности производится на основе неравенства

$$w_1(\alpha) \leq W_l \leq w_2(\alpha),$$

где $w_2(\alpha)$, $w_1(\alpha)$ — верхняя и нижняя границы статистики Вилкоксона при уровне значимости α ($0 < \alpha < 1$). Если неравенство выполняется, то гипотеза H_0 принимается. Нижняя и верхняя границы вычисляются исходя из соотношений:

$$w_1(\alpha) = \frac{m(2m+1)-1}{2} - \psi \sqrt{\frac{m^2(2m+1)}{12}};$$

$$w_2(\alpha) = m(2m-1) - w_1(\alpha),$$

где $\psi = \arg \Phi(1+\alpha/2)$; $\arg \Phi(x)$ означает функцию, обратную функции нормального закона распределения.

Необходимо обратить внимание на трудность выбора контролируемого параметра, поскольку эта процедура имеет неформальный характер. В ряде случаев исследователи строят графики изменения значений контролируемого параметра в модельном времени t_0 и по виду этого графика устанавливают, как отделить пе-

реходный режим. Сбор статистики моделирования начинается именно с этого значения модельного времени. Некоторые средства автоматизации моделирования [51] позволяют для определения момента входа модели в стационарный режим использовать специальные процессы-контролеры. Эти процессы функционируют по различным алгоритмам, описываемым самим пользователем и реализуемым в виде специальных процедур.

Пример. В случае моделирования производственного участка в качестве контролируемого параметра можно выбрать длину очереди деталей ко второму станку (l_2). В самом деле, с помощью очереди к первому станку длиной l_1 осуществляется развязка между данным производственным участком и внешним окружением (другими производственными участками). Поэтому l_1 может определяться несогласованностью длительностей обработки деталей и интенсивностью их поступления на обработку. Очереди к третьему и четвертому станкам длиной соответственно l_3 и l_4 характеризуют скорость окончательной обработки только своего типа деталей. Поэтому для выбора остается параметр l_2 . Именно за его значением и следует наблюдать в ходе имитации с заданным шагом проверки Δt в модельном времени t_0 . На рис. 4.2 представлен график изменения l_2 в модельном времени t_0 . По виду графика находим, что моментом завершения переходного периода является t_n . Для $t_0 \geq t_n$ значение l_2 колеблется вокруг его среднего значения \bar{l}_2 с некоторой допустимой амплитудой изменения.

Не всегда целью моделирования является исследование поведения системы в стационарном режиме. Бывают случаи, когда предметом имитации является переходный режим, имеющий место в реальной системе. Но и в этих случаях важно уметь уловить момент окончания переходного режима в модели, чтобы завершить имитацию, не затратив лишних ресурсов ЭВМ. Исследователь может оказаться и в такой ситуации, когда за максимально возможное в реальных условиях время моделирования варианта системы не удастся достичь стационарного режима. Вместе с тем он знает, что все результаты моделирования могут иметь смысл только при достижении стационарного режима в системе. Несмотря на то, что в такой ситуации исследователь мало доверяет результатам имитации, он все же вынужден ими пользоваться, по-

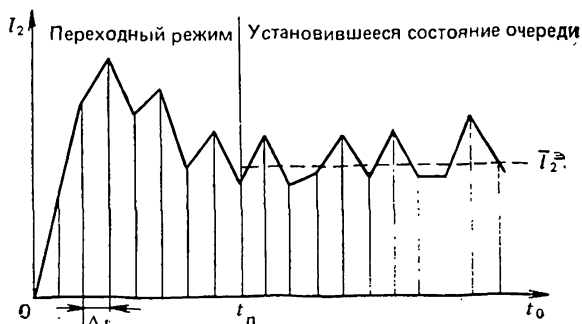


Рис. 4.2. Пример изменения во времени длины очереди деталей ко второму обрабатывающему станку

сколько зачастую другого средства для изучения явления у него нет. Чтобы как-то повысить доверие к такой модели, ему приходится строить последовательность вложенных моделей, отличающихся друг от друга уровнем детализации, но позволяющих достигать стационарного состояния. Более подробно об использовании для подобных целей последовательностей вложенных имитационных моделей можно найти в [67].

4.3.3. ОЦЕНКА УСТОЙЧИВОСТИ РЕЗУЛЬТАТОВ ИМИТАЦИИ

Под устойчивостью результатов имитации будем понимать степень нечувствительности ее к изменению входных условий. Универсальной процедуры подобной проверки не существует. Поэтому создатели моделей вынуждены прибегать к методам «для данного случая», частичным тестам и к здравому смыслу. Часто бывает полезна апостериорная проверка. Она состоит в сравнении предсказаний модели и результатов измерений после того, как будет произведено какое-либо изменение в установке, предусмотренное в исследовании. Если модель окажется приемлемо точной, уверенность пользователей в ее устойчивости оправданно возрастает.

В общем случае можно следовать принципу: чем ближе структура модели к структуре системы и чем выше степень ее детальности, тем обширнее область пригодности модели. Однако иногда в процессе структурной калибровки модели достигается точка, за которой устойчивость результатов моделирования может ухудшаться из-за чрезмерной подстройки модели к частным входным условиям, использованным для ее калибровки. К сожалению, крайне трудно определить момент, когда эта точка достигнута. Практика моделирования вычислительных систем показала, что устойчивость результатов моделирования можно оценивать дисперсией значений отклика (по выбранной компоненте). Если эта дисперсия при увеличении времени моделирования T_n не увеличивается, значит, результаты моделирования устойчивы.

Методика оценки дисперсии может быть следующей. В модельном времени t_0 задаются шаг Δt контролирования показателя качества Y и число шагов h для контроля, а также экспертное значение изменения контролируемого параметра ΔY_s . По достижении стационарного состояния оценивается амплитуда изменений параметра Y и вычисляется модуль амплитуды как функции от h . Каждый раз при этом проверяется выполнимость неравенства $|\Delta Y| \leq \Delta Y_s$. Если на всем интервале исследования Y окажется в заданных пределах, то считается, что модель находится в устойчивом состоянии. Рост разброса контролируемого параметра от начального значения при изменении h указывает на неустойчивый характер имитации исследуемого процесса.

4.3.4. ИССЛЕДОВАНИЕ ЧУВСТВИТЕЛЬНОСТИ ИМИТАЦИОННОЙ МОДЕЛИ

Исследователь должен установить диапазон изменения отклика модели Y при изменении каждой компоненты вектора параметров X . В зависимости от диапазона изменения откликов Y определяется стратегия планирования экспериментов на ИМ. Если при значительной амплитуде изменений некоторой компоненты вектора параметров модели X отклик Y меняется незначительно, то это означает, что точность представления этой компоненты в ИМ не играет существенной роли. Кроме того, в планировании имитационных экспериментов эта компонента не будет использоваться как основная. Если же отклик модели Y окажется высокочувствительным к изменению некоторой компоненты вектора X , то это служит прямым указанием на необходимость представления ее в модели с максимально возможной точностью.

Вторым, не менее важным моментом исследования чувствительности модели является проверка зависимости отклика модели Y от изменений параметров внешней среды G . Меняя характеристики G в обе стороны на некоторую величину ΔG , исследователь оценивает диапазон изменений вектора отклика модели δY . Если δY незначителен, то требования к точности задания модели внешней среды могут быть несущественными. В противном случае измерение характеристик G^* , по которым строится модель внешней среды G , и сами способы стабилизации значения G в модели должны быть высокоточными. Анализ поведения приращенного отклика модели δY при колебаниях X и G иногда позволяет скорректировать алгоритмы ИМ в сторону их упрощения.

Обычно пространство значений параметра X задано и определяется целями моделирования и степенью осведомленности исследователя о компонентах объекта моделирования. Определение чувствительности ИМ параметров легче всего проводить в центральной точке пространства значений параметра. Выбор центральной точки осуществляется, как правило, на основании априорных суждений [67, 81] и носит неформальный характер. Для вычисления отклика применяются процедуры, описываемые самим пользователем на языке реализации модели, результаты работы которых затем уже используются стандартной процедурой оценки чувствительности. Исходя из тех же априорных сведений пользователь должен задать интервал изменения каждой компоненты вектора параметров X модели системы.

Например, в [51] специальная стандартная процедура находит область изменения показателя качества модели следующим образом. Каждая q -я компонента вектора X отклоняется от значения его в центральной точке в обе стороны на длину выбранного интервала его изменения ($\min X_q, \max X_q$). Остальные компоненты вектора X остаются неизменными и соответствуют центральной точке. При указанных значениях вектора параметров X проводится пара модельных экспериментов и вычисляются отклики модели ($\min Y, \max Y$), где $\min Y$ и $\max Y$ означают соответст-

венно векторы отклика, полученные при минимальном и максимальном значениях q -й компоненты вектора параметров X . Вычисляется приращение q -й компоненты вектора параметров модели

$$\delta X_q^0 = \frac{(\max X_q - \min X_q) \cdot 2}{(\max X_q + \min X_q)} 100\%, \quad (4.9)$$

которое и будет приращением вектора параметров X при изменении только одной компоненты q . Находится приращение n -й компоненты вектора отклика

$$\delta Y_n = \frac{|\max Y_n - \min Y_n| \cdot 2}{(\max Y_n + \min Y_n)} 100\%. \quad (4.10)$$

Изменение вектора отклика Y можно определять либо модулем вектора приращений (δY_n , $n=1, L$; L — размерность вектора отклика), либо максимальным значением из δY_n . Чаще всего исследователи останавливаются на втором способе, когда в качестве изменения вектора отклика выбирают $\delta Y_q^0 = \max\{\delta Y_n\}$, где δY_n вычисляются по формуле (4.10).

Итак, чувствительность модели по q -й компоненте вектора параметров X определяется парой значений (δX_q^0 , δY_q^0). Эта пара чисел показывает, на сколько процентов может измениться отклик модели при увеличении q -й компоненты параметров на δX_q^0 процентов. Затем подобным образом поступают с остальными компонентами вектора параметров X . В результате получают множество пар значений $\{\delta X_q^0, \delta Y_q^0\}$, $q=1, h$, где h — размерность вектора параметров модели X . Этой информации, как правило, бывает достаточно для ранжирования компонент вектора параметров X по значению чувствительности вектора отклика модели. Если модель оказывается малочувствительной по какой-либо q -й компоненте вектора параметров модели X_q , то зачастую исследователь не включает в план имитационного эксперимента изменение X_q , чем достигается экономия ресурса времени моделирования.

Пример. Рассмотрим применение указанной процедуры для анализа чувствительности модели производственного участка. Допустим, что середина вектора параметров соответствует интенсивностям поступления деталей первого и второго типа $\lambda_1=0,108$ дет./мин и $\lambda_2=0,071$ дет./мин. Длительность интервала моделирования составляет $T_n=600$ мин. В табл. 4.2 представлены значения компонент вектора откликов $Y=(T_1, T_2, \bar{\eta}_1, \bar{\eta}_2, \bar{\eta}_4, \bar{\eta}_5)$ при соответствующих значениях вектора параметров модели $X=(\lambda_1, \lambda_2)$. Диапазоны изменений интенсивностей поступления деталей на обработку выбираем: от 0,50 дет./мин до 0,150 дет./мин для деталей первого типа; от 0,035 дет./мин до 0,105 дет./мин для деталей второго типа. Как видим, выбрано 100 % значения приращений компонент вектора параметров модели производственного участка ($\delta \lambda_q=100\%$). Было проведено четыре варианта имитационных экспериментов.

Приращения компонент вектора параметров ($\delta \lambda_q$) вычислялись по формуле (4.9). Для каждой компоненты вектора откликов Y по формуле (4.10) оп-

Таблица 4.2

Пример проверки чувствительности модели производственного участка

Номер варианта	λ_1	λ_2	T_1 , с	T_2 , с	η_1	η_2	η_3	η_4
1	0,050	0,071	15,7	19,7	0,60	0,57	0,30	0,71
2	0,150	0,071	22,7	26,7	1,00	1,00	0,90	0,71
3	0,108	0,035	12,7	21,7	0,71	0,65	0,65	0,33
4	0,108	0,105	22,7	26,7	1,00	1,00	0,65	1,00

Номер варианта	$\delta\lambda_q$, %	δT_1 , %	δT_2 , %	$\delta\eta_1$, %	$\delta\eta_2$, %	$\delta\eta_3$, %	$\delta\eta_4$, %	$\delta\bar{x}_q$, %
1								
2	100	36,4	30,0	50,0	54,7	100,0	0	100,0
3								
4	100	57,9	20,0	33,9	42,4	0	100,0	100,0

ределялись приращения в процентах. Результаты расчетов также представлены в табл. 4.2. Наибольшая чувствительность модели при изменении λ_1 достигается по отклику η_3 (коэффициенту загрузки третьего обрабатывающего станка), максимальная чувствительность модели при изменении λ_2 — по отклику η_4 (коэффициенту загрузки четвертого обрабатывающего станка). Поскольку максимальная чувствительность модели к обоим интенсивностям одинакова, то можно сделать вывод о том, что модель одинаково чувствительна к изменению обоих параметров моделирования. Причем 100%-ному изменению диапазона значений любого из параметров (λ_1 , λ_2) соответствует 100%-ное изменение одной из компонент вектора откликов модели.

У каждого пользователя может быть большое число подобных процедур исследования чувствительности модели. Причем каждая такая процедура хороша для конкретной задачи моделирования.

В заключение отметим, что если результаты испытаний и исследования свойств ИМ согласуются с имеющимся опытом использования подобных систем, то никого не волнуют детали построения модели и тщательность выполнения проверки. Однако без выполнения указанных выше процедур исследователь обойтись не может, поскольку они дают необходимую информационную базу обеспечения доверия к ИМ и перехода к следующим этапам работы с моделью.

4.4. ПЛАНИРОВАНИЕ ИМИТАЦИОННЫХ ЭКСПЕРИМЕНТОВ

4.4.1. ПРОБЛЕМЫ, РЕШАЕМЫЕ НА ЭТАПЕ ПЛАНИРОВАНИЯ ЭКСПЕРИМЕНТОВ

После того как установлены цели эксперимента, принято решение использовать машинное имитационное моделирование и определена система автоматизации моделирования, полезно провести предварительное планирование предстоящего эксперимента. Исследователь должен иметь подробный план эксперимента для целенаправленного и эффективного получения требуемых данных. Плановые ограничения на время и затраты должны быть приведены в соответствии с имеющимися в распоряжении исследователя ресурсами. Чем дороже и сложнее эксперимент, тем большее внимание следует обращать на этот этап. Зачастую ограничения в ресурсах опытов настолько жесткие, что не позволяют воспользоваться традиционными статистическими методами.

Обычно используются три типа экспериментов: сравнение средних и дисперсий различных альтернатив; определение важности учета или значимости влияния переменных и ограничений, наложенных на эти переменные; отыскание оптимальных значений на некотором множестве возможных значений переменных.

Эксперименты первого типа являются, как правило, однофакторными и довольно простыми. Экспериментатор решает вопросы о размере выборки, начальных условиях и наличии или отсутствии автокорреляции. Достаточно хорошие ответы на эти вопросы имеются в [81]. Экспериментам второго типа посвящено большинство книг по планированию экспериментов и анализу их результатов. Основными методами истолкования результатов этих экспериментов являются дисперсионный и регрессионный анализы. Для начального знакомства можно рекомендовать [8, 19]. Более подготовленному читателю следует обратиться к [24, 29, 61]. Третий тип экспериментов обычно предполагает использование последовательных или поисковых методов построения экспериментов. Для начала можно рекомендовать [17, 46, 81], а затем, по приобретении необходимых знаний и опыта постановки экспериментов, желательно ознакомиться с [26, 41, 61, 77, 78].

Предполагая соответствующую подготовку читателя и возможность использования обширной литературы по планированию экспериментов, в дальнейшем мы остановимся только на следующих вопросах: определение требуемого размера выборки; определение интервалов изменения параметров модели; планирование поиска источников погрешностей в имитационном эксперименте.

Наш опыт постановки модельных экспериментов на ЭВМ показывает, что удачное решение указанных вопросов при планировании экспериментов зачастую позволяет получить большую экономию в ресурсах.

4.4.2. ОПРЕДЕЛЕНИЕ ТРЕБУЕМОГО РАЗМЕРА ВЫБОРКИ

При ограниченном ресурсе времени моделирования первойшей задачей исследования является получение ответа на вопрос: как много выборочных значений следует взять во время моделирования, чтобы обеспечить достаточную статистическую значимость? Поскольку разбросы выборочных значений случайны, то обусловленная ими некоторая неточность результата эксперимента в значительной мере определяется размером выборки. Задача определения такого размера выборки, который позволяет обеспечить желаемый уровень точности и в то же время минимальную стоимость моделирования, весьма трудна, но чрезвычайно важна.

К сожалению, на практике размер выборки обычно является в первую очередь функцией количества средств, отпущенных на моделирование. Поскольку полученная в результате модельного эксперимента информация в дальнейшем используется для принятия решений, связанных с изменением поведения реальной системы, т. е. является очень важной, то и должна она быть как можно более точной или хотя бы должна быть известной степень ее неточности.

Многие методы анализа используют предположение о независимости и нормальном распределении откликов модели. Это предположение основано на применении центральной предельной теоремы теории вероятностей. Действительно, часто отклик представляет собой сумму большого числа небольших эффектов, и переменная отклика сложной имитационной модели является результатом аддитивного действия большого числа случайных переменных. Если учесть, что каждое выборочное значение представляет собой также сумму большого числа небольших эффектов, то центральная предельная теорема применима, и мы должны предполагать, что приближенно отклик имеет нормальное распределение. В условиях применимости центральной предельной теоремы и отсутствия автокорреляции мы можем использовать для определения размера выборки, необходимой для оценивания параметров с заданной точностью, правило «автоматической остановки». Оцениваемыми параметрами могут быть среднее значение и среднеквадратическое отклонение совокупности.

Правило «автоматической остановки» основано на методе доверительных интервалов. Метод доверительных интервалов предполагает задание точности представления d_n математического ожидания μ_n или b_n дисперсии σ_n^2 n -й компоненты вектора отклика \mathbf{Y} и уровня значимости α , гарантирующего попадание μ_n , σ_n^2 внутрь интервалов $(\bar{Y}_n \pm d_n)$, $(D_n \pm b_n)$ с вероятностью $(1-\alpha)$. Здесь \bar{Y}_n и D_n представляют собой среднее значение и дисперсию, вычисленные по выборке объема N , и являются оценками соответственно μ_n и σ_n^2 .

В ходе испытания и исследования свойств ИМ исследователь определяет векторы точностей $(d_1, \dots, d_n, \dots, d_L)$ или $(b_1, \dots, b_n, \dots, b_L)$ представления компонент вектора отклика \mathbf{Y} . Выполнение

правила «автоматической остановки» представляет собой итеративную процедуру, суть которой состоит в следующем. До начала проведения серии экспериментов известны уровень значимости α и векторы точностей представления компонент вектора отклика (b_n) или (d_n). Из априорных сведений, например из опыта, полученного в ходе исследования свойств ИМ, устанавливается количество начальных экспериментов N_1 , необходимых для получения выборки значений компонент откликов модели $\{Y_{nk}\}$, $n=\overline{1, L}$; $k=\overline{1, N_1}$. Если исследователь предполагает, что в ходе исследования свойств ИМ число опытов было слишком большим, то в качестве начального значения полагают $N_1=5$. Далее алгоритм выбора числа экспериментов N состоит из следующих шагов.

Шаг 1. По выборке $\{Y_{n,k}\}$ по формулам (4.3) находят средние значения \bar{Y}_n и дисперсии D_n .

Шаг 2. Для очередного номера n компоненты вектора отклика модели Y определяют достигнутую точность оценок \bar{Y}_n и D_n при выполненных N_1 экспериментах. Здесь может быть несколько случаев.

Если выборка малого объема ($N_1 \leq 30$), для вычисления доверительного интервала используют t -статистику, имеющую распределения Стьюдента $d_{1n} = t_{кр} \sqrt{D_n / (N_1 - 1)}$, где $t_{кр}$ — критическое значение t -статистики, которое находится по таблице распределения Стьюдента [13] при $N_1 - 1$ степенях свободы и заданном уровне значимости α .

Если размер выборки большой ($N_1 > 30$), то для вычисления доверительного интервала μ_n используют двустороннюю статистику с нормированным нормальным распределением

$$d_{1n} = z_{\alpha/2} \sqrt{D_n / N_1},$$

где $z_{\alpha/2}$ — значение нормированного нормального распределения, которое находится по таблице [13] при заданном уровне значимости $\alpha/2$.

Если нормальность Y_n предположить нельзя, но число очень большое, применяют неравенство Чебышева [13]

$$p \{ |\bar{Y}_n - \mu_n| \geq h \sigma_n / \sqrt{N_1} \} \leq 1/h^2,$$

где h — некоторая наперед заданная константа, означающая число среднеквадратических отклонений, удовлетворяющих исследователя.

Тогда доверительный интервал можно с достаточной точностью вычислить по формуле

$$d_{1n} = \sqrt{D_n / (N_1 (1 - \alpha))}.$$

При оценивании дисперсии задача отыскания оценки D_n с достоверностью $(1 - \alpha)$ имеет вид

$$p \{ (1 - b_n) \sigma_n^2 \leq D_n \leq (1 + b_n) \sigma_n^2 \} = 1 - \alpha.$$

Ее имеет смысл решать только при большом числе опытов N_1 . В

этом случае используют хи-квадрат-статистику. Поскольку N_1 достаточно велико, то эту статистику аппроксимируют нормальным распределением и получают следующую формулу для определения достигнутой точности оценки σ_n^2 в N_1 опытах:

$$b_{1n} = z_{\alpha/2} \sqrt{2/(N_1 - 1)},$$

где $z_{\alpha/2}$ — значение нормированного нормального распределения, определяемое по таблице [13] при заданном уровне значимости $\alpha/2$.

Шаг 3. Сравнивают достигнутой точности d_{1n} или b_{1n} оценок μ_n или σ_n^2 в N_1 опытах с заданным значением d_n или b_n . Если выполняется неравенство $d_{1n} \leq d_n$ или $b_{1n} \leq b_n$, то отмечают, что по n -й компоненте вектора откликов модели Y достигнута требуемая точность оценки в N_1 экспериментах. Если данное неравенство не выполняется, то переходят к шагу 4. При выполнении данного неравенства переходят к шагу 5.

Шаг 4. Выполняют еще один модельный эксперимент: N_1 увеличивают на единицу ($N_1 \equiv N_1 + 1$) и переходят к шагу 1.

Шаг 5. Проверяют, все ли компоненты вектора откликов проверены на удовлетворение точности оценки μ_n или σ_n^2 . Если проверена достижимость точности по всем компонентам вектора точностей $(d_1, \dots, d_n, \dots, d_L)$ или $(b_1, \dots, b_n, \dots, b_L)$, то эксперименты завершаются. В противном случае меняют номер компоненты вектора откликов модели и переходят к шагу 2.

Предложенная процедура, реализующая правило «автоматической остановки», имеется в составе МК PLSIM [51] в качестве стандартной, и ее можно использовать не только для оценки откликов модели и управления количеством имитационных экспериментов. Если в ходе испытания и исследования модели исследователь установил необходимую точность измерения статистик моделирования, то он может использовать рассмотренную выше процедуру, реализующую правило «автоматической остановки», для определения момента завершения имитации. Достижимость заданной точности d_n или b_n оценки некоторой статистики (или группы статистик) может быть одним из условий окончания имитации реального явления.

4.4.3. ОПРЕДЕЛЕНИЕ ИНТЕРВАЛОВ ИЗМЕНЕНИЯ ПАРАМЕТРОВ

В любом имитационном эксперименте важно определить интервалы изменения параметров модели. В частности, до начала имитации необходимо найти предельные значения статистик моделирования. Существуют три основных фактора, влияющих на выбор интервалов изменения параметров модели: необходимость получения одинаковой относительной точности статистик на разных участках области измерения параметров модели; характер функции отклика; назначение имитационного эксперимента.

Если анализ погрешностей показывает, что на каком-то участке моделируемого процесса данные вызывают сомнения, то в ходе имитации на этом участке необходимо увеличить частоту фик-

сации данных. Необходимо стремиться к тому, чтобы точность функции отклика модели была на всех участках одинаковой. Для этого необходимо соответствующим образом спланировать эксперимент.

Если известно, что в модельном эксперименте имеются особенности, которые можно обнаружить при получении данных в регулярной последовательности, используют классический последовательный план. Все параметры модели полагают постоянными, а один из них изменяют по всей области значений. Выбор интервалов между соседними значениями ведется с учетом баланса точностей. Аналогично изменяют второй параметр, третий и т. д. Затем составляют план изменения параметров в имитационном эксперименте.

В теории планирования экспериментов моделирования X называют факторами, а значения компонент вектора параметров — уровнями факторов. Перед планированием экспериментов исследователь устанавливает границы областей определения факторов. При этом следует учитывать несколько типов ограничений: принципиальные, которые нельзя нарушать в ходе эксперимента из-за их физической сущности; обуславливаемые технико-экономическими соображениями; связанные с условиями проведения имитационного эксперимента. Каждая комбинация уровней факторов является многомерной точкой в пространстве параметров, называемом факторным пространством.

Построение плана эксперимента сводится к выбору экспериментальных точек, симметричных относительно основного уровня. В зависимости от вида функции отклика $Y = \varphi(X, G)$ исследователь может располагать различными (априорными) сведениями об области наилучших значений Y . Априорная информация об этой области существенно влияет на выбор основного уровня по каждому фактору. Если у исследователя имеются сведения о координатах только одной точки и нет информации о границах факторов, то ему ничего не остается, как рассматривать эту точку в качестве основного уровня. Когда границы изменения факторов известны исследователю и он знает, что наилучшие значения Y находятся внутри области изменения факторов, то в качестве основного уровня он может выбрать любую из точек факторного пространства.

Хуже обстоит дело, когда известная точка лежит на границе факторного пространства, поскольку исследователю приходится выбирать основной уровень со сдвигом от наилучших значений Y . Если исследователю известно, что имеется несколько наилучших значений Y , то в качестве основного уровня он может выбрать любую случайную точку внутри факторного пространства. Наконец, исследователю до постановки имитационного эксперимента известна подобласть в факторном пространстве, где исследуемый процесс протекает оптимально и эксперимент ведется только для уточнения места этого оптимума. Тогда в качестве основного уровня он выбирает центр этой подобласти.

Назначение имитационного эксперимента также существенно влияет на методику выбора интервалов изменения параметров модели. Если предметом моделирования является поиск узких мест в функционировании объекта исследования или выбор применяемых гипотез о механизме явлений, то применяется многоуровневое факторное планирование. В этом случае на выбор уровней факторов влияет в основном относительная точность статистик моделирования на различных участках измерения параметров модели. Когда предметом моделирования является либо поиск оптимальных условий функционирования объекта исследования, либо выбор существенных факторов для управления объектом моделирования, либо построение вида функции, аппроксимирующей поведение объекта исследования, то задача сводится к выбору для каждого фактора кроме основного уровня еще двух уровней (нижнего и верхнего). В этом случае выбирается интервал варьирования фактора (число, прибавление которого к основному уровню дает верхний, а вычитание — нижний уровни фактора).

На выбор интервалов варьирования факторов накладываются ограничения сверху и снизу. Интервал варьирования не может быть меньше той погрешности, с которой исследователь фиксирует уровни, чтобы они были различимы. Но интервал варьирования не может быть настолько большим, чтобы эти уровни оказались за пределами области определения факторов. Для выбора интервала варьирования исследователь обычно использует следующую априорную информацию: сведения о точности фиксации факторов; о кривизне функции отклика (линейная или нелинейная); о чувствительности отклика модели Y к изменениям параметров модели X . Зачастую исследователи проводят предварительные эксперименты на модели для знакомства с исследуемым объектом и получения недостающей априорной информации о влиянии выбранных интервалов варьирования факторов на вид искомой функции откликов.

4.4.4. ПЛАНИРОВАНИЕ ПОИСКА ИСТОЧНИКОВ ПОГРЕШНОСТЕЙ В ИМИТАЦИОННОМ ЭКСПЕРИМЕНТЕ

При составлении плана анализа результатов моделирования исследователь должен иметь в виду одну или несколько проверок точности и приемлемости результатов моделирования. В ряде случаев для определения источников погрешностей можно использовать уравнение баланса¹. Суть такой методики рассмотрим на следующем примере.

Пусть в нашем модельном эксперименте измеряются переменные A , B , X , Y , которые связаны между собой уравнением сохранения между парами переменных

$$A \cdot B = X \cdot Y.$$

¹ Шенк Х. Теория инженерного эксперимента. — М.: Наука, 1975. — 158 с.

Установлено, что из-за неточности описания реального процесса одна из этих переменных (но неизвестно какая) является причиной систематической погрешности. Для коррекции ИМ нужно выявить не только виновника погрешности, но и место ее возникновения в алгоритме модели. Пусть в ходе моделирования не меняются A , а остальные переменные изменяются следующим образом: $m \cdot X$; $n \cdot Y$; $m \cdot n \cdot B$. Уравнение баланса при этих условиях имеет вид

$$A \cdot m \cdot n \cdot B = m \cdot X \cdot n \cdot Y.$$

Используется следующее правило. Если одна из переменных в уравнении баланса может быть представлена в виде суммы ($A + f(A)$), где $f(A)$ — систематическая погрешность, то эту переменную можно обнаружить, рассматривая поочередно случаи с фиксированным значением каждой переменной. Переменная, относительная погрешность изменения которой при фиксированном ее значении не меняется, и является причиной систематической погрешности. Исключением оказывается случай, когда переменная имеет вид $A + k \cdot A$ (k — постоянная величина). В этом случае погрешность с помощью данного метода обнаружить невозможно.

Часто для поиска погрешности имитации используется предварительный эксперимент. По результатам этого эксперимента исследователь строит график зависимости $Y = f(X)$. Система координат и сама функция выбираются такими, чтобы график был линейным или хотя бы не имел большой кривизны вблизи начала координат для последующей экстраполяции. Обычно используются линейные и полулогарифмические шкалы, позволяющие строить прямую. В том случае, когда в области малых значений имеет место резкое отклонение от линейного вида, экстраполяцию данных проводить нельзя. Поэтому исследователь должен следить за этой ситуацией.

Пусть исследователю известно, что один из линейных графиков отклика модели имел систематическую погрешность. Необходимо определить, какой из графиков верен, и оценить эту погрешность. Например, каждый набор результатов моделирования характеризуется одинаковым показателем степени, но графики не совпадают. Однако известно, что в области малых значений зависимость должна проходить через начало координат. Тогда для идентификации систематической погрешности модельного эксперимента строят графики данных в области малых значений. Тот график, который не удовлетворяет условию прохождения через начало координат, и имеет систематическую погрешность. Среднее значение ординаты при $X = 0$ является оценкой этой погрешности.

4.5. ЭКСПЛУАТАЦИЯ МОДЕЛИ

4.5.1. ЗАДАЧИ, РЕШАЕМЫЕ НА ЭТАПЕ ЭКСПЛУАТАЦИИ МОДЕЛИ

На этапе эксплуатации ИМ исследователю предстоит выполнять две взаимосвязанные процедуры: обработку результатов имитационного эксперимента; представление и документирование полученных экспериментальных данных. Любая имитационная модель не имеет ценности до тех пор, пока она не будет использована теми, для кого разработана. Для большинства руководителей разработки сложной системы интерес представляют только их собственные проблемы и способы их решения. Поэтому информация, получаемая с помощью имитационной модели, должна быть приемлемой для них. Критерии приемлемости прежде всего включают в себя надежность и полезность информации для проектирования.

Кроме того, исследователь должен понимать, как необходимо поступить или как можно использовать результаты моделирования в трудных ситуациях. Если ему не ясно, как эти данные могут помочь ему или кому-то другому принимать проектные решения, то он их будет просто игнорировать, и вся работа по созданию имитационной модели окажется безрезультатной. Поэтому для создателей ИМ весьма важно уметь представить результаты в наглядной форме, широко применяя графические способы анализа данных моделирования.

