

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Государственное образовательное учреждение высшего профессионального образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

О.М. Замятина

МОДЕЛИРОВАНИЕ СЕТЕЙ

*Рекомендовано в качестве учебного пособия
Редакционно-издательским советом
Томского политехнического университета*

Издательство
Томского политехнического университета
2011

УДК 004.72(075.8)
ББК 32.973.202я73
З-269

Замятина О.М.

З-269

Моделирование сетей: учебное пособие / О.М. Замятина: Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2011. – 168 с.

В учебном пособии кратко изложены основы теории моделирования систем, приведены различные виды классификации моделирования и моделей, рассмотрена математическая основа моделирования сетей, средства моделирования сетей, а также разработан практический курс в рамках дисциплины «Моделирование сетей ЭВМ и систем телекоммуникаций».

Пособие подготовлено на кафедре оптимизации систем управления Национального исследовательского Томского политехнического университета, соответствует рабочей программе дисциплины «Моделирование сетей ЭВМ и систем телекоммуникаций» и предназначено для студентов, обучающихся по магистерской программе 230100 Информатика и вычислительная техника (специализация «Сети ЭВМ и телекоммуникации»).

**УДК 004.72(075.8)
ББК 32.973.202я73**

Рецензенты

Доктор технических наук,
профессор кафедры вычислительной техники
Сибирского федерального университета
А.И. Легалов

Доктор технических наук, профессор кафедры
вычислительной техники Томского политехнического университета
В.Г. Спицын

Кандидат технических наук,
начальник отдела корпоративной сети ОАО «Востокгазпром»
П.М. Острась

© ГОУ ВПО НИ ТПУ, 2011
© Замятина О.М., 2011
© Оформление. Издательство Томского
политехнического университета, 2011

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
1. ОСНОВЫ МОДЕЛИРОВАНИЯ СИСТЕМ.....	8
1.1. Модель и моделирование.....	8
1.2. Классификация моделей.....	9
1.2.1. Классификация моделей по степени абстрагирования модели от оригинала	10
1.2.2. Классификация моделей по степени устойчивости.....	18
1.2.3. Классификация моделей по отношению к внешним факторам	18
1.2.4. Классификация моделей по отношению ко времени	18
1.3. Этапы разработки моделей	19
1.4. Выводы по главе	24
1.5. Вопросы по главе	25
2. ОСНОВЫ ТЕОРИИ ВЕРОЯТНОСТЕЙ И МАТЕМАТИЧЕСКОЙ СТАТИСТИКИ.....	26
2.1. Случайные величины и их свойства.....	26
2.2. Основные законы распределения случайных величин.....	31
2.2.1. Нормальное распределение	31
2.2.2. Биномиальный закон распределения	33
2.2.3. Закон распределения Пуассона	34
2.2.4. Равномерный закон распределения.....	36
2.2.5. Экспоненциальный закон распределения	38
2.2.6. Геометрическое распределение	39
2.2.7. Треугольное распределение (распределение Симпсона).....	40
2.3. Выходные данные и стохастические процессы моделирования	41
2.4. Планирование экспериментов.....	42
2.5. Выводы по главе	43
2.6. Вопросы по главе	44

3. МОДЕЛИРОВАНИЕ КОМПЬЮТЕРНЫХ СЕТЕЙ.....	45
3.1. Аналитическое моделирование	46
3.1.1. Аналитическое моделирование на основе систем массового обслуживания	47
3.1.2. Сети Петри.....	54
3.2. Имитационное моделирование.....	59
3.3. Выводы по главе	63
3.4. Вопросы по главе	63
4. СРЕДСТВА МОДЕЛИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ.....	65
4.1. Система Prophecy	67
4.2. Система Modeler.....	70
4.3. Система COMNETIII.....	74
4.4. Система NetMaker XA	76
4.5. Система SES/Strategizer	77
4.6. Система Arena.....	78
4.7. Выводы по главе	80
4.8. Вопросы по главе	81
5. ПРИМЕР МОДЕЛИРОВАНИЯ СЕТИ.....	82
5.1. Введение в предметную область.....	82
5.2. Постановка задачи	83
5.3. Описание модели суперкомпьютерного кластера	83
5.4. Анализ вариантов распределения приоритетов обработки заданий.....	89
5.5. Выводы по главе	94
6. ПРАКТИЧЕСКАЯ ЧАСТЬ ДИСЦИПЛИНЫ.....	95
6.1. Лабораторный практикум.....	95
6.1.1. Лабораторная работа № 1. Модули Create, Resource, Process и Schedule	96
6.1.2. Лабораторная работа № 2. Генераторы случайных чисел и законы распределения	101
6.1.3. Лабораторная работа № 3. Модуль Decide.....	104

6.1.4. Лабораторная работа № 4. Модули Batch, Separate и Match	107
6.1.5. Лабораторная работа № 5. Приоритеты	111
6.1.6. Лабораторная работа № 6. Модули Assign и Hold, построитель выражений (Build Expression)..	114
6.1.7. Лабораторная работа № 7. Статистика.....	118
6.2. Курсовое проектирование	120
6.3. Семинар.....	126
6.4. Выводы по главе	127
ЗАКЛЮЧЕНИЕ	128
Приложение.	
МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО РАБОТЕ С ПП ARENA	129
1. Начало работы с программным пакетом Arena 7.0	129
1.1. Создание модели с помощью ПП Arena 9.0	129
2. Basic Process Panel (панель основных процессов).....	130
2.1. Схемные модули	130
2.2. Модули данных	138
3. Advanced Process Panel (панель усовершенствованных процессов)	141
3.1. Схемные модули	142
3.2. Модули данных	149
4. Advanced Transfer Panel (панель перемещения)	152
4.1. Схемные модули	152
4.2. Модули данных	160
5. Панель отчетов	162
6. Панель навигации.....	163
7. Построитель выражений	163
СПИСОК ЛИТЕРАТУРЫ	166

ВВЕДЕНИЕ

Данное учебное пособие ориентировано на студентов, обучающихся по направлению магистерской подготовки 230100 «Информатика и вычислительная техника» со специализацией в области сетей и систем телекоммуникаций (СТК).

Учебное пособие «Моделирование сетей» ориентировано на формирование у студентов навыков и знаний в теории моделирования систем и сетей с целью последующего их анализа и оптимизации с использованием современных компьютерных технологий.

Теоретическая часть пособия дает сведения об основных понятиях моделирования, о возможных методах классификации моделей, этапах построения имитационных моделей. Также в качестве повторения и закрепления материала приведены основные понятия теории вероятностей и математической статистики, без знания которых невозможно заниматься математическим моделированием сетей. Эти знания являются основой всех программных средств моделирования сетей. В учебном пособии подробно рассмотрены два вида моделирования сетей: *аналитическое* и *имитационное*. Третья глава учебного пособия содержит краткий обзор современного рынка компьютерных средств моделирования сетей.

Учебный материал, ставший основой этого учебного пособия, уже в течение нескольких лет читается студентам Национального исследовательского Томского политехнического университета и апробирована его теоретическая часть. В конце глав приведены вопросы, которые позволят на практике освоить теоретический материал, ответы на эти вопросы можно найти, проанализировав и осмыслив предшествующую главу.

Практическая часть курса разработана с использованием современных образовательных методик, таких как работа в команде, учебные дискуссии, семинары, мастер-классы, case-study метод. Практическая часть состоит из лабораторного практикума, семинара, курсового проектирования и мастер-класса. Все это позволяет студентам освоить и практически применять теоретические знания. В рамках данного учебного курса «Моделирование сетей ЭВМ и систем телекоммуникаций (СТК)» студентам предложено использовать одно из самых современных пакетов имитационного моделирования Arena, в основу которого заложен математический аппарат раскрашенных сетей Петри и систем массового обслуживания. Описание модулей, их свойств, правил моделирования приведено в приложении. Отдельной главой рассмотрен пример разработки имитационной модели суперкомпьютерного кластера, с полученной имитационной моделью проведен ряд экспериментов.

От автора

Автор выражает благодарность всем тем, кто принял участие в подготовке этого пособия, особенно хочется выделить Ушакову Евгению, плодотворное сотрудничество с которой позволило значительно дополнить ряд глав.

Ваши замечания, предложения и вопросы отправляйте по адресу электронной почты zamyatina@tpu.ru. Я буду рада узнать Ваше мнение!

1. ОСНОВЫ МОДЕЛИРОВАНИЯ СИСТЕМ

1.1. Модель и моделирование

Слово «модель» (от лат. *modelium*) означает «мера», «способ», «сходство с какой-то вещью».

Термин «модель» широко используется в различных сферах человеческой деятельности и имеет множество смысловых значений.

Модель – это объект или описание объекта, системы для замещения (при определенных условиях, предложениях, гипотезах) одной системы (т.е. оригинала) другой системой для лучшего изучения оригинала или воспроизведения каких-либо его свойств [1, 8].

Модель – это результат отображения одной структуры (изученной) на другую (малоизученную). Любая модель строится и исследуется при определенных допущениях, гипотезах. Модель должна строиться так, чтобы она наиболее полно воспроизводила те качества объекта, которые необходимо изучить в соответствии с поставленной целью [2, 4].

Мы под «моделью» будем понимать некий материальный или мысленно представляемый объект, который в процессе исследования замещает объект-оригинал так, что его непосредственное изучение дает новые знания об объекте-оригинале.

Во всех отношениях модель должна быть проще объекта и удобнее его для изучения. Таким образом, для одного и того же объекта могут существовать различные модели, классы моделей, соответствующие различным целям его изучения. Необходимым условием моделирования является подобие объекта и его модели. В этом случае мы должны говорить об **адекватности** модели объекту-оригиналу.