Наконец, любая имитационная модель должна позволять разработчикам систем оценивать те решения, которые удовлетворяют их собственным понятиям рациональности проекта или системы, и возможные результаты применения других стратегий проектирования. Весьма важно, чтобы они сами принимали личное участие в создании и разработке модели.

Таким образом, модель должна быть понятной заказчику-пользователю; способной давать разумные ответы на его вопросы; способной давать информацию, которая может быть в дальнейшем использована им при проектировании; легко модифицируемой; недорогой при ее применении. Результаты моделирования должны тщательно документироваться и представляться в понятной и четкой форме. Подача результатов столь же важна, сколь и само проведение работы. Весьма важно организовать обсуждение результатов моделирования. При этом одному большому формальному изложению результатов работы с моделью лучше предпочесть серию небольших неформальных обсуждений с заказчиком. Исследователь должен владеть методами подачи результатов моделирования и их графического представления.

Все традиционные методики обработки результатов наблюдений приемлемы и для анализа результатов имитации. Для начального знакомства можно рекомендовать [13, 33]. Более подготовленному читателю рекомендуем обратиться к [6, 41, 47, 77]. Остановимся на следующих вопросах представления результатов

моделирования: исключение резко отклоняющихся значений при натурных экспериментах для получения исходной информации моделирования; выбор системы координат при графическом представлении данных имитации; анализ регрессионной зависимости функции отклика от параметров моделирования; документирование результатов имитации.

4.5.2. ИСКЛЮЧЕНИЕ РЕЗКО ОТКЛОНЯЮЩИХСЯ ЗНАЧЕНИЙ ПРИ НАТУРНЫХ ЭКСПЕРИМЕНТАХ С ПРОТОТИПАМИ

Чтобы задать характеристики поведения компонент реальной системы, зачастую приходится ставить серии экспериментов с прототипами компонент модели. Однако в ряде случаев поведение компоненты реальной системы может резко отличаться от предполагаемого исследователем. В таких случаях исследователь стоит перед проблемой: или игнорировать некоторые замеры как ошибочные, или пересмотреть свое представление о поведении компонент реальной системы. Для отбраковки ошибочных значений предполагается процедура, суть которой поясняется следующим примером.

На рис. 4.3 представлена зависимость $Y=f(X)$. Несколько точек резко отклоняется от аппроксимирующей прямой. Ставится вопрос: какие из них можно считать грубыми погрешностями опыта и исключить из дальнейшего рассмотрения? Весьма вероятно, что точка *A* является сомнительной, поскольку и для меньших, и для больших значений *X* ординаты близки к аппроксимирующей прямой линии. Точка *B* может быть ошибочной, но, возможно, она

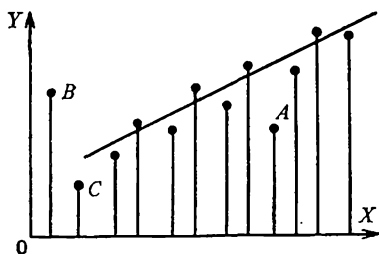


Рис. 4.3. Пример представления эмпирических данных, содержащих резко отклоняющиеся значения от линейной зависимости

является важной и отражает какой-либо реальный физический эффект (например, переход системы в область других состояний). Точку *C* также следует сохранить до тех пор, пока не будут получены дополнительные сведения в области малых значений. Можно рекомендовать правило отбраковки: отклоняющиеся точки следует исключать, пользуясь статистическими критериями, и только в том случае, если они находятся в средней части графика.

4.5.3. ВЫБОР СИСТЕМЫ КООРДИНАТ

В экспериментальной работе часто может возникать следующая ситуация. Имеются данные, которые в определенной системе координат могут быть представлены в виде прямой, уравнение которой можно найти. Тогда задачей исследователя является выбор такой системы координат или преобразование выбранной си-

стемы координат в такую, в которой полный набор данных моделирования давал бы по возможности прямую линию. Для этой цели строят график в логарифмических координатах (например, $Y=kX^a$ имеет вид $\lg Y=\lg k+a \lg X$) либо в полулогарифмических координатах (шкала Y — логарифмическая, X — линейная). Иногда гиперболическую функцию $Y=X/(a+bX)$ можно представить в виде прямой, построив в линейных координатах зависимость X/Y от X или зависимость $1/Y$ от $1/X$. Вообще графики позволяют представить данные в наглядной форме при минимальной обработке экспериментального материала. Как средство выдачи максимума информации при минимальном пространстве графики незаменимы.

При графическом представлении данных моделирования необходимо придерживаться трех простых правил, которые позволяют получить графики с минимальной неопределенностью:

минимальное деление шкалы графика должно соответствовать вероятной погрешности измеряемой величины, т. е. вероятная погрешность должна быть примерно равна половине наименьшего деления шкалы графика;

по возможности для любых данных надо пытаться строить линейный график;

выбор длины шкалы должен производиться из условия максимального использования формата графической бумаги.

В противном случае мы рискуем либо не уловить основной характер кривой и не установить закономерность ее изменения, либо не получить желаемой точности, так как все случайные отклонения сгладятся.

4.5.4. АНАЛИЗ РЕГРЕССИОННОЙ ЗАВИСИМОСТИ ФУНКЦИИ ОТКЛИКА МОДЕЛИ ОТ ПАРАМЕТРОВ МОДЕЛИРОВАНИЯ

Получив в ходе обработки результатов эксперимента параметры линейной регрессионной модели поведения отклика на изменения параметров модели, исследователь приступает к ее анализу. Коэффициенты полинома являются частными производными функции отклика по соответствующим переменным. Геометрический смысл коэффициентов — это тангенсы углов наклона гиперплоскости к соответствующей оси. Большой по абсолютной величине коэффициент соответствует большому углу наклона и, следовательно, более существенному изменению отклика при изменении данного фактора. Анализ уравнения регрессии осуществляется в три этапа.

На первом этапе устанавливается, в какой мере каждый из факторов влияет на значение отклика. Коэффициенты регрессии представляют собой количественную меру этого влияния. Чем больше коэффициент, тем сильнее влияние фактора. О характере влияния факторов говорят знаки коэффициентов. При знаке «+» с ростом фактора растет величина отклика; при знаке «—» отклик убывает.

На втором этапе совокупность факторов располагают в ряд по силе их влияния на отклик. Факторы, у которых коэффициенты незначимы, не интерпретируются. Обычно исследователь следует правилу: если коэффициент регрессии отрицателен, то для увеличения отклика модели надо уменьшить значение фактора; если он положителен, то значение фактора необходимо увеличить. При минимизации отклика модели можно изменить знаки коэффициентов на обратные (кроме свободного члена b_0) в уравнении регрессии и затем поступить, как в случае максимизации параметра оптимизации.

На третьем этапе производится оценка эффектов взаимодействия факторов. Необходимо придерживаться следующего правила: если эффект взаимодействия имеет положительный знак, то для увеличения отклика модели требуется одновременное увеличение или уменьшение значений факторов. Для уменьшения отклика модели факторы должны одновременно изменяться в разных направлениях. Если же эффект взаимодействия имеет отрицательный знак, то для увеличения отклика модели требуется одновременное уменьшение факторов. В каждом случае имеются два варианта. Какому из вариантов отдать предпочтение? Прежде всего нужно учесть знаки линейных эффектов соответствующих факторов. Если эффект взаимодействия имеет знак «+» и соответствующие линейные эффекты отрицательны, то выбор однозначен (оба фактора устанавливаются на нижнем уровне). Если же знаки различны, то приходится учитывать численные значения коэффициентов в уравнении регрессии и жертвовать тем фактором, значение коэффициента которого меньше.

4.5.5. ДОКУМЕНТАЦИЯ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ

На всех этапах разработки и использования моделей необходима документация как для лица, принимающего решение, так и для самих разработчиков ИМ. При подготовке документации следует решить: какие виды документов необходимы для создания полной и адекватной документации на модель; кто ее пишет и каким образом; как меняется качество документов по мере использования модели.

Документация на модель должна содержать в себе следующую информацию: точную формулировку цели моделирования; математическое определение, состав предположений и формулировку задачи моделирования; полный набор текущих входных данных моделирования; полный набор схем программы модели; описание различных стандартных программ и методик, используемых в модели; описание программ модели с комментариями; имена программистов и руководителей, ответственных за разработку модели. По мере того как описание модели создается, оно постепенно пополняется и детализируется.

На этапе определения необходимости имитационного моделирования создаются документы, которые потом носят исторический

характер. В них указываются: начальная идея модели; состав лиц, инициировавших построение модели; пользователи модели и их потребности в моделировании; общее описание проблемы; доводы при выборе типа имитационной модели с описанием проблем, стоящих перед моделированием, и доказательством невозможности использования аналитических решений для этой цели.

Затем, по мере изучения модели, составляется второй документ, в котором учитывается цель моделирования, составляется список участников разработки модели, формулируется набор альтернативных решений, составляются план работы над созданием модели и график ее выполнения, уточняются требования заказчика к составу разработчиков модели, ресурсам ЭВМ и оборудования, а также устанавливается роль пользователей модели в организации моделирования.

На этапе формулировки модели разрабатываются рабочие документы на весь жизненный цикл моделирования. В этих документах содержатся: описание всех деталей модели; теоретическое и аналитическое обоснование модели; состав предположений, гипотез и ограничений; описание процедур оценки параметров; основные требования к заданию исходных данных моделирования; список ограничений по использованию имитационной модели.

На этапе реализации имитационной модели формируется набор эксплуатационных документов. Кроме того, приводятся описание задачи и общее описание модели для решения. Суммируются результаты первичного анализа проблемы с указанием: допущений, ограничений и предостережений; ожидаемых результатов моделирования; плана эксплуатации модели, а также предварительного плана обучения заказчика работе с моделью. Отметим, что эксплуатационная документация на программу модели оформляется в соответствии с требованиями ГОСТ 19.301—79 и 19.507—79.

Результаты испытания модели **сложной системы** также представляют собой важный документ. Должен быть подробно описан план проверки модели, согласованный с лицом, принимающим решение. Приводятся тесты и входные данные для испытания модели. В состав документации этапа испытания должны входить: план верификации модели, порядок исследования свойств модели, план проверки ее адекватности объекту моделирования. По результатам испытания составляется пояснительная записка, в которой указываются размеры области изменения параметров, в которой модель дает удовлетворительные результаты.

Убедившись в пригодности модели для решения поставленных задач моделирования, составляют документы для обучения заказчика работе с моделью. В них должно быть приведено описание процедуры привлечения пользователя к работам с моделью. Составляются: руководство пользователя, в котором указываются все необходимые сведения для понимания и работы с моделью; руководство аналитика с необходимой информацией для работы с моделью в будущем и ее модификации; резюме исполнителя,

облегчающее интерпретацию результатов лицу, принимающему проектные решения.

Этап моделирования тщательно документируется и сопровождается большим числом иллюстрированного материала. Все выводы, полученные в ходе моделирования, оформляются в виде отчетов или рабочих записок, которые также составляют неотъемлемую часть документации этапа использования модели и могут служить источником для разработки новой имитационной модели системы.

Глава 5

ПОВЫШЕНИЕ УРОВНЯ ТЕХНОЛОГИИ ПОСТАНОВКИ ИМИТАЦИОННЫХ ЭКСПЕРИМЕНТОВ

5.1. СПОСОБЫ ПОВЫШЕНИЯ УРОВНЯ ТЕХНОЛОГИИ ПОСТРОЕНИЯ МОДЕЛЕЙ

5.1.1. ВОЗМОЖНОСТИ АВТОМАТИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ ЭТАПОВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Трудности создания моделей сложных систем, которые были описаны в предыдущих главах, стимулировали поиски путей повышения уровня технологии постановки имитационных экспериментов на ЭВМ. Этими вопросами занимаются в настоящее время практически все разработчики систем моделирования. Поскольку затраты на разработку имитационных моделей достаточно велики, сравнимы с созданием самих реальных систем, то даже малейшее расширение возможностей в технологии имитационного эксперимента на ЭВМ за счет введения того или иного средства автоматизации дает такому средству большую перспективу использования.

Этап составления содержательного описания объекта моделирования практически невозможно автоматизировать, поскольку на данном этапе большая часть работы выполняется исследователем непосредственно и к тому же состав и содержание работ определяются данным конкретным объектом, а также задачами и целями моделирования. По той же причине трудно автоматизировать этап составления концептуальной модели объекта моделирования. Кроме того, на этом этапе осуществляется выбор способа формализации, что определяет специфику реализации и использования имитационной модели на последующих технологических этапах.

На этапе формализации в ряде случаев удается унифицировать следующие функциональные действия: описание компонент объекта моделирования; представление взаимодействия компо-

нент системы по информации; отделение параметров модели от алгоритмов, реализующих функции компонент системы; представление взаимодействия компонент друг с другом по управлению. Это ускоряет выполнение этапа, но такие действия не поддаются автоматизации.

Этап перехода от формального описания к описанию имитационной модели системы также трудно автоматизировать из-за необходимости участия исследователя. Однако в ряде случаев возможно ускорение выполнения работ благодаря стандартизации при организации внутренней синхронизации между компонентами модели, выборе мест для установки операторов внешней синхронизации компонент модели, использовании общих ресурсов компонент модели и организации обмена информацией между компонентами модели.

Самая большая степень автоматизации работ может быть достигнута на наиболее трудоемком этапе программирования имитационных моделей. Во многих современных системах моделирования удастся автоматизировать следующие основные работы: кодировку программ компонент модели; обеспечение автономной отладки, при этом вводятся дополнительные средства диагностики ошибок программирования и отладки компонент модели; обеспечение параллельной разработки программ ИМ большими коллективами; обеспечение комплексной отладки ИМ; документирование этапа программирования ИМ.

Этап испытания ИМ обычно трудно поддается автоматизации, поскольку большинство функциональных действий требует участия исследователя. Однако в некоторых системах моделирования в распоряжение исследователя предоставляются программные средства, обеспечивающие автоматизацию выполнения ряда функциональных действий. В результате удается автоматизировать: оценку погрешности имитации, обусловленной наличием в ИМ неидеальных генераторов псевдослучайных чисел; нахождение момента входа ИМ в стационарный режим; калибровку ИМ.

Для автоматизации выполнения этапа исследования свойств ИМ в ряде СМ имеются средства: выделения переходного режима в работе ИМ; оценки устойчивости результатов имитации; определения чувствительности ИМ.

Весьма трудно автоматизируется этап проверки адекватности ИМ объекту моделирования. Для ускорения выполнения этого этапа в некоторых СМ в распоряжение исследователя предоставляются средства проведения: независимых прогонов ИМ и сравнения откликов; проверки гипотез о близости средних значений отклика в ИМ и реальной системе; наблюдения за отклонением отклика ИМ от откликов, получаемых при исследовании реальной системы.

Этап планирования экспериментов содержит с точки зрения возможности автоматизации два вида функциональных действий. Первый — это трудно стандартизуемые действия, требующие учас-

тия исследователя: уменьшение имитационных экспериментов, определение интервалов изменения параметров, планирование поиска источников ошибок. Второй — это функции, которые обеспечивают классическими методами планирования экспериментов. Они хорошо поддаются автоматизации (составление планов эксперимента, планирование для поиска регрессионных зависимостей и т. д.) и во многих СМ оформлены в виде библиотек стандартных процедур, позволяющих широко их использовать при выполнении работ на этапе планирования имитационных экспериментов.

Наибольшее развитие получили способы автоматизации этапа эксплуатации имитационных моделей. Для этой цели ряд СМ предоставляет в распоряжение исследователя средства автоматизации представления результатов моделирования, организации серии модельных экспериментов, сбора статистики и обработки результатов имитации, выдачи сообщений о ходе имитации, ввода-вывода информации в виде, удобном для анализа результатов.

5.1.2. СРЕДСТВА ОБЕСПЕЧЕНИЯ ВЫСОКОГО УРОВНЯ ТЕХНОЛОГИИ МОДЕЛИРОВАНИЯ

Рассмотрим некоторые СМ, которые повышают уровень технологии модельного эксперимента на ЭВМ. Для сравнения выбраны СМ, базирующиеся на двух наиболее распространенных способах формализации реальных объектов: транзактном (ИМСС [66], МК АСИМ [52]) и процессном (МПЛ/1 [69], МК PLSIM [51]). Кроме того, большой интерес представляют параметрические способы повышения уровня технологии имитационного моделирования, реализованные в СМ CSS II [83].

При рассмотрении каждой из перечисленных СМ будем обращать внимание только на наиболее интересные пути автоматизации перечисленных в предыдущем параграфе функциональных действий каждого из этапов. Имеющиеся в большинстве СМ возможности реализации технологических этапов создания и эксплуатации ИМ здесь рассматривать не будем. Частично о них сказано в гл. 3 при сравнительном анализе СМ. Выбор СМ для анализа определялся степенью знакомства автора с технологическими возможностями существующих средств автоматизации моделирования сложных систем.

Система моделирования МПЛ/1. Эта система представляет собой попытку объединения в одной системе следующих возможностей: построения ИМ с непрерывным и дискретным характером изменения модельного времени t_0 ; представления объекта моделирования в виде взаимодействующих друг с другом автоматов; построения ИМ для анализа в проектировании вычислительных систем на всех стадиях их разработки с помощью набора описаний фрагментов ВС на разных уровнях детализации, который имеется в библиотеке МПЛ/1.

Для ускорения выполнения этапа *формализации* ВС система моделирования МПЛ/1 обладает средствами: моделирования (ти-

па языка CSS); имитации непрерывных систем (типа языков CSCL и НЕДИС); моделирования структурных и функциональных схем цифровой аппаратуры; описания алгоритмов ЭВМ на языке программирования PL/1. Для моделирования ВС в СМ МПЛ/1 имеется ряд эквивалентов языка CSS, которые обеспечивают:

- описание структуры и функционирование схем цифровой аппаратуры с помощью нового типа данных РЕГИСТР;

- задание специальных переключательных функций, которые позволяют идентифицировать события в объекте моделирования;

- представление временных функций при описании переходных процессов в электронных схемах;

- описание специальных объектов для конструирования вычислительных систем и сетей передачи данных (ХОСТ-ЭВМ, задача, пакет). В результате появляется возможность оперативного анализа: алгоритмов диспетчеризации в сети ЭВМ, маршрутизации потоков сообщений, топологии сетей передачи данных; конфигурации узлов связи; протоколов передачи информации.

Технология работ этапа *программирования ИМ* определяется возможностями языка высокого уровня. Все указанные средства реализуются как макрорасширение языка PL/1. Препроцессоры формируют описания моделей на языке программирования PL/1, а затем уже на втором шаге происходит компилирование программы в объектный код. Двухступенчатый характер создания имитационных моделей затрудняет диагностику ошибок в динамике имитации и снижает оперативность создания имитационных моделей. Для отладки и диагностики компонент ИМ используются средства языка PL/1.

Этап *эксплуатации* модели зачастую сокращается за счет возможности пополнения состава препроцессоров без изменения существующих средств. В результате достигается постоянное расширение множества предметных областей использования МПЛ/1 и, таким образом, повышение скорости создания ИМ из библиотек стандартных подмоделей. Для сбора статистики СМ МПЛ/1 предлагает исследователю специальный набор операторов. Для автоматизации представления результатов имитации существует набор специальных операторов ОМ МПЛ/1.

На остальных этапах создания ИМ МПЛ/1 не имеет дополнительных технологических средств ускорения работ.

Система моделирования ИМСС. Эта система позволяет перейти исследователю от алгоритмического к декларативному способу описания имитационных моделей сложных систем. В основу реализации квазипараллельной организации имитации положено транзактное представление процесса моделирования. Вместе с тем сохранена фортраноподобная форма представления описания объектов модели. Предметная область представления объектов моделирования ограничена сетями массового обслуживания. За счет некоторой универсальности описания сложных систем поль-

зователь ИМСС существенно выигрывает в оперативности построения имитационных моделей.

Для ускорения выполнения этапа *формализации* сложных систем в виде сетей массового обслуживания в распоряжение исследователя предоставляются следующие возможности. При описании входных потоков транзактов введены четыре типа источников: независимые, источники с последствием, зависимые и групповые источники транзактов. Каждому транзакту присваивается тот тип, который присущ его источнику. Транзакты могут обладать абсолютным и относительным приоритетом. Параметры поведения источника транзактов задаются пользователем при описании источника. При этом кроме признака типа источника, имени источника, определяющего тип транзактов, абсолютного и относительного приоритетов задаются еще имя и параметры закона распределения длин интервалов между формированиями транзактов. Для источников с последствием указывается также описание имени и параметров закона распределения интервалов блокировки работы источника. Зависимые источники формируют очередной транзакт на входе модели системы только в тех случаях, когда к нему возвращается предыдущий транзакт от системы. Групповые источники формируют транзакты на входе модели группами указанного размера. Для ликвидации транзактов используется стандартное средство — приемники. Причем в этот момент и фиксируются времена пребывания транзактов в системе.

Отражение ресурсных возможностей моделируемой сети массового обслуживания осуществляется описаниями устройств и памяти. Каждое из устройств описывается предложением подобного типа, в котором указываются: имя устройства, число каналов обслуживания, дисциплина обслуживания, тип транзакта, подлежащего обслуживанию, название и параметры закона распределения времени обслуживания транзакта данным устройством. Отметим, что каждое устройство может обслуживать только одну из очередей. Устройство представляет собой типовой обслуживающий прибор системы массового обслуживания.

Память также описывается предложением, в котором указываются: имя памяти, общий размер памяти, выделяемый транзактом, дисциплина обслуживания транзактов, имя и параметры закона распределения длин сегментов, выделяемых транзактом. Любой транзакт, обслуживаемый какой-либо l -й памятью, захватывает определенное количество памяти от общего количества и освобождает его, только попадая в специальный блок, занимающийся освобождением памяти. При этом указываются: имя элемента, отбирающего у заявок память, имя освобождаемой памяти, типы транзактов, которые отдают память.

Для задания маршрутов движения транзактов по схеме модели ИМСС предоставляет возможность использовать шесть типов узлов:

D — узлы, позволяющие размножать один транзакт на заданное число копий;

S — узлы для переключения транзактов в соответствии с указанными типами транзактов, возможно при этом разделение потока транзактов на части;

P — узлы, организующие вероятностное переключение транзактов, причем вероятность перехода транзакта в другое русло задается в описании *P*-узла;

M — узлы, в которых осуществляется модификация параметров транзактов заданного типа; допускается изменение типа абсолютного и относительного приоритета транзактов;

Q — узлы, обеспечивающие выбор направления транзактов в зависимости от длины очереди к устройству; при этом указываются длина очереди, с которой начинается изменение направления, и имена устройств, куда направляются транзакты;

C — узлы, организующие циклическое прохождение транзакта по некоторым маршрутам; указываются числа повторений маршрута транзактов и имена устройств, куда необходимо направить транзакты внутри и вне цикла обслуживания.

Для задания коммутации элементов сети перечисляются имена всех устройств, которые направляют транзакты на вход данного устройства. Время моделирования устанавливается специальным предложением. Возможны два способа задания времени моделирования: явное с указанием времени моделирования T_n и с помощью контрольных элементов. Структура модели задается четырьмя типами описаний: источников, устройств, узлов, связей элементов сети.

Специальных средств ускорения этапа перехода от формального описания к описанию имитационной модели объекта СМ ИМСС исследователю не предоставляют, поскольку формальное описание практически совпадает с описанием имитационной модели объекта моделирования.

Технология *программирования* ИМ определяется возможностями языка программирования ФОРТРАН. Текст описания ИМ интерпретируется в ходе моделирования. Специальные средства для организации отладки ИМ в распоряжение исследователя не предоставляются. Дополнительные средства для организации параллельного создания ИМ большими коллективами не имеются.

Этапы испытания, исследования, проверки адекватности модели и планирования имитационных экспериментов практически не поддерживаются технологическими средствами.

Этап *эксплуатации* ИМ зачастую сокращается из-за декларативного способа описания моделей и возможности представления сложной системы множеством элементов четырех типов. Пользователь может модифицировать поведение отдельных элементов сети за счет параметрического представления характеристик функционирования этих элементов. ИМСС предоставляет пользователю возможность сбора следующей статистики моделирования: времен пребывания транзактов в системе; времен ожидания обслуживания заявок в очередях; длин очередей; степени загрузки устройств; коэффициентов использования памяти. Имеется

также возможность автоматизации представления результатов имитации с помощью стандартизации вывода результатов моделирования.

Пакет прикладных программ моделирования вычислительных систем (МВС) [83]. Пакет МВС представляет собой достаточно мощное средство автоматизации моделирования с использованием принципов параметрического задания структуры модели сложных систем. Предметная ориентация на моделирование ВС на базе ЕС ЭВМ и возможность отделения параметров моделирования от самого алгоритма ИМ привели к широкому использованию пакета МВС среди разработчиков вычислительных комплексов на базе ЕС ЭВМ.

Этап *формализации* ИМ существенно убыстряется за счет использования специализированного языка CSS, предназначенного для имитации функционирования различных конфигураций ЭВМ единой серии. Язык CSS разработан в терминах элементов вычислительных систем — таких, как процессоры, устройства управления, каналы, магнитные диски (МД), магнитные ленты (МЛ), линии связи, терминалы, дисциплины опроса и т. д. Модели на языке CSS представляют собой описание характеристик аппаратуры, конфигурации системы, операционной логики и среды, в которой система должна функционировать. Главные достоинства моделей, разрабатываемых на языке CSS, — возможность детального представления вычислительного процесса на ЕС ЭВМ и наличие компонент для построения моделей вычислительных систем на различных уровнях детализации.

Недостатком пакета МВС является необходимость разработки моделей управляющих программ. Проектировщик должен разрабатывать кроме моделей прикладных программ еще и модели операционной системы, а это процесс достаточно сложный и трудоемкий. Кроме того, отсутствие конструкций, обеспечивающих требуемую логику функционирования отдельных компонент ВС, приводит к необходимости разработки программ на языке Ассемблера, что создает известные трудности. Поэтому была разработана вторая версия CM CSS II [83].

Версия CSS II ориентирована на моделирование ВС фирмы IBM, однако она пригодна для описания и других ВС. Ориентация CSS II на фирменное оборудование в основном заключается в наборе специализированных операторов моделирования конкретных типов стандартных устройств, выпускаемых фирмой IBM. В этих операторах параметры устройств могут быть не указаны, а компилятор CSS II автоматически подставляет их номинальные значения. Если какое-либо устройство объекта моделирования имеет нестандартные значения параметров, то на соответствующих позициях операторов описания характеристик устройств указываются необходимые значения. CSS II позволяет в случае необходимости задавать не только вероятностные характеристики оборудования, но и детерминированные значения параметров устройств. Для описания нестандартного оборудования

или устройств, выпускаемых другими фирмами, введены специальные операторы. Таким образом, CSS II ориентирован на так называемое репрезентативное моделирование, т. е. для описания функциональных возможностей каждого элемента (устройства) ВС имеет свой оператор.

Этап перехода от формального описания к описанию имитационной модели объекта при использовании СМ CSS II практически отсутствует, поскольку при параметрическом задании ИМ формальное описание объекта моделирования является одновременно и описанием имитационной модели.

Этап *программирования* ИМ практически автоматизирован, создание репрезентативных моделей выполняется достаточно оперативно. Однако емкость основной памяти, занимаемой этими моделями, и время имитации отдельных вариантов при моделировании больших систем могут оказаться весьма значительными, что снижает эффективность использования СМ CSS II. Специальные средства отладки ИМ для тех случаев, когда необходимо объединять подмодели, исследователю не предоставляются, и он вынужден обходиться возможностями Ассемблера. Средства документирования информации стандартизованы, что снимает с исследователя заботу об организации вывода результатов.

Этапы испытания и исследования свойств ИМ практически не поддерживаются дополнительными технологическими средствами. И это является наиболее существенным недостатком пакета МВС.

Этап *эксплуатации* модели вырождается в процедуру изменения параметров ИМ. Это обеспечивает высокую оперативность компоновки моделей из готовых элементов с помощью параметрического задания характеристик этих элементов. Наличие стандартных средств сбора статистики, ввода-вывода информации и выдачи стандартных сообщений в ходе имитации обеспечивают дополнительные возможности исследователю при использовании СМ CSS II в ходе проектного моделирования вычислительных систем, конструируемых на базе ЕС ЭВМ.

5.2. ПОВЫШЕНИЕ УРОВНЯ ТЕХНОЛОГИИ МОДЕЛИРОВАНИЯ С ПОМОЩЬЮ МОДЕЛИРУЮЩЕГО КОМПЛЕКСА PLSIM

5.2.1. ВОЗМОЖНОСТИ МОДЕЛИРУЮЩЕГО КОМПЛЕКСА ДЛЯ АВТОМАТИЗАЦИИ ЭТАПОВ ПОСТРОЕНИЯ ИМ СЛОЖНЫХ СИСТЕМ

В тех случаях, когда активности в составе процессов модели имеют сложные алгоритмы выполнения, для описания которых необходим специальный язык, можно использовать моделирующий комплекс МК PLSIM. Он построен на базе универсального языка программирования PL/I ОС ЕС [51].

При использовании МК PLSIM на этапе *формализации* ИМ основными компонентами модели являются объекты. Различают два типа объектов: активные и пассивные. Активными объектами явля-

ются процессы. Они имеют собственный алгоритм функционирования и могут воздействовать на другие объекты. Пассивные объекты — это очереди, средства и трассы. Динамика их поведения определяется активными объектами. Модель в МК PLSIM представляет собой набор взаимосвязанных активных и пассивных объектов. Под процессом понимаем набор связанных активностей, рассматриваемый как отдельный объект. МК PLSIM допускает использование уникальных и размножаемых процессов. Уникальному процессу соответствует своя область основной памяти (ОП), где помещаются алгоритм процесса и данные, используемые этим алгоритмом. Размножаемый процесс является реентерабельной процедурой, в которой имеется один алгоритм на все копии процесса, но для каждой копии отведена своя область данных.

Очереди также бывают уникальными и размножаемыми. По способу хранения и передачи информации различаем два типа очередей: без потерь и с потерями требований. Требования имеют фиксируемый формат в пределах одной очереди и представляются собой PL-структуры языка PL/1, полностью описываемые пользователем.

Средства в МК PLSIM служат инструментом синхронизации взаимодействующих процессов модели. Различаем четыре типа средств: устройства, памяти, флаги и сигналы. Устройства и памяти выполняют те же функции, что и в других языках моделирования. Любое устройство может быть захвачено и освобождено полностью любым процессом. Память имеет емкость, которая может выделяться процессам (и освобождаться) по частям. В случае занятости устройства или нехватки памяти процессы выстраиваются в соответствующие очереди и ожидают их освобождения. Для моделирования механизмов синхронизации процессов друг с другом используются флаги. Процессы могут реагировать на выставленный флаг и могут его игнорировать. При наличии флага и реакции на флаг процессы также выстраиваются в соответствующие очереди и ожидают момента снятия флага. Сигналы являются средством организации имитации прерывания выполнения алгоритма одного процесса со стороны другого процесса. При описании сигналов указывается, какие из процессов обслуживают пересекajícíеся множества сигналов. Любой процесс может выставить маску на обслуживание сигнала. Процессы, которые обслуживают некоторый сигнал, при появлении маски прерывают свое выполнение. Процессы могут активизировать и деактивизировать сигналы. В случае отсутствия маскирования сигналов управляющая программа активизирует обслуживающие процессы, реагирующие на эти сигналы. С помощью сигналов пользователь МК PLSIM получает возможность моделировать динамику функционирования системы прерывания ВС.

Трассы используются в тех случаях, когда необходимо исследовать различные алгоритмы работы объекта моделирования. Трасса представляет собой набор точек, в которых фиксируются моменты активизации отдельных фаз работы алгоритма объекта

моделирования. При этом запоминаются лишь времена прохождения точек трассы.

На этапе описания имитационной модели используются операторы, определяющие количественный состав и структуру всей модели. Для этой цели в МК PLSIM служат операторы описания процессов, очередей, средств и трасс. При описании процессов исследователь в этих операторах указывает тип и приоритет процесса. Описание очередей содержит информацию о дисциплине обслуживания требований, структуре требований и размерности очереди. Для описания состава устройств, флагов, сигналов и емкостей памяти в соответствующих операторах указываются списки имен. Описание трасс сводится к указанию в специальном операторе количества трасс и описанию размещения точек трасс в алгоритме модели.

Описание процесса состоит из заготовки и описания алгоритма его функционирования. Поскольку процессы в МК PLSIM представляют собой PL-процедуры, то заголовок процесса должен содержать заголовок процедуры, операторы сохранения параметров процесса средствами языка PL/I и оператор готовности процесса к моделированию. При описании самого алгоритма процесса используются возможности языка PL/I. Кроме того, возможно применение операторов моделирования синхронизации процессов (ожидание активизации по времени и условию) и управление объектами (немедленного, запланированного по времени, по условию). Операторы управления процессами (запуск, останов, пассивизация, продолжение) и набор операторов синхронизации работы процессов традиционны. Для описания работ очередей в МК PLSIM имеются операторы, обеспечивающие запись и выбор требований из очереди, копирование и уничтожение требований, очистку всей очереди. Устройства могут закрепляться за некоторыми процессами с помощью операторов захвата и освобождаться этими или другими процессами. Динамика изменения размеров памяти регулируется операторами выделения и освобождения заданной емкости памяти.

Этап *перехода к описанию имитационной модели* объекта исследования с помощью МК PLSIM предполагает выполнение исследователем правил распределения описания модели по стадиям будущей имитации: синтез модели, организация взаимодействия ее компонент, обработка результатов. Вначале реализуются описание статической структуры модели и набор операторов планирования экспериментов. Далее следуют описания алгоритмов процессов и набор операторов управления процессом имитации. Завершает описание ИМ набор операторов обработки результатов моделирования и операторов планирования экспериментов.

Для организации внутренней синхронизации между процессами в ИМ можно применять операторы выставления и снятия флагов. Механизмы флагов особенно эффективны при имитации использования общих ресурсов сложной системы и поэтому особенно удобны при моделировании функционирования операционных

систем вычислительных комплексов. Для имитации синхронизации с помощью флагов пользователю предоставляется следующий набор операторов: выставления флага; назначения и отмены реагирования процесса на указанный флаг. В любом месте алгоритма процесса могут встретиться операторы управления сигналами: активизации и деактивизации сигналов, одиночного и группового маскирования сигналов. При организации обмена информацией между процессами МК PLSIM предоставляет в распоряжение исследователя возможность использования в качестве требований PL-структуры.

Этап *программирования* облегчается благодаря наличию в МК PLSIM библиотеки процедур, позволяющей исследователю автоматизировать компоновку ИМ из набора стандартных процессов. В отличие от известных попыток расширения языка программирования PL/1 до языка моделирования путем реализации препроцессора [19, 79], в МК PLSIM использован процедурный принцип реализации программы модели, базирующийся на ряде соглашений, которым должен следовать пользователь. Это позволило отказаться от препроцессора и избежать недостатков этого способа описания модели.

В МК PLSIM повышение уровня технологии создания имитационной модели возможно в случае его применения для узкой предметной области исследования. Создание стандартных процессов и формирование на их основе библиотеки процессов позволяют ускорить и упростить описание модели. Например, ориентация на моделирование вычислительного процесса в ВС потребовала создания ряда стандартных процессов: генераторов задач, решаемых на ВС; имитаторов оборудования ВС; имитаторов программных компонент МО ВС. В таких случаях возможно создание «конструкторов» для оперативной сборки имитационной модели ВС из набора стандартных процессов.

Особенно большой эффект в ускорении построения моделей ВС достигается при большом числе типовых процессов. МК PLSIM допускает использование процессов, каталогизированных в его библиотеке в качестве стандартных, без представления алгоритма этого процесса на языке PL/1. В этом случае пользователю достаточно перечислить в описании состава модели имена стандартных процессов. Система моделирования МК PLSIM сама вызовет эти PL-процедуры и организует для них соответствующие рабочие области. Таким образом, появляется возможность оперативного пополнения библиотеки стандартных процессов и постановки в библиотеку других процессов-полуфабрикатов для заданной предметной области исследований. Вообще любой уникальный процесс после окончания отладки может быть каталогизирован в библиотеке процессов МК PLSIM и стать, таким образом, стандартным. Тем самым существенно повышается уровень автоматизации построения модели ВС.

Система моделирования МК PLSIM предоставляет пользователю дополнительные технологические возможности построения мо-

дели: отладку динамики функционирования объектов ИМ с помощью средств трассировки состояния объектов и выдачи на печать информации о смене состояний объектов, диагностику при реализации этапов синтеза и имитации модели. Для упрощения организации управления отладкой вводится маркировка процессов соответствующими операторами на этапе синтеза модели. Для маркированных процессов собирается стандартная статистика, что снимает с пользователя заботу об организации сбора традиционной статистики моделирования. Большой набор сообщений МК PLSIM обеспечивает исследователю возможность локализации как статических (в теле описания модели), так и динамических ошибок (в ходе имитационного эксперимента).

Поскольку имитационная модель ВС большого размера предназначена для поиска узких мест в алгоритме вычислительного процесса ВС, то и средства отладки ИМ в МК PLSIM специализированы на обеспечение этих функциональных возможностей. Для этой цели в распоряжении пользователя имеется возможность экспресс-обработки результатов моделирования на любом шаге имитации. Сюда входят возможности МК PLSIM по выдаче временных диаграмм поведения процессов и проведению пробных расчетов с выдачей общей и стандартной статистики. Для уточнения характеристик процессов в ходе моделирования возможно использование в алгоритме процесса ряда операторов-функций, позволяющих определить: номер процесса, состояние процесса и величину оставшегося времени до момента активизации процесса. Для отладки ИМ большими коллективами с помощью МК PLSIM имеются средства размножения процессов и создания библиотек моделей и подмоделей. Объявление любого автономно отлаженного процесса в качестве стандартного дает возможность использовать его в других подмоделях разрабатываемой ИМ.

5.2.2. СРЕДСТВА МК PLSIM ДЛЯ ОБЕСПЕЧЕНИЯ ЭТАПОВ ИСПЫТАНИЯ И ЭКСПЛУАТАЦИИ МОДЕЛЕЙ СЛОЖНЫХ СИСТЕМ

На этапе *испытания* ИМ, созданной на базе МК PLSIM, автоматизированы многие функции исследователя. Так, при задании исходной информации имеется возможность применять процедуры, облегчающие: отбор независимых переменных, классификацию факторов, определение области варьирования и уровней изменения факторов, предварительное построение регрессионной модели. При первичной статистической обработке возможно использование процедур: отбраковки результатов, анализа выполнения предпосылок регрессионного анализа, проверки гипотезы о значимости взаимодействий факторов. Для проведения оценок и параметризации моделей в МК PLSIM имеются процедуры проведения оценок по методу наименьших квадратов.

На этапе *проверки адекватности* ИМ можно автоматизировать процесс получения N независимых прогонов и сравнения откликов модели. Поскольку программная реализация модели представля-

ет собой PL-программу, скомпонованную таким образом, что стадии генерации ИМ, имитации и обработки отделены друг от друга, то организация многовариантного или итеративного моделирования обеспечивается простым зацикливанием с конца на начало модели. Для проверки гипотез о близости средних значений откликов в ИМ и в реальной системе существует несколько процедур, использование которых обеспечивается выполнением операторов процедур обработки МК PLSIM.

На этапе *исследования свойств* ИМ МК PLSIM позволяет управлять начальным периодом моделирования. Для оценки точности имитации исследователь может из набора процедур оценивания результатов моделирования выбрать наиболее подходящую для данного случая.

В МК PLSIM для определения момента входа имитационной модели в стационарный режим имеется возможность использования специальных процессов-контролеров. Алгоритм функционирования этих процессов задается самим пользователем и они реализуются в виде специальных процедур. С момента входа модели в стационарный режим при выполнении заданных условий в этих процедурах УПМ МК PLSIM осуществляет автоматический запуск сбора статистики, который и будет продолжаться до заданного момента времени или выполнения другого условия окончания имитации.

Для исследования чувствительности ИМ в МК PLSIM исследователь может применять процедуры вычисления отклика, описываемые самим пользователем. Результаты работы этих процедур используются стандартной процедурой оценки чувствительности. Из априорных сведений пользователь должен задать интервал изменения каждой компоненты вектора параметров модели ВС. В тех случаях, когда исследование объекта на ИМ ведется по нескольким критериям, процедура оценки чувствительности модели по указанию пользователя может обращаться к разным процедурам, вычисляющим соответствующие показатели качества, и затем уже проводить ранжировку нескольких показателей качества по значению чувствительности модели. Такой прием зачастую позволяет выбрать соответствующий критерий оценки вариантов моделирования при наличии нескольких альтернативных показателей качества имитационной модели ВС.

На этапе *составления плана эксперимента* МК PLSIM позволяет автоматизировать такие традиционные процедуры, как аппроксимация гистограмм распределений, вычисление квантилей распределений, проверка гипотез, построение временных рядов. Язык планирования экспериментов МК PLSIM дает исследователю возможность применять следующие стратегии планирования: композиционную схему экстремального планирования, последовательное Д-оптимальное планирование, априорное планирование с предварительным отсеиванием факторов, последовательную дискриминацию взаимодействия факторов и отклика. В распоряжение исследователя МК PLSIM предоставляет набор процедур

облегчающих реализацию алгоритмов регрессионного и дисперсионного анализа. Весьма широк набор процедур обработки результатов активного и пассивного эксперимента.

Как показывает практика исследований имитационных моделей ВС, чаще всего реализуются два типа имитационного эксперимента: обработка временных рядов и активный эксперимент для построения регрессионных зависимостей между факторами модели ВС и статистиками моделирования. Обработка временных рядов включает в себя определение по статистике параметров временного ряда и нахождение передаточной функции. Активный эксперимент ориентирован на проведение отсеивающего регрессионного эксперимента, дробного факторного эксперимента, композиционного планирования, планирования в условиях дрейфа, дискриминации моделей и построения нелинейных моделей.

Особенность эксперимента на ИМ сложных систем состоит в том, что исследователю часто приходится решать многокритериальные и многофакторные задачи исследования в условиях ограничений на ресурсы проведения эксперимента. Существует большое число планов, обладающих разными свойствами и оптимальных по широкому набору критериев, и выбор плана, удовлетворяющего ограничениям на ресурс опыта, уже становится проблемой. В технологическом аспекте весьма актуально автоматизировать процесс выбора плана по заказу исследователя, не обладающего специальной подготовкой в планировании экспериментов.

Комплекс МК PLSIM позволяет автоматизировать цикл исследований на имитационной модели: моделирование варианта, обработка данных, формирование планов и повторение опыта. В качестве базовых в МК PLSIM были отобраны планы и разработаны алгоритмы синтеза планов, обеспечивающие малое время синтеза при ограниченной емкости основной памяти для программ, реализующих эти алгоритмы. За основу синтеза были взяты двухуровневый факторный эксперимент и неполноблочные сбалансированные планы.

На этапе *эксплуатации* ИМ МК PLSIM позволяет автоматизировать некоторые функции. Например, возможна автоматизация представления результатов эксперимента. Она достигается благодаря стандартным средствам отображения информации выдачи на АЦПУ графиков и временных диаграмм, протоколирования процесса имитации, выдачи стандартных таблиц, содержащих результаты имитации. МК PLSIM обладает необходимым набором средств управления имитацией. Пользователь может каталогизировать в качестве стандартных любое число процессов-контролеров, в функции которых входит управление процессом моделирования. С помощью процесса-лидера пользователь может организовать любой режим «разгона» модели в стационарное состояние.

Наличие средств управления сбором статистики позволяет фиксировать статистику по любому алгоритму, проводить разделение статистик переходного и стационарного состояний моде-

ли. МК PLSIM имеет средства останова имитации и продолжения ее с места прерывания, что существенно при организации модельного эксперимента в условиях ограничения времени работы ЭВМ на один заход. Это позволяет исследовать действительно большие имитационные модели, требующие для своей работы много машинного времени и больших размеров памяти. Наличие условных операторов в МК PLSIM позволяет запрограммировать любые процедуры управления процессом имитации в модели. Множество операторов управления объектами модели существенно расширяет возможности МК PLSIM по управлению: процессами, устройствами, трассами, очередями, флагами, сигналами. Статистическая база МК PLSIM достаточно разнообразна: оперативная статистика, обрабатываемая в ходе имитации; стандартная статистика, обрабатываемая по окончании варианта моделирования; общая статистика, обрабатываемая в другое время специально написанной для этой цели программой. Таким образом, пользователю облегчается решение проблемы построения ИМ больших систем при наличии ограничений на оперативную память. Ориентация библиотеки специальных процессов МК PLSIM на моделирование архитектуры вычислительных систем позволяет существенно ускорить все технологические этапы построения и эксплуатации их ИМ.

5.3. ПОВЫШЕНИЕ УРОВНЯ ТЕХНОЛОГИИ МОДЕЛИРОВАНИЯ С ПОМОЩЬЮ МОДЕЛИРУЮЩЕГО КОМПЛЕКСА АСИМ

5.3.1. ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ МОДЕЛЕЙ СЛОЖНОЙ СИСТЕМЫ

При разработке ИМ сложных вычислительных систем, одной из проблем является обеспечение достаточно простого перехода от временной диаграммы взаимодействия компонент системы к модели системы. В моделирующем комплексе АСИМ [71] предложен один из возможных способов такого перехода. В основе этого способа лежит представление вычислительных процессов, позволяющее отразить конкуренцию задач за ресурсы вычислительной системы в мультипрограммном режиме. При этом выделяются конкуренции за центральный процессор (ЦП), каналы обмена, основную память. Для организации имитации такой конкуренции необходима строгая регламентация порядка работы устройств, что достигается введением в язык МК АСИМ специальных средств управления, обеспечивающих разрешение, запрет и прерывание обслуживания требований устройствами.

Следует заметить, что хотя МК АСИМ разрабатывался для моделирования ВП в ВС, однако использованный в нем метод универсален. Поэтому МК АСИМ может быть применен при создании ИМ любых сложных систем. В силу наглядности графиче-

ского представления ВП открываются широкие возможности этого метода при построении моделей.

Сущность графического представления компонент модели сложной системы состоит в следующем. Любую компоненту сложной системы можно представить прибором массового обслуживания, поведение которого изображается графически, как показано на рис. 5.1. Возможное поведение каждого такого прибора существенно отличается от поведения обычного прибора массового обслуживания. Во-первых, на входе и выходе возможно изображение любого числа очередей транзактов. Во-вторых, любой такой прибор может обслуживать любое число транзактов по индивидуальному для каждого из них закону распределения времени их обслуживания $F(t_{ij})$, здесь i — идентификатор транзакта, а j — идентификатор элемента сложной системы. В-третьих, каждый такой прибор имеет три управляющих входа: «включить»,

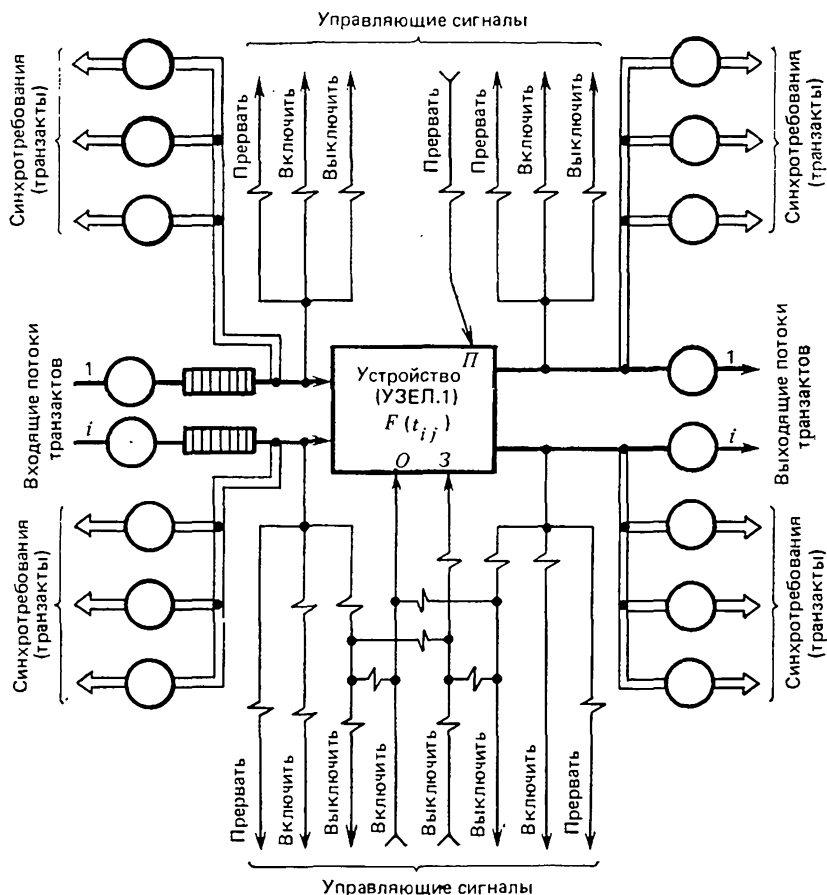


Рис. 5.1. Обобщенный узел сети СМО

«выключить» и «прервать». На эти управляющие входы могут поступать соответствующие управляющие сигналы. Если прибор выключен, на входе его образуются очереди, несмотря на то, что прибор может быть свободен от обслуживания. Это состояние может продолжаться до тех пор, пока не поступит сигнал «Включить» прибор. Если во время обслуживания i -го транзакта прибором j приходит сигнал «Прервать», то происходит прерывание этого обслуживания. Прибор переходит в состояние «Выключен», а i -й транзакт возвращается первым в очередь, где ожидает продолжения обслуживания. Прибор будет находиться в состоянии «Выключен» до прихода управляющего сигнала «Включить» прибор j . В-четвертых, на выходе прибора указывается дальнейшая траектория i -го транзакта по окончании его обслуживания j -м прибором. Возможен либо вероятностный выбор направления i -го транзакта по вектору p_{ij} , либо детерминированный выбор маршрута его движения. В-пятых, и на входе, и на выходе прибора допускаются следующие переходы:

размножение i -го транзакта на любое указанное заранее число транзактов-копий;

формирование из i -го транзакта любого указанного заранее числа управляющих сигналов всех трех типов («Включить», «Выключить» и «Прервать»).

В результате изменение состава управляющих сигналов, мест их генерации при постоянном составе обслуживающих приборов позволяет изменять логику функционирования сложной системы. Отсутствие элементов логического управления в объекте моделирования приводит к типовым моделям систем массового обслуживания [42].

На рис. 5.1 введены следующие обозначения. Синхротребования означают копии транзактов, формируемых из входных и выходных потоков транзактов. Кружками обозначены источники транзактов. Источниками могут быть как другие узлы, так и данный узел при формировании синхротребований. Двойными линиями показаны маршруты движения синхротребований. Жирными линиями отмечены маршруты транзактов по сети СМО. Управляющие сигналы обозначаются тонкими линиями; $F(t_{ij})$ означает функцию распределения длительности обслуживания i -го транзакта j -м узлом сети СМО.

Для организации эксперимента с моделями сложных систем необходимо было разработать способ перевода графического представления в набор утверждений (терминов). В этих терминах должны быть полностью представлены объект моделирования и логика его функционирования. В [54] предложен двухэтапный способ перевода графических моделей в программы имитационных моделей. Вначале графическое представление сложной системы записывается набором декларативных утверждений, отражающих все аспекты функционирования модели. К их числу относятся: состав устройств проектируемого объекта, связи между ними, логика функционирования и взаимодействия устройств. За-

тем полученное описание объекта исследования автоматически переводится в машинную имитационную модель с помощью специального программного обеспечения. Поэтому задача исследователя состоит лишь в построении графической модели и описании ее набором стандартных терминов. Напомним еще раз, что совокупность таких терминов не представляет собой программу ЭВМ в общепринятом представлении.

Таким образом, для создания имитационной модели с помощью МК АСИМ необходимо последовательно описать устройства набором предложений, каждое из которых предназначено для задания конкретных свойств узла сети. Эти предложения характеризуют следующие общеизвестные аспекты теории массового обслуживания: ресурс памяти [30, 36], входящие потоки транзактов [31], очереди и дисциплины их обслуживания [16], времена обслуживания транзактов [40, 42], входящие потоки [52], маршруты транзактов [13, 33, 82], логику управления состоянием устройств [36, 40]. Совокупность описаний всех устройств в соответствии с установленными синтаксическими правилами представляет собой текст, готовый для автоматического перевода в машинную имитационную модель.

5.3.2. СРЕДСТВА МК АСИМ, ПОВЫШАЮЩИЕ УРОВЕНЬ ТЕХНОЛОГИИ СОЗДАНИЯ ИМИТАЦИОННЫХ МОДЕЛЕЙ

На этапе *формализации* при создании ИМ сложных систем с помощью МК АСИМ используется специально разработанный для этих целей декларативный язык АСИМ. Каждый элемент системы на языке АСИМ описывается комбинацией из следующих девяти предложений: УСТРОЙСТВО, ВХОДЯЩИЙ ПОТОК, ОЧЕРЕДЬ, МЕХАНИЗМ ОБСЛУЖИВАНИЯ, ВРЕМЯ ОБСЛУЖИВАНИЯ, МЕХАНИЗМ ПОДКЛЮЧЕНИЯ, ВРЕМЯ ОТКАЗА, МЕХАНИЗМ ДООБСЛУЖИВАНИЯ, ВЫХОДЯЩИЙ ПОТОК.

Модель представляется в виде сети СМО, состоящей из узлов (обслуживающих устройств) и связей между ними (потоков транзактов, называемых требованиями). Проводится декомпозиция сложной сети на отдельные узлы, а затем отождествляются компоненты сети с соответствующими языковыми единицами комплекса.

Как уже было сказано, узел описывается комбинацией из девяти типов предложений. Причем первое предложение идентифицирует узел сети, а остальные описывают его характерные особенности. Совокупность описания всех узлов в соответствии с установленными синтаксическими правилами МК АСИМ представляет собой общий текст модели, который завершается предложением ВРЕМЯ МОДЕЛИРОВАНИЯ.

Предложение УСТРОЙСТВО идентифицирует каждый из узлов с помощью указателя имени устройства ТИП, который может иметь смысловую нагрузку, и указателя приоритета устройства ИНДЕКС. При моделировании систем с ненадежными элементами

ми, в которых возможно использование резервных устройств, включаемых в работу при выходе из строя основного устройства, резервное устройство задается с помощью ключевого слова РЕЗЕРВ в предложении УСТРОЙСТВО и последующего описания идентификатора резервного устройства.

Предложение ВХОДЯЩИЙ ПОТОК характеризует процесс поступления требований на вход обслуживающего устройства (ОУ). Описание входного потока транзактов состоит из перечисления источников, поставляющих транзакты на обслуживание. Источники транзактов подразделяются на внесистемные и внутрисистемные. Внесистемные источники задают порядок поступления транзактов в систему извне и определяются уникальным номером и одним из возможных законов распределений: равномерным, экспоненциальным, нормальным, Эрланга, Вейбулла, Парето или табличным. При исследовании СМО с приоритетами во входящем потоке каждому источнику присваиваются определенный класс и приоритет согласно предусматриваемым дисциплинам обслуживания. Для регулирования процесса поступления транзактов во времени с помощью конструкции ПЕРИОД ЗАПУСКА задаются активный и пассивный периоды источника транзактов.

Внутрисистемные источники задают порядок поступления транзактов из других узлов сети. Любой из внутрисистемных источников характеризуется уникальным номером, заданным заранее, и идентификатором узла, из которого поступают транзакты. При моделировании систем с ненадежными элементами и резервированием для описания входов дополнительно используются конструкции ДЛЯ ОГРАНИЧЕНИЙ. Наличие этой конструкции указывает на то, что поступления транзактов от этих источников на резервное ОУ будут планироваться только в случае отказа основного устройства. Для имитации систем с внутренним управлением МК АСИМ позволяет переводить устройство из состояния ГОТОВО в состояние ЗАНЯТО. В первом случае устройство потенциально готово к обслуживанию поступающих на его вход транзактов, а во втором — обработка запрещена. Перевод узла из одного состояния в другое производится сигналами «Включить» и «Выключить». Генерация этих сигналов может быть проведена любым из внутрисистемных потоков транзактов в момент их поступления на вход к ОУ. Задание режима генерации сигналов производится посредством конструкции С РАЗДЕЛЕНИЕМ НА СИГНАЛЫ ТИПА.

Предложение МЕХАНИЗМ ОБСЛУЖИВАНИЯ задает совокупность правил, которые определяют порядок обработки поступающих на устройство транзактов. Предложение содержит описатели дисциплин обслуживания и ряд дополнительных указателей в случае моделирования систем с отказами. Возможны следующие дисциплины обслуживания: FIFO, LIFO, RANDOM (случайный выбор из очереди), RR, FB (круговые циклические алгоритмы), приоритетные и с отказами. Для приоритетных дисциплин

указывается тип приоритета с помощью указателя ДИНАМИЧЕСКИЙ или ВНЕШНИЙ. Кроме того, возможна дополнительная идентификация приоритета с помощью указателей АБСОЛЮТНЫЙ, ОТНОСИТЕЛЬНЫЙ, СМЕШАННЫЙ, АБСОЛЮТНО-ОТНОСИТЕЛЬНЫЙ и МНОГОУРОВНЕВЫЙ. Задаaniem конструкции с ПОТЕРЯМИ или ПОВТОРНОЕ ОБСЛУЖИВАНИЕ указывается порядок обработки транзакта после прерывания его обслуживания из-за прихода более приоритетного. При этом характер повторного обслуживания задается конструкцией С НОВЫМ ВРЕМЕНЕМ или С ПРЕЖНИМ ВРЕМЕНЕМ.

Предложение ОЧЕРЕДЬ описывает состав входных очередей к устройству. Возможно задание трех типов очередей к устройству с помощью указателей: ОБЩАЯ, ОТДЕЛЬНАЯ, СМЕШАННАЯ. Общая очередь предполагает постановку в одну очередь транзактов от всех источников, перечисленных в предложении ВХОДЯЩИЙ ПОТОК. Конструкция ОТДЕЛЬНАЯ предполагает создание очереди для каждого из источников, а смешанная очередь является комбинацией предыдущих. Место, занимаемое поступающими транзактами в очереди, определяется механизмом обслуживания. На формирование очередей могут налагаться ограничения: длины очереди, времени пребывания в очереди и вероятности постановки транзакта в очередь. Для задания этих ограничений в предложении имеются указатели: ДЛИНА, ВРЕМЯ, ВЕРОЯТНОСТЬ. При отказе от обслуживания транзакт либо покидает систему, либо поступает на обслуживание к устройству, описанному следом за ограничением. Для СМО с несколькими очередями на входе к ОУ возможно задание режима циклического обслуживания очередей с помощью конструкции ОПРОС ЦИКЛИЧЕСКИЙ. На устройствах с ограниченными очередями при их переполнении может быть задан специальный режим конструкции БЕЗ ПОТЕРЬ. В этом случае транзакт, застав ОУ занятым и очередь переполненной, не покидает предыдущее устройство до тех пор, пока не появится свободное место в очереди.

Предложение МЕХАНИЗМ ПОДКЛЮЧЕНИЯ описывает различные стратегии подключения свободных устройств. Возможны следующие стратегии подключения, задаваемые конструкциями: В ПОРЯДКЕ ПРИОРИТЕТА, В ПОРЯДКЕ ОСВОБОЖДЕНИЯ, В СЛУЧАЙНОМ ПОРЯДКЕ, В ПОРЯДКЕ ГРУППОВОГО ПРИОРИТЕТА. В некоторых СМО поступивший транзакт не может тотчас же быть принятым на обслуживание, так как для включения или подготовки ОУ к работе требуется некоторое время «разогрева». Если у транзактов разных потоков законы распределения времени «разогрева» различны, то имеет место «разогрев первого рода», а если одинаковы — «разогрев второго рода». Вид «разогрева» указывает пользователь в конце предложения с помощью соответствующих конструкций: РАЗОГРЕВ 1 РОДА; РАЗОГРЕВ 2 РОДА. В этих конструкциях используются указатели типа РАСПРЕДЕЛЕНИЕ. Для моделирования СМО с малой нагрузкой вводится описатель ПОРОГ ВКЛЮЧЕНИЯ. В

этом режиме ОУ начинает обслуживание, когда на входе скопится число транзактов, превышающее пороговое значение.

Предложение ВРЕМЯ ОБСЛУЖИВАНИЯ описывает длительность обслуживания транзактов устройством. Описатель РАСПРЕДЕЛЕНИЕ задает тип и параметры закона распределения длительности обслуживания транзактов. Если для каждого класса транзактов обслуживание различно, то необходимы описатели законов распределения времени обслуживания для каждого источника отдельно. Поскольку в СМО широкое распространение получил способ обслуживания с использованием квантования, МК АСИМ позволяет использовать конструкцию С РЕЖИМОМ КВАНТОВАНИЯ, в которой указывается квант времени, выделяемый для обслуживания.

Предложение ВРЕМЯ ОТКАЗА используется для описания СМО с ненадежными элементами. Задается длительность безотказной работы, распределенная по закону, указанному описателем РАСПРЕДЕЛЕНИЕ. В течение определенного времени отказавшее ОУ может быть восстановлено, после чего процесс обслуживания ИМ возобновляется. Длительность восстановления также является случайной величиной, распределенной по закону, указанному вторым описателем РАСПРЕДЕЛЕНИЕ. Задание времени восстановления не является обязательным, и в случае его отсутствия считается, что отказавшее устройство не восстанавливается до конца моделирования. Требования, поступившие на вход отказавшего ОУ, заполняют очередь по обычным правилам, если для последнего не было задано резервное устройство.

Предложение МЕХАНИЗМ ДООБСЛУЖИВАНИЯ описывает судьбу транзакта, в момент обработки которого произошел отказ ОУ. Это предложение имеет смысл только при наличии предложения ВРЕМЯ ОТКАЗА. Предложение конструируется из описателей: ТРЕБОВАНИЯ ДООБСЛУЖИВАЮТСЯ, ТРЕБОВАНИЯ ТЕРЯЮТСЯ ИЗ-ЗА ОТСУТСТВИЯ СВОБОДНЫХ УСТРОЙСТВ, ТРЕБОВАНИЯ ПОВТОРНО ОБСЛУЖИВАЮТСЯ. В случае повторного обслуживания используются описатели С ПРЕЖНИМ ВРЕМЕНЕМ и С НОВЫМ ВРЕМЕНЕМ. Характер повторного обслуживания задается описателями В ПЕРВУЮ ОЧЕРЕДЬ, НА ОБЩИХ ОСНОВАНИЯХ.

Предложение ВЫХОДЯЩИЙ ПОТОК применяется в тех случаях, когда после прохождения этапа обслуживания транзакт является источником новых транзактов. Процесс деления задается посредством конструкции с РАЗДЕЛЕНИЕМ НА ЗАЯВКИ. Правила планирования новых транзактов на дальнейшую обработку отличны от стандартных. Маршруты их следования могут быть как детерминированными, так и вероятностными и задаются комбинациями конструкций СПИСОК УСТРОЙСТВ и С ВЕРОЯТНОСТЬЮ. Особенностью выходящего потока является свойство генерации управляющих сигналов «Включить» и «Выключить», переводящих устройства в состояние ГОТОВО или ЗАНЯТО. Задание управляющих сигналов осуществляется с помощью кон-

струкции С РАЗДЕЛЕНИЕМ НА СИГНАЛЫ ТИПА. Задается также способ определения дальнейшего маршрута следования обслуженных транзактов. Имеется возможность выбора маршрута на основе вероятностей, приписанных каждому из альтернативных устройств. Множества устройств и их весов составляют условный список, а задание случайного способа планирования осуществляется посредством конструкции ПЛАНИРОВАНИЕ ПО ВЕРОЯТНОСТИ.