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

Под адекватной моделью понимается модель, которая с определенной степенью приближения на уровне понимания моделируемой системы разработчиком модели отражает процесс ее функционирования во внешней среде. Под **адекватностью** (от лат. *adaequatus* – приравненный) будем понимать степень соответствия результатов, полученных по разработанной модели, данным эксперимента или тестовой задачи. Если система, для которой разрабатывается модель, существует, то сравнивают выходные данные модели и этой системы. В том случае, когда два набора данных оказываются подобными, модель существующей систе-

мы считается адекватной. Чем больше общего между существующей системой и ее моделью, тем больше уверенность в правильности модели системы.

Проверка адекватности модели необходима для того, чтобы убедиться в справедливости совокупности гипотез, сформулированных на первом этапе разработки модели, и точности полученных результатов, требуемых техническим заданием.

Для моделей, предназначенных для приблизительных расчетов, удовлетворительной считается точность 10–15 %, а для моделей, предназначенных для использования в управляющих и контролирующих системах, – 1–2 % [2].

Любая модель обладает следующими свойствами:

- конечностью: модель отображает оригинал лишь в конечном числе его отношений;
- упрощенностью: модель отображает только существенные стороны объекта;
- приблизительностью: действительность отображается моделью грубо или приблизительно;
- адекватностью: модель успешно описывает моделируемую систему;
- информативностью: модель должна содержать достаточную информацию о системе в рамках гипотез, принятых при построении модели [3].

Процесс построения, изучения и применения моделей будем называть моделированием, т.е. можно сказать, что **моделирование** – это метод исследования объекта путем построения и исследования его модели, осуществляемый с определенной целью, и состоящий в замене оригинала на модель.

Моделирование базируется на математической теории подобия, согласно которой абсолютное подобие может иметь место лишь при замене одного объекта другим, точно таким же. При моделировании большинства систем (за исключением, возможно, моделирования одних математических структур другими) абсолютное подобие невозможно, и основная цель моделирования – получить модель, достаточно хорошо отображающую функционирование моделируемой системы [3].

1.2. Классификация моделей

В общем случае все модели, независимо от областей и сфер их применения, бывают трех типов: познавательные, прагматические и инструментальные.

Познавательная модель – форма организации и представления знаний, средство соединения новых и старых знаний. Познавательная модель обычно подгоняется под реальность и является теоретической моделью.

Прагматическая модель – средство организации практических действий, рабочего представления целей системы для ее управления. Реальность в них подгоняется под некоторую прагматическую модель. Это, как правило, прикладные модели.

Инструментальная модель – средство построения, исследования и использования прагматических или познавательных моделей.

Познавательные отражают существующие, а прагматические – хоть и не существующие, но желаемые и, возможно, исполнимые отношения и связи.

Вся остальная классификация моделей выстраивается по отношению к объекту-оригиналу, методам изучения и т.п.

1.2.1. Классификация моделей по степени абстрагирования модели от оригинала

По степени абстрагирования от оригинала (рис. 1.1) модели могут быть разделены на материальные (физические) и идеальные. К **материальным** относятся такие способы, при которых исследование ведется на основе модели, воспроизводящей основные геометрические, физические, динамические и функциональные характеристики изучаемого объекта. Основными разновидностями физических моделей являются [4]:

- натурные;
- квазинатурные;
- масштабные;
- аналоговые.

Натурные – это реальные исследуемые системы, которые являются макетами и опытными образцами. Натурные модели имеют полную адекватность с системой-оригиналом, что обеспечивает высокую точность и достоверность результатов моделирования; другими словами, модель натурная, если она есть материальная копия объекта моделирования. Например, глобус – натурная географическая модель земного шара.

Квазинатурные (от лат. «квази» – почти) – это совокупность натурных и математических моделей. Этот вид моделей используется в случаях, когда математическая модель части системы не является удовлетворительной или когда часть системы должна быть исследована во взаимодействии с остальными частями, но их еще не существует либо их включение в модель затруднено или дорого.



Рис. 1.1. Схема классификации моделей по степени абстрагирования от объекта-оригинала

Масштабные модели – это системы той же физической природы, что и оригинал, но отличающиеся от него размерами. В основе масштабных моделей лежит математический аппарат теории подобия, который предусматривает соблюдение геометрического подобия оригинала и модели и соответствующих масштабов для их параметров. Примером масштабного моделирования являются любые разработки макетов домов, а порой и целых районов при проведении проектных работ при строительстве. Также масштабное моделирование используется при проектировании крупных объектов в самолетостроении и кораблестроении.

Аналоговое моделирование основано на аналогии процессов и явлений, имеющих различную физическую природу, но одинаково описываемых формально (одними и теми же математическими уравнениями, логическими схемами и т.п.). В качестве аналоговых моделей используются механические, гидравлические, пневматические системы, но наиболее широкое применение получили электрические и электронные аналоговые модели, в которых сила тока или напряжение является аналогами физических величин другой природы. Например, является общеизвестным, что математическое уравнение колебания маятника имеет эквивалент при записи уравнения колебаний тока.

Идеальное моделирование носит теоретический характер. Различают два типа идеального моделирования: интуитивное и знаковое.

Под **интуитивным** будем понимать моделирование, основанное на интуитивном представлении об объекте исследования, не поддающемся формализации либо не нуждающимся в ней. В этом смысле, на-

пример, жизненный опыт каждого человека может считаться его интуитивной моделью окружающего мира.

Знаковым называется моделирование, использующее в качестве моделей знаковые преобразования различного вида: схемы, графики, чертежи, формулы, наборы символов и т.д., включающие совокупность законов, по которым можно оперировать с выбранными знаковыми элементами. Знаковая модель может делиться на лингвистическую, визуальную, графическую и математическую модели.

Модель **лингвистическая**, если она представлена некоторым лингвистическим объектом, формализованной языковой системой или структурой. Иногда такие модели называют вербальными, например, правила дорожного движения – языковая, структурная модель движения транспорта и пешеходов на дорогах.

Модель **визуальная**, если она позволяет визуализировать отношения и связи моделируемой системы, особенно в динамике. Например, на экране компьютера часто пользуются визуальной моделью объектов, клавиатуры в программе-тренажере по обучению работе на клавиатуре.

Модель **графическая**, если она представлена геометрическими образами и объектами, например, макет дома является натурной геометрической моделью строящегося дома.

Важнейшим видом знакового моделирования является **математическое** моделирование, классическим примером математического моделирования является описание и исследование основных законов механики И. Ньютона средствами математики.

Классификация математических моделей

Математические модели классифицируются:

- по принадлежности к иерархическому уровню;
- характеру отображаемых свойств объекта;
- способу представления свойств объекта;
- способу получения модели;
- форме представления свойств объекта;
- по содержанию вероятностных компонентов.

• **По принадлежности к иерархическому уровню** математические модели делятся на модели микроуровня, макроуровня, метауровня (рис. 1.2).

Математические модели на **микроуровне** процесса отражают физические процессы, протекающие, например, при резании металлов. Они описывают процессы на уровне перехода (прохода).

Математические модели на **макроуровне** процесса описывают технологические процессы.

Математические модели на **метауровне** процесса описывают технологические системы (участки, цехи, предприятие в целом).



Рис. 1.2. Схема классификации математических моделей по принадлежности к иерархическому уровню

• **По характеру отображаемых свойств объекта** модели можно классифицировать на структурные и функциональные (рис. 1.3).



Рис. 1.3. Схема классификации математических моделей по характеру отображаемых свойств объекта

Модель **структурная**, если она представима структурой данных или структурами данных и отношениями между ними; например, структурной моделью может служить описание (табличное, графовое, функциональное или другое) трофической структуры экосистемы. В свою очередь, структурная модель может быть иерархической или сетевой.

Модель **иерархическая** (древовидная), если представима некоторой иерархической структурой (деревом); например, для решения задачи нахождения маршрута в дереве поиска можно построить древовидную модель, приведенную на рис. 1.4.

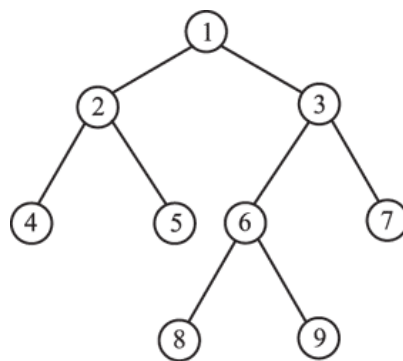


Рис. 1.4. Модель иерархической структуры

Модель *сетевая*, если она представима некоторой сетевой структурой. Например, строительство нового дома включает операции, приведенные в нижеследующей таблице. Эти операции можно представить в виде сетевой модели, приведенной на рис. 1.5 и в табл. 1.1.

Таблица 1.1

Таблица работ при строительстве дома

№	Операция	Время выполнения (дни)	Предшествующие операции	Дуги графа
1	Расчистка участка	1	Нет	–
2	Закладка фундамента	4	Расчистка участка (1)	1–2
3	Возведение стен	4	Закладка фундамента (2)	2–3
4	Монтаж электропроводки	3	Возведение стен (3)	3–4
5	Штукатурные работы	4	Монтаж электропроводки (4)	4–5
6	Благоустройство территории	6	Возведение стен (3)	3–6
7	Отделочные работы	4	Штукатурные работы (5)	5–7
8	Настил крыши	5	Возведение стен (3)	3–8

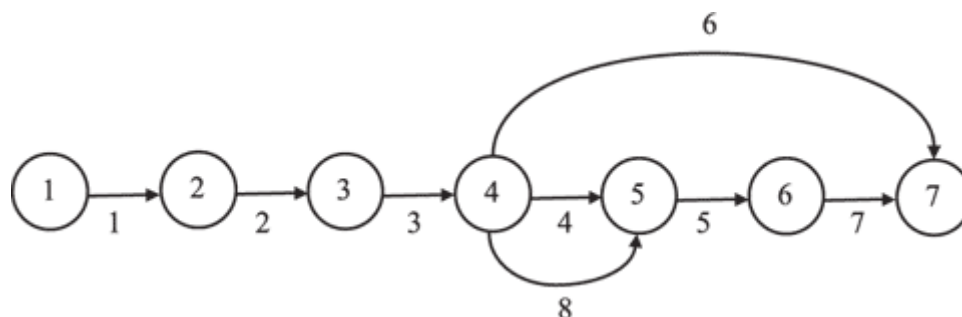


Рис. 1.5. Сетевой график строительства работ

Модель **функциональная**, если она представлена в виде системы функциональных соотношений. Например, закон Ньютона и модель производства товаров – функциональные.