Таким образом, в распоряжение исследователя предоставляется декларативный неалгоритмический язык описания одного класса сложных систем большой размерности и стохастических сетей массового обслуживания АСИМ, который имеет следующие особенности:

- наличие средств управления структурой сети, технологии связей, алгоритмов маршрутизации транзактов, стартстопного режима работы узлов сети;

- возможность описания систем с ненадежными элементами (учитываются отказы, резервирование и восстановление устройств);

- широкий спектр законов распределения входящих потоков и дисциплин обслуживания, наиболее часто встречающихся в практике исследования сложных систем;

- наличие средств организации сложной системы очередей, допускающей выбор транзактов произвольными подмножествами узлов сети.

Указанные особенности МК АСИМ предоставляют исследователям ряд дополнительных возможностей построения имитационных экспериментов по управлению сетью СМО, организации стартстопной работы устройств, интегрированному описанию сложных систем, отражению ресурсных свойств узлов сети, управлению процессом генерации транзактов.

Для сокращения записи модели используется понятие «ссылка», которое позволяет взять за основу описание идентичного предложения устройства, на которое указывает ссылка. Допускается задание ссылок на любые узлы без ограничения их числа. Возможно формирование описания узла с помощью текста двух и более различных устройств. Основной недостаток языка МК АСИМ заключается в громоздкости текста описания сети. Это приводит к увеличению времени построения модели на внутреннем языке МК АСИМ и времени интерпретации моделируемого процесса. Возникают очень большие объемы информации при описании сетей СМО даже средней размерности. Особенно это существенно при описании множества входящих потоков.

Пример. Рассмотрим в качестве примера двухуровневую вычислительную систему (рис. 5.2), которая на первой ступени имеет n мини-ЭВМ, к входам которых подключено m абонентов (АБОН. j), $j = \overline{1, m}$. Информация от абонентов через мини-ЭВМ поступает в центральную ЭВМ по любому из q каналов. Таким образом, число входов в $(n+1)$ -ю очередь будет равно (qmn) .

Кроме того, в результате обменов на центральную ЭВМ поступают требования от a накопителей на магнитных дисках (МД. l), $l=\overline{1, a}$ и от b накопителей на магнитных лентах (МЛ. k), $k=\overline{1, b}$. Общее число входов к центральной ЭВМ будет равно $mn(q+b+a)$. Для сети средней размерности (например, $m=6$, $n=15$, $q=6$, $a=12$, $b=6$) число потоков требований к центральной ЭВМ будет несколько тысяч (в нашем случае 2160).

Для сокращения текста модели в подобных ситуациях в МК АСИМ разработаны специальные конструкции группового описания. Например, описание 2160 входящих потоков центральной ЭВМ (см. рис. 5.2) представляется следующим образом:

ВХОДЯЩИЙ ПОТОК: (ИП=1—90 У (КАНАЛ.1)2—6),
(ИП=1—9 У (МД.1) 2—12), (ИП=1—9 У (МЛ.1) 2—6);

Средства группового описания применимы для отображения практически всех аспектов функционирования узла, включая различные типы входов, системы очередей, времена обслуживания и свойства выходящего потока. Преимущества применения средств группового описания очевидны. Они обеспечивают снижение трудоемкости составления текста модели (в данном примере на три порядка). При этом достигается значительная экономия оперативной памяти ЭВМ при отображении описания модели на внутреннем языке МК АСИМ (вместо 2160 структур данных достаточно использовать 3). Существенно уменьшается расход памяти на отображение дополнительных свойств входящих и выходящих потоков (генерации сигналов включения и выключения устройств, деления сигналов и требований). Уменьшение объема информационных списков приводит к значительному сокращению реального времени моделирования за счет ускорения поиска необхо-

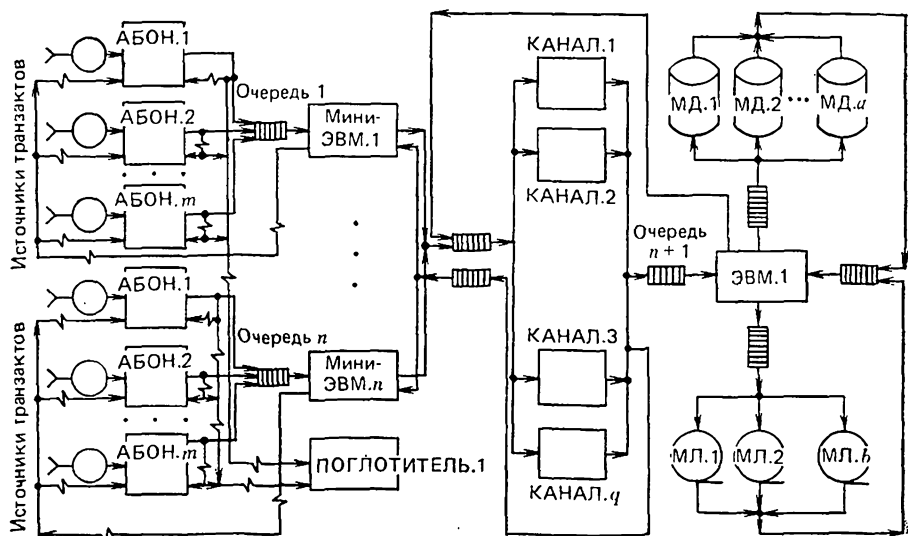


Рис. 5.2. Схема ИМ двухуровневой ВС

димого элемента списка. Пропорционально уменьшается и время интерпретации.

Этап *перехода к описанию имитационной модели* при использовании МК АСИМ практически отсутствует, поскольку формальное описание на языке АСИМ представляет собой описание имитационной модели. Исследователю не приходится заботиться об установке операторов синхронизации компонент ИМ. В тех случаях, когда в ИМ необходимо отразить конкуренцию за ресурсы, язык МК АСИМ представляет возможность закреплять за узлами СМО некоторые «ресурсы» в условных единицах. Поступающие на вход такого устройства транзакты обслуживаются только в том случае, если устройства свободны и имеют необходимый свободный «ресурс».

Ресурс задается либо детерминированно, либо согласно установленному закону распределения вероятностей. При наличии альтернативных законов выбор конкретного типа распределения может производиться согласно вероятности, предписанной каждому из них. Если необходимый ресурс отсутствует, то транзакты устанавливаются в очередь к ресурсу и находятся в ней до тех пор, пока значение свободного ресурса не превысит минимального уровня, необходимого очередному транзакту. Удовлетворение запроса на ресурс приведет к переводу транзакта из фазы ожидания ресурса в фазу обслуживания (одновременно с этим уменьшается текущее значение свободного ресурса). Тем не менее перевод транзакта в фазу обслуживания не означает, что транзакт немедленно поступит на обслуживающее устройство.

Обслуживание транзакта производится только тогда, когда устройство свободно. В противном случае транзакт устанавливается в очередь к устройству. По завершении обслуживания транзакт может вернуть занятый «ресурс». При этом значение свободного ресурса увеличивается, что может привести к переводу одного или нескольких транзактов из фазы ожидания ресурса в фазу обслуживания. Отметим, что ресурс может быть освобожден на выходе любого из устройств, входящих в структуру сети. Это означает, что один транзакт может занимать одновременно несколько ресурсов разных типов.

В ряде случаев удобно использовать возможности выбора ресурсов в узлах сети в целях блокировки обслуживающих устройств. Захват транзактом ресурса может привести к тому, что по завершении обслуживания и при отсутствии конструкций освобождения ресурса устройство фактически будет заблокировано. Снятие блокировки может быть произведено на выходе любого узла сети с помощью конструкции **ОСВОБОДИТЬ РЕСУРС**.

Пример. В качестве примера, иллюстрирующего аспекты создания ресурсной модели, рассмотрим ВС, представленную на рис. 5.2. Пусть устройство (ЭВМ.1) обладает ресурсом в 768 единиц. На входе к ЭВМ формируются четыре очереди. Транзакты, поступающие от мини-ЭВМ через каналы связи, выдают запросы на ресурс. Предположим, что в центральной ЭВМ решаются

три класса задач, каждый из которых выставляет свой запрос на ресурс ЭВМ. Пусть задачи выставляют запросы равновероятно для каждого класса в пределах 20—60; 80—120; 120—320 единиц соответственно для классов А, В и С. Причем вероятности поступления транзактов указанных классов равны соответственно 0,6; 0,3; 0,1. Тогда описание входящего потока на устройство (ЭВМ, 1) можно представить следующим образом:

УСТРОЙСТВО (ЭВМ, 1) РЕСУРС=768;
 ВХОДЯЩИЙ ПОТОК: (ИП=1—90, У (КАНАЛ, 1) 2—6)
 ЗАХВАТИТЬ РЕСУРС (0,6 РАВНОМЕРНОЕ (20, 60),
 0,3 РАВНОМЕРНОЕ (60, 20), 0,1 РАВНОМЕРНОЕ (120, 320)),
 (ИП=1—90 У (МД, 1) 2—12), (ИП=1—90 У (МЛ, 1) 2—6);

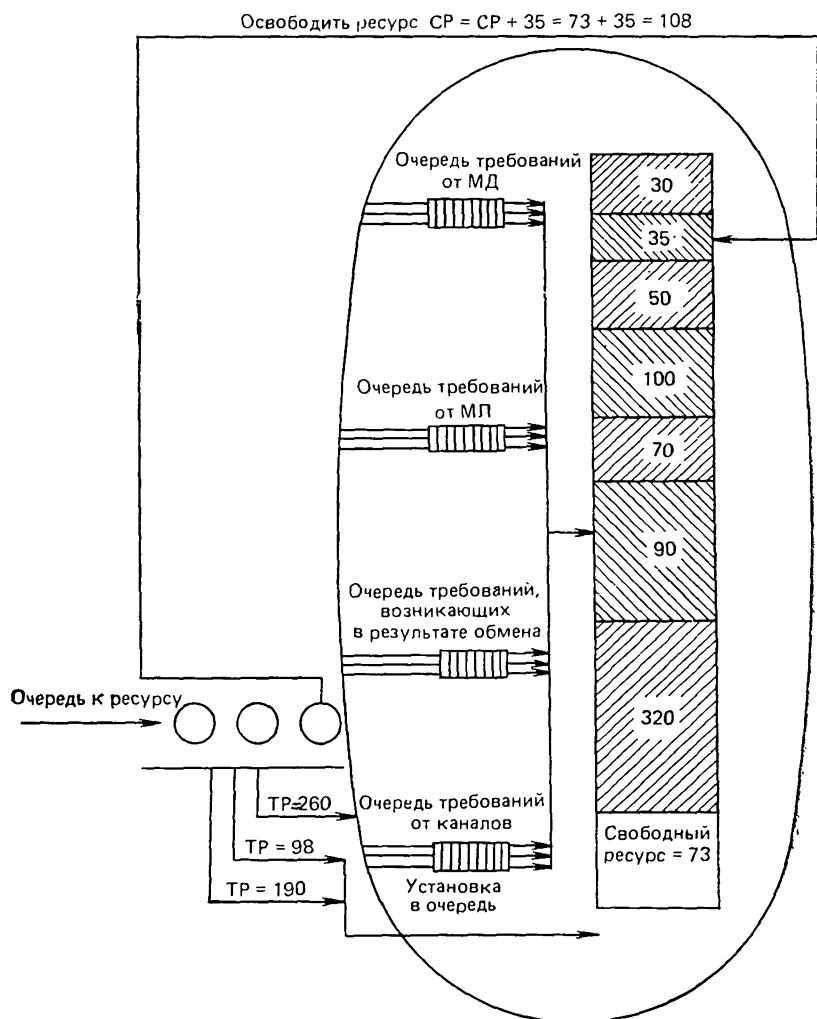


Рис. 5.3. Схема ресурсной ИМ сети СМО

Схематически алгоритм выделения и освобождения ресурса проиллюстрирован на рис. 5.3. К моменту рассмотрения основную память ЭВМ занимают семь активных задач, и им выделены области, равные соответственно 30, 35, 50, ..., 320 единиц. Хотя свободный ресурс памяти составляет 73 единицы, транзакты, находящиеся в очереди к ресурсу, не могут быть переведены в фазу обработки, поскольку выставленные ими запросы на ресурс превышают неиспользованный остаток памяти. Окончание обработки транзакта, захватившего ресурс объемом 35 единиц, увеличит свободный ресурс до 108 единиц. Будет проведен просмотр очереди к устройству. Транзакт, выставивший запрос на ресурс объемом 98 единиц, будет переведен в фазу обработки. При этом объем свободного ресурса уменьшится до 10 единиц. Дальнейшая обработка транзакта, только что получившего ресурс памяти, будет производиться согласно механизму обслуживания, учитывающему приоритетность входных очередей. Отметим, что модифицированный алгоритм захвата и освобождения памяти может учитывать и фрагментацию памяти.

Этап *программирования* при создании ИМ с помощью МК АСИМ практически исключается, поскольку превращение математической модели в программу ИМ состоит в автоматической кодировке предложений языка МК АСИМ на внутренний язык программной-интерпретатором. Отладка ИМ сводится к анализу синтаксиса конструкций входного языка. О всех ошибках синтаксиса программа-интерпретатор МК АСИМ уведомляет пользователя. Только при отсутствии синтаксических ошибок исходный текст ИМ преобразуется в списковые структуры внутреннего языка МК АСИМ. Для документирования МК АСИМ выдает исследователю листинг с описанием текста модели и список ошибок описания модели. Даже неправильно сформированная, но записанная синтаксически верно, ИМ запускается на имитацию автоматически. Поэтому процесс комплексной отладки ИМ, записанных на языке МК АСИМ, носит характер анализа полученных данных.

5.3.3. СРЕДСТВА МК АСИМ, ПОВЫШАЮЩИЕ УРОВЕНЬ ТЕХНОЛОГИИ ИСПЫТАНИЯ И ЭКСПЛУАТАЦИИ ИМИТАЦИОННЫХ МОДЕЛЕЙ

На этапе *испытания* ИМ исследователь выполняет основные функциональные действия на основании анализа содержания стандартизованных выданных статистики моделирования узлов СМО и времен жизни транзактов. При проверке адекватности МК АСИМ предоставляет исследователю возможность задать однопрогонный либо многопрогонный режим работы. Первый предполагает одноразовое исследование модели на интервале $(0, T_n)$, а второй — N -кратное повторение однопрогонного режима с использованием независимых реализаций внешних случайных воздействий. Средства проверки гипотез о близости средних значений отклика в модели и в реальной системе в МК АСИМ отсутствуют. Однако, поскольку выдачи результатов моделирования стандартизованы и имеется возможность их записи во внешнюю

память ЭВМ, при проверке адекватности модели реальному объекту можно использовать любые стандартные процедуры обработки результатов моделирования.

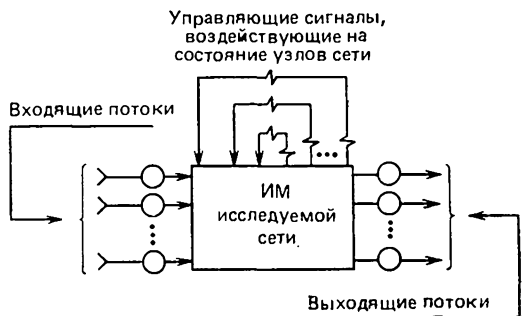
Этап исследования свойств ИМ обеспечивается в МК АСИМ наличием предложения ВРЕМЯ МОДЕЛИРОВАНИЯ, задающим интервал времени, в течение которого необходимо имитировать поведение СМО. Наличие в этом предложении специальной конструкции СБОР СТАТИСТИКИ В СТАЦИОНАРНОМ РЕЖИМЕ обеспечивает автоматическое наблюдение за моментом входа контролируемого параметра в стационарный режим путем оценки средней длины очереди. Контроль за указанной величиной может быть осуществлен на любом устройстве, входящем в состав модели. Для этого фиксируются идентификаторы контролируемого устройства посредством конструкции УСТРОЙСТВО. Конструкцией ПАРАМЕТР задается временной интервал Δt , по истечении которого происходит вычисление оценки средней длины очереди. Проверка ведется согласно критерию Вилкоксона, который требует задания уровня значимости с помощью конструкции ДОВЕРИТЕЛЬНАЯ ВЕРОЯТНОСТЬ.

Для достижения необходимой надежности заключения о стабилизации оценки средней длины очереди вводится коэффициент подтверждения, который позволяет отложить до следующей проверки заключение о достижении стационарного состояния. Для этой цели используется описатель КОЭФФИЦИЕНТ ПОДТВЕРЖДЕНИЯ.

В ряде случаев кроме требования сбора статистики в стационарном режиме необходимо также оценить характер реального переходного процесса. Наличие в предложении ВРЕМЯ МОДЕЛИРОВАНИЯ конструкции ПЕЧАТЬ ГРАФИКА СРЕДНЕГО позволяет осуществить в конце моделирования выдачу на печать графика, отражающего зависимость средней длины очереди от времени. Остальные функциональные действия исследователя в МК АСИМ не автоматизированы. Более того, в силу специфики декларативного описания ИМ эти действия практически неформализуемы и требуют использования эвристических способностей исследователя.

Специальных средств для обеспечения этапа *планирования экспериментов* МК АСИМ не имеет. Однако в МК АСИМ достаточно развиты средства прямого управления ИМ. Процессы управления отдельными узлами в сложных сетях, как правило, очень трудно охарактеризовать и тем более реализовать в виде имитационных моделей с помощью традиционных средств автоматизации моделирования в силу следующих свойств сетей СМО: сложной логики функционирования узлов сети; динамической структуры сети; взаимозависимости состояния узлов сети, непосредственно не связанных между собой. Разработка имитационных моделей подобных сетей предполагает использование специальных средств управления, отсутствующих в большинстве известных специализированных языков моделирования. Основу

Рис. 5.4. Имитационная модель с внутренним управлением



практической реализации управления в МК АСИМ составляет совокупность следующих возможностей входного языка по генерации: управляющих сигналов «Включить», «Выключить», «Прервать» и синхронизирующих транзактов с помощью операции деления транзактов по детерминированному либо случайному расписанию. В зависимости от способа применения этих средств управление сетью подразделяется на внутреннее, внешнее и смешанное.

Внутреннее управление стохастическими сетями массового обслуживания достигается транзактами определенных потоков, генерирующих сигналы, которые воздействуют на структуры и состояние узлов сети. Схематически модель с внутренним управлением представлена на рис. 5.4. Генерация сигналов происходит на входе или выходе любого узла сети, предназначенного непосредственно для обслуживания транзактов произвольного множества входящих потоков. Управляющие сигналы могут воздействовать не только на состояния других узлов, но и на состояние исходного узла.

Внешнее управление предполагает разработку независимого блока управления, не входящего в структуру имитационной модели сети (рис. 5.5). Последний также состоит из совокупности обслуживающих устройств. Однако, в отличие от функций обычных ОУ, имитирующих процессы обслуживания, функции узлов управления заключаются в генерации множества управляющих сигналов, которые в качестве сигналов обратной связи по управлению воздействуют на процесс работы имитационной модели исследуемой сети. Основой возникновения управляющих сигналов могут быть либо синхротребования, поступающие от устройств ИМ исследуемой сети, либо внешние независимые источники, либо и те, и другие транзакты вместе. Синхронизирующие транзакты поступают в блок управления в результате операций деления на входах или выходах ОУ имитационных моделей. По сути они информируют блок управления о наступлении тех или иных событий в модели. Блок управления генерирует ответные сигналы, воздействующие определенным образом на структуру и состояние узлов модели сети путем включения, выключения и прерывания работы соответствующих устройств. Наличие независимых тран-

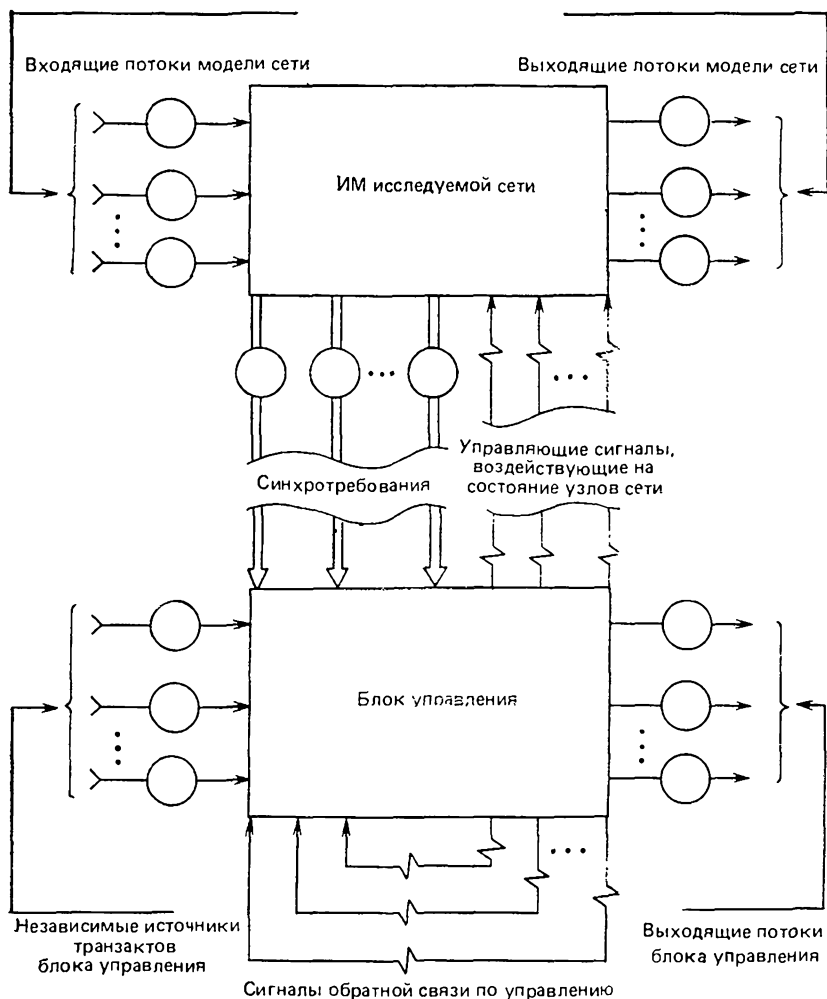


Рис. 5.5. Имитационная модель с внешним управлением

зактов определяется множеством специально предусмотренных потоков синхротребований, циркулирующих только в блоке управления. Расписание генерации независимых транзактов определяет, в свою очередь, характер управления. Регулярное появление независимых транзактов согласно жесткому расписанию реализует детерминированное управление сетью СМО. Случайный характер появления транзактов задается описанием законов распределения вероятностей периодов активности и пассивности источников транзактов.

Совместное использование обоих вышеназванных типов транзактов позволяет имитировать весьма сложный в логическом от-

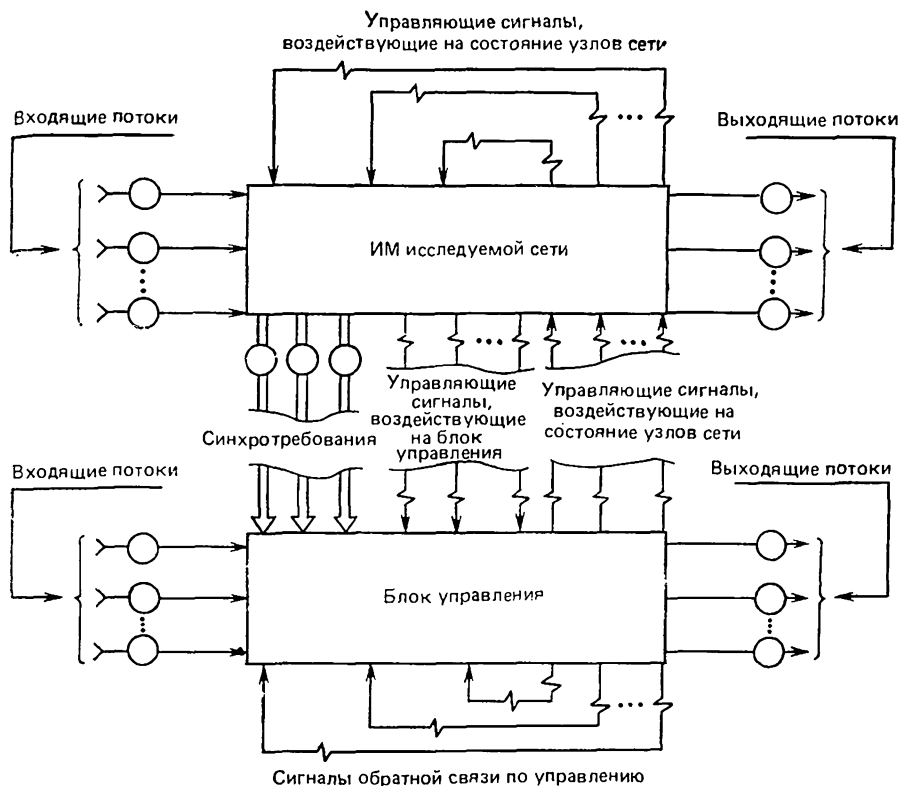


Рис. 5.6 Имитационная модель со смешанным управлением

ношении процесс управления. Смешанный способ управления предполагает зависимость состояния логических элементов блока управления от процессов, протекающих в моделируемой сети (рис. 5.6). Конкретное выражение этой зависимости реализуется в виде сигналов, поступающих из имитационной модели в блок управления. Последние производят установку состояния логических элементов блока управления соответственно ситуации, сложившейся в модели. С помощью синхронизирующих транзактов (синхротребований) либо транзактов, поступающих от независимых источников, происходит срабатывание соответствующих управляющих сигналов, воздействующих определенным образом на другие узлы сети СМО. Реакция блока управления на сложившуюся ситуацию в модели путем выработки соответствующих управляющих сигналов представляет собой некоторую форму выражения обратной связи и может реализовать синхронное и асинхронное управление. Сигналы, выдаваемые блоком управления, могут динамически изменять структуру сети путем включения, выключения и прерывания работы устройств, вследствие чего воз-

можно перераспределение потоков сети в зависимости от ее текущего состояния.

Пример. В качестве примера применения синхронного управления ИМ рассмотрим фрагмент сети (рис. 5.7), состоящий из 15 обслуживающих устройств с идентификаторами (ПРИБОР.1), ..., (ПРИБОР.15). Совокупность этих устройств представляет собой некий общий ресурс, например разделы основной памяти ЭВМ. Полагаем, что одновременно процессор может обрабатывать задание только одного из разделов; обслуживание производится в строгой очередности независимо от порядка поступления задач транзактов. Тем самым определяем приоритет обслуживания для каждого типа задач-транзактов. Регулирование очередности выполнения задач производится блоком управления, представленным устройством (УПР.1). На него поступают синхронизирующие транзакты, возникающие в результате операций деления транзактов на синхротребования на входе приборов. Предположим, что в стартовом состоянии модели все устройства находятся в состоянии «Занято». Первый транзакт-копия, поступивший в блок управления, например, с устройства (ПРИБОР.1) будет обработан, и на выходе блока управления появятся два сигнала: «Выключить» устройство (УПР.1) и «Включить» первое устройство. Очередные синхронизирующие транзакты будут устанавливаться в очередях к блоку управления до тех пор, пока последний не включится. Перевод устройства (УПР.1) в состояние «Готово» осуществляется сигналами «Включить», выдаваемыми устройствами, завершившими обслуживание. В момент завершения обслуживания устройства вновь переводятся в состояние «Занято». Одновременно блок управления производит последовательный опрос очередей, сформированных из синхронизирующих транзактов. Наличие требования в очереди

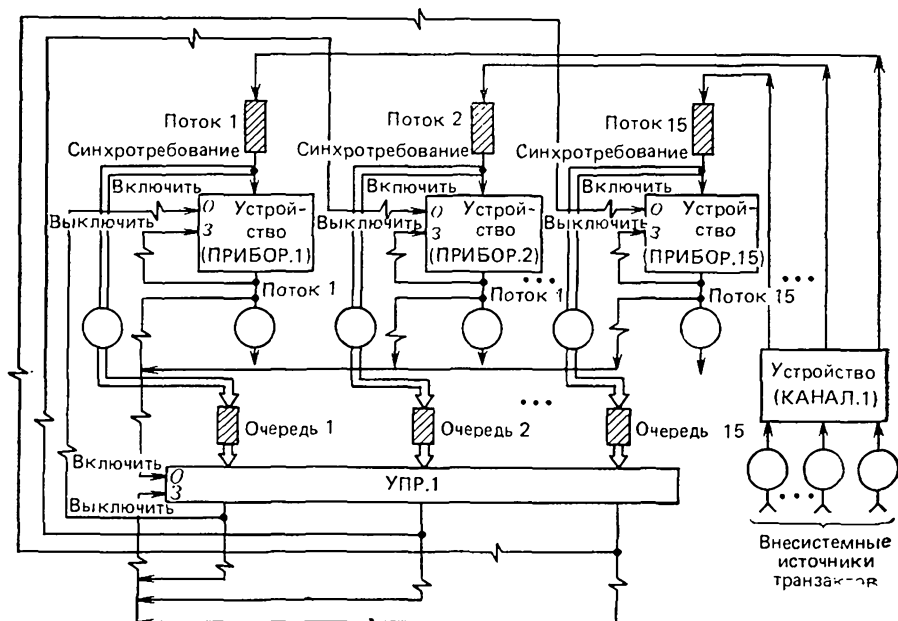


Рис. 5.7. Синхронное управление сетью СМО

немедленно приводит к ответной реакции блока управления, которая заключается в генерации двух сигналов, аналогичных по смыслу сигналам, генерированным первым синхронизирующим транзактом. Порядок опроса очередей устанавливается либо механизмом обслуживания, либо описанием очереди на устройстве (УПР. 1), а скорость реакции — временем обслуживания транзактов.

Из сказанного выше вытекает, что МК АСИМ дает пользователю достаточно средств для конструирования логических схем управления сетью СМО любой сложности. При этом выбор вариантов управления полностью зависит от разработчика. В отличие от способа организации управления на основе алгоритмических языков, здесь вопросы конструирования схем управления не затеняются алгоритмическим представлением, что существенно облегчает поиск и проверку на модели необходимого варианта управления сетью СМО.

Этап *эксплуатации* ИМ, построенной с помощью МК АСИМ, обеспечивается возможностями: управления процессом генерации транзактов, автоматизации представления результатов, сбора стандартной статистики. Введение средств управления процессом генерации транзактов упрощает проектирование логических схем управления сетью. Генерация новых транзактов в СМО осуществляется посредством включения управляемых источников транзактов. Число транзактов регулируется числом сигналов «Включить» источник. Тем самым возможна имитация группового поступления транзактов на обслуживание. Управляемые источники могут быть сгруппированы на одном устройстве, что приводит к экономии памяти и уменьшению объема описания модели на исходном языке. Имитация группового поступления транзактов на обработку возможна также путем задания свойств деления транзактов в выходящем потоке. Нередко эти свойства используются в моделях сетей передачи данных. В ряде случаев при построении ИМ с помощью МК АСИМ необходимо объединить множество копий в один транзакт (осуществить «сборку» транзактов). Для реализации операции сборки транзактов необходимо указать только объем сборки транзактов при описании каждого конкретного входа в устройство, на котором производится сборка транзактов.

Пример. Для иллюстрации применения указанных средств на рис. 5.8 приведен фрагмент модели СМО, состоящий из управляемых источников, имитационной модели самой сети и одного из узлов сети, обладающего ресурсом, к которому транзакты выстраиваются в очередь. Управляемые источники ИП=1 — ИП=3 сгруппированы на устройстве (УПР. 1). Предполагается, что порядок прихода управляющих сигналов на каждый из управляемых источников различен. Так, на источник ИП=2 одновременно поступают 20 сигналов запуска источника, что приводит к генерации на выходе устройства (УПР. 1) 20 транзактов, поступающих в имитационную модель сети. На источник ИП=1 поступают одиночные управляющие сигналы «Включить», поэтому и на выходе устройства (УПР. 1) генерируются одиночные транзакты с индекс-потоком, равным 1. На управляющий источник ИП=3 поступает групповой управляю-

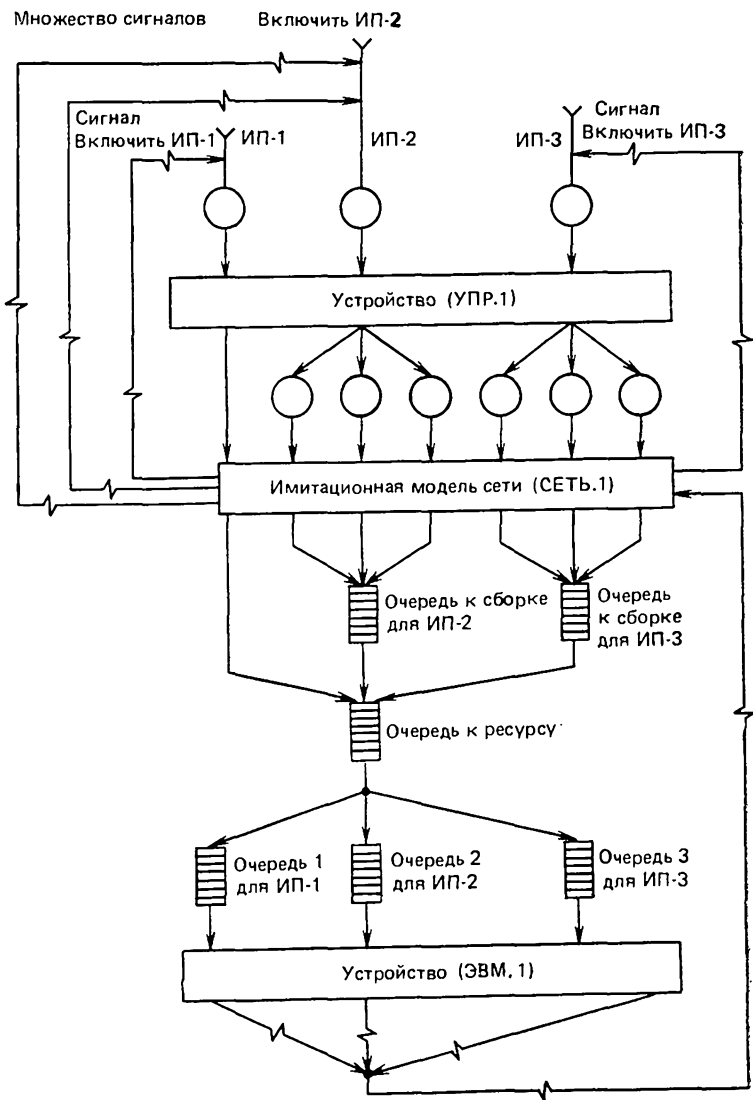


Рис. 5.8. Фрагмент модели СМО с управляемыми источниками

щий сигнал с некоторого узла модели сети. Допустим, что по этому групповому управляющему сигналу необходимо генерировать 25 транзактов, которые затем поступают в модель сети на очередной этап обслуживания.

Не рассматривая подробно структуру и связи узлов модели сети, предположим, что все копии транзактов поступают на устройство (ЭВМ.1) и требуют захвата общего ресурса, объем которого перед началом имитации составляет 400 единиц. На входе узла сети (устройство (ЭВМ.1)) имеются три очереди для каждого из типов транзактов. На рис. 5.8 показано, что операция

сборки транзактов, рождаемых одним и тем же источником, реализуется с помощью соответствующих очередей транзактов. Поскольку все транзакты используют единый ресурс, то они выстраиваются в единую очередь на захват части ресурса. После выделения ресурса некоторому транзакту последний поступает в соответствующую очередь устройства (ЭВМ.1). Удовлетворение запросов транзактов на ресурс производится по алгоритму, описанному выше. Потоки транзактов обозначены на рис. 5.8 ИП=1—ИП=3. Таким образом, возможно неявное создание трехуровневой системы очередей (на сборку, к ресурсу, на обслуживание). Причем явно описываются только очереди на обслуживание, а остальные очереди формируются УПМ МК АСИМ автоматически. Указателями для организации сборки транзактов служат конструкции вида СБОРКА=ИНДЕКС. Описание приведенного фрагмента сети на языке МК АСИМ имеет вид:

УСТРОЙСТВО (УПР.1);

ВХОДЯЩИЙ ПОТОК: (ИП=1—3, УПР);

ВРЕМЯ ОБСЛУЖИВАНИЯ: ПОСТОЯННОЕ (1Е-12);

ВЫХОДЯЩИЙ ПОТОК: (ИП=3) С РАЗДЕЛЕНИЕМ НА ЗАЯВКИ
(25 УСТРОЙСТВО (СЕТЬ.1));

〈ДАЛЕЕ СЛЕДУЕТ ТЕКСТ ОПИСАНИЯ ИМИТАЦИОННОЙ МОДЕЛИ СЕТИ〉

УСТРОЙСТВО (ЭВМ.1) РЕСУРС (400);

ВХОДЯЩИЙ ПОТОК: (ИП=1 У (КАНАЛ.1))

РЕСУРС (ПОСТОЯННОЕ (80)), (ИП=2 У (КАНАЛ.1))

СБОРКА=20 РЕСУРС (РАВНОМЕРНОЕ (20, 320)), ...

(ИП=3 У (КАНАЛ.1))

СБОРКА=25 РЕСУРС (ТАБЛИЧНОЕ (60, 20, 0.5, 0.3, 0.15, 0.1, 0.05));

ОЧЕРЕДЬ: ОТДЕЛЬНАЯ;

〈ДАЛЕЕ СЛЕДУЮТ ОСТАЛЬНЫЕ ПРЕДЛОЖЕНИЯ, СОСТАВЛЯЮЩИЕ ОПИСАНИЕ ДАННОГО УЗЛА〉.

Этот текст означает следующее. Время обслуживания любого из транзактов ИП=1—ИП=3 постоянное и равно нулю. На выходе устройства (УПР. 1) каждый транзакт, поступающий с источника ИП=3, разделяется на 25 копий, каждая из которых поступает на устройство (СЕТЬ.1), входящее в состав модели сети. Далее аналогично описываются все устройства, входящие в состав модели сети. При этом в местах выделения из транзактов (на входе или выходе устройств) управляющих сигналов «Включить» соответствующий источник используются соответствующие конструкции языка МК АСИМ, в которых указываются имена управляемых источников транзактов. Последним в описании данного фрагмента сети стоит описание устройства (ЭВМ.1). При идентификации узла с помощью конструкции РЕСУРС указывается начальное значение объема общего ресурса, равное 400 единицам. Транзакты, рожденные источником ИП=1, поступают из устройства (КАНАЛ.1). Каждый из этих транзактов требует постоянного значения ресурса, равного 80. Транзакты, рожденные источником ИП=2, также поступают на устройства (КАНАЛ.1). Все они поступают в очередь к сборке для ИП=2; для этой очереди должна быть зарезервирована «длина», равная 20 транзактам. Каждый из этих транзактов требует ресурса, размер которого подчиняется равномерному распределению вероятностей на интервале от 20 до 320 единиц. Транзакты, рожденные источником ИП=3, поступают из устройства (КАНАЛ.1) в очередь к сборке для

ИП=3; для этой очереди должна быть зарезервирована «длина», равная 25 транзактам. Каждый из этих транзактов требует ресурса, размер которого подчиняется табличному закону распределения вероятностей. Начальное значение табличной функции распределения ресурса равно 60 единицам, интервал изменения величины ресурса равен 20. Гистограмма распределения вероятностей имеет пять интервалов группирования, вероятности выбора которых соответственно равны 0,5; 0,3; 0,15; 0,1 и 0,05.

Для каждого типа транзактов используется своя очередь. По окончании времени моделирования управление передается процедурам окончательной обработки и выдачи на печать оценок исследуемой сети. Печать производится в стандартной форме для всех структурных элементов сети. При этом для каждого узла оценки формируются отдельно и включают в себя оценки для устройства и для очередей. К основным оценкам, характеризующим обслуживаемое устройство, относятся: количество обслуженных транзактов; количество транзактов, получивших отказ в обслуживании; времена работы и простоя; коэффициент загрузки и простоя; коэффициент занятости и время простоя из-за отказов. Основные статистические оценки, формируемые относительно очередей, включают: количество транзактов, побывавших в очереди; количество транзактов, находящихся в ней; количество транзактов, получивших отказ; максимальную и среднюю длины очереди; среднее время пребывания транзакта в очереди. К выходной информации, характеризующей систему в целом, относятся среднее время пребывания транзакта в системе и количество транзактов данного типа, прошедших через систему. Блок статистики предназначен для сбора, обработки и выдачи на печать результатов независимых прогонов ИМ. Выдаются характеристики распределения случайной величины: оценки моментов, эксцесса, асимметрии, функция распределения вероятностей, а также доверительная область для оценок с заданным коэффициентом доверия.

5.4. ПЕРСПЕКТИВА РАЗВИТИЯ СРЕДСТВ МОДЕЛИРОВАНИЯ

5.4.1. РАСШИРЕНИЕ ОПИСАТЕЛЬНЫХ ВОЗМОЖНОСТЕЙ СРЕДСТВ АВТОМАТИЗАЦИИ МОДЕЛИРОВАНИЯ

Декларативный способ описания имитационных моделей, используемый в моделирующих комплексах ИМСС и АСИМ, существенно облегчает написание имитационных моделей в терминах СМО. Однако, как показала практика построения имитационных моделей ВС, достоинства этих комплексов переходят в их недостатки в тех случаях, когда объект моделирования имеет сложную алгоритмическую природу. Например, с помощью МК АСИМ очень сложно описывать поведение систем, логику функционирования которых трудно представить в графическом виде из-за параметрической многофункциональности поведения элементов моделируемого объекта. В таких случаях исследователь

попадает в сложные условия. С одной стороны, при алгоритмическом подходе к описанию системы на создание модели приходится тратить много усилий при программировании и отладке модели (низок уровень технологии создания модели). С другой стороны, при использовании декларативного подхода к описанию модели усложняется из-за необходимости представления алгоритма системы с помощью дополнительных устройств и логических цепей из этих устройств. Это увеличивает размеры модели и усложняет отладку и организацию модельных экспериментов.

На практике чаще всего имеет место следующая ситуация: около 90% системы удобно описывать декларативным способом, а для описания остальных компонент требуется алгоритмический способ. У существующих систем моделирования совмещение обоих способов описания модели не допускается. Выход из положения найден в разработке средств, обеспечивающих пользователя набором так называемых «алгоритмических примитивов». С их помощью пользователь может в рамках описания элемента сложной системы конструировать алгоритм его поведения. В отличие от общепринятого подхода, в этом случае пользователю не нужно заботиться о программистских аспектах (соответствие типов, адресация, размерности массивов и т. д.), достаточно лишь «набрать» алгоритм из последовательности алгоритмических примитивов, указав их параметры. Отладка таких алгоритмов из последовательностей примитивов сводится к обеспечению синтаксиса описания примитива. Все внутренние аспекты программной стыковки этих примитивов друг с другом берет на себя система моделирования.

Состав алгоритмических примитивов может быть различен и может постоянно расширяться. Например, в состав примитивов могут входить: наиболее типичные арифметические операторы, операторы передачи управления на другой участок текста модели, операторы обработки общих переменных модели. Введением алгоритмических примитивов может быть ликвидирован недостаток декларативного подхода к описанию функциональных возможностей узлов сети СМО. В результате основное преимущество алгоритмического способа описания имитационных моделей сложных систем перед декларативным будет сведено к минимуму при сохранении всех достоинств декларативного способа.

Следствием использования алгоритмических примитивов в составе декларативного описания являются: введение общих переменных модели и наличие функциональных устройств (приборов), которые используют и обрабатывают эти переменные модели. Некоторые из общих переменных можно использовать в качестве «флагов», регулирующих движение потоков транзактов между функциональными устройствами. Причем набором специальных операторов-деклараций организуются выставление и сброс этих «флагов» различными приборами. Возможно применение «флагов», на которые реагируют либо транзакты, либо устройства, либо и те, и другие одновременно.

Большим недостатком существующих систем моделирования декларативного типа является практическое отсутствие «подкраски» транзактов. Каждый транзакт определенного типа представляет собой идентичную копию, генерируемую источником на входе в сеть СМО и в лучшем случае содержащую в себе приоритеты. Сущность «подкраски» транзактов состоит в допущении у транзактов набора описываемых и формируемых атрибутов. С помощью алгоритмических примитивов можно легко организовать обработку атрибутов «подкрашенных» транзактов. Наличие «подкраски» транзактов расширяет возможности системы моделирования по организации управления имитацией сложных систем с динамической структурой [52]. Исследователь таким способом получает в свое распоряжение инструмент для моделирования массивов и, как следствие, возможность моделировать декларативным способом различные варианты организации базы данных. Кроме того, с помощью «подкраски» можно получить дополнительные возможности обмена информацией между функциональными устройствами, а это уже существенный шаг к процессному способу описания модели, но на более высоком технологическом уровне.

В существующих системах моделирования сетей СМО обслуживающие приборы (устройства) однофункциональны. Это означает, во-первых, что обслуживание транзакта имитируется временной задержкой, длительность которой определяется по указанному закону распределения, во-вторых, что любой функциональный узел в любой момент может обслужить только один транзакт. Однако практика использования МК АСИМ выявила необходимость организации параллельного обслуживания одним прибором нескольких транзактов, каждый из которых требует различной временной задержки, формируемой по различным законам распределения. Дальнейшим развитием декларативного способа описания имитационных моделей сетей СМО в этом направлении является предоставление в распоряжение исследователя средств конструирования алгоритма обслуживания транзактов узлом сети СМО. Речь идет об операторном наборе функций обслуживания прибора. Сюда входят:

- запуск других приборов с помощью сигналов «Включить», «Выключить» и «Прервать»;

- при выполнении логических условий или при соответствующих значениях «флага» преждевременная отсылка транзактов или копий транзактов по указанному адресу;

- формирование адреса по информации, находящейся в транзакте;

- пооператорная организация разделения маршрута следования транзактов из данного функционального узла либо по детерминированному признаку (например, по именам транзактов), либо по вектору вероятностей разделения маршрута движения.

При этом одно предложение, описывающее в МК АСИМ время обслуживания устройства, превращается в последовательность

предложений, которые транзакт пытается последовательно «пройти», пока не покинет данный функциональный узел сети. Вторым важным моментом является возможность группирования этих функциональных предложений по типам транзактов (с однотипным обслуживанием). Такой подход расширяет и упрощает конструктивные возможности описания логики поведения сложных функциональных узлов сети. В результате увеличивается уровень детализации деклараций, связанных с характером и способом обслуживания транзактов. Здесь сделан шаг в сторону «алгоритмизации» описания узла сети, однако на другом качественном уровне. Приведенные описания структуры функционирования узла сети не затеняются различными аспектами, связанными с программированием алгоритма узла и информационной стыковкой с другими узлами, и понятны другим специалистам, владеющим методами описания сетей СМО.

В большинстве систем моделирования используется понятие общего ресурса памяти в модели. Этот ресурс памяти выделяется либо транзактам, либо процессам. В МК АСИМ [52] любому функциональному узлу сети может быть приписан любой ресурс индивидуальной памяти, который можно выделять транзактам по двум дисциплинам. Для организации общего ресурса памяти приходится использовать служебные устройства и сложные схемы взаимодействия этих устройств. В МК PLSIM [51] можно использовать несколько типов общего ресурса памяти. Однако в ряде практических моделей возникает необходимость одновременного использования нескольких типов общей памяти модели наряду с возможностью приписывания каждому узлу сети своей индивидуальной памяти. Речь идет об объединении возможностей МК АСИМ и МК PLSIM в рамках одной системы автоматизации моделирования. Способов выделения этих ресурсов памяти на одном узле сети также должно быть несколько. По-видимому, легче всего пользователю описывать поведение таких функциональных устройств набором специальных предложений декларативного типа. При этом должны быть реализованы предложения, задающие способ и места выделения, освобождения и передачи общих и индивидуальных ресурсов памяти транзактам.

Практика построения моделей вычислительного процесса с помощью МК АСИМ показала, что в ряде случаев для организации стартстопной работы узлов сети требуется использование сигналов, включающих узел, на выходе которого они формируются. Гораздо проще при описании узла пометить, что он работает в режиме автоостанова, и не требовать от пользователя описания сигналов, закрывающих этот узел. У всех подобным образом помеченных узлов режим автоостанова реализуется автоматически системой моделирования. В результате существенно сокращается текст описания имитационной модели, особенно для случая, когда средства управления интенсивно используются.

5.4.2. ПРОГРАММИРУЕМЫЙ ВЫБОР МАРШРУТА ТРАНЗАКТА

Введение «подкраски» транзактов и оперативного набора функций узла модели сети позволяет организовать новый тип выбора маршрута транзакта в модели по окончании его обслуживания. В МК АСИМ и ИМСС [66] используются два способа выбора маршрута транзактов: вероятностный и детерминированный. Фактически траектория движения транзакта определяется либо путем розыгрыша по заданному вектору вероятностей для всех возможных направлений, либо четко каждому типу транзакта раз и навсегда определяется направление следования (адресат) по окончании обслуживания его узлом сети. Предлагается способ определения адресата следования на основе программы маршрута, заложенной в самом теле транзакта («подкраска»). Программа маршрута следования транзакта по окончании обслуживания прибором составляется из последовательности команд. Различаем следующие типы команд:

- выбор адресата следования транзакта по условию (в качестве условия может использоваться любая общая переменная модели);
- безусловный выбор адресата следования транзакта;
- вероятностный выбор адресата следования;
- условный и безусловный переходы внутри «программы маршрута»;
- организация цикла просмотра «команд» внутри тела транзакта.

В результате любой транзакт кроме приоритета и тела, содержащего «программу маршрута следования» по модели, содержит еще указатель текущей команды, которую надлежит анализировать при последующем обслуживании данного транзакта на функциональном узле, снабженном специальным предложением, задающим программируемый режим выбора маршрута транзакта. С помощью программы маршрута можно задать для каждого транзакта любую траекторию движения по модели. Например, легко организуется циклическое прохождение транзактов с любым чередованием функциональных узлов в траектории любой сложности. Подобный способ описания модели прост и позволяет оперативно заменять «тела» транзактов без изменения самого текста имитационной модели.

5.4.3. СПОСОБЫ УВЕЛИЧЕНИЯ ТЕХНОЛОГИЧЕСКИХ ВОЗМОЖНОСТЕЙ СИСТЕМ МОДЕЛИРОВАНИЯ

Вернемся к идее отделения параметров от текста модели, впервые выдвинутой в языке моделирования ДИС [64]. С усложнением описания модели на МК АСИМ очень трудно организовать оперативное изменение параметров модели. Как правило, для этой цели необходимо переписывать заново текст модели, что связано с новыми ошибками, необходимостью их исключения и т. д. Словом, это равносильно написанию новой имитационной модели.

Чтобы избежать лишней работы, предлагается использовать формальное задание параметров модели. Это означает, что в описании модели указываются не сами значения параметров, а адреса памяти, где находятся эти значения. В результате становится возможной оперативная замена текста параметров модели без изменения текста самой имитационной модели. Для реализации подобного принципа система моделирования должна опираться на мощную современную базу данных.

Вторым важным аспектом системы моделирования ДИС является возможность коммутации входов и выходов компонент модели друг с другом и отделения этой коммутации от текста модели. Хотя эту идею легче высказать, чем реализовать, тем не менее открывающиеся возможности конструирования и модификации модели при этом подходе настолько привлекательны, что отказаться от них невозможно. Реализация этих принципов существенно поднимает уровень технологии создания и модификации имитационных моделей. Предлагаются два типа коммутации: связей транзактов и связей сигналов управления работой устройств. В описании коммутации связей устанавливается соответствие выходов одних функциональных узлов входам других. С помощью коммутации связей указывается возможное число связей узлов независимо от типа выбора маршрута следования транзактов по модели (вероятностного, детерминированного, программируемого).

Существенную помощь в повышении уровня технологии моделирования может оказать библиотека процедур для испытания моделей. В состав такой библиотеки должны входить процедуры: оценки точности моделирования; параметрической верификации модели для выбранной предметной области исследования; оценки чувствительности откликов модели к изменению параметров моделирования для различных вариантов модели. На все случаи жизни предусмотреть набор процедур невозможно. Однако при длительном использовании метода моделирования в определенной области исследований такая библиотека накапливается. Весь вопрос в придании ей стандартного вида.

При алгоритмическом представлении моделей наиболее употребительным способом повышения скорости написания имитационных моделей является использование макропроцессоров. Этот способ действительно ускоряет этап составления описания модели и существенно уменьшает количество ошибок статического характера. Но при наличии ошибок динамического характера, проявляющихся только в ходе имитации, их локализация чрезвычайно затруднена, поскольку компилятор базового языка программирования обычно указывает место ошибки в уже развернутом макропроцессором тексте модели. Поэтому зачастую трудно осуществить перевод сообщений компилятора с базового языка программирования на язык макропроцессора, на котором пользователь записал свою программу. А для случая, когда компилятор имеет в своем составе алгоритмы оптимизации текста программы, поиск

ошибок динамического характера представляет собой практически неразрешимую задачу. По этой причине весьма заманчивый путь использования макропроцессоров не столько облегчает жизнь исследователям, сколько порождает новые проблемы и снижает доверие к построенной модели (такая модель хороша только при отсутствии динамических ошибок).

Более прогрессивным приемом повышения скорости создания имитационных моделей является использование специальных конструкторов моделей [73, 74]. Исследователь, долго работающий в данной предметной области, создает и накапливает библиотеку подмоделей («заготовок»), из которых затем конструирует различные варианты модели сложной системы. Весь вопрос состоит в соблюдении интерфейса между подмоделями и в последующей параметрической настройке компонент модели. Особенно удобен подобный прием в случае, когда в системе моделирования имеются средства отделения параметров от тела модели. Например, при проектировании вычислительных комплексов, ориентированных на обслуживание систем автоматизации проектирования, разработчики архитектуры использовали специальный конструктор моделей вычислительного процесса [73], построенный на базе моделирующего комплекса АСИМ [56].

Наметилась тенденция использования диалогового режима работы с ЭВМ в ходе имитационного эксперимента [7, 9, 25, 58]. Однако, по нашему мнению, пока удачны только попытки задания параметров модели и корректировки текста модели. Это означает, что с точки зрения динамики взаимодействия пользователя с моделью применение системных диалоговых средств никаких принципиальных преимуществ по сравнению с пакетным режимом отладки ИМ пользователю не дает.

Практика моделирования ВС показала необходимость расширения возможностей диалогового взаимодействия пользователя с имитационной моделью. Для задания темпа взаимодействия пользователя с моделью и управляющей программой моделирования вводятся так называемые точки диалогового взаимодействия, назначение которых состоит в следующем. В моменты, когда алгоритм любой компонентой доходит до данной точки, приостанавливается работа управляющей программы моделирования. Происходит копирование памяти модели, и управление передается диалоговому монитору. Диалоговый монитор проверяет ситуацию «был запрос к модели от пользователя» и формирует на экране дисплея соответствующий текст-сообщение пользователю о том, что имитация остановлена и пользователь может формировать диалоговому монитору директиву воздействия на процесс имитации. Диалоговый монитор анализирует директиву и выполняет действия, предусмотренные этой директивой. В частности, желательно, чтобы система могла реагировать на такие директивы пользователя, как «стоп-кадр имитации», «продолжение имитации с остановленного места», «запуск имитации» с указанной точки в алгоритме компоненты модели системы. Возможность последовательного копирова-

ния памяти модели и сохранения этих копий в течение заданного числа просмотров точек диалогового взаимодействия позволяет пользователю организовать возврат к предыдущему состоянию. При этом должны указываться количество возвратов к предыдущему состоянию (число точек диалогового взаимодействия) и характер управляющего воздействия на моделируемый процесс.

Очевидно, что пользователь должен иметь возможность изменять в динамике имитации общие переменные модели и осуществлять запуски-остановы некоторых элементов модели. Таким способом пользователь сможет действительно вмешиваться в процесс имитации и постоянно его контролировать. Для постоянного контроля ситуации в модели и изучения поведения компонент модели в ходе имитации исследователь должен иметь возможность установки временных наблюдателей за очередью, обслуживающим устройством, транзактами, управляющими сигналами.

Результаты наблюдения должны высвечиваться на экране дисплея. Тогда у пользователя появится возможность наблюдения за ходом реального процесса в модели подобно тому, как это делается в ходе телевизионных передач на спортивную тему (возврат к прошлому, стоп-кадр, постепенный переход к последующим состояниям объекта). Очевидно, что в моменты остановки имитации пользователь набором директив должен быть в состоянии выполнять с дисплея все те операции по управлению имитацией, о которых ранее упоминалось. Восстановление собственно процесса имитации осуществляется по соответствующей директиве пользователя. Далее имитация продолжается до момента активизации следующей точки диалогового взаимодействия. Снова останов имитации и контроль ситуации. Таким образом пользователь сможет контролировать весь процесс имитации, если он предусмотрел необходимое количество точек диалогового взаимодействия и правильно их расставил по алгоритму модели.

Глава 6

СОЗДАНИЕ ИМИТАЦИОННОЙ МОДЕЛИ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

6.1. ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС В ОС ЕС КАК ОБЪЕКТ МОДЕЛИРОВАНИЯ

Для иллюстрации предложенных ранее методик построения имитационных моделей в качестве объекта моделирования возьмем вычислительную систему. Обычно вычислительная система состоит из одной или нескольких ЭВМ (аппаратная часть) и набора программ, организующих автоматическую обработку информации в соответствии с заданным алгоритмом (программная часть). Назначение аппаратной части ВС — выполнение операций ввода,

хранения, преобразования и вывода информации. Программная часть ВС обеспечивает эффективное функционирование аппаратной части и определенный сервис при постановке и решении пакета задач или коллективном обслуживании пользователей. Согласно [66] ВС как конкретный объект исследования характеризуется:

- 1) множеством данных, вводимых в систему, и множеством результатов, формируемых на выходе системы;
- 2) выполняемой функцией, устанавливающей правило отображения множества данных в множество результатов;
- 3) структурой системы, определяющей состав и функциональное назначение входов и выходов системы: номенклатуру устройств и связи между ними; номенклатуру алгоритмов и порядок их взаимодействия между собой и с аппаратурой системы.

Если первые две группы сведений характеризуют назначение ВС, то последняя группа определяет способ внутренней организации ВС. В состав аппаратной части ВС входят: процессоры, каналы ввода-вывода, устройства управления и периферийное оборудование. Программное обеспечение включает в себя прикладные и управляющие программы. Прикладные программы реализуют алгоритмы, обеспечивающие выполнение некоторых системных функций. Между прикладными программами существуют информационные связи в тех случаях, когда результаты, формируемые одной программой, являются данными для другой программы. Управляющие программы (УП) обеспечивают необходимый порядок взаимодействия прикладных программ между собой и с аппаратной частью ВС. Таким образом, УП организуют вычислительный процесс в пространстве (между устройствами ЭВМ) и во времени. Порядок распределения ВП в пространстве и во времени определяется стратегией управления, принятой при реализации данной ВС. Стратегию управления задает алгоритм планирования работ в системе, который реализуют управляющие программы.

Зачастую при создании ВС исходя из сведений о назначении системы определяют ее структуру и состав алгоритмов, наилучшим образом отвечающие назначению системы. Чтобы решить такую задачу, необходимо располагать знаниями о том, как влияют различные способы структурной организации ВС и управления ВП на эффективность функционирования компонент ВС. Сложность ВС, разнообразие их аппаратных и программных средств требуют для решения этой задачи использования методов имитационного моделирования. В системах общего назначения комплекс управляющих программ называют операционной системой. Структура и функции ОС определяются не только характером задач, решаемых системой управления, но и логической структурой ВС.

Рассмотрим работу ОС, которая наиболее широко распространена в нашей стране и реализована ЭВМ Единой системы. Структурно ОС ЕС представляет собой многоуровневую систему управ-

ляющих программ, каждая из которых имеет вполне определенное назначение. Динамика взаимодействия модулей ОС ЕС зависит от режима ее работы (пакетная обработка заданий, обслуживание взаимодействий пользователей с системными программами в режиме диалога, удаленная пакетная обработка и т. д.) и имеющихся на ЕС ЭВМ ресурсов для решений задач [32]. Вычислительный процесс в ОС ЕС представляет собой достаточно удобный объект для исследования на имитационной модели. В зависимости от уровня детализации можно построить имитационные модели ВП в ОС ЕС различной степени сложности.

Вычислительный процесс на ЕС ЭВМ невозможно полностью точно отобразить одной моделью. Целесообразнее описывать динамику организации вычислений в ЕС ЭВМ семейством моделей, каждая из которых отображает поведение системы на различных уровнях детализации. Для каждого уровня можно выделить характерные особенности системы, переменные, законы и принципы, с помощью которых затем описывать поведение системы. Выбор уровня детализации зависит от целей исследования. В [68], например, предложены четыре основных уровня детализации ВС. Причем на некоторых уровнях выделены подуровни более детального описания. На первом уровне система описывается в терминах физических законов, которые используются при создании отдельных устройств ВС — таких, как процессоры, память, каналы ввода-вывода и т. д. На втором описывается структура технического обеспечения, т. е. функции, выполняемые перечисленными выше устройствами, и существующие между ними связи. На третьем уровне описывается структура программного обеспечения, которое управляет работой устройств. На четвертом уровне описывается поведение пользователей ВС, т. е. поток задач, которые должны решаться на ВС.

Процесс проектирования ВС в терминах системы иерархических моделей должен происходить сверху вниз. Исходными требованиями при проектировании являются требования пользователей. По характеру взаимодействия ВС с внешними объектами (пользователями, абонентами), для обслуживания которых она предназначена, их модели можно разделить на две группы:

ориентированные на определение технических характеристик и описание режимов функционирования ВС, их устройств и операционных систем. Такие модели находят применение обычно при проектировании ЭВМ, сетей ЭВМ, при разработке ОС и алгоритмов управления;

ориентированные на определение характеристик обслуживания пользователей ВС.

Обычно модели второй группы применяются при проектировании управляющих систем, где важнейшим показателем является способность ВС справиться с имеющейся работой в требуемые сроки, соблюдая налагаемые ограничения. Ко второй группе относятся также модели для исследования режимов коллективного

обслуживания абонентов сетью ЭВМ и вычислительными центрами коллективного пользования (ВЦКП).

При построении моделей ВС моделируемый объект представляется как совокупность обслуживающих устройств и входного потока, создающей нагрузку на ВС. В зависимости от целей исследования конкретных ВС (вычислительных сетей [62], многомашинных вычислительных комплексов [1], мультипроцессорных ЭВМ [18], однопроцессорных ЭВМ [21]) и наличия соответствующих исходных данных исследователи рассматривают разные режимы и различные уровни детализации внутреннего функционирования ЭВМ.