- **По способу представления свойств объекта** (рис. 1.6) модели делятся на аналитические, численные, алгоритмические и имитационные [2, 4, 6, 8, 16].



Рис. 1.6. Схема классификации математических моделей по способу представления свойств объекта

Аналитические математические модели представляют собой явные математические выражения выходных параметров как функций от параметров входных и внутренних и имеют единственные решения при любых начальных условиях. Например, процесс резания (точения) с точки зрения действующих сил представляет собой аналитическую модель. Также квадратное уравнение, имеющее одно или несколько решений, будет аналитической моделью.

Модель будет **численной**, если она имеет решения при конкретных начальных условиях (дифференциальные, интегральные уравнения).

Модель **алгоритмическая**, если она описана некоторым алгоритмом или комплексом алгоритмов, определяющим ее функционирование и развитие. Введение данного типа моделей (действительно, кажется, что любая модель может быть представлена алгоритмом ее исследования) вполне обосновано, т.к. не все модели могут быть исследованы или реализованы алгоритмически. Например, моделью вычисления суммы бесконечного убывающего ряда чисел может служить алгоритм вычисления конечной суммы ряда до некоторой заданной степени точности. Алгоритмической моделью корня квадратного из числа X может служить алгоритм вычисления его приближенного, сколь угодно точного значения по известной рекуррентной формуле.

Модель **имитационная**, если она предназначена для испытания или изучения возможных путей развития и поведения объекта путем варьирования некоторых или всех параметров модели, например модель экономической системы производства товаров двух видов. Такую модель можно использовать в качестве имитационной с целью определения и варьирования общей стоимости в зависимости от тех или иных значений объемов производимых товаров.

- **По способу получения модели** делятся на теоретические и эмпирические (рис. 1.7).

Теоретические математические модели создаются в результате исследования объектов (процессов) на теоретическом уровне. Например, существуют выражения для сил резания, полученные на основе обобщения физических законов. Но они неприемлемы для практического использования, т.к. очень громоздки и не совсем адаптированы к реальным процессам обработки материалов.



Рис. 1.7. Схема классификации математических моделей по способу получения модели

Эмпирические математические модели создаются в результате проведения экспериментов (изучения внешних проявлений свойств объекта с помощью измерения его параметров на входе и выходе) и обработки их результатов методами математической статистики.

- **По форме представления свойств объекта** модели делятся на логические, теоретико-множественные и графовые (рис. 1.8).

Модель **логическая**, если она представлена предикатами, логическими функциями, например совокупность двух логических функций может служить математической моделью одноразрядного сумматора.

Модель **теоретико-множественная**, если она представлена с помощью некоторых множеств и отношений принадлежности к ним и между ними.

Модель *графовая*, если она представлена графом или графами и отношениями между ними.



Рис. 1.8. Схема классификации математических моделей по форме представления свойств объекта

• **По содержанию вероятностных компонентов** модели делятся на детерминированные и стохастические (рис. 1.9).

Если модель не содержит вероятностных (стохастических) компонентов, она называется **детерминированной**. Примером такой модели является система дифференциально-разностных уравнений, описывающих химическую реакцию либо класс систем, описываемых уравнениями в частных производных параболического типа. В детерминированной модели результат можно получить, когда для нее заданы все входные величины.

Однако множество систем моделируются с несколькими случайными входными величинами, в результате чего создается **стохастическая (вероятностная)** модель. Примерами таких моделей являются системы массового обслуживания и управления запасами. Стохастические модели выдают результат, который является случайным сам по себе, и поэтому может он рассматриваться как оценка истинных характеристик модели [9].



Рис. 1.9. Схема классификации математических моделей по содержанию вероятностных компонентов

1.2.2. Классификация моделей по степени устойчивости

Все модели могут быть разделены на устойчивые и неустойчивые.

Устойчивой является такая система, которая, будучи выведена из своего исходного состояния, стремится к нему. Она может колебаться некоторое время около исходной точки подобно обычному маятнику, приведенному в движение, но возмущения в ней со временем затухают и исчезают.

В **неустойчивой** системе, находящейся первоначально в состоянии покоя, возникшее возмущение усиливается, вызывая увеличение значений соответствующих переменных или их колебания с возрастающей амплитудой.

1.2.3. Классификация моделей по отношению к внешним факторам

По отношению к внешним факторам модели могут быть разделены на открытые и замкнутые.

Замкнутой моделью является модель, которая функционирует вне связи с внешними (экзогенными) переменными. В замкнутой модели изменения значений переменных во времени определяются внутренним взаимодействием самих переменных. Замкнутая модель может выявить поведение системы без ввода внешней переменной. Пример: информационные системы с обратной связью являются замкнутыми системами. Это самонастраивающиеся системы, и их характеристики вытекают из внутренней структуры и взаимодействий, которые отражают ввод внешней информации.

Модель, связанная с внешними (экзогенными) переменными, называется **открытой**.

1.2.4. Классификация моделей по отношению ко времени

Существуют две классификации моделей по отношению к временному фактору. Модели могут быть:

- непрерывными или дискретными;
- статическими или динамическими.

Непрерывная модель описывает систему во времени с помощью представления, в котором переменные состояния меняются непрерывно по отношению ко времени. Примером непрерывной модели является сложная система дифференциальных уравнений, которые устанавливают отношения для скоростей изменения переменных состояния во вре-

мени. В *дискретной* модели значения переменных можно определить только в конкретные моменты времени [6].

По отношению к временному фактору модели делятся на динамические и статические (рис. 1.10).

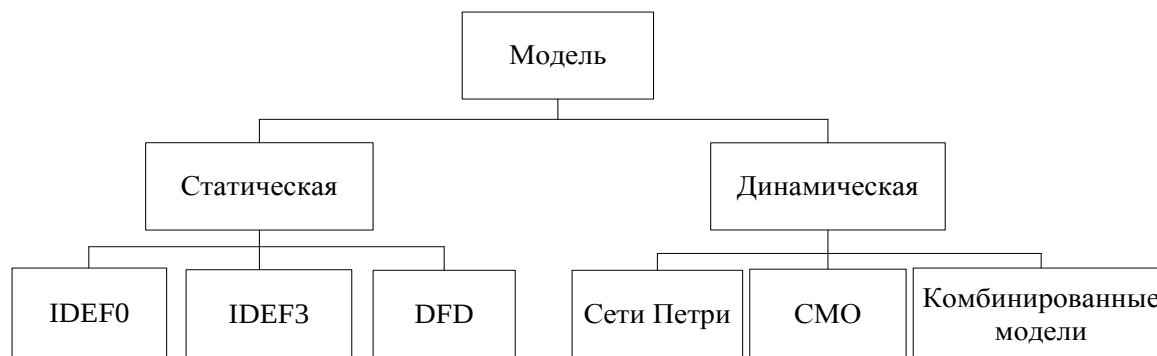


Рис. 1.10. Схема классификации математических моделей по отношению ко времени

Модель называется *статической*, если среди параметров, участвующих в ее описании, нет временного параметра. Статическая модель в каждый момент времени дает лишь «фотографию» системы, ее срез. Одним из видов статических моделей являются структурные модели.

Динамической моделью называется модель, если среди ее параметров есть временной параметр, т.е. она отображает систему (процессы в системе) во времени.

1.3. Этапы разработки моделей

Процесс моделирования имеет итерационный характер и проводится в рамках ранее сформулированных целей и с соблюдением границ моделирования. Построение начинается с изучения (обследования) реальной системы, ее внутренней структуры и содержания взаимосвязей между ее элементами, а также внешних воздействий, и завершается разработкой модели.

Моделирование – от постановки задачи до получения результатов – проходит следующие этапы:

I. Анализ требований и проектирование:

1. Постановка и анализ задачи и цели моделирования.
2. Сбор и анализ исходной информации об объекте моделирования.
3. Построение концептуальной модели.
4. Проверка достоверности концептуальной модели.

II. Разработка модели:

1. Выбор среды моделирования.
2. Составление логической модели.
3. Назначение свойств модулям модели.
4. Задание модельного времени.
5. Верификация модели.

III. Проведение эксперимента:

1. Запуск модели, прогон модели.
2. Варьирование параметров модели и сбор статистики.
3. Анализ результатов моделирования.

IV. Подведение итогов моделирования согласно поставленной цели и задачи моделирования

Схема этапов моделирования представлена на рис. 1.11.



Рис. 1.11. Схема создания модели

Необходимо отметить, что при разработке конкретных моделей с определенными целями и границами моделирования необязательно все подэтапы должны выполняться. Например, при разработке статических моделей IDEF0, DFD 3 и 4 подэтапы «Разработки модели» не выполняются, т.к. эти методологии не предусматривают задание временных параметров модели.

На первом этапе моделирования – *«Анализ требований и проектирование»* – формулируется концептуальная модель, строится ее формальная схема и решается вопрос об эффективности и целесообразности моделирования системы.

Концептуальная модель (КМ) – это абстрактная модель, определяющая состав и структуру системы, свойства элементов и причинно-следственные связи, присущие анализируемой системе и существенные для достижения целей моделирования. В таких моделях обычно в словесной форме приводятся сведения о природе и параметрах (характеристиках) элементарных явлений исследуемой системы, о виде и степени взаимодействия между ними, о месте и значении каждого элементарного явления в общем процессе функционирования системы. При создании КМ практически параллельно формируется область исходных данных (информационное пространство системы) – этап подготовки исходных данных. На данном этапе выявляются количественные характеристики (параметры) функционирования системы и ее элементов, численные значения которых составят исходные данные для моделирования. Очевидно, что значительная часть параметров системы – это случайные величины. Поэтому особое значение при формировании исходных данных имеют выбор законов распределения случайных величин, аппроксимация функций и т.д. В результате выявления свойств модели и построения концептуальной модели необходимо проверить адекватность модели.