Как было сказано ранее, в качестве объекта моделирования выбран ВП в ОС ЕС. Цель моделирования состоит в исследовании на модели вопросов организации режима пакетной обработки в ОС ЕС и в определении наиболее существенных факторов, влияющих на пропускную способность ЕС ЭВМ при заданном входном потоке заданий. Другими словами, ограничимся рассмотрением только одного, наиболее распространенного, режима обработки заданий в ОС ЕС и возможностей его адаптации к условиям внешней среды. Под внешней средой в нашем случае будем понимать устойчивый по составу и структуре поток заданий пользователей, решаемых в режиме пакетной обработки. Поставленная цель исследования определяет уровень детализации ВП. Нас будут интересовать поведение ВС на третьем уровне детализации и частично ее взаимодействие с потоком задач, описываемым на четвертом уровне детализации.

В данной главе иллюстрируется изложенная ранее методика построения (гл. 2) имитационных моделей. При этом будем выделять каждый этап реализации имитационных моделей вычислительного процесса в операционной системе ЕС. Несмотря на достаточную сложность объекта моделирования, сформулированные цели моделирования и выбранный уровень детализации позволяют, по нашему мнению, проиллюстрировать основные аспекты методики построения имитационных моделей.

6.2. ЦЕЛЬ И ПАРАМЕТРЫ МОДЕЛИРОВАНИЯ

Предположим, что нам необходимо решить типовую задачу организации эффективной обработки информации на ЕС ЭВМ, работающей под управлением ОС ЕС версии 6.1 в режиме пакетной обработки заданий. Пусть существует достаточно устойчивый состав часто решаемых заданий (внешняя нагрузка G^* на ЕС ЭВМ) и требуется адаптировать работу ОС ЕС под этот поток заданий при имеющемся ограниченном объеме V основной памяти ЕС ЭВМ.

Выберем *параметры моделирования* ВП в ЕС ЭВМ. Согласно поставленной задаче внешняя нагрузка G характеризуется набором характеристик F , которые будем считать неизменными в ходе выбора вариантов организации ВП в ОС ЕС. Будем также считать,

что в распоряжение исследователя для организации мультипрограммной обработки заданий в пакетном режиме выделены постоянная емкость основной памяти, которая распределена по j -м разделам $\{V_{0j}\}$, и фиксированный состав внешних устройств с заданной емкостью внешней памяти V_b . Эти два параметра также будем считать неизменными в ходе моделирования. В распоряжении исследователя остаются четыре основных управляемых параметра:

λ_b — интенсивность поступления заданий из множества G для обработки на ЕС ЭВМ (заданий/с);

v_R — число параллельно работающих системных программ RDR , организующих ввод заданий в систему;

v_W — число параллельно работающих системных программ WTR , организующих вывод результатов обработки информации из системы;

v_I — максимально возможное число параллельно работающих проблемных задач пользователей, иницируемых системной программой $INIT$.

В качестве целевой функции моделирования вариантов организации ВП в ОС ЕС выберем следующие зависимости:

1. Зависимость среднего времени обработки задания в пакетном режиме T_3 от значений управляемых параметров при фиксированных значениях задаваемых параметров модели:

$$T_3 = \varphi_1(\lambda_b, v_R, v_W, v_I, \{V_{0j}\}, F). \quad (6.1)$$

2. Зависимость коэффициента загрузки процессора ЕС ЭВМ Q_{Π} от значений тех же управляемых параметров при фиксированных значениях задаваемых параметров модели:

$$Q_{\Pi} = \varphi_2(\lambda_b, v_R, v_W, v_I, \{V_{0j}\}, F). \quad (6.2)$$

Необходимо минимизировать показатель T_3 и максимизировать коэффициент Q_{Π} . В качестве задания на моделирование установим нахождение вида и коэффициентов регрессионных зависимостей φ_1 и φ_2 . Итак, цель моделирования поставлена, параметры моделирования и целевые функции определены. Уровень детализации ВП в ОС ЕС выбран.

6.3. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В ОС ЕС

На этапе составления концептуальной модели ВП в ОС ЕС необходимо установить ограничения и зависимости между параметрами, которые используются при создании имитационной модели. Рассматриваются следующие аспекты отражения реальных процессов: последовательность состояний решаемой задачи, функционирование супервизора, разделение ресурса ЦП, выделение ре-

сурса основной памяти (ОП), разделение ресурса внешней памяти.

В основу концептуальной модели ВП положим временные диаграммы изменения состояний задач и графический метод представления моделей выделения задачам ресурсов системы, изложенный в гл. 5. Для примера представим себе следующий случай. Пусть в системе решаются две задачи (1 и 2). Задача 2 находится во входной очереди системы, поэтому ресурс ОП, занимаемый ею, равен $R_{21}=0$. Задача 1 находится в одном из разделов ОП и занимает ресурс R_{11} . Ядро ОС ЕС постоянно занимает ресурс памяти R_0 . Поэтому в начальный момент времени система располагает свободной ОП, равной S_1 . Рассмотрение процесса использования ресурсов ОП, ЦП и внешней памяти начинается с момента выполнения задачи 1 на процессоре. По окончании обслуживания кванта на ЦП управление передается супервизору, поскольку задаче 1 необходимо организовать обмен с магнитным диском. При этом освобождается некоторый ресурс ОП. Супервизор организует этот обмен, загружает в ОП задачу 2 и передает ей управление.

По окончании кванта обслуживания на ЦП задача 2 обращается к супервизору для организации обмена с тем же самым МД, который использует задача 1. По окончании обмена с МД супервизор организует операцию обмена в интересах задачи 2 и возобновляет выполнение на ЦП задачи 1. Задаче 1 необходимо выполнить ряд системных операций, для чего она обращается к супервизору, и после их выполнения возобновляется выполнение на ЦП задачи 1. По окончании обмена происходит прерывание счета задачи 1 и управление передается супервизору. Последний освобождает часть ресурса ОП, занятой задачей 2, и, таким образом, появляется свободный ресурс памяти S_3 . Далее управление передается на выполнение задачи 2, а затем после очередной передачи управления супервизору возобновляется выполнение задачи 1.

Динамику использования ресурсов ВС задачами можно представить в виде временной диаграммы взаимодействия программных и технических средств. Для этой цели определим следующие состояния задачи i :

AK_i — активное, когда осуществляется ее реализация на ЦП согласно алгоритму задачи;

SU_i — ожидание завершения обработки супервизором SVC-прерывания или прерывания по концу операций обмена или ввода-вывода;

OB_i — ожидание завершения обмена с внешней памятью или операции ввода-вывода, необходимых для i -й задачи;

OA_i — ожидания активизации выполнения задачи на ЦП из-за того, что супервизор обслуживает другие задачи;

BX_i — ожидание задачи i во входной очереди заданий;

OB_i — ожидание окончания обмена с внешней памятью или операции ввода-вывода, необходимых для решения другой задачи;

OZ_i — ожидание выделения требуемого ресурса ОП или загрузки задачи i в ОП.

Процесс решения задачи можно представить в виде последовательности состояний, длительность пребывания задачи в которых определяется либо алгоритмом выполнения задачи, либо влиянием внешних воздействий (параллельно решаемых задач).

В активном состоянии AK_i задача i может находиться Ω_i раз с различной длительностью каждого отдельного кванта использования ЦП $\tau_{AK\ i k} (k=\overline{1, \Omega_i})$ — порядковый номер кванта использования ЦП в задаче i). В дальнейшем порядковые номера квантов времени на временной диаграмме, соответствующих одному и тому же состоянию, будем обозначать через k .

За время решения задачи i может иметь место ω_i прерываний ее выполнения на ЦП и нахождения задачи в состоянии SU_i с различной длительностью $\tau_{SU\ i k} (k=\overline{1, \omega_i})$. При обработке на ЦП SVC-прерываний задачи i или прерываний по концу обмена супервизор обслуживает вначале либо запрос задачи i , либо прерывание от устройства обмена, а затем переходит к обслуживанию запроса другой задачи. Длительность кванта обслуживания на ЦП супервизора $\tau_{os\ k}$ зависит от состава выполняемых работ супервизором. При этом могут иметь место выделение задачам ресурсов ОП и возврат этих ресурсов в общее поле свободной ОП. Кроме того, при наличии очереди обменов и свободных соответствующих ресурсов внешней памяти супервизор инициализирует эти обмены в интересах той или иной задачи.

В тех случаях, когда алгоритм решения задачи требует организации обмена информацией с внешней памятью и необходимое устройство обмена свободно, задача i переходит в состояние ожидания завершения обмена OB_i . За время решения задачи она может находиться в этом состоянии s_i раз с различной длительностью каждого отдельного кванта $\tau_{OB\ i k} (k=\overline{1, s_i})$. В некоторых случаях обмены задачи i удобно идентифицировать по каждому физическому устройству. Тогда под $\tau_{OB\ i k}$ понимают величину $\tau^{B\ i k}$.

В состоянии OA_i задача попадает в том случае, когда супервизор обслуживает запросы более приоритетных задач или прерывания от устройств обмена или ввода-вывода. В этом состоянии задача i может находиться n_i раз с различной длительностью $\tau_{OA\ i k} (k=\overline{1, n_i})$.

В состоянии BX_i задача i может находиться во входной и выходной очередях длительностями соответственно $\tau_{BX\ i1}$ и $\tau_{BX\ i2}$. В первом случае задача только формируется инициатором ОС ЕС по заказу задания, а во втором случае задача завершила этап вычислений и задание ожидает продолжения обработки на ЭВМ в одной из очередей освобождения системной программы вывода WTR.

В состоянии OB_i задача i может находиться h_i раз с различной длительностью ожидания завершения операций обмена или ввода-вывода другой задачей $\tau_{OB\ i\ k} (k=\overline{1, h_i})$.

Наконец, задача i в состоянии OZ_i может ожидать m_i раз выделения ресурсов ОП длительностью $\tau_{OZ\ i\ k} (k=\overline{1, m_i})$ в тех случаях, когда имеющийся ресурс свободной основной памяти меньше запрашиваемого задачей ресурса ОП.

На рис. 6.1 для нашего примера представлена ВД взаимодействия задач 1 и 2 с супервизором и общим устройством обмена (магнитным диском). Пока задача 1 находилась в состоянии AK_1 длительностью τ_{AK11} , задача 2 пребывала в состоянии BX_2 длительностью τ_{BX21} . Затем по SVC-прерыванию управление передается супервизору и задача 1 переходит в состояние CU_1 . Поскольку задача 2 для своей загрузки в ОП требует ресурса R_{22} , который больше свободного ресурса памяти системы S_1 , то задача 2 пребывает в состоянии OZ_2 до тех пор, пока супервизор по запросу задачи 1 не освободит часть ресурса ОП. В результате в распоряжении задачи 1 останется R_{12} ресурса ОП, а супервизор выделит задаче 2 требуемый ресурс R_{22} и свободного ресурса ОП у системы не останется. При этом супервизор организует операцию обмена задачи 1 с внешней памятью, а задачу 2 переводит в активное состояние AK_2 . Все эти операции супервизор осуществляет за время, равное τ_{OC1} . Таким образом, задача 1 переходит в состояние OB_1 на время выполнения обмена, равное τ_{OB11} . Задача 2 находится в состоянии AK_2 длительностью τ_{AK21} и по SVC-прерыванию передает управление супервизору, переходя таким образом в состояние CU_2 на время обслуживания супервизором запросов задачи 2, равное τ_{CU21} . В данном случае задаче 2 необходимо организовать обмен с тем же МД, который находится в состоянии обмена по запросу задачи 1. Поэтому супервизор устанавливает запрос на обмен задачи 2 в очередь обменов к данному МД, и, поскольку других работ нет, супервизор переходит в состояние ожидания завершения обмена, в котором он находится в течение кванта времени τ_{OK1} . Все это время ЦП простаивает, а задача 2 находится в состоянии OB_2 , ожидая освобождения МД.

По завершении обмена происходит возобновление работы на ЦП супервизора, который инициализирует команду обмена с МД в интересах задачи 2, а задачу 1 переводит в состояние AK_1 . Таким образом, задачи 1 и 2 находятся в одном и том же состоянии (CU_1 и CU_2) с одной и той же длительностью $\tau_{CU12} = \tau_{CU22} = \tau_{OC2}$ до тех пор, пока супервизор не передаст управление задаче 1. За время нахождения задачи 2 в состоянии OB_2 длительностью $\tau_{OB21} = \tau_{B21}$ задача 1 дважды переходит в состояние AK_1 соответственно с длительностями τ_{AK12} и τ_{AK13} .

В момент завершения обмена с МД происходит прерывание выполнения задачи 1 на ЦП, она переходит в состояние CU_1 , а управление передается супервизору. Супервизор завершает операцию обмена задачи 2, задачу 1 переводит в очередь задач, ожи-

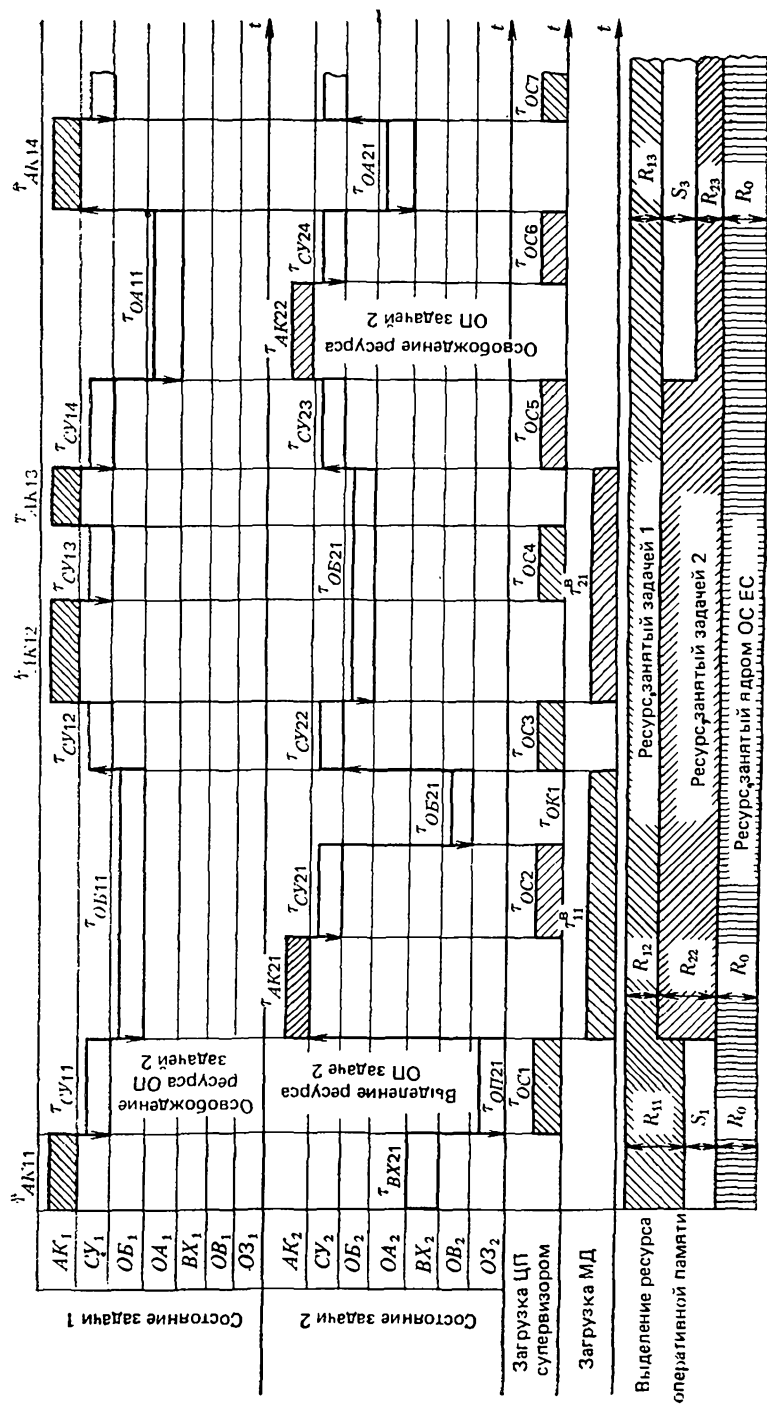


Рис. 6.1. Временная диаграмма взаимодействия задач с супервизором и устройствами обмена

дающих активизации, и активизирует задачу 2. Поэтому задача 2 по окончании обмена с МД сначала находится в состоянии $СУ_2$ длительностью $\tau_{СУ23}$, а затем переходит в состояние $АК_2$. Задача 1, наоборот, из состояния $АК_1$ вначале переводится супервизором в состояние $СУ_1$, а затем она переходит в состояние $ОА_1$, где пребывает в течение кванта длительностью $\tau_{ОА11}$. За это время задача 2 побывает в двух состояниях: $АК_2$ длительностью $\tau_{АК22}$ и $СУ_2$ длительностью $\tau_{СУ21}$. Наконец, супервизор продолжает выполнение прерванного кванта на ЦП, переводя соответственно задачу 1 в состояние $АК_1$, а задачу 2 в состояние $ОА_2$. После завершения обмена с МД задачи 2 супервизор по указанию задачи 2 возвращает часть ресурса ОП и в системе появляется ресурс свободной ОП, равный S_3 , а задачи 1 и 2 занимают ресурс ОП, равный соответственно R_{13} и R_{23} .

Таким образом, общее время решения задачи T_i представляется в виде суммы квантов времени нахождения задачи i в указанных состояниях:

$$T_i = \sum_{k=1}^{\Omega_i} \tau_{АК ik} + \sum_{k=1}^{\omega_i} \tau_{СУ ik} + \sum_{k=1}^{s_i} \tau_{ОБ ik} + \sum_{k=1}^{h_i} \tau_{ОБ ik} + \\ + \sum_{k=1}^{n_i} \tau_{ОА ik} + \sum_{k=1}^{m_i} \tau_{ОЗ ik} + \tau_{ВХ i1} + \tau_{ВХ i2}. \quad (6.3)$$

Все параметры решения задачи i можно разделить на два класса: характеристики монопольного режима решения задачи, определяемые классом задачи ($\tau_{АК ik}$, Ω_i , $\tau_{СУ ik}$, ω_i , $\tau_{ОБ ik}$, S_i);

характеристики, связанные с организацией мультипрограммного режима обработки задач ($\tau_{ОА ik}$, n_i , $\tau_{ОВ ik}$, h_i , $\tau_{ОЗ ik}$, m_i , $\tau_{ВХ i1}$, $\tau_{ВХ i2}$).

Поскольку имеют место равенства квантов времен $\tau_{СУ ik} = \tau_{ОС k}$ и $\tau_{ОБ ik} = \tau_{В ik}$ и обычно исследуется достаточно устойчивый класс заданий, то можно предположить, что при этом порождается типовой состав системных задач. Тогда для описания процесса использования ресурса ЦП в реальных условиях достаточно определить функции распределения характеристик монопольного режима решения типового состава задач $F(\tau_{АК i})$, $F(\Omega_i)$, $F(\tau_{ОС k})$, $F(\omega_i)$, $F(\tau_{В i})$, $F(s_i)$. Эти параметры можно использовать в качестве задаваемой внешней нагрузки G на вход модели системы. Остальные характеристики ВП, связанные с организацией мультипрограммного режима работы ВС, можно определять в ходе моделирования ВП и использовать их как для проверки адекватности имитационной модели реальному ВП, так и для вычисления откликов модели при постановке имитационных экспериментов с помощью ИМ вычислительного процесса.

В ряде случаев при имитации процесса выделения задачам определенного класса ресурса ЦП и внешних устройств вместо функций распределения $F(\Omega_i)$ и $F(s_i)$ достаточно использовать интегральные частотные характеристики:

$P^*_{0i} = 1/\overline{\Omega}_i$ — частота завершения i -й задачей использования квантов ЦП в активном состоянии;

$P^*_{1i} = \bar{s}_i/\Omega_i$ — частота обращения i -й задачи к внешним устройствам;

$\overline{\Omega}_i, \bar{s}_i$ — средние значения параметров Ω_i и s_i при решении i -й задачи.

При использовании достаточно большого числа задач, входящих в некоторый класс задач, можно считать, что частоты P^*_{0i} и P^*_{1i} приближаются к соответствующим вероятностям P_{0i} и P_{1i} и их можно использовать в качестве исходной информации при моделировании внешней нагрузки G на вход модели ВС. Отметим при этом, что на каждый квант решения задачи на ЦП имеет место обращение к супервизору. Правда, число обращений к супервизору ω_i больше числа квантов решения задачи Ω_i из-за появления прерываний ЦП от устройств ввода-вывода. Из-за прерываний счета на ЦП при окончании работы устройств обмена или ввода-вывода интервал $\tau_{ак i}$ нахождения задачи i в состоянии AK_i уменьшается по сравнению с требованиями алгоритма самой задачи. Поэтому при вычислении P^*_{0i} и P^*_{1i} для повышения точности имитации используют вместо параметра Ω_i значения параметра $\bar{\omega}_i$: $P^*_{0i} = 1/\bar{\omega}_i$; $P^*_{1i} = \bar{s}_i/\bar{\omega}_i$.

Поскольку в ЕС ЭВМ имеет место разделение ресурса внешней памяти между супервизором, системными и проблемными задачами, то при построении ИМ необходимо отразить процесс доступа запросов от функционирующих программных модулей к информации, записанной на внешней памяти, например на магнитных дисках. Кроме того, нужно отразить селекторный принцип обмена, который состоит в том, что если какая-нибудь задача получила право на обмен с данным МД, то до тех пор, пока его не закончит, другие программы не могут осуществлять обмены с данным МД. На рис. 6.1 такая ситуация отражена. Она имеет место, когда супервизор ожидает окончания обмена первой задачи с внешней памятью и находится в этом состоянии в течение $\tau_{окт}$.

В соответствии с выбранным уровнем детализации ВП в ОС ЕС не будем описывать функционирование программы загрузки. При моделировании режим загрузки ОС также не будет рассматриваться. Супервизор памяти также нас мало интересует, поскольку все механизмы выделения и освобождения памяти вырождаются соответственно до операторов «выделить ресурс» BR_i и «освободить ресурс» OR_i у каждой модели задач. Алгоритм функционирования супервизора ввода-вывода также не рассматривается, поскольку функции выбора каналов и их взаимодействия будут рассмотрены в дальнейшем при описании функционирования внешней памяти.

6.4. ФОРМАЛЬНОЕ ОПИСАНИЕ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В ОС ЕС

На этапе составления формального описания ВП в ОС ЕС в режиме пакетной обработки прежде всего необходимо обратить внимание на следующие два основных аспекта функционирования ЕС ЭВМ:

организацию внешней нагрузки на ВП (дать описание инвариантов решаемых задач);

взаимодействие супервизора ОС ЕС с задачами и организацию внешней памяти при обслуживании запросов системных и проблемных задач.

Весьма существенна также формализация технологии режима пакетной обработки. Завершается этап формализации ВП определением состава параметров и статистик имитационной модели и составлением целевых функций моделирования ВП в ОС ЕС.

6.4.1. ФОРМАЛЬНОЕ ОПИСАНИЕ ПРОЦЕССА РЕШЕНИЯ ЗАДАЧ В ВС

С помощью графического способа представления имитационных моделей (см. гл. 5) описание использования задач ресурсов ЦП, основной и внешней памяти можно представить в виде сети массового обслуживания. В основу формального описания процесса поступления и реализации i -й задачи на ЕС ЭВМ положен транзактный способ построения имитационных моделей. На рис. 6.2 представлена типовая подмодель расхода j -й задачей ресурсов ЕС ЭВМ с учетом существующей конкуренции за ресурсы ЦП, оперативной и внешней памяти ЭВМ. Подмодель представляет собой многофазную систему массового обслуживания транзактов, состоящую из двух приборов — функциональных узлов (ФУ)

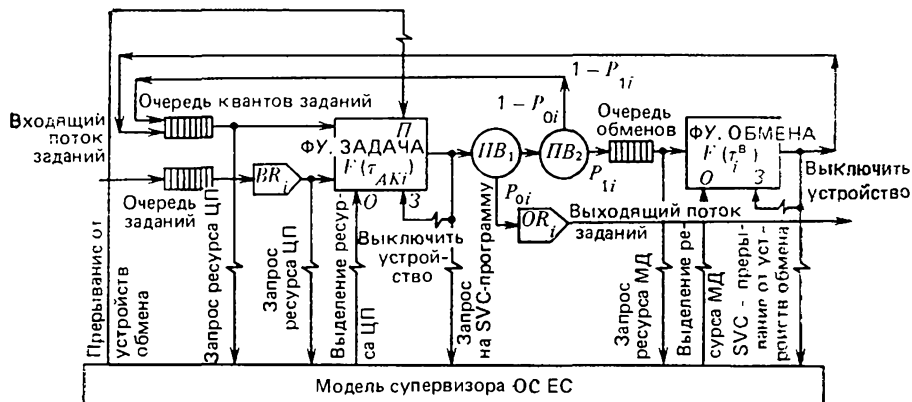


Рис. 6.2. Модель взаимодействия i -й задачи с супервизором

и трех очередей. Каждому типу заданий соответствует свой тип транзактов.

Прибор ФУ. ЗАДАЧА имитирует нахождение задачи i в состоянии AK_i . Моделирование конкуренции использования ресурса ЦП различными задачами достигается с помощью управляющих сигналов у прибора ФУ. ЗАДАЧА. При выделении ресурса ЦП именно данной задаче модель супервизора посылает на вход O управляющие сигналы «Выделение ресурса ЦП», переводящие прибор ФУ. ЗАДАЧА в состояние «Готово». Имитация прерывания выполнения задачи осуществляется сигналами, поступающими с модели супервизора на вход P , переводящими прибор ФУ. ЗАДАЧА в состояние «Занято». Для инициирования указанных управляющих сигналов после завершения очередного обслуживания транзакта на выходе прибора ФУ. ЗАДАЧА формируются два управляющих сигнала. Один из сигналов поступает на вход Z и переводит прибор ФУ. ЗАДАЧА в состояние «Занято», а другой сигнал представляет собой запрос на SVC-программу и поступает в модель супервизора. При каждом поступлении транзакта в любую из очередей к прибору ФУ. ЗАДАЧА формируется сигнал «Запрос ресурса ЦП», поступающий в модель супервизора ОС ЕС.

Прибор ФУ. ОБМЕНА имитирует нахождение задачи i в состоянии OB_i . Длительность нахождения задачи i в этом состоянии τ^B_{ik} определяется по функции распределения $F(\tau^B_i)$. Моделирование конкуренции использования ресурса внешней памяти различными задачами достигается формированием следующих трех управляющих сигналов:

«Выделение ресурса МД», поступающего из модели супервизора ОС ЕС на вход O прибора ФУ. ОБМЕНА и переводящего прибор в состояние «Готово»;

«SVC-прерывание от устройств обмена», поступающего на модель супервизора для организации прерывания выполнения другой задачи по окончании операций обмена и ввода-вывода;

«Выключить устройство», поступающего на вход Z прибора ФУ. ОБМЕНА и переводящего его в состояние «Занято» по окончании обслуживания транзакта прибором ФУ. ОБМЕНА.

При каждом поступлении заданий (транзактов) в очередь обменов формируется сигнал «Запрос ресурса МД». По этому сигналу модель супервизора в соответствии с установленной дисциплиной обслуживания формирует управляющий сигнал «Выделение ресурса МД».

Нахождение задачи i в состояниях BX_i и OZ_i имитируется ожиданием в очереди заданий. Пребывание задачи i в состояниях CU_i и OA_i моделируется ожиданием в очереди квантов заданий. Нахождение задачи i в состоянии OB_i имитируется ожиданием в очереди обменов.

Данная подмодель использования ресурсов ЕС ЭВМ задачами отражает тот факт, что в начале каждого шага обслуживания заданий ОС ЕС создает соответствующую задачу. Вся последовательность обслуживания указанными приборами некоторого тран-

закта в подмодели соответствует временной диаграмме использования ресурсов ЕС ЭВМ i -й задачей, созданной ОС ЕС для данного шага задания. Таким образом, реальному заданию соответствует транзакт в подмодели. Последовательность его обслуживания отражает процесс использования ресурсов соответствующей задачей. Задание становится в данной подмодели эквивалентом задачи.

Итак, входящий поток заданий (транзактов) поступает во входную очередь заданий прибора ФУ. ЗАДАЧА. В моменты завершения имитации решения предыдущей задачи модель супервизора посылает сигнал «Выделение ресурса ЦП», прибор ФУ. ЗАДАЧА осуществляет выбор i -го задания из очереди заданий. В этом случае оператором BR осуществляется выделение i -му заданию требуемого ресурса основной памяти R_i . Если ресурса основной памяти ВС не хватает, то задание находится в очереди ОП до тех пор, пока другая задача не возвратит системе свой ресурс памяти и суммарного свободного ресурса ОП окажется достаточно для инициализации i -го задания.

После получения ресурса основной памяти R_i начинается цикл обслуживания i -й задачи. Нахождение задачи в состоянии AK_i длительностью $\tau_{AK\ i}$ имитируется прибором ФУ. ЗАДАЧА. Длительность кванта обслуживания задания этим прибором $\tau_{AK\ i}$ разыгрывается по функции распределения $F(\tau_{AK\ i})$. По окончании обслуживания задания i на выходе прибора ФУ. ЗАДАЧА вырабатываются два управляющих сигнала. Один сигнал переводит этот прибор в состояние «Занято», а другой сигнал «Запрос на SVC-программу супервизора» поступает в модель супервизора. Таким способом имитируется строго последовательное выделение ресурса ЦП супервизору после окончания кванта обслуживания задачи, имеющее место при появлении запросов на SVC-программы супервизора по инициативе i -й задачи. В соответствии с установленной дисциплиной обслуживания запроса на SVC-программу длительностью $\tau_{OS\ k}$ модель супервизора выработает сигнал «Выделение ресурса ЦП». По этому сигналу снова прибор ФУ. ЗАДАЧА начинает цикл обслуживания задания (транзакта) i .

С выхода прибора ФУ. ЗАДАЧА задание (транзакт) поступает на оператор $PВ_1$, в котором осуществляется выбор дальнейшей траектории движения задания по модели. С вероятностью P_0 , завершается цикл обслуживания i -го задания данной подмоделью. Это означает завершение решения i -й задачи. В этот момент оператором OR осуществляется возврат ресурса оперативной памяти R_i , выделенного i -й задаче. Задания поступают в выходной поток заданий. С вероятностью $1 - P_0$ задание переходит на оператор $PВ_2$. Здесь осуществляется новый выбор дальнейшей траектории движения заданий по модели: либо с вероятностью P_{1i} задание поступает в очередь обменов, либо с вероятностью $1 - P_{1i}$ задание направляется в очередь квантов заданий. Очередь квантов заданий имеет высший приоритет, и поэтому, если она не пуста, когда

приходит от модели супервизора сигнал «Выделение ресурса ЦП» i -й задаче, выбор задания (транзакта) осуществляется сначала из нее. Таким способом описывается динамика выделения ресурса ЦП, когда не требуется организация обмена со стороны супервизора.

6.4.2. ФОРМАЛЬНОЕ ОПИСАНИЕ СУПЕРВИЗОРА

На рис. 6.3 представлена модель взаимодействия трех супервизоров ОС ЕС ЭВМ (прерываний, задач, ввода-вывода) и диспетчера в виде многофазной системы массового обслуживания. Работа диспетчера и супервизора прерываний моделируется соответственно приборами ФУ. ДИСП. и ФУ. ЗАПР., а также множеством управляющих источников ИУ₀, ИУ₁₁—ИУ_{1 k} (k — число подмоделей задач, обслуживаемых моделью супервизора). Супервизор задач представлен тремя взаимодействующими приборами ФУ. ЯДРО, ФУ. ОБМОС, ФУ. ВЫД и множеством управляемых источников ИУ₂₁—ИУ_{2 k} . Функционирование супервизора ввода-вывода имитируется множеством управляемых источников ИУ₃₁—ИУ_{3 k} и приборов ФУ. СВВ₁—ФУ. СВВ _{k} .