На втором этапе моделирования – *«Разработка модели»* – происходит уточнение или выбор программного пакета моделирования. Выбор средств моделирования: программные и технические средства выбираются с учетом ряда критериев. Непременное условие при этом – достаточность и полнота средств для реализации концептуальной модели. Среди других критериев можно назвать доступность, простоту и легкость освоения, скорость и корректность создания программной модели.

После выбора среды проектирования концептуальная модель, сформулированная на предыдущем этапе, воплощается в компьютерную модель, т.е. решается проблема алгоритмизации и детализации модели.

Модель системы представляется в виде совокупности частей (элементов, подсистем). В эту совокупность включаются все части, которые обеспечивают сохранение целостности системы, с одной стороны, а с другой – достижение поставленных целей моделирования (получения необходимой точности и достоверности результатов при проведении компьютерных экспериментов над моделью). В дальнейшем производится окончательная детализация, локализация (выделение системы из окружающей среды), структуризация (указание и общее описание

связей между выделенными элементами системы), укрупненное описание динамики функционирования системы и ее возможных состояний.

Для того чтобы выполнить подэтап «Задание модельного времени», введем понятие модельного времени. В компьютерной модели переменная, обеспечивающая текущее значение модельного времени, называется *часами модельного времени*.

Существует два основных подхода к продвижению модельного времени: *продвижение времени от события к событию* и *продвижение времени с постоянным шагом* [6].

Подход, использующий продвижение времени в модели от события к событию, применяется всеми основными компьютерными программами и большинством разработчиков, создающих свои модели на универсальных языках (рис. 1.12) [6].

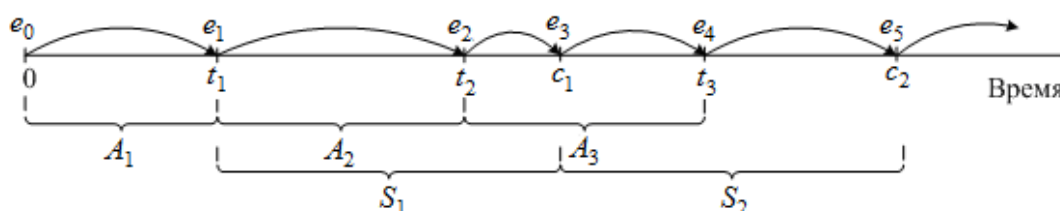


Рис. 1.12. Механизм продвижения модельного времени от события к событию

При использовании продвижения времени от события к событию часы модельного времени в исходном состоянии устанавливаются в 0, и определяется время возникновения будущих событий. После этого часы модельного времени переходят на время возникновения ближайшего события, и в этот момент обновляются состояние системы, с учетом произошедшего события, а также сведения о времени возникновения будущих событий. Затем часы модельного времени продвигаются ко времени возникновения следующего нового ближайшего события, обновляется состояние системы и определяется время будущих событий и т.д. Процесс продвижения модельного времени от времени возникновения одного события ко времени возникновения другого продолжается до тех пор, пока не будет выполнено какое-либо условие останова, указанное заранее. Поскольку в дискретно-событийной имитационной модели все изменения происходят только во время возникновения событий, периоды бездействия системы просто пропускаются, и часы переводятся со времени возникновения одного события на время возникновения другого. При продвижении времени с постоянным шагом такие периоды бездействия не пропускаются, что приводит к большим затратам компьютерного времени. Следует отметить, что длительность интервала продви-

жения модельного времени от одного события к другому может быть различной [6].

При продвижении времени с постоянным шагом Δt часы модельного времени продвигаются точно на Δt единиц времени для какого-либо соответствующего выбора значения Δt . После каждого обновления часов выполняется проверка, чтобы определить, произошли какие-либо события в течение предыдущего интервала времени Δt или нет. Если на этот интервал запланированы одно или несколько событий, считается, что данные события происходят в конце интервала, после чего состояние системы и статистические счетчики соответствующим образом обновляются. Продвижение времени посредством постоянного шага показано на рис. 1.13, где изогнутые стрелки показывают продвижение часов модельного времени, а e_i ($i = 1, 2, \dots$) – это действительное время возникновения события i любого типа, а не значение часов модельного времени. На интервале $[0, \Delta t)$ событие происходит в момент времени e_1 , но оно рассматривается как произошедшее в момент времени Δt . На интервале $[\Delta t, 2\Delta t)$ события не происходят, но все же модель выполняет проверку, чтобы убедиться в этом. На интервале $[2\Delta t, 3\Delta t)$ события происходят в моменты времени e_2 и e_3 , однако считается, что они произошли в момент времени $3\Delta t$ и т.д. В ситуациях, когда принято считать, что два или несколько событий происходят в одно и то же время, необходимо применение ряда правил, позволяющих определять, в каком порядке обрабатывать события. Таким образом, продвижение времени посредством постоянного шага имеет два недостатка: возникновение ошибок, связанных с обработкой событий в конце интервала, в течение которого они происходят, а также необходимость решать, какое событие обрабатывать первым, если события, в действительности происходящие в разное время, рассматриваются как одновременные. Подобного рода проблемы можно частично решить, сделав интервалы Δt менее продолжительными, но тогда возрастает число проверок возникновения событий, что приводит к увеличению времени выполнения задачи. Принимая во внимание это обстоятельство, продвижение времени с помощью постоянного шага не используют в дискретно-событийных имитационных моделях, когда интервалы времени между последовательными событиями могут значительно отличаться по своей продолжительности [6].

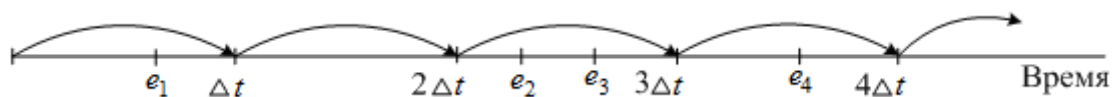


Рис. 1.13. Пример продвижения модельного времени посредством постоянного шага

В основном этот подход предназначен для систем, в которых можно допустить, что все события в действительности происходят в один из моментов n времени Δt ($n = 0, 1, 2, \dots$) для соответственно выбранного Δt . Так, в экономических системах данные часто предоставляются за годовичные промежутки времени, поэтому естественно в имитационной модели установить продвижение времени с шагом, равным одному году. Следует заметить, что продвижение времени посредством постоянного шага может быть выполнено с помощью механизма продвижения времени от события к событию, если планировать время возникновения событий через Δt единиц времени, т.е. данный подход является разновидностью механизма продвижения времени от события к событию.

Третий этап – *«Проведение эксперимента»* – является решающим, на котором, благодаря процессу имитации моделируемой системы, происходит сбор необходимой информации, ее статической обработки в интерпретации результатов моделирования, в результате чего принимается решение: либо исследование будет продолжено, либо закончено. Если известен результат, то можно сравнить его с полученным результатом моделирования. Полученные выводы часто способствуют проведению дополнительной серии экспериментов, а иногда и изменению модели. Основой для выработки решения служат результаты тестирования и экспериментов. Если результаты не соответствуют целям моделирования (реальному объекту или процессу), значит, допущены ошибки на предыдущих этапах или входные данные не являются лучшими параметрами в изучаемой области, поэтому разработчик возвращается к одному из предыдущих этапов.

Подэтап *«Анализ результатов моделирования»* представляет собой всесторонний анализ полученных результатов с целью получения рекомендаций по проектированию системы или ее модификации.

На этапе *«Подведение итогов моделирования согласно поставленной цели и задачи моделирования»* проводят оценку проделанной работы, сопоставляют поставленные цели с полученными результатами и создают окончательный отчет по выполненной работе.

1.4. Выводы по главе

В данной главе были рассмотрены основные понятия теории моделирования. Даны определения модели и моделирования. Показано, что в зависимости от выбранного критерия модели могут быть классифицированы различным образом. Приведена классификация моделей по степени абстрагирования модели от оригинала, по степени устойчивости, по отношению к внешним факторам и по отношению ко времени. Приведены и описаны этапы моделирования.

Для моделирования сетей на высоком уровне необходимо понимание математических основ моделирования. Для этого в следующей главе рассмотрим основы теории вероятностей и математической статистики.

1.5. Вопросы по главе*

1. Что такое модель и моделирование?
2. Что такое адекватная модель?
3. Перечислите, какими свойствами должна обладать модель.
4. Расскажите о классификации моделей.
5. Что такое математическая модель?
6. В чем разница задания модельного времени «продвижение времени от события к событию» и «продвижение времени с постоянным шагом»?
7. На какие типы можно разделить материальные и идеальные модели? Дайте определение каждому типу моделей.
8. Дайте объяснение «детерминированной» и «стохастической» моделям.
9. Что такое аналитическая модель системы?
10. Что такое имитационная модель системы?

* Данные вопросы могут быть использованы для учебной дискуссии.

2. ОСНОВЫ ТЕОРИИ ВЕРОЯТНОСТЕЙ И МАТЕМАТИЧЕСКОЙ СТАТИСТИКИ

Для успешного исследования системы посредством моделирования недостаточно просто создать блок-схему системы, преобразовать ее в компьютерную программу и выполнить один или несколько повторных прогонов для каждой из предложенных конфигураций. Такое исследование требует задания математических параметров модели, что невозможно сделать без применения теории вероятностей и статистики. В частности, теория вероятностей и статистика нужны, чтобы понимать, как моделируется вероятностная система, проверять достоверность имитационной модели, выбирать входные распределения вероятностей, генерировать случайные выборки из этих распределений, выполнять статистический анализ выходных данных моделирования и планировать имитационные эксперименты.