На выбранном уровне детализации супервизора предполагаем, что диспетчер, а также супервизоры прерываний и ввода-вывода практически не расходуют на диспетчеризацию ресурс ЦП. Поэтому приборы ФУ. ДИСП, ФУ. ЗАПР, ФУ. СВВ₁—ФУ. СВВ _{k} и ФУ. ВЫД обслуживают управляющие транзакты с нулевой задержкой ($\tau=0$). Весь ресурс времени ЦП супервизором расходуется только на обслуживание задач (транзактов) прибором ФУ. ЯДРО. Длительность обслуживания транзактов прибором ФУ. ЯДРО для каждой задачи i может быть различной $\tau_{ос\ k}$ и формируется по функциям распределения $F(\tau_{ос\ k})$.

Работа супервизора прерываний имитируется следующим образом. Сигнал «SVC-прерывание от устройства обмена» открывает соответствующий управляемый источник из множества ИУ₁₁—ИУ_{1 k} , что приводит к генерированию очередного управляющего транзакта в одну из очередей. Если приходит одновременно несколько «SVC-прерываний от устройств обмена», то они попадают каждый в свою очередь супервизора прерываний. В соответствии с установленной дисциплиной обслуживания очередей прибор ФУ. ЗАПР выбирает l -й управляющий транзакт ($l=\overline{1, k}$) и обслуживает его с нулевой задержкой. По окончании обслуживания транзакта на выходе прибора ФУ. ЗАПР формируются три управляющих сигнала: «Включить прибор ФУ. СВВ _{l} », «Включить прибор ФУ. ДИСП» и «SVC-прерывание от устройств обмена».

Первым сигналом инициируется работа супервизора ввода-вывода. Второй сигнал инициирует работу диспетчера. Третий сигнал прерывает выполнение всех подмоделей задач, поступая на входы Π приборов ФУ. ЗАДАЧА, и затем переводит эти приборы в состояния «Занято» (см. рис. 6.2).

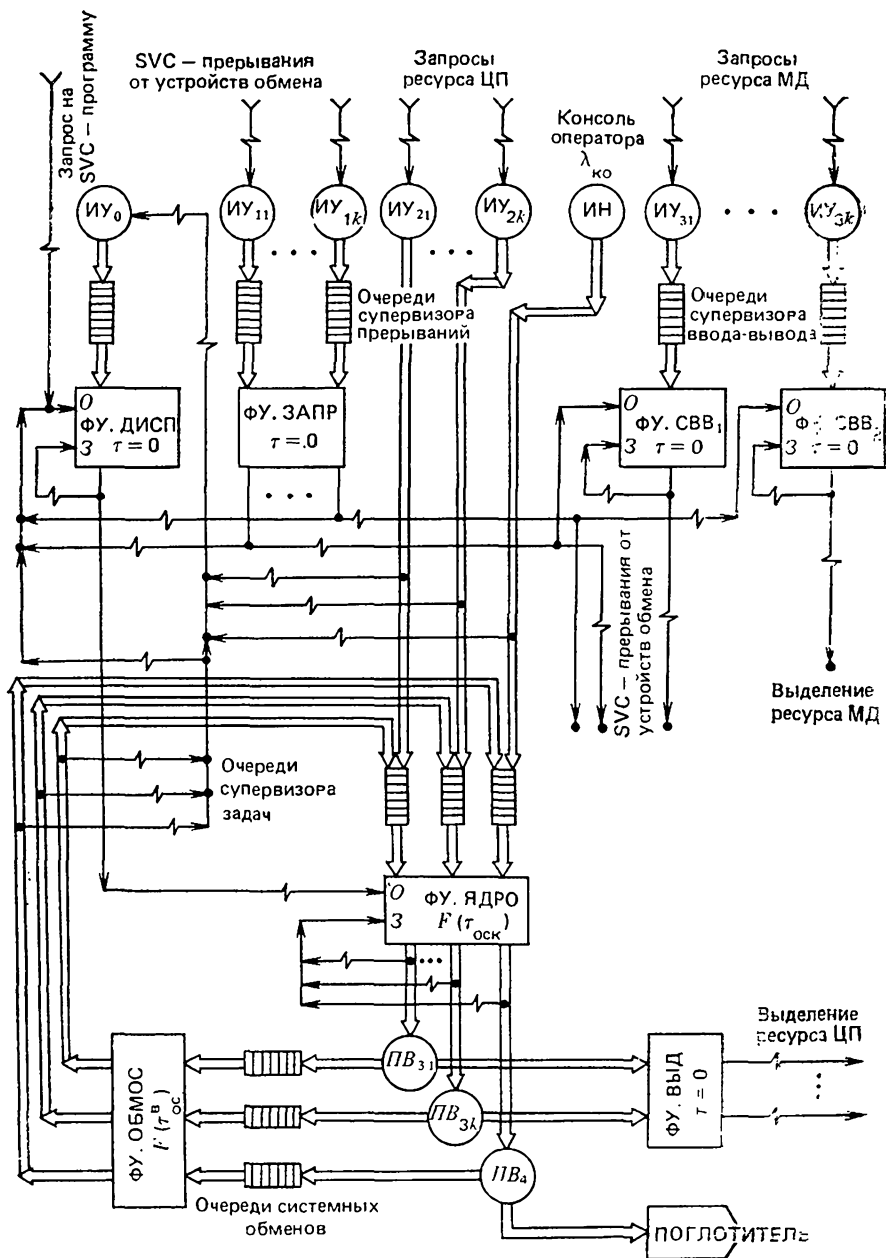


Рис. 6.3. Модель супервизора ЕС ЭВМ

Работа супервизора ввода-вывода моделируется следующим образом. Сигнал «Запрос ресурса МД» открывает соответствующий управляемый источник из множества $IY_{31}—IY_{3k}$, что приводит к генерированию очередного управляющего транзакта и поступлению его в одну из очередей супервизора ввода-вывода. Каждый запрос на ресурс МД попадает в одну из очередей супервизора ввода-вывода. Любому l -му физическому устройству обмена или ввода-вывода в супервизоре ввода-вывода соответствует своя l -я очередь запросов на ресурс МД, которая обслуживается с нулевой задержкой ($\tau=0$) соответствующим прибором ФУ. СВВ $_l$ ($l=\overline{1,k}$). Поступление от супервизора прерываний сигнала «Включить прибор ФУ. СВВ $_l$ » инициирует его работу. По окончании обслуживания на выходе прибора ФУ. СВВ $_l$ формируются два управляющих сигнала. Первый сигнал поступает на вход $З$ прибора ФУ. СВВ $_l$, переводя его в состояние «Занято» и запрещая таким образом обслуживание l -го устройства обмена до поступления с супервизора прерываний нового сигнала «Включить ФУ. СВВ $_l$ ». Второй управляющий сигнал «Выделение ресурса МД» разрешает выделение l -го ресурса внешней памяти и поступает на вход $О$ прибора ФУ. ОБМЕНА, который находится в модели задачи, пославшей запрос ресурса МД в l -ю очередь, и таким образом переводит его в состояние «Готово» (см. рис. 6.2).

Работа супервизора задач моделируется следующим образом. Сигнал «Запрос ресурса ЦП», поступающий от i -й ($i=\overline{1,k}$) модели задачи, открывает i -й управляемый источник из множества $IY_{21}—IY_{2k}$, что приводит к генерированию очередного управляющего транзакта, который попадает в i -ю очередь супервизора задач. При поступлении в любую из очередей супервизора задач управляющих транзактов генерируются управляющие сигналы, открывающие управляемый источник IY_0 . Таким образом, каждому управляющему транзакту, находящемуся в одной из очередей супервизора задач, в очереди диспетчера задач соответствует его копия, поступившая с источника IY_0 . Входная очередь прибора ФУ. ДИСП играет роль счетчика запросов ресурса ЦП, поступающих от моделей задач.

Все управляющие сигналы запроса моделей задач на SVC-программу поступают на вход $О$ прибора ФУ. ДИСП, разрешая таким образом выбор на обслуживание прибором ФУ. ДИСП управляющих транзактов из очереди. Транзакты обслуживаются прибором ФУ. ДИСП с нулевой задержкой ($\tau=0$), после чего на выходе прибора ФУ. ДИСП формируются два управляющих сигнала. Первый управляющий сигнал поступает на вход $З$ прибора ФУ. ДИСП, запрещая выбор из очереди следующего транзакта либо до прихода сигнала от супервизора SVC-прерываний, либо до поступления из модели любой задачи управляющего сигнала «Запрос на SVC-программу». Второй управляющий сигнал поступает на вход $О$ прибора ФУ. ЯДРО, разрешая последнему начать обслуживать управляющие транзакты, находящиеся в оче-

реди супервизора задач. В соответствии с установленным алгоритмом обслуживания супервизором задач запросов ресурса ЦП прибор ФУ. ЯДРО выбирает из очереди на обслуживание управляющий транзакт. Длительность обслуживания транзактов $t_{оск}$ разыгрывается по функции распределения $F(t_{оск})$.

По окончании обслуживания очередного управляющего транзакта выполняются следующие действия. Во-первых, генерируется управляющий сигнал, переводящий прибор ФУ. ЯДРО в состояние «Занято». Во-вторых, выбирается по вероятности дальнейший маршрут $P^{oc}_{i_i}$ следования управляющего транзакта ($PВ_{31}—PВ_{3k}$). С вероятностью $P^{oc}_{i_i}$ управляющий транзакт поступает на прибор ФУ. ВЫД, на выходе которого формируется сигнал «Выделение ресурса ЦП» i -й модели задачи. С вероятностью $(1—P^{oc}_{i_i})$ управляющие транзакты поступают в очереди системных обменов к прибору ФУ. ОБМОС для имитации механизмов организации обменов в самом супервизоре ОС ЕС. Обслуживание управляющего транзакта прибором ФУ. ОБМОС осуществляется в течение времени $\tau^{бос}$, разыгрываемого по функции распределения $F(\tau^{бос})$.

По завершении обслуживания транзакт поступает в соответствующую очередь супервизора задач, а на выходе прибора ФУ. ОБМОС из управляющих транзактов формируются два управляющих сигнала. Первый управляющий сигнал открывает управляющий источник ИУ₀, обеспечивая таким способом поступление в очередь диспетчера еще одного управляющего сигнала (копии транзакта, поступившего в очередь к ФУ. ЯДРО). Второй управляющий сигнал переводит прибор ФУ. ДИСП в состояние «Готово», разрешая ему сформировать сигнал «Включить прибор ФУ. ЯДРО», что означает очередную инициализацию супервизора задач.

Взаимодействие супервизора с консолью оператора ЭВМ моделируется работой неуправляемого источника транзактов ИН, который с интенсивностью $\lambda_{ко}$ генерирует управляющие транзакты, поступающие в самую неприоритетную очередь супервизора задач. Порядок обслуживания управляющих транзактов, поступивших с источников ИН и ИУ_{2i}, практически одинаков. Отличие заключается в том, что после обслуживания управляющего транзакта прибором ФУ. ЯДРО, имитирующего поведение консоли оператора, вероятностный выбор дальнейшей траектории транзакта осуществляется в операторе $PВ_4$. Этот оператор направляет управляющий транзакт либо в очередь системных обменов, либо на поглотитель транзактов (ПОГЛ).

Выбором дисциплины обслуживания очередей супервизора задач определяется порядок синхронизации обслуживания приоритетных и неприоритетных запросов к супервизору на выделение ресурса ЦП задачами использования ресурса ЦП самим супервизором. Изменением вероятностей $P^{oc}_{i_i}$ достигается регулирование накладных расходов на реализацию мультипрограммирования и времени реакции супервизора на запросы системных и пользовательских задач.

6.4.3. ФОРМАЛЬНОЕ ОПИСАНИЕ ТЕХНОЛОГИИ РЕЖИМА ПАКЕТНОЙ ОБРАБОТКИ

За основу формализации описания технологии режима пакетной обработки заданий в ОС ЕС выберем схему прохождения потока заданий в режиме мультипрограммирования [32].

Любое трехшаговое задание обслуживается последовательно следующими задачами: программой системного ввода, инициатором, транслятором, редактором, проблемной задачей, терминатором и программой системного вывода. На рис. 6.4 представлена графическая модель обслуживания потока заданий в режиме пакетной обработки ОС ЕС. Каждое задание (транзакт) последовательно обслуживается следующими подмоделями задач:

M. RDR — имитирует системный ввод заданий во входную очередь системы;

M. INIT — моделируют работу задачи инициатора при загрузке задания в оперативную память и организации очередного шага задания;

M. TRAN — имитирует работу любого транслятора с языка программирования высокого уровня в объектный код ЕС ЭВМ;

M. REDA — моделирует процесс редактирования задачи, состоящей из нескольких объектных модулей для ЕС ЭВМ;

M. TASK — имитирует процесс решения задач пользователя на ЭВМ;

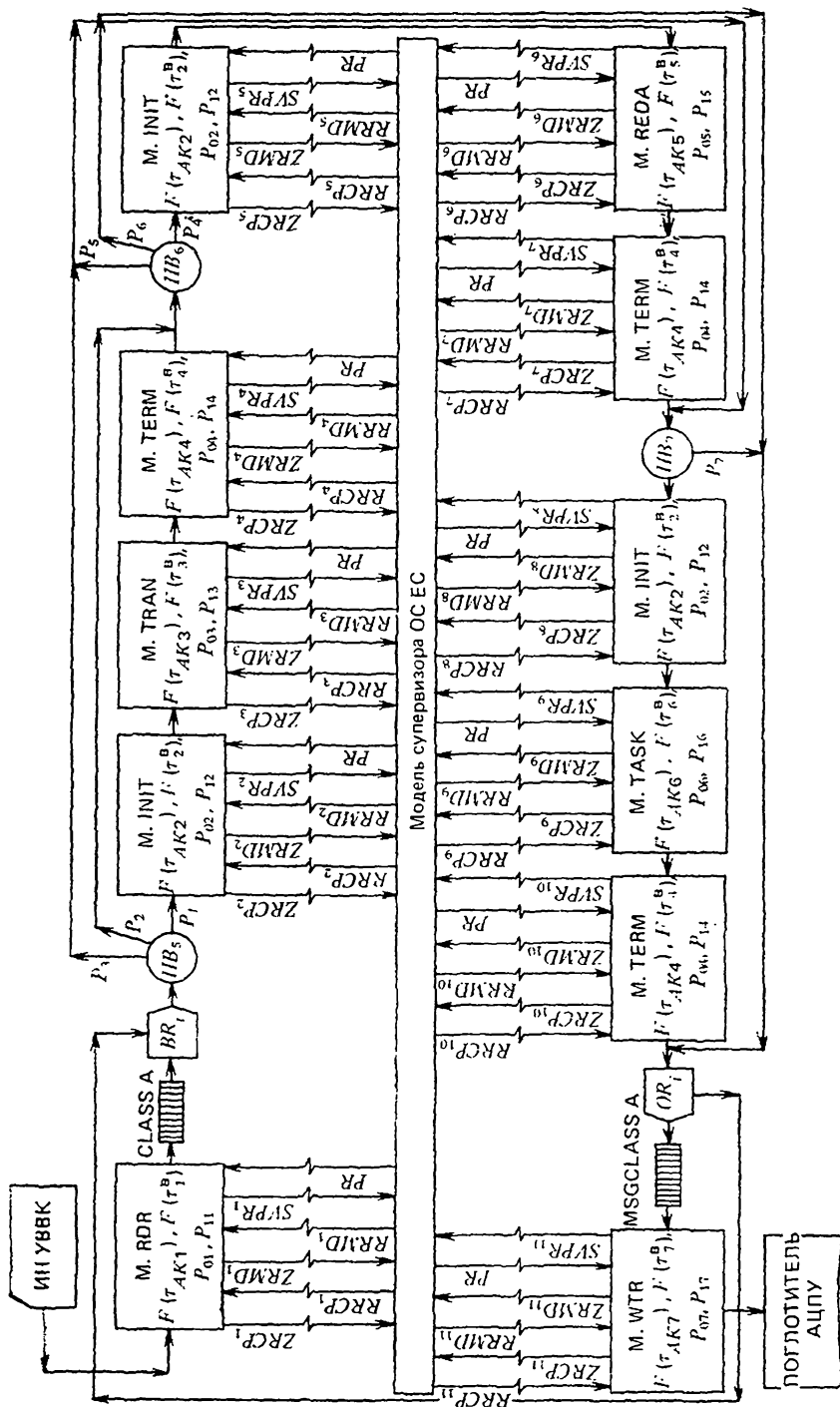
M. TERM — моделирует работу системной задачи инициатора-терминатора, относящейся к завершению очередного шага задания;

M. WTP — имитирует вывод результатов решения задачи пользователя программой системного вывода.

В ходе обслуживания в режиме пакетной обработки любое задание состоит из различного числа шагов. На каждом шаге может оказаться любая из задач: трансляции, редактирования, решения. Порядок следования задач произвольный. Для простоты изложения допустим, что задание обрабатывается согласно схеме, представленной на рис. 6.4.

Рождаются транзакты неуправляемым источником ИН УВВК, имитирующим ввод заданий с УВВК в систему с заданной интенсивностью λ_b . В устройстве ПОГЛОТИТ АЦПУ транзакты прекращают свое существование, имитируя таким образом конец решения задач. При этом фиксируется среднее время жизни транзакта T_z , означающее среднее время пропуска задания в пакетном режиме. Каждая задача использует собственные наборы данных на магнитных дисках, а поэтому ей соответствует собственное функциональное устройство обмена с МД.

Перед началом обслуживания транзакта подмодель i -й задачи запрашивает у супервизора памяти с помощью оператора BR_i требуемый ресурс памяти. По окончании обслуживания транзакта моделью i -й задачи этот ресурс освобождается соответственно оператором OR_i . Как было указано ранее, очередь квантов обслужива-



живания заданий у каждой подмодели i -й задачи более приоритета, чем входная очередь заданий. Поэтому выбор очередного транзакта на обслуживание из очереди заданий возможен только при условии окончания обслуживания предыдущего транзакта и пустой очереди квантов заданий (см. рис. 6.2). Любая подмодель i -й задачи последовательно может посылать модели супервизора два запроса на ресурсы. Первый запрос означает «Запрос ресурса ЦП» ($ZRCP_i$), а второй — «Запрос ресурса МД» ($ZRMD_i$). По первому запросу в соответствии с операционной обстановкой в системе модель супервизора вырабатывает разрешающий сигнал «Выделение ресурса ЦП» ($RRCP_i$), по второму — разрешающий сигнал «Выделение ресурса МД» ($RRMD_i$). При завершении какого-либо обмена с внешней памятью i -й подмоделью задачи она посылает в модель супервизора сигнал «SVC-прерывание от устройств обмена» ($SVPR_i$). По этому сигналу модель супервизора посылает всем подмоделям задач сигнал «Прервать счет» (PR).

Транзакты, имитирующие задания на обработку в системе, поступают с интенсивностью λ_b на обслуживание подмоделью системной программы ввода заданий ($M.RDR$). Алгоритм обслуживания транзактов соответствует функционированию типовой модели использования ресурсов ЕС ЭВМ i -й задач (см. рис. 6.2). По окончании обслуживания выходящий поток заданий (транзактов) поступает, например, во входную очередь системы CLASS A, которая соответствует очереди заданий на рис. 6.2. По оператору BR_i осуществляется выделение ресурса основной памяти заданию. Это выделение управляемое, и оно не может быть начато до тех пор, пока не поступит сигнал разрешения обслуживания следующего задания. Такой сигнал вырабатывается оператором освобождения ресурса основной памяти OR_i (см. рис. 6.4) по окончании обслуживания всех шагов задания перед поступлением задания в выходную очередь системы MSGCLASS A.

Способ выбора и возвращения ресурсов ОП операторами BR_i и OR_i определяется тем, какой режим обработки заданий ОС ЕС моделируется. Эти операторы используют одну общую переменную, означающую размер свободной ОП, когда моделируется режим MVT. При моделировании режима MFT эти операторы используют множество общих переменных $\{V_{0j}\}$ ($j=1, L$, где L — число разделов ОП, используемых для режима MFT; $L \leq 15$), отражающее разбиение ОП на фиксированные разделы.

Оператор BR_i моделирует также и загрузку задачи в ОП. Каждому заданию соответствует своя потребность в ресурсах ОП. Эта потребность должна быть задана с помощью функции распределения $F(\theta_i)$, и затем в ходе имитации формируется оператором BR_i (здесь θ_i означает размер ресурса ОП в килобайтах, требуемого для начальной загрузки i -й задач). В данной модели ВП принято следующее допущение: при решении i -й задачи в j -м разделе ОП выделяется вся требуемая емкость памяти θ_i в

начале обслуживания задания с помощью оператора BR_i . При имитации режима MVT в операторе BR_i из V_0 вычитается величина θ_i . Для моделирования выделения ресурса ОП оператором BR_i в случае режима MFT производится сравнение V_{0j} у тех разделов ОП, которые в данный момент свободны. Затем из всех свободных разделов ОП, для которых выполняется неравенство $V_{0j} \geq \theta_i$, выбирается раздел, имеющий минимальное значение V_{0j} , и он закрепляется за i -м заданием с помощью установки признака «Раздел j занят i -й задачей». По окончании имитации всех фаз обслуживания задания i в операторе OR_i осуществляется возврат всей затребованной емкости ОП θ_i . Этот возврат ОП моделируется оператором OR_i либо прибавлением величины ресурса ОП θ_i к V_0 (для режима MVT), либо снятием признака «Раздел j занят i -й задачей» (для режима MFT).

После оператора BR_i имеет место вероятностный выбор дальнейшего маршрута следования транзакта по модели, который осуществляется с помощью оператора ПБ₅. В этом операторе по вектору вероятностей (P_1, P_2, P_3) осуществляется розыгрыш направления движения транзакта. Так, с вероятностью P_1 транзакт попадает последовательно на обслуживание тремя подмоделями: M. INIT, M. TRAN, M. TERM, — имитируя таким образом фазу трансляции задания в реальной системе. С вероятностью P_2 транзакт поступает на последовательное обслуживание другой тройкой подмоделей: M. INIT, M. REDA, M. TERM, — имитируя фазу редактирования задания в реальной системе. С вероятностью P_3 транзакт переходит на обслуживание третьей тройкой подмоделей: M. INIT, M. TASK, M. TERM, — имитируя фазу непосредственного решения проблемной задачи на ЦП.

В начале имитации каждой из фаз обслуживания задания i транзакт, его имитирующий, обслуживается подмоделью системной программы-инициатора (M. INIT). Завершается имитация каждой фазы обслуживания задания i обслуживанием транзакта подмоделью системной программы-терминатора (M. TERM). Собственно имитация каждой фазы обслуживания i -го задания осуществляется соответствующей подмоделью задач: трансляции (M. TRAN), редактирования (M. REDA), выполнения проблемных задач (M. TASK). Таким образом, имитация каждой фазы обслуживания задания осуществляется соответствующей моделью, каждая из которых состоит из трех подмоделей (см. рис. 6.4).

Длительность обслуживания транзакта, имитирующего поведение i -го задания, подмоделями задач определяется двумя функциями распределения: $F(\tau_{AKi})$ и $F(\tau^B_i)$, а также двумя вероятностями: P_{0i} и P_{1i} (см. § 6.3, рис. 6.5 и 6.7).

По окончании имитации фазы трансляции оператором ПБ₆ осуществляется розыгрыш направления дальнейшего движения транзакта по вектору (P_4, P_5, P_6). Так, с вероятностью P_4 транзакт попадает на имитацию фазы редактирования. С вероятностью P_5 задание, минуя имитацию фазы редактирования, переходит на имитацию фазы решения проблемной задачи. А с вероят-

ностью P_6 имитируется завершение обслуживания задания. Аналогичным образом по окончании имитации редактирования оператором $ПВ_7$ осуществляется розыгрыш направления движения транзакта по вероятности P_7 . С вероятностью P_7 транзакт попадает на имитацию фазы решения проблемной задачи, а с вероятностью $1 - P_7$ имитация процесса обслуживания задания завершается.

Как сказано ранее, по окончании имитации обслуживания задания системе возвращается ресурс основной памяти и транзакт поступает в одну из выходных очередей, например MSGCLASS A. Из этой очереди задания (транзакты) выбираются на обслуживание подмоделью системной программы вывода (M. WTR), а по завершении процесса имитации обслуживания задания транзакт поступает в ПОГЛОТИТЕЛЬ АЦПУ.

6.4.4 ПАРАМЕТРИЗАЦИЯ МОДЕЛИ ВП В ОС ЕС

На рис. 6.7 для обслуживания потока заданий используется по одной подмодели системных программ ввода и вывода (M. RDR и M. WTR). На самом деле число таких подмоделей может быть любым (в разумных пределах). В зависимости от числа устройств системного ввода v_R в модели ВП устанавливается соответствующее число подмоделей M. RDR. Аналогичным образом число устройств системного вывода v_W определяет в модели ВП число подмоделей M. WTR, а число входных очередей v_I определяет число подмоделей обслуживающих задач (M. INIT; M. TRAN; M. REDA; M. TASK; M. TERM). Приступим к установлению состава управляемых параметров моделирования.

Параметры v_R , v_W , v_I определяют число и состав системных задач, обслуживающих поток заданий (транзактов) в модели вычислительного процесса. Модель ВП строится из расчета реально допустимых максимальных значений этих параметров. Поэтому, задавая различные значения v_R , v_W и v_I , приходим к усеченной модели ВП.

Из емкости ОП, имеющейся на исследуемой ЭВМ, вычитается емкость, необходимая для размещения ядра ОС ЕС и системных программ RDR и WTR. Остаток ОП разбивается по разделам аналогично тем значениям, которые были во время натурного эксперимента. В результате формируется множество значений $\{V_{0j}\}$, которое затем используется операторами BR_i и OR_i в процессе имитации технологии обработки заданий в системе.

Важным параметром моделирования в нашем исследовании будет интенсивность поступления заданий в систему, определяемая как интенсивность рождения транзактов λ_b неуправляемым источником транзактов ИН УВБК. В качестве задаваемых параметров ВП в нашем исследовании будут:

$F(\tau_{AKi})$ — функция распределения длительности квантов обслуживания ЦП i -й системной или проблемной задачи ($i = \overline{1, 7}$, см. рис. 6.4);

$F(\tau^B_i)$ — функция распределения длительности обменов с внешней памятью для решения j -й системной или проблемной задачи;

$(P_1 - P_7)$ — вектор вероятностей выбора количества шагов обслуживания заданий, определяющий структуру задания;

P_{0i} — вероятность окончания обслуживания заданий i -й подмоделью задачи (определяет число квантов обслуживания задачи i на ЦП);

P_{1i} — вероятность обращения к устройствам обмена с МД при обращении i -й задачи к супервизору;

$F(\theta_i)$ — функция распределения требуемых объемов ОП при решении задачи i .

Все перечисленные функции распределения и вероятности определяются для каждой задачи индивидуально. Они зависят от исследуемого класса задач. Эта работа достаточно сложна и требует создания и использования специального математического обеспечения. В состав задаваемых характеристик, определенных ранее как G , входят: $G\{F(\tau_{AK\ i}), F(\theta_i), F(\tau^B_i), P_{0i}, P_{1i}, P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$. Они устанавливаются на этапе калибровки модели и в ходе моделирования не меняются. Эти характеристики определяют фактически структуру запросов заданий на ресурсы ВС и должны быть инвариантными при решении задач моделирования.

В качестве основной статистики будем использовать среднее время жизни транзакта T_3 . Второй важной статистикой моделирования будет коэффициент загрузки процессора ЕС ЭВМ Q_n . Он равен сумме коэффициентов загрузки приборов, имитирующих работу задач и супервизора:

$$Q_n = \sum_{j=1}^{VR} \eta_{Rj} + \sum_{j=1}^{VW} \eta_{Wj} + \sum_{j=1}^{VI} (\eta_{Ij} + \eta_{TRj} + \eta_{REj} + \eta_{TSj} + \eta_{TEj}) + \eta_{\pi},$$

где η_{Rj} , η_{Wj} , η_{Ij} , η_{TRj} , η_{REj} , η_{TSj} , η_{TEj} , η_{π} — коэффициенты использования ЦП j -ми М. RDR, М. WTR, М. INIT, М. TRAN, М. REQA, М. TASK, М. TERM и супервизором соответственно.

Согласно постановке задачи на моделирование нам необходимо найти зависимость T_3 и Q_n от управляемых параметров при фиксированных значениях задаваемых характеристик моделирования G :

$$T_3 = \varphi_1(\lambda_B, \lambda_R, v_W, v_I, \{V_{0j}\});$$

$$Q_n = \varphi_2(\lambda_B, v_R, v_W, v_I, \{V_{0j}\}).$$

Вопрос сводится к нахождению вида и коэффициентов регрессионных зависимостей φ_1 и φ_2 в ходе имитационных экспериментов согласно составленному плану.

6.5. ПРЕОБРАЗОВАНИЕ ФОРМАЛЬНОГО ОПИСАНИЯ В ОПИСАНИЕ ИМИТАЦИОННОЙ МОДЕЛИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА

Выбор графического способа формализации ВС определяет необходимость использования в качестве языка реализации имитационной модели ВП моделирующего комплекса АСИМ [54]. В соответствии с методикой, изложенной в § 2.4, для превращения графического описания ВП в ОС ЕС в имитационную модель, удобную для описания средствами языка МК АСИМ, необходимо провести декомпозицию ВП на отдельные составляющие. Как видно из рис. 6.7, формальное описание ВП в ОС ЕС в виде сети массового обслуживания представляет собой обобщенную графическую схему приборов (узлов сети) и связей между ними.

На схеме производится нумерация внесистемных потоков путем присвоения им определенных индексов и устанавливаются связи между отдельными узлами (устройствами в терминологии МК АСИМ [54]). Затем уточняются аспекты, характеризующие узел: входящие потоки и их характеристики (тип распределения, параметры, класс, приоритет, период запуска); порядок формирования очередей (общая, отдельная, смешанная), их приоритеты и ограничения; дисциплина обслуживания (FIFO, LIFO и т. д.); характер распределения времени обслуживания и его параметры; способ подключения к обслуживанию и правила выбора свободного устройства; возможность появления отказа в обслуживающем устройстве, правила восстановления обслуживающего устройства, порядок резервирования.

Приступаем к независимому описанию каждого из узлов. Узлы сети отображаются предложениями УСТРОЙСТВО, а особенности функционирования каждого из них — набором предложений языка МК АСИМ. Декомпозиция предполагает условное разбиение формального описания на отдельные узлы с указанием для каждого из них всех связей по входу и выходу, сохраняющих исходную картину распределения транзактов.

Переход от детального графического описания к описанию имитационной модели на языке МК АСИМ достаточно прост. При этом сохраняется наглядность текста как всей модели, так и отдельных узлов сети. Это обеспечивает легкий обратный переход от описания модели на языке АСИМ к ее графическому представлению на этапе проверки правильности составления описания модели. Для составления детального графического описания, пригодного для кодирования на языке МК АСИМ, будем использовать ряд запирающих вентилей, дополнительных устройств, специальные средства планирования транзактов, а также средства управления моделью.

Запирающие вентили будем применять, как правило, при управлении процессом поступления транзактов в сеть извне. После обработки транзактов устройство переводится в состояние «Занято» до поступления с некоторого другого устройства сигнала

«Включить устройство». При этом поступление транзактов на занятое устройство в зависимости от типа узлов (с отказом или с ожиданием) приводит соответственно либо к потере транзактов, либо к установке их в очередь.

Введение дополнительных (фиктивных) устройств позволяет организовать такую обработку транзактов, когда транзакты с выхода устройства через фиктивный узел с нулевой задержкой поступают на его вход, а прямая организация циклической обработки недопустима правилами МК АСИМ. Кроме того, введение дополнительных устройств в ряде мест модели приводит к необходимости конкретизации свойств каждого потока в отдельности. Параметризация модели существенно облегчается при использовании языка МК АСИМ. Русский текст предложений, описывающих устройства, позволяет применять семантику обозначений узлов и параметров их обслуживания для отладки модели. Поэтому легко выделяются состав параметров моделирования и характеристики, задаваемые в качестве исходной информации для моделирования.