2.1. Случайные величины и их свойства

Эксперимент – это процесс, результат которого точно не известен. Совокупность всех возможных результатов эксперимента называется *пространством выборки* и обозначается S . Сами результаты называются *элементами выборки* в пространстве выборки [6, 9, 17].

Случайная величина – это функция (или правило), которая определяет вещественное число (любое число больше $-\infty$ и меньше ∞) каждому элементу в пространстве выборки S [17, 18]. Случайные величины обозначают прописными буквами X, Y, Z , а значения, которые принимают случайные величины, строчными буквами x, y, z .

Функция распределения вероятностей (иногда именуемая также *интегральной функцией распределения вероятностей*) $F(x)$ случайной величины X определяется для каждого вещественного числа x следующим образом:

$$F(x) = P(X \leq x) \text{ для } -\infty < x < \infty, \quad (2.1)$$

где $P(X \leq x)$ – вероятность, связанная с событием $\{X \leq x\}$.

Следовательно, $F(x)$ – это вероятность того, что после выполнения эксперимента случайная величина X получит значение, не превышающее число x . Функция распределения $F(x)$ имеет такие свойства:

- 1) $0 \leq F(x) \leq 1$ для всех значений x ;
- 2) $F(x)$ является неубывающей функцией; т.е., если $x_1 < x_2$, тогда $F(x_1) \leq F(x_2)$;

3) $\lim_{x \rightarrow \infty} F(x) = 1$ и $\lim_{x \rightarrow \infty} F(x) = 0$ (поскольку X принимает только конечные значения) [17, 18].

Случайная величина X считается *дискретной*, если она может принимать значения из счетного множества, например: $x_1, x_2, \dots$ (счетное множество – это множество возможных значений переменной, взаимно однозначно соответствующих множеству положительных целых чисел; примером несчетного множества являются вещественные числа между 0 и 1). Следовательно, случайная величина, принимающая только конечное число значений x_1, x_2, x_n , является дискретной.

Вероятность, с которой дискретная случайная величина X принимает значение x_i , задается как

$$p(x_i) = P(X = x_i) \text{ для } i = 1, 2, \dots \quad (2.2)$$

и для нее должно выполняться равенство

$$\sum_{i=1}^{\infty} P(x_i) = 1. \quad (2.3)$$

Все вероятностные характеристики величины X могут быть вычислены с помощью функции $p(x)$, именуемой *вероятностной мерой* дискретной случайной величины X . Если $I = [a, b]$, где a и b – вещественные числа, для которых $a \leq b$, то

$$P(X \in I) = \sum_{a \leq x_i \leq b} P(x_i), \quad (2.4)$$

где символ $\in$ обозначает «содержится в», а суммирование означает сложение всех величин $p(x_i)$, для которых $a \leq x_i \leq b$. Функция распределения дискретной случайной величины X

$$F(x) = \sum_{x_i \leq x} P(x_i) \quad (2.5)$$

для всех $-\infty < x < \infty$.

Рассмотрим случайные величины, которые могут принимать только несчетно-бесконечное число различных значений (например, все неотрицательные вещественные числа). Случайная величина X считается *непрерывной*, если существует такая неотрицательная функция $f(x)$, при которой для любого множества вещественных чисел B (например, B может включать все вещественные числа между 1 и 2)

$$P(X \in B) = \int_B f(x) dx \text{ и } \int_{-\infty}^{\infty} f(x) dx = 1. \quad (2.6)$$

Таким образом, общая площадь под функцией $f(x)$ равна 1. Если X – неотрицательная случайная величина, что часто встречается при моделировании, вторая область интегрирования будет в пределах от 0 до ∞ .

Все вероятностные характеристики величины X могут (в принципе) вычисляться с помощью функции $f(x)$, которая называется *плотностью распределения вероятностей* непрерывной случайной величины X .

Для дискретной случайной величины X функция $p(x)$ – это действительная вероятность, связанная со значением x . Однако функция $f(x)$ не является вероятностью того, что непрерывная случайная величина X равна x . Для любого вещественного числа x

$$P(X = x) = P(X \in [x, x]) = \int_x^x f(y) dy = 0. \quad (2.7)$$

Так как вероятность, связанная с каждым значением x , равна 0, можно дать следующую интерпретацию функции $f(x)$. Если x это любое число, а $\Delta x > 0$, тогда

$$P(x, x + \Delta x) = \int_x^{x+\Delta x} f(y) dy = 0, \quad (2.8)$$

что равно площади под функцией $f(x)$ между x и $x + \Delta x$. Отсюда следует, что с большей вероятностью непрерывная случайная величина X попадет в интервал, где функция $f(x)$ имеет большое значение, чем в интервал, где функция $f(x)$ имеет небольшое значение.

Функция распределения непрерывной случайной величины X

$$F(x) = P(X \in [-\infty, x]) = \int_{-\infty}^x f(y) dy \quad (2.9)$$

для всех $-\infty < x < \infty$.

Таким образом, с некоторыми формальными нестрогими допущениями $f(x) = F'(x)$, где $F'(x)$ – производная от функции $F(x)$. Кроме того, если $I = [a, b]$, где a и b – любые вещественные числа, для которых $a < b$, то

$$P(X \in I) = \int_a^b f(y) dy = F(b) - F(a). \quad (2.10)$$

Последнее равенство представляет применение фундаментальной теоремы вычислений, поскольку $F'(x) = f(x)$.

При моделировании обычно приходится иметь дело с n (n – положительное целое число) случайными величинами $X_1, X_2, \dots, X_n$ одновременно. Примем $n = 2$, т.е. воспользуемся двумя случайными величинами X и Y .

Если X и Y являются дискретными случайными величинами, тогда $p(x, y) = P(X = x, Y = y)$ для всех x, y , где $p(x, y)$ называется *совместной вероятностной мерой функции* величин X и Y . При этом величины X и Y будут *независимыми*, если $P(x, y) = P_x(x)P_y(y)$ для всех x, y , где функции

$$P_x(x) = \sum_{\text{всех } y} P(x, y); \quad (2.11)$$

$$P_y(y) = \sum_{\text{всех } x} P(x, y) \quad (2.12)$$

есть безусловные вероятностные меры величин X и Y .

Случайные величины X и Y называются *совместно непрерывными*, если для них существует неотрицательная функция $f(x, y)$, именуемая *совместной функцией плотности распределения вероятностей* величин X и Y , определенная для всех множеств вещественных чисел A и B

$$P(X \in A, Y \in B) = \iint_{BA} f(x, y) dx dy. \quad (2.13)$$

В этом случае величины X и Y являются независимыми, если $f(x, y) = f_x(x) f_y(y)$ для всех x, y , где функции

$$f_x(x) = \int_{-\infty}^{\infty} f(x, y) dy; \quad (2.14)$$

$$f_y(y) = \int_{-\infty}^{\infty} f(x, y) dx \quad (2.15)$$

представляют собой плотности безусловного распределения вероятностей соответственно величин X и Y .

Иными словами, случайные величины X и Y (как дискретные, так и непрерывные) являются независимыми, если известное значение, которое может принимать одна величина, не сказывается на распределении другой величины. Также, если величины X и Y не являются независимыми, их называют *зависимыми*.

Рассмотрим еще раз случай с n случайными величинами $X_1, X_2, \dots, X_n$. В частности, обратимся к некоторым характеристикам отдельной случайной величины X_i и некоторым показателям зависимости, которая может существовать между двумя случайными величинами X_i и X_j .

Среднее значение, или *математическое ожидание*, случайной величины X_i (где $i = 1, 2, \dots, n$) обозначается μ_i или $E(X_i)$ и определяется как

$$\mu_i = \begin{cases} \sum_{j=1}^{\infty} x_j P_{x_i}(x_j), & \text{если } X_i \text{ — дискретная величина;} \\ \int_{-\infty}^{\infty} x f_{X_i}(x) dx, & \text{если } X_i \text{ — непрерывная величина.} \end{cases} \quad (2.16)$$

Средние значения обладают такими важными свойствами (c и c_i обозначают константу — вещественное число):

$$1) E(cX) = cE(X);$$

$$2) E\left(\sum_{i=1}^n c_i X_i\right) = \sum_{i=1}^n c_i E(X_i), \text{ даже если } X_i \text{ зависимые.}$$

Дисперсия случайной величины X_i обозначается σ_i^2 или $\text{Var}(X_i)$. Она определяется как

$$\sigma_i^2 = E[(X_i - \mu_i)^2] = E(X_i^2) - \mu_i^2. \quad (2.17)$$

Дисперсия является показателем рассеяния случайной величины по отношению к ее среднему значению. Чем больше дисперсия, тем более вероятно, что случайная величина будет принимать значения, далекие от среднего.

Дисперсия имеет такие свойства:

- 1) $\text{Var}(X) \geq 0$;
- 2) $\text{Var}(cX) = c^2 \text{Var}(X)$;
- 3) $\text{Var}(\sum_{i=1}^n X_i) = \sum_{i=1}^n \text{Var}(X_i)$,

если значения X_i являются независимыми (или некоррелированными).

Стандартное отклонение случайной величины X_i определяется как $\sigma_i = \sqrt{\sigma_i^2}$. Наиболее точное толкование стандартного отклонения может быть дано, когда X_i имеет нормальное распределение.

Показателем линейной зависимости между случайными величинами X_i и X_j (где $i = 1, 2, \dots, n$; $j = 1, 2, \dots, n$) является *ковариация*, которая обозначается C_{ij} или $\text{Cov}(X_i, X_j)$ и определяется как

$$C_{ij} = E(X_i - \mu_i)(X_j - \mu_j) = E(X_i X_j) - \mu_i \mu_j. \quad (2.18)$$

Ковариации симметричны, т.е. $C_{ij} = C_{ji}$, и если $i=j$, то $C_{ij} = C_{ji} = \sigma_i^2$.

При $C_{ij} = 0$ случайные величины X_i и X_j считаются *некоррелированными*. Легко доказать, что если X_i и X_j являются независимыми случайными величинами, то $C_{ij} = 0$. Однако обратное утверждение не является справедливым. Тем не менее, если X_i и X_j являются совместно нормально распределенными случайными величинами с $C_{ij} = 0$, то они являются также и независимыми.