Важной особенностью создаваемой модели является отсутствие «подкраски» транзактов. В модели циркулирует только один тип транзактов с одним и тем же индексом, что также существенно упрощает описание модели на языке МК АСИМ. Кроме того, наличие средств сбора стандартной статистики в МК АСИМ, которая по существу совпадает с требуемым составом статистик данной модели, снимает проблему выбора мест установки операторов сбора статистики в формальном описании объекта моделирования. Поскольку МК АСИМ автоматически организует синхронизацию обслуживания транзактов, то снимаются также вопросы установки операторов синхронизации компонент модели. Вероятностные переходы транзактов по окончании обслуживания их устройствами организуются в МК АСИМ соответствующей конструкцией предложения **ВЫХОДЯЩИЙ ПОТОК**. Рождение управляющих сигналов из транзактов также описывается соответствующей конструкцией этого предложения. Однако для описания двух следующих друг за другом вероятностных переходов транзактов необходимо использовать дополнительные устройства.

На рис. 6.5 представлено описание модели i -й задачи, по которому можно вести кодировку текста модели на МК АСИМ. Отличие детального описания от представленного на рис. 6.2 состоит в следующем. Введен входной вентиль (ВХВ.1), который запрещает выбор следующего транзакта на обслуживание задачей i до момента поступления предыдущего транзакта в выходящий поток. Для организации вероятностных переходов транзакта $ПВ_1$ и $ПВ_2$ служат вспомогательная очередь и три дополнительных распределительных устройства (ВЫУ.1, ПП.1, РУ.1). На приборе (ВЫУ.1) разыгрывается вероятностный переход $ПВ_1$. Прибор (ПП.1) служит для согласования моделей, и на его выходе формируется сигнал, означающий завершение решения задачи i и разрешение поступления нового задания на обслуживание моде-

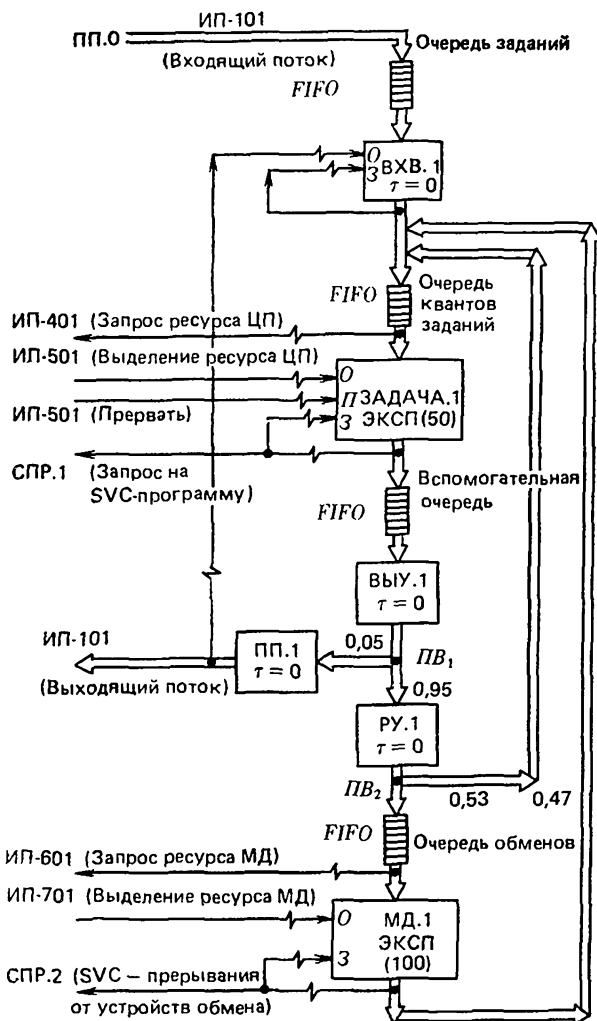


Рис. 6.5. Детальное графическое описание использования ресурсов ЕС ЭВМ задачей I

люю задачи. Прибор (ПУ.1) позволяет реализовать вероятностный переход $ПВ_2$.

Каждой очереди указывается дисциплина обслуживания. На рис. 6.5 показано, что у всех очередей установлена дисциплина выбора транзактов FIFO. Поскольку МК АСИМ не допускает параметризованного представления в тексте описания устройств, то распределения $F(\tau_i)$ и $F(\tau^B_i)$, а также вероятности P_{0i} и P_{1i} должны быть явно указаны на детальном графическом описании. Поэтому на рис. 6.5 $F(\tau_i)$ и $F(\tau^B_i)$ соответствуют описания распределений ЭКСП (50) и ЭКСП (100), которые означают, что

используются экспоненциальные распределения с параметрами, равными соответственно 50 и 100. Аналогично вместо буквенных значений P_{0i} и P_{1i} необходимо указать цифры. На рис. 6.5 в качестве примера показано: $P_{0i}=0,05$; $1-P_{0i}=0,95$; $P_{1i}=0,53$; $1-P_{1i}=0,47$.

Указываются также имена входящих и выходящих потоков транзактов, а также имена устройств, на которые поступают управляющие сигналы от данной модели, и имена потоков, из которых рождаются управляющие сигналы в модели супервизора. Так, на рис. 6.5 показано следующее: входящий и выходящий потоки транзактов имеют имя ИП=101. Управляющие сигналы «Запрос ресурса ЦП» и «Запрос ресурса МД» поступают на управляемые источники с именами соответственно ИП=401 и ИП=601. Управляющие сигналы «Выделение ресурса ЦП» и «Выделение ресурса МД» рождаются из управляющих транзактов с именами соответственно ИП=501 и ИП=701. Сигналы «Запрос на SVC-программу» и «SVC-прерывание от устройств обмена» поступают соответственно на устройства СПР.1 и СПР.2 в модели супервизора. Указание всей этой дополнительной информации необходимо для последующей кодировки описания имитационной модели задачи на языке МК АСИМ. Аналогичные вспомогательные устройства и очереди используются и при составлении детального описания остальных компонент ВП в ОС ЕС.

Составленное подобным образом детальное графическое описание компонент ВП в ОС ЕС готово для описания модели на языке МК АСИМ. Использование МК АСИМ в качестве средства реализации имитационной модели освобождает нас от составления блоков задания начальных условий контроля за ходом моделирования эксперимента. Функции перечисленных блоков выполняет системная часть МК АСИМ.

6.6. ПРОГРАММИРОВАНИЕ И ОТЛАДКА ИМИТАЦИОННОЙ МОДЕЛИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В ОС ЕС

Пятый этап реализации имитационной модели начинается с кодирования компонент описания имитационной модели ВП в ОС ЕС. Каждое функциональное устройство описывается набором декларативных утверждений, отражающих все аспекты его поведения в ходе имитации. Поскольку при моделировании надежностные аспекты не рассматриваются, то каждое функциональное устройство описывается набором из шести основных предложений языка МК АСИМ: УСТРОЙСТВО, ВХОДЯЩИЙ ПОТОК, ОЧЕРЕДЬ, МЕХАНИЗМ ОБСЛУЖИВАНИЯ, ВРЕМЯ ОБСЛУЖИВАНИЯ, ВХОДЯЩИЙ ПОТОК.

Пример. Рассмотрим фрагмент описания на языке МК АСИМ нескольких функциональных устройств, входящих в состав подмодели использования

ресурсов задачи 1. По детальному графическому описанию, представленному на рис. 6.5, составляется следующий текст описания:

УСТРОЙСТВО (ВХВ.1);

ВХОДЯЩИЙ ПОТОК: (ИП=101 У (ПП, 0));

ОЧЕРЕДЬ: ОБЩАЯ;

МЕХАНИЗМ ОБСЛУЖИВАНИЯ: FIFO;

ВРЕМЯ ОБСЛУЖИВАНИЯ: ПС (1Е-12);

ВЫХОДЯЩИЙ ПОТОК: СРН СТ ВЫКЛ (У (ВХВ.1));

УСТРОЙСТВО (ЗАДАЧА.1);

ВХОДЯЩИЙ ПОТОК:

(ИП=101 У (ВХВ.1)) СРН СТ ВКЛ * ((ИП=401));

(ИП=101 У (РУ.1)) СРН СТ ВКЛ * ((ИП=401));

(ИП=101 У (МД.1)) СРН СТ ВКЛ * ((ИП=401));

ОЧЕРЕДЬ: ОБЩАЯ;

МЕХАНИЗМ ОБСЛУЖИВАНИЯ: FIFO;

ВРЕМЯ ОБСЛУЖИВАНИЯ: ЭКСП (50);

ВЫХОДЯЩИЙ ПОТОК:

СРН СТ ВКЛ * (У (СРП.1)) ((ИП=501) ВЫКЛ (У (ЗАДАЧА.1)));

УСТРОЙСТВО (ВЫУ.1);

ВХОДЯЩИЙ ПОТОК:

(ИП=101 У (ЗАДАЧА.1));

ОЧЕРЕДЬ: ОБЩАЯ;

МЕХАНИЗМ ОБСЛУЖИВАНИЯ: FIFO;

ВРЕМЯ ОБСЛУЖИВАНИЯ: ПС (1Е-12);

ВЫХОДЯЩИЙ ПОТОК:

ППВ (0.05 У (ПП.1), 0.95 У (РУ.11));

УСТРОЙСТВО (ПП.1);

ВХОДЯЩИЙ ПОТОК:

(ИП=101 У (ВЫП.1));

ВРЕМЯ ОБСЛУЖИВАНИЯ: ПС (1Е-12);

ВЫХОДЯЩИЙ ПОТОК:

СРН СТ ВКЛ (У (ВХВ.1));

УСТРОЙСТВО (РУ.1);

ВХОДЯЩИЙ ПОТОК:

(ИП=101 У (ВЫУ.1));

ВРЕМЯ ОБСЛУЖИВАНИЯ: ПС (1Е-12);

ВЫХОДЯЩИЙ ПОТОК:

ППВ (0.47 У (ЗАДАЧА.1), 0.53 У (МД.1));

УСТРОЙСТВО (МД.1);

ВХОДЯЩИЙ ПОТОК:

(ИП=101 У (РУ.1)) СРН СТ ВКЛ * ((ИП=601));

ОЧЕРЕДЬ: ОБЩАЯ;

МЕХАНИЗМ ОБСЛУЖИВАНИЯ: FIFO;

ВРЕМЯ ОБСЛУЖИВАНИЯ: ЭКСП (100);

ВЫХОДЯЩИЙ ПОТОК:

СРН СТ ВКЛ (У (СРП.2)) ВЫКЛ (У (МД.1));

Поясним некоторые узловые моменты представленного выше описания фрагмента модели на языке МК АСИМ. Информационные транзакты, имеющие обозначение ИП=101, поступают извне с устройства (ПП, 0). Время обслу-

живания на входном вентиле (ВХВ.1) постоянное и равно нулю. Обозначение * (ИП=401) означает включение управляемого источника управляющих требований с индексом (ИП=401). Транзакты обслуживаются на устройстве (ЗАДАЧА.1) с длительностью, определяемой по экспоненциальному закону с параметром, равным 50. Сигналы «Выключить» и «Прервать» устройство (ЗАДАЧА.1) объединены, поступают из модели супервизора и рождаются управляющими транзактами с индексом (ИП=501). Конструкция ППВ в выходящем потоке устройства (РУ.1) указывает, что с вероятностью 0,47 транзакты поступают в очередь квантов заданий, а с вероятностью 0,53 они направляются в очередь обменов. При поступлении в очередь обменов транзакты рожают сигналы «Включить управляемый источник» с индексом (ИП=601) в модели супервизора, имитируя таким способом запрос к супервизору ввода-вывода на ресурс МД.

Машинный эксперимент с имитационной моделью, разработанной на языке МК АСИМ, включает следующие этапы: перфорация и ввод текста модели, запуск модели, отладка синтаксиса, проверка работоспособности модели. Запуск модели заключается в вызове из библиотеки загрузочного модуля МК АСИМ и подаче на его вход описания модели. В результате выполнения программы «Интерпретатор МК АСИМ» на печать выдается список ошибок и предупреждений с конкретной их привязкой к устройству и типу предложения.

Следующий этап заключается в исправлении допущенных ошибок и приведении описания предложений в соответствие с синтаксисом входного языка МК АСИМ. Наиболее эффективный способ отладки с помощью МК АСИМ возможен на основе диалоговых систем редактирования текстовых библиотек. При этом описание модели ВП заносится в библиотеку исходных модулей и дальнейшая корректировка текста производится с дисплеев.

Отсутствие ошибок в описании модели ВП позволяет реализовать пробный эксперимент. На основе его результатов производится оценка процессов функционирования как всей модели, так и ее отдельных узлов. С этого момента считаем этап построения имитационной модели завершенным и переходим к испытанию модели ВП. Выбор декларативного языка описания модели существенно облегчил создание модели и за счет практического исключения этапа программирования в традиционном применении повысил уровень технологии создания имитационной модели ВП в ОС ЕС.

В качестве выходного документа фазы создания имитационной модели ВП в ОС ЕС может служить текст описания модели на языке МК АСИМ. Практически описание модели представляет собой набор русских предложений с соответствующей семантикой, который является исходной информацией для последующего перевода описания на внутренний язык списковых структур — сплетений [70], используемых управляющей программой МК АСИМ в ходе имитационного эксперимента.

СПИСОК ЛИТЕРАТУРЫ

1. Авен О. И., Гурин Н. Н., Коган Я. А. Оценка качества и оптимизация вычислительных систем. — М.: Наука, 1982. — 464 с.
2. Автоматизация проектирования вычислительных систем. Языки, моделирование и базы данных/Под ред. М. Брейера. — М.: Мир, 1979. — 463 с.
3. Андрианов А. Н., Бычков С. П., Хорошилов А. И. Средства раздельной компиляции СИМУЛА-программ на БЭСМ-64 ЕС ЭВМ. — М.: ИПМ АН СССР, 1982. — 40 с.
4. Андрианов А. И., Бычков С. П. и др. Система моделирования на базе языка СИМУЛА-67 для БЭСМ-6 и ЕС ЭВМ//Моделирование дискретных управляющих и вычислительных систем: Тез. докл. III Всесоюз. семинара. — Свердловск, 1981. — С. 44, 45.
5. Аксенов А. С., Галиев Р. С. и др. Пакет OSSTAN в имитационном моделировании вычислительных комплексов//Моделирование дискретных управляющих и вычислительных систем: Тез. докл. IV Всесоюз. семинара. — Свердловск, 1984. — С. 4—6.
6. Афифи А., Эйзен С. Статистический анализ: Подход к использованию ЭВМ. — М.: Мир, 1982. — 480 с.
7. Ахламов А. Г., Басва И. Н. и др. Диалоговые процедуры управления ходом имитационного эксперимента на ЭВМ//Теория и практика имитационного моделирования сложных систем: Тез. докл. республик. науч.-техн. конф. — Одесса, 1983. — С. 16, 17.
8. Базенов В. И., Стрельченко А. М. Основы планирования и моделирования в теории инженерного эксперимента. Учеб. пособие факультета повыш. квалиф. ИТР. — М.: МАИ, 1983. — 58 с.
9. Беспалов В. М., Копелев Ю. И. Система интерактивного комбинированного имитационного моделирования СИКИМ//Теория и практика имитационного моделирования сложных систем: Тез. докл. republ. науч.-техн. конф. — Одесса, 1983. — С. 17, 18.
10. Бусленко Н. П., Калашников В. В., Коваленко И. Н. Лекции по теории сложных систем. — М.: Сов. радио, 1973. — 439 с.
11. Бусленко В. Н. Автоматизация имитационного моделирования сложных систем. — М.: Наука, 1977. — 240 с.
12. Браверман Э. М., Мучник И. Б. Структурные методы обработки эмпирических данных. — М.: Наука, 1983. — 464 с.
13. Бородюк В. П., Волошин А. П., Иванова А. З. Статистические методы в инженерных исследованиях. Лаб. практикум (для вузов): Под ред. Г. К. Круга. — М.: Высшая школа, 1983. — 216 с.
14. Бежикина И. И., Галицкая В. И. и др. Структура новой версии агрегативной имитационной системы//Теоретические и прикладные вопросы моделирования/МНИПУ. — М., 1984. — С. 5—13.
15. Бычков С. П., Перевицкий О. В., Фисун В. А. Входной язык компилятора с языка ДИНАМО для БЭСМ-6//Математическое обеспечение моделирования сложных систем: Тез. докл. Всесоюз. конф. Ч. 1. — Киев, 1977. — С. 119.
16. Волковинский М. И., Кабалевский А. Н. Анализ приоритетных очередей с учетом времени переключения. — М.: Энергоатомиздат, 1981. — 168 с.
17. Вознесенский В. А. Статистические методы планирования эксперимента в технико-экономических исследованиях. — М.: Финансы и статистика, 1981. — 263 с.
18. Вычислительные системы/Под ред. Э. А. Евреинова. — М.: Статистика, 1980. Вып. 1. — 158 с.
19. Волгин В. В., Усенко В. В. Конспект лекций по курсу «Теория эксперимента». Проверка статистических гипотез. — М.: МЭИ, 1981. — 58 с.
20. Глушков В. М., Калинин А. А. и др. СЛЕНГ — система программирования для моделирования дискретных систем. — Киев: Ин-т кибернетики АН УССР, 1969. — 320 с.
21. Головкин Б. А. Расчет характеристик и планирование параллельных вычислительных процессов. — М.: Радио и связь, 1983. — 272 с.

22. Гуревич Е. И., Татарникова Ю. А. Система функционального моделирования цифровых систем МОДИС-ВЕС//Моделирование дискретных управляющих и вычислительных систем: Тез. докл. IV Всесоюз. семинара. — Свердловск, 1984. — С. 31—34.
23. Данилошкин В. П. Справочник системного программиста по операционной системе ОС ЕС. — М.: Финансы и статистика, 1982. — 224 с.
24. Джонсон Н., Лион Ф. Статистика и планирование эксперимента в технике и науке. Методы обработки данных. — М.: Финансы и статистика, 1980. — 114 с.
25. Дризо В. Е. Применение средств диалогового взаимодействия в имитационной системе ПРОГНОЗ//Теория и практика имитационного моделирования сложных систем: Тез. докл. республик. научн.-технич. конф. — Одесса, 1983. — С. 13, 14.
26. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. — М.: Финансы и статистика, 1986. — 365 с.
27. Демуськов А. Б., Максимей И. В. Пакет программ измерения характеристик функционирования операционной системы ЕС ЭВМ//Теория сложных систем и методы их моделирования: Труды семинара. — М.: ВНИИСИ, 1984. — С. 85—95.
28. Ермаков С. М., Михайлов Г. А. Статистическое моделирование: Учеб. пособие для вузов, 2-е изд., перераб. — М.: Наука, 1982. — 296 с.
29. Ермаков С. М., Бродский В. З., Жиглевский А. А. Математическая теория планирования эксперимента. — М.: Наука, 1983. — 391 с.
30. Зайцев В. И., Насонов А. С., Сериков Г. С. Моделирование систем управления передачей и обработкой данных//Управляющие системы и машины. — 1982. — № 2. — С. 72—74.
31. Зайцев В. Г., Егизарян В. Т. Способ автоматизированного проектирования программного обеспечения АСУТП//Управляющие системы и машины. — 1979. — № 6. — С. 35—40.
32. Земленский А. А., Персиц М. Г. Основы операционной системы ЕС ЭВМ. — М.: Сов. радио, 1980. — 144 с.
33. Ивченко Г. И., Медведев Ю. И. Математическая статистика: Учеб. пособие для вузов. — М.: Высшая школа, 1984. — 248 с.
34. Иберла К. Факторный анализ. — М.: Статистика, 1980. — 398 с.
35. Имитационное моделирование производственных систем//Под ред. А. А. Вавилова. — М.: Машиностроение, Берлин: Техника, 1983. — 416 с.
36. Калашников В. В., Лутков В. И. и др. Вопросы разработки имитационных систем//Электронная техника. Сер. Экономика и системы управления. — 1983. — Вып. 1. — С. 71—87.
37. Калашников В. В., Немчинов Б. В., Петров С. Т. КОМПАС — программный язык описания агрегативных систем//Моделирование дискретных управляющих и вычислительных систем: Тез. докл. IV Всесоюз. семинара. — Свердловск, 1984. — С. 44—46.
38. Калашников В. В. Организация моделирования сложных систем//Математика и кибернетика. — М.: Знание, 1982. — № 3/82. — С. 64—72.
39. Калинин Л. А., Щербин В. М. СКИФ — независимый от языков программирования пакет программ, ориентированный на имитационное моделирование систем с дискретными событиями//Управляющие системы и машины. — 1976. — № 4. — С. 37—42.
40. Киндлер Е. Языки моделирования. — М.: Энергоатомиздат, 1985. — 288 с.
41. Клейнен Дж. Статистические методы в имитационном моделировании: В 2-х т. — М.: Статистика, 1978. — 221, 335 с.
42. Клейнрок Л. Теория массового обслуживания. — М.: Машиностроение, 1979. — 430 с.
43. Климов Г. П. и др. Математические модели систем с разделением времени. — Кишинев: Штиинца, 1983. — 110 с.
44. Комплекс программ сопровождения и обработки статистики по ОБСП и мониторингной системе «Дубна»/С. Ю. Капральченко, М. Н. Кюса, А. В. Рядченко//Математическое обеспечение и технические средства СКП ЭВМ МГУ/МГУ. НИВЦ. М., 1984. С. 115—118.

45. Кононенко И. А., Ситников О. И. Базовый язык моделирования DISLIN// Математическое обеспечение автоматического проектирования (моделирование дискретных систем): Труды ИММ УНЦ АН СССР. — Свердловск, 1981. — С. 11—38.
46. Короткевич В. А., Максимей И. В., Демуськов А. Б. Об исследовании вычислительного процесса в ОС ЕС//Управляющие системы и машины. — 1984. — № 1. — С. 42—47.
47. Кокс Д. Р., Снел Э.-Д. Прикладная статистика: Принципы и примеры. — М.: Мир, 1984. — 200 с.
48. Красовский Г. И., Филаретов Г. Ф. Планирование эксперимента. — Минск. Изд-во БГУ, 1982. — 302 с.
49. Линник Ю. В. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений. — М.: Физматгиз, 1982. — 320 с.
50. Литвинов В. В. Математическое обеспечение проектирования вычислительных систем и сетей. — Киев: Техника, 1982. — 176 с.
51. Максимей И. В., Хвещук В. И. и др. Моделирующий комплекс PLSIM// Теория сложных систем и методы их моделирования: Труды семинара/ВНИИСИ. — М., 1982. — С. 107—111.
52. Максимей И. В., Семишин Ю. А. Об автоматизации моделирования сетей СМО с динамической структурой//Управляющие системы и машины. — 1981. — № 6. — С. 16—22.
53. Максимей И. В., Аксенов А. С. и др. Возможности пакета задач планирования экспериментов ПЭОН//Автоматизация проектирования технологических процессов/ИТК АН БССР. — Минск, 1984. — Вып. 1. — С. 97—110.
54. Максимей И. В., Семишин Ю. А. Вопросы моделирования сетей массового обслуживания с динамической структурой//Электронная техника. Сер. Экономика и системы управления. — 1981. — № 1. — С. 45—50.
55. Максимей И. В., Семишин Ю. А. Технологические возможности имитационного моделирования сетей массового обслуживания//Электронная техника. Сер. Экономика и системы управления. — 1982. — № 1. — С. 45—50.
56. Максимей И. В., Семишин Ю. А. Моделирующий комплекс сетей массового обслуживания с динамической структурой//Моделирование дискретных управляющих и вычислительных систем: Тез. докл. III Всесоюз. семинара. — Свердловск, 1981. — С. 94, 95.
57. Максимей И. В., Семишин Ю. А. Языковые и технологические возможности МК АСИМ для построения имитационных моделей больших систем//Моделирование дискретных управляющих и вычислительных систем. Тез. докл. IV Всесоюз. семинара. — Свердловск, 1984. — С. 61—63.
58. Максимей И. В., Хвещук В. И. Вопросы технологии диалогового моделирования//Теория и практика имитационного моделирования сложных систем: Тез. докл. республ. науч.-техн. конф. — Одесса, 1983. — С. 4, 5.
59. Мановицкий В. И., Сурков Е. М. Система имитационного моделирования дискретных процессов ДИСМ. — Киев, Одесса: Виша школа, 1981. — 94 с.
60. Марьянович Т. П., Сахнюк М. А., Шемшур А. В. Пакет моделирования НЕДИС—ЕС//Теория и методы математического моделирования: Тез. докл. VII Всесоюз. совещания. — Куйбышев: Наука, 1978. — С. 165, 166.
61. Маркова Е. В., Лысенков Л. Н. Комбинированные планы в задачах многофакторного эксперимента. — М.: Наука, 1979. — 348 с.
62. Митрофанов Ю. И., Кубангулов В. Х. Исследование ВЦКП СОАН СССР (модели и результаты). — Новосибирск, 1979. — 50 с. (ВЦ СОАН СССР, № 143, 144).
63. Наумов В. В., Пеледов Г. В. и др. Супервизор ОС ЕС ЭВМ. — М.: Статистика, 1975. — 87 с.
64. Нечипуренко М. И., Хайрутдинов А. Х., Шабров Г. В. Язык моделирования ДИС//Тез. докл. 5-й Межвуз. конф. по физическому и математическому моделированию/МЭИ. — М., 1968. — С. 77—84.
65. Околынишников В. В. Система автоматизации дискретного моделирования МОДЕЛЬ-6/Математическое обеспечение моделирования сложных систем: Тез. докл. Всесоюз. конф. — Киев, 1977. — С. 140, 141.
66. Основы теории вычислительных систем/Под ред. С. А. Майорова: Учеб. пособие для вузов. — М.: Высшая школа, 1978. — 403 с.

67. Поляк Ю. Г. Вероятностное моделирование на электронных вычислительных машинах. — М.: Сов. радио, 1971. — 399 с.
68. Пранявичюс Г. Модели и методы исследования вычислительных систем. — Вильнюс: Мокслас, 1982. — 227 с.
69. Поляков А. К. Архитектура многофункциональной системы моделирования МПЛ/1//Моделирование дискретных управляющих и вычислительных систем: Тез. докл. 3-го Всесоюз. семинара. — Свердловск, 1981. — С. 42, 43.
70. Семишин Ю. А. Входной язык автоматизированной системы имитационного моделирования. — Одесса, 1978. — 26 с. — Деп. в ВИНТИ, № 3655—78.
71. Семишин Ю. А. Автоматизация процесса разработки моделей вычислительных систем: Дис. канд. техн. наук. — Киев, 1983. — 240 с.
72. Система математического обеспечения ЕС ЭВМ/Под ред. А. М. Ларпонова. — М.: Статистика, 1980. — 216 с.
73. Семенков О. И., Волошин Ю. И., Максимей И. В. Конструктор имитационных моделей вычислительных систем и комплексов//Тез. докл. 6-й Всесоюз. школы-семинара по вычислительным сетям. Ч. 2. — М., 1983. — С. 242—247.
74. Семенков О. И., Волошин Ю. И., Максимей И. В. Об использовании системы вложенных имитационных моделей при проектировании ВЦКП систем автоматизации проектирования//Проблемы проектирования создания вычислительных центров и систем коллективного пользования: Тез. докл. IV Всесоюз. научн.-техн. конференции. — Минск, 1981. — С. 65—68.
75. Смит Дж. М. Математическое и цифровое моделирование для инженеров и исследователей. — М.: Машиностроение, 1980. — 271 с.
76. Солодовников И. В. Языки, программное обеспечение и организация систем имитационного моделирования. — М.: Машиностроение, 1982. — 49 с.
77. Себер Дж. Линейный регрессионный анализ. — М.: Мир, 1980. — 456 с.
78. Справочник по типовым программам моделирования/А. Г. Ивахненко, Ю. В. Коппа и др.; Под ред. А. Г. Ивахненко. — Киев: Техника, 1980. — 183 с.
79. Стогний А. А., Паньшин Б. Н. Программное обеспечение управления вычислительным процессом в ВЦ и сетях ЭВМ. — Киев: Наукова думка, 1983. — 305 с.
80. Феррари Д. Оценка производительности вычислительных систем. — М.: Мир, 1981. — 576 с.
81. Шеннон Р. Имитационное моделирование систем. — Искусство и наука. — М.: Мир, 1978. — 417 с.
82. Шрайбер Т. Дж. Моделирование на GPSS. — М.: Машиностроение, 1980. — 592 с.
83. CSS/EC: general description, 1985. — 47 p.
84. Прицкер А. Введение в имитационное моделирование и язык СЛЭМ-II. — М.: Мир, 1987. — 647 с.

ОГЛАВЛЕНИЕ

Предисловие	3
Введение	5
Глава 1. Общие вопросы имитационного моделирования	7
1.1. Сущность имитационного моделирования	7
1.2. Понятие о модельном времени	8
1.3. Способы организации квазипараллелизма в имитационных моделях	14
1.4. Технология моделирования сложных систем	37
Глава 2. Построение моделей сложных систем	49
2.1. Составление содержательного описания объекта моделирования	49
2.2. Разработка концептуальной модели объекта моделирования	52
2.3. Формализация объекта моделирования	61
2.4. Преобразование формального языка в описание имитационной модели	81
2.5. Программирование и отладка имитационной модели	91
Глава 3. Анализ существующих средств автоматизации моделирования	99
3.1. Назначение и виды языков моделирования	99
3.2. Инструментальные и технологические возможности средств автоматизации моделирования	106
3.3. О выборе системы моделирования	118
Глава 4. Испытания и эксплуатация имитационных моделей	121
4.1. Технологические этапы испытания и эксплуатации имитационных моделей	121
4.2. Испытание имитационной модели	123
4.3. Исследование свойств имитационной модели	134
4.4. Планирование имитационных экспериментов	144
4.5. Эксплуатация модели	151
Глава 5. Повышение уровня технологии постановки имитационных экспериментов	156
5.1. Способы повышения уровня технологии построения моделей	156
5.2. Повышение уровня технологии моделирования с помощью моделирующего комплекса PLSIM	163
5.3. Повышение уровня технологии моделирования с помощью моделирующего комплекса АСИМ	170
5.4. Перспектива развития средств моделирования	190
Глава 6. Создание имитационной модели вычислительной системы	197
6.1. Вычислительный процесс в ОС ЕС как объект моделирования	197
6.2. Цель и параметры моделирования	200
6.3. Концептуальная модель вычислительного процесса в ОС ЕС	201
6.4. Формальное описание вычислительного процесса в ОС ЕС	208
6.5. Преобразование формального описания в описание имитационной модели вычислительного процесса	221
6.6. Программирование и отладка имитационной модели вычислительного процесса в ОС ЕС	224
Список литературы	227

Научное издание

МАКСИМЕЙ ИВАН ВАСИЛЬЕВИЧ

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ НА ЭВМ

Заведующая редакцией Г. И. Козырева. Научный редактор Е. А. Северякова.
Редактор издательства Т. М. Любимова. Художественный редактор Н. С. Шейн.
Обложка художника В. В. Третьякова. Технический редактор А. Н. Золотарева.
Корректор Н. Л. Жукова

ИБ № 1004

Сдано в набор 16.12.87	Подписано в печать 26.02.88			
Т-05084	Формат 60×90/16	Бумага кн.-журн. № 2	Гарнитура литературная	
Печать высокая	Усл. печ. л. 14,5	Усл. кр.-отт. 14,75	Уч.-изд. л. 16,77	Тираж 15 000 экз.
Изд. № 20953	Зак. № 244	Цена 1 р. 10 к.		
Издательство «Радио и связь». 101000 Москва, Почтамт, а/я 693				

Московская типография № 5 ВГО «Союзучетиздат». 101000 Москва, ул. Кирова, д. 40

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ НА ЭВМ