Приведем два определения, которые помогут уяснить значение ковариации. Если $C_{ij} > 0$, то X_i и X_j считаются *положительно коррелированными* величинами. Тогда имеет место тенденция возникать совместно $X_i > \mu_i$ и $X_j > \mu_j$, а также $X_i < \mu_i$ и $X_j < \mu_j$. Таким образом, если одна из положительно коррелированных случайных величин имеет большое значение, другая, скорее всего, тоже будет иметь большое значение.

Если $C_{ij} < 0$, то X_i и X_j считаются *отрицательно коррелированными* величинами. В этом случае тенденцию возникать совместно имеют $X_i > \mu_i$ и $X_j < \mu_j$, а также $X_i < \mu_i$ и $X_j > \mu_j$. Таким образом, если одна из отрицательно коррелированных случайных величин имеет большое значение, другая, скорее всего, будет иметь маленькое значение.

Если $X_1, X_2, \dots, X_n$ представляют собой выходные данные моделирования, часто нужно знать не только среднее значение и дисперсию σ_i^2 при $i = 1, 2, \dots, n$, но и показатель зависимости между X_i и X_j при $i \neq j$. Однако сложность использования ковариации C_{ij} в качестве показателя зависимости между X_i и X_j заключается в том, что она не является безразмерной величиной, что усложняет ее толкование. (Если X_i и X_j измеряются, например, в минутах, то ковариация C_{ij} будет измеряться в минутах в квадрате.)

В связи с этим в качестве основного показателя линейной зависимости между C_{ij} используется *корреляция* r_{ij} , определяемая по формуле

$$r_{ij} = \frac{C_{ij}}{\sqrt{\sigma_i^2 \sigma_j^2}} \quad i=1,2,\dots,n; \quad j=2,\dots,n. \quad (2.19)$$

Корреляцию между X_i и X_j можно обозначать и как $\text{Corr}(X_i, X_j)$, так как знаменатель в формуле имеет положительное значение, естественно, что корреляция r_{ij} будет иметь тот же знак, что и ковариация C_{ij} . Более того, $-1 \leq r_{ij} \leq 1$ при всех i и j . Если r_{ij} имеет значение, близкое к $+1$, то X_i и X_j – сильно положительно коррелированные величины. Если r_{ij} близко к -1 , то X_i и X_j – сильно отрицательно коррелированные величины [6, 5, 9, 17, 18].

2.2. Основные законы распределения случайных величин

2.2.1. Нормальное распределение

Непрерывная случайная величина X имеет нормальный закон распределения с параметрами a и σ , если её плотность вероятности $f(x)$ имеет вид:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}}. \quad (2.20)$$

Кривая нормального распределения $f(x)$ (нормальная кривая или кривая Гаусса) приведена на рис. 2.1.

Нормальный закон распределения случайной величины с параметрами $a = 0$ и $\sigma = 1$ называется стандартным или *нормированным*, а соответствующая нормальная кривая – *стандартной* или *нормированной*.

Математическое ожидание случайной величины X , распределённой по нормальному закону, равно параметру a этого закона, а её дисперсия – квадрату параметра σ , т.е. $M(X) = a$, $D(X) = \sigma^2$.

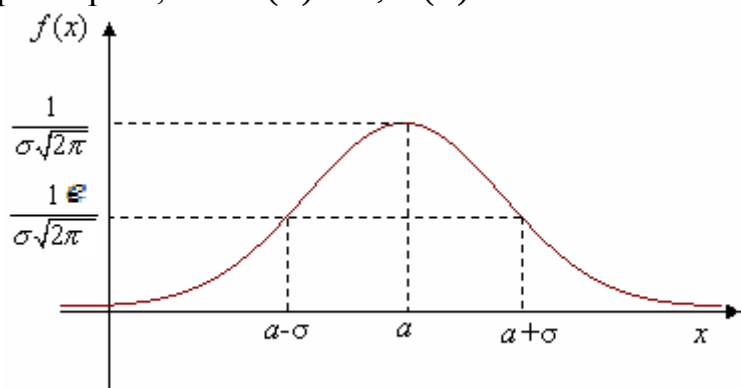


Рис. 2.1. Кривая нормального распределения

Наиболее важные свойства случайной величины, распределённой по нормальному закону:

1. Вероятность попадания случайной величины в интервал (x_1, x_2) , равна

$$P(x_1 < X < x_2) = \Phi(t_2) - \Phi(t_1), \quad (2.21)$$

где $t_1 = \frac{x_1 - a}{\sigma}$, $t_2 = \frac{x_2 - a}{\sigma}$.

2. Вероятность того, что отклонение случайной величины X , распределённой по нормальному закону, от математического ожидания a не превысит по абсолютной величине величину $\Delta > 0$, равна

$$P(|X - a| \leq \Delta) = 2\Phi(t), \quad (2.22)$$

где $t = \frac{\Delta}{\sigma}$.

3. «Правило трёх сигм». Если случайная величина X распределена нормально (с параметрами a и σ), то практически достоверно, что абсолютная величина её отклонения от математического ожидания не превосходит утроенного среднего квадратического отклонения, т.е.

$$P(|X - a| \leq 3\sigma) \approx 1. \quad (2.23)$$

4. Если случайная величина X имеет нормальный закон распределения с параметрами a и σ , то практически достоверно, что её значения заключены в интервале $(a - 3\sigma, a + 3\sigma)$.

5. Коэффициент асимметрии и эксцесс нормально распределённой случайной величины равны нулю [19].

Нормальный закон распределения наиболее часто встречается на практике. Главная особенность, выделяющая его среди других законов, состоит в том, что он является предельным законом, к которому приближаются другие законы при весьма часто встречающихся типичных условиях.

2.2.2. Биномиальный закон распределения

Дискретная случайная величина X имеет биномиальный закон распределения, если она принимает значения $0, 1, 2, \dots, m, \dots, n$ с вероятностями

$$p(X = m) = C_n^m p^m q^{n-m}. \quad (2.24)$$

где $0 < p < 1, q = 1 - p, m = 0, 1, 2, \dots, n$.

Биномиальный закон распределения представляет собой закон распределения числа $X = m$ наступлений события A в n независимых испытаниях, в каждом из которых событие A может осуществиться с одной и той же вероятностью $P(A) = p = \text{const}$. Кроме события A может произойти также противоположное событие $\bar{A}$, вероятность которого $P(\bar{A}) = 1 - p = q$.

Ряд распределения биномиального закона имеет вид:

x_i	0	1	...	2	m	...	n
p_i	q^n	$C_n^1 p q^{n-1}$	$C_n^2 p^2 q^{n-2}$	...	$C_n^m p^m q^{n-m}$	...	p^n

Вероятности любого числа событий соответствуют членам разложения бинома Ньютона в степени, равной числу испытаний:

$$(p + q)^n = p^n + np^{n-1}q + \frac{n(n-1)}{1 \cdot 2} \times \\ \times p^{n-2}q^2 + \dots + C_n^m p^m q^{n-m} + \dots + npq^{n-1} + q^n,$$

где p^n – вероятность того, что при n испытаниях событие A наступит n раз;

q^n – вероятность того, что при n испытаниях событие A не наступит ни разу;

$c_n^m p^m q^{n-m}$ – вероятность того, что при n испытаниях событие A наступит m раз, а событие $\bar{A}$ наступит $n - m$ раз;

c_n^m – число сочетаний (комбинаций) появления события A и $\bar{A}$.

Числовые характеристики биномиального распределения:

- 1) $M(m) = np$ – математическое ожидание частоты появления события A при n независимых испытаниях;
- 2) $D(m) = npq$ – дисперсия частоты появления события A ;
- 3) $\sigma(m) = \sqrt{npq}$ – среднее квадратическое отклонение частоты.

На рис. 2.2 приведены многоугольники (полигоны) распределения случайной величины X , имеющей биномиальный закон распределения с параметрами $n = 5$ и p (для $p = 0,2; 0,3; 0,5; 0,7; 0,8$).

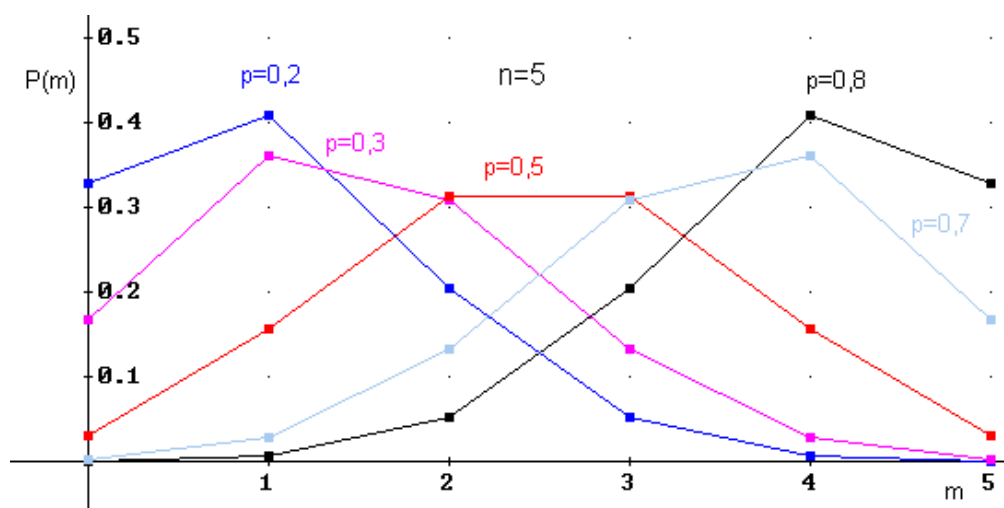


Рис. 2.2. Кривые биномиального распределения

Биномиальный закон широко используется в теории и практике статистического контроля качества продукции, при описании функционирования систем массового обслуживания, в теории стрельбы и в других областях [7].

2.2.3. Закон распределения Пуассона

Дискретная случайная величина X имеет закон распределения Пуассона, если она принимает значения $0, 1, 2, \dots, m, \dots$ (бесконечное, но счётное множество значений) с вероятностями

$$P(X = m) = \frac{\lambda^m e^{-\lambda}}{m!}, \quad (2.25)$$

где $m = 0, 1, 2, \dots$

Ряд распределения закона Пуассона имеет вид:

x_i	0	1	2	...	m	...
p_i	$e^{-\lambda}$	$\lambda e^{-\lambda}$	$\frac{\lambda^2 e^{-\lambda}}{2!}$	...	$\frac{\lambda^m e^{-\lambda}}{m!}$	...

На рис. 2.3 приведены многоугольники (полигоны) распределения случайной величины X , имеющей закон распределения Пуассона с параметром λ (для $\lambda = 0,5; 1; 2; 3,5; 5$).

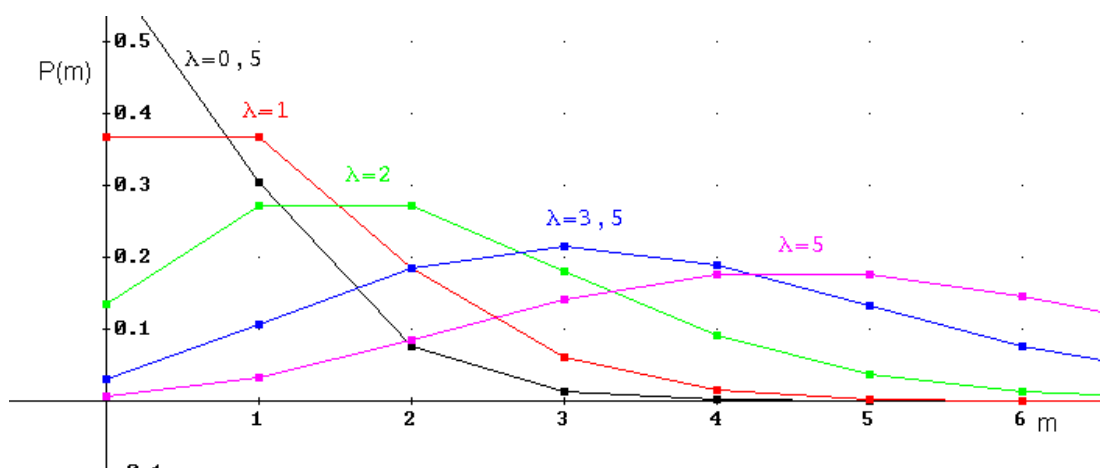


Рис. 2.3. Кривые распределения Пуассона

Математическое ожидание и дисперсия случайной величины X , распределённой по закону Пуассона, совпадают и равны значению параметра этого закона, т.е. $M(X) = \lambda$, $D(X) = \lambda$.

При условии $p \rightarrow 0$, $n \rightarrow \infty$, $np \rightarrow \lambda = \text{const}$ закон распределения Пуассона является предельным случаем биномиального закона. Так как при этом вероятность p события A в каждом испытании мала, то закон распределения Пуассона называют часто *законом редких явлений*.

Наряду с «предельным» случаем биномиального распределения закон Пуассона может возникнуть и в ряде других случаев. Так для простейшего потока событий число событий, попадающих на произвольный отрезок времени, есть случайная величина, имеющая пуассоновское распределение. Также по закону Пуассона распределены, например: число рождений четверней, число сбоев на автоматической линии, число отказов сложной системы в «нормальном режиме», число «требоований на обслуживание», поступивших в единицу времени в системах массового обслуживания, и др.

Если случайная величина представляет собой сумму двух независимых случайных величин, распределённых по закону Пуассона, то она также распределена по закону Пуассона [19].

2.2.4. Равномерный закон распределения

Непрерывная случайная величина X имеет равномерный закон распределения на отрезке $[a, b]$, если её плотность вероятности $f(x)$ постоянна на этом отрезке и равна нулю вне его, т.е.

$$f(x) = \begin{cases} 0 & \text{при } a < x; \\ C & \text{при } a \leq x \leq b; \\ 0 & \text{при } x > b. \end{cases} \quad (2.26)$$

Определим значение константы C , исходя из свойства плотности вероятности

$$\int_{-\infty}^{\infty} f(x) dx = 1. \quad (2.27)$$

$$\int_{-\infty}^{\infty} f(x) dx = \int_{-\infty}^a 0 \cdot dx + \int_a^b C dx + \int_b^{+\infty} 0 \cdot dx = \quad (2.28)$$

$$0 + C \int_a^b dx + 0 = Cx \Big|_a^b = C(b - a) = 1,$$

отсюда $C = \frac{1}{b - a}$.

Таким образом, непрерывная случайная величина X имеет равномерный закон распределения, на отрезке $[a, b]$, если её плотность вероятности

$$f(x) = \begin{cases} 0 & \text{при } x < a; \\ \frac{1}{b - a} & \text{при } a \leq x \leq b; \\ 0 & \text{при } x > b. \end{cases} \quad (2.29)$$

Кривая распределения $f(x)$ приведена на рис. 2.4.

Функция распределения случайной величины X , распределённой по равномерному закону, есть

$$F(x) = \begin{cases} 0 & \text{при } x < a; \\ \frac{x - a}{b - a} & \text{при } a \leq x \leq b; \\ 1 & \text{при } x > b, \end{cases} \quad (2.30)$$

её математическое ожидание

$$M(X) = \frac{a+b}{2}, \quad (2.31)$$

её дисперсия

$$D(X) = \frac{(b-a)^2}{12} \quad (2.32)$$

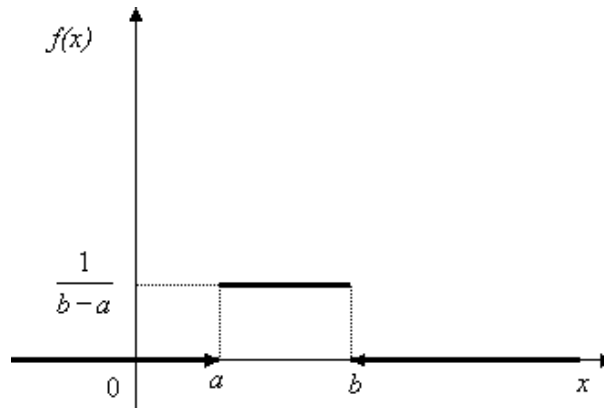


Рис. 2.4. Равномерный закон распределения

График функции распределения $F(x)$ равномерно распределённой случайной величины X представлен на рис. 2.5.

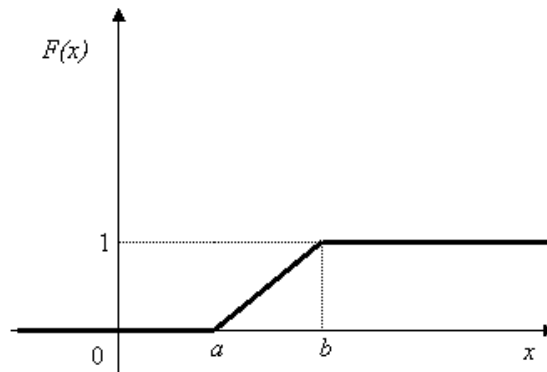


Рис. 2.5. График функции распределения $F(x)$ равномерно распределённой случайной величины X

Равномерный закон распределения используется при анализе ошибок округления при проведении числовых расчётов (например, ошибка округления числа до целого распределена равномерно на отрезке $[-0,5; 0,5]$), в ряде задач массового обслуживания, при статистическом моделировании наблюдений, подчинённых заданному распределению. Так, случайная величина X , распределённая равномерно на отрезке $[0; 1]$, называемая «случайным числом от 0 до 1», служит исходным материалом для получения случайных величин с любым законом распределения [9].

2.2.5. Экспоненциальный закон распределения

Непрерывная случайная величина X имеет экспоненциальный (показательный) закон распределения с параметром λ , если её плотность вероятности $f(x)$ имеет вид:

$$f(x) = \begin{cases} 0 & \text{при } x < 0; \\ \lambda e^{-\lambda x} & \text{при } x \geq 0. \end{cases} \quad (2.33)$$

Кривая распределения $f(x)$ приведена на рис. 2.6.

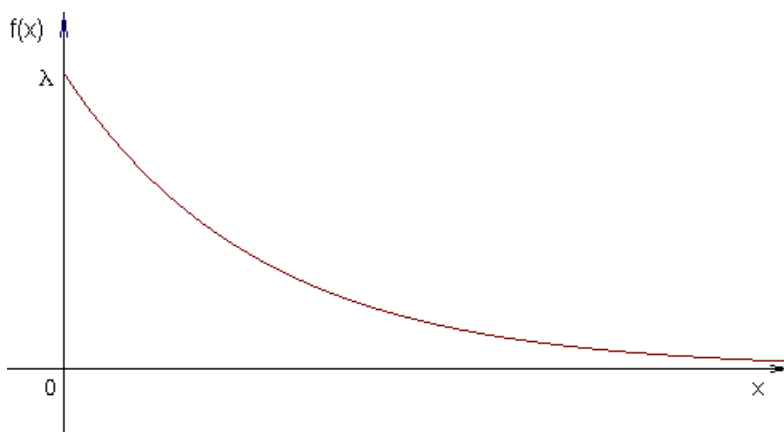


Рис. 2.6. Кривая экспоненциального распределения

Функция распределения случайной величины X , распределённой по экспоненциальному закону, есть

$$F(x) = \begin{cases} 0 & \text{при } x < 0; \\ 1 - e^{-\lambda x} & \text{при } x \geq 0. \end{cases} \quad (2.34)$$

её математическое ожидание

$$M(X) = \frac{1}{\lambda}, \quad (2.35)$$

а её дисперсия

$$D(X) = \frac{1}{\lambda^2}. \quad (2.36)$$

График функции распределения $F(x)$ случайной величины X , имеющей экспоненциальное распределение представлен на рис. 2.7.

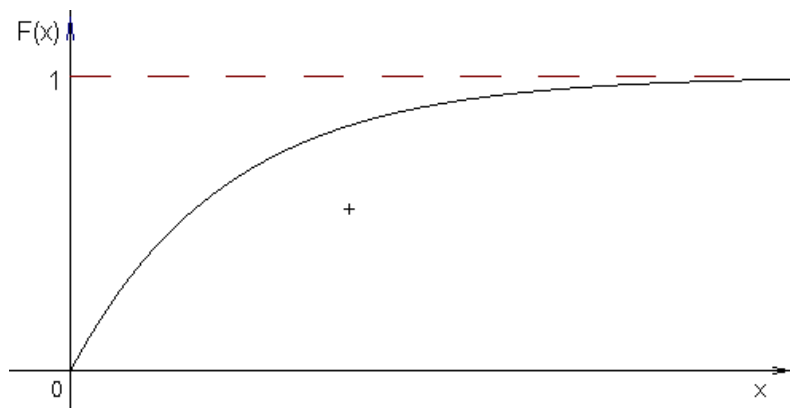


Рис. 2.7. График функции распределения $F(x)$ случайной величины X , имеющей экспоненциальное распределение

Экспоненциальный закон распределения играет большую роль в теории массового обслуживания и теории надёжности. Так, например, интервал времени T между двумя соседними событиями в простейшем потоке событий имеет экспоненциальное распределение с параметром λ – интенсивностью потока [17].

2.2.6. Геометрическое распределение

Дискретная случайная величина X имеет геометрическое распределение, если она принимает значения $1, 2, \dots, m, \dots$ (бесконечное, но счётное множество значений) с вероятностями

$$P(X = m) = pq^{m-1}, \quad (2.37)$$

где $0 < p < 1$, $q = 1 - p$, $m = 1, 2, \dots$

Ряд геометрического распределения имеет вид:

x_i	1	2	3	...	m	...
p_i	p	pq	pq^2	...	pq^{m-1}	...

Очевидно, что вероятности p_i образуют геометрическую прогрессию с первым членом p и знаменателем q (отсюда и название «геометрическое распределение»).

Случайная величина $X = m$, имеющая геометрическое распределение, представляет собой число m испытаний, проведённых по схеме Бернулли, с вероятностью p наступления события в каждом испытании до первого положительного исхода.

Математическое ожидание случайной величины X , имеющей геометрическое распределение с параметром p , равно $M(X) = \frac{1}{p}$, а её дис-

персия $D(X) = \frac{q}{p^2}$, где $q = 1 - p$.

2.2.7. Треугольное распределение (распределение Симпсона)

Случайная величина ξ имеет треугольное распределение (распределение Симпсона) на отрезке $[a, b]$ ($a < b$), если её плотность вероятности вычисляется по формуле

$$f(x) = \begin{cases} \frac{2}{b-a} - \frac{2}{(b-a)^2} |a+b-2x|, & x \in [a, b]; \\ 0, & x \notin [a, b]. \end{cases} \quad (2.38)$$

Характеристическая функция треугольного распределения имеет вид:

$$\varphi(t) = \left[\frac{2(e^{itb/2} - e^{ita/2})}{(b-a)it} \right]^2. \quad (2.39)$$

Дисперсия имеет вид:

$$D\xi = \frac{(b-a)^2}{24}. \quad (2.40)$$

Если ξ_1 и ξ_2 – независимые случайные величины, равномерно распределенные на отрезке $[a/2, b/2]$, то случайная величина $\xi = \xi_1 + \xi_2$ имеет треугольное распределение.

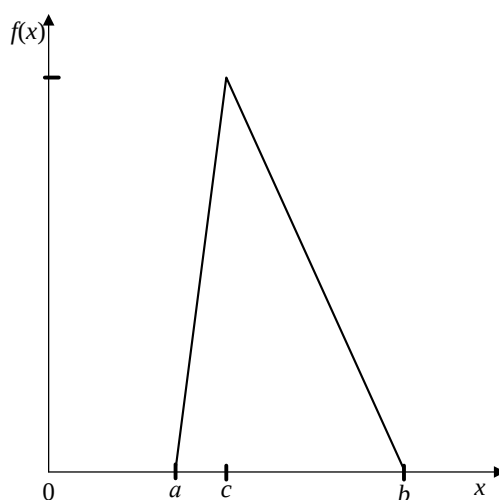


Рис. 2.8. Функция плотности распределения вероятностей треугольного распределения

Треугольное распределение часто используется при моделировании случайных явлений при отсутствии достаточных данных, позволяющих сформулировать гипотезу об ином распределении. Однако использование треугольного распределения ограничивает исследователя невозможностью независимого варьирования такими параметрами как мода, медиана, математическое ожидание [13].

2.3. Выходные данные и стохастические процессы моделирования

В большинстве имитационных моделей в качестве входных данных используются случайные величины, поэтому выходные данные имитационного моделирования также носят случайный характер. В связи с этим нужно осторожно делать выводы относительно действительных характеристик модели (например, об ожидаемой средней задержке требований в системе массового обслуживания). Правильное проведение анализа выходных данных невозможно без ознакомления со стохастическими процессами.

Стохастический процесс представляет совокупность «однородных» случайных величин, которые упорядочены во времени и определены в общем, выборочном пространстве. Множество всех возможных значений, которые могут принимать эти случайные величины, называется *пространством состояний*. Если совокупность величин представлена как $X_1, X_2, \dots$, то речь идет о *дискретном* стохастическом процессе, если же как $\{X(t), t \geq 0\}$, то о *непрерывном* [17].

Иногда, для того чтобы сделать статистические выводы о стохастическом процессе по совокупности выходных данных, необходимо принять относительно него допущения, которые на практике не являются строго справедливыми (без таких допущений статистический анализ выходных данных может оказаться невыполнимым). Например, можно допустить, что стохастический процесс является ковариационно стационарным. Дискретный стохастический процесс $X_1, X_2, \dots$ считается *ковариационно стационарным*, если

$$\begin{aligned}\mu_i &= \mu && \text{для } i = 1, 2, \dots \text{ и } -\infty < \mu < \infty; \\ \sigma_i^2 &= \sigma^2 && \text{для } i = 1, 2, \dots \text{ и } \sigma_i^2 < \infty,\end{aligned}$$

а $C_{i,i+j} = \text{Cov}(X_i, X_{i+j})$ не зависит от i для $j = 1, 2, \dots$

Таким образом, для ковариационно стационарного процесса среднее значение и дисперсия являются неизменными в течение всего времени (общее среднее значение и общая дисперсия обозначаются соот-

ветственно как μ и σ^2), а ковариация между двумя наблюдениями X_i и X_j зависит только от интервала j , а не от действительных значений времени i и $i + j$.

Для ковариационно стационарного процесса ковариация и корреляция между X_i и X_{i+j} обозначаются соответственно как C_j и p_j и имеют такую зависимость:

$$p_i = \frac{C_{i,i+j}}{\sqrt{\sigma_i^2 \sigma_{i+j}^2}} = \frac{C_j}{\sigma^2} = \frac{C_j}{C_0} \text{ для } j = 1, 2, \dots$$

Если $X_1, X_2, \dots$ – стохастический процесс, начавшийся в момент времени 0, вполне вероятно, что такой процесс *не* является ковариационно стационарным. Однако в *некоторых* имитационных моделях процесс $X_{k+1}, X_{k+2}, \dots$ будет приблизительно ковариационно стационарным, если k имеет достаточно большое значение; k – это продолжительность *переходного периода* работы системы [6].

2.4. Планирование экспериментов

Часто возникают ситуации, в которых требуется определить, какие из множества параметров и структурных допущений имеют наибольшее влияние на показатели работы или какой набор параметров модели позволяет получить оптимальную характеристику. В таких ситуациях, при которых цель исследования системы посредством моделирования менее структурирована, весьма эффективным оказывается предварительное планирование эксперимента.

В терминологии планирования экспериментов входные переменные и структурные допущения, составляющие модель, называются *факторами*, а выходные показатели работы – *откликами* [6]. Решение о том, какие параметры и структурные допущения считать фиксированными показателями модели, а какие – экспериментальными факторами, зависит скорее от целей исследования, а не от внутреннего вида модели. Кроме того, при имитационных исследованиях часто интерес представляют несколько различных откликов или показателей работы.

Факторы могут быть либо *количественными*, либо *качественными*. Количественные факторы, как правило, предполагают численные значения, тогда как качественные факторы обычно являют собой структурные допущения, которые не измеряются количественно. При этом у некоторых факторов различие может быть не ясно.

При проведении имитационных экспериментов различают *управляемые* и *неуправляемые* факторы в зависимости от того, может ли руководство соответствующих реальных систем управлять ими.

Управляемые факторы имитационных экспериментов имеют наибольшее влияние на решения, которые могут быть приняты относительно реализации реальных систем. Тем не менее, неуправляемые факторы тоже представляют интерес при проведении имитационных экспериментов, поскольку может понадобиться оценить, каким образом резкое увеличение интенсивности прибытия заданий (клиентов и т.п.) повлияет на загруженность системы. В математическом моделировании (таком как выполняемое нами имитационное моделирование) доступно управление *всеми факторами*, независимо от их практической управляемости.

При моделировании перед выполнением прогонов модели *планирование экспериментов* позволяет решить, какие именно конфигурации следует создавать, чтобы получить нужную информацию при наименьшем объеме моделирования. Тщательно проработанные эксперименты гораздо более эффективны, чем выполняемая наугад последовательность прогонов, в ходе которых просто не систематически проверяется ряд альтернативных конфигураций, чтобы увидеть, что при них происходит.

Планирование особенно эффективно на ранних этапах экспериментов, когда еще не ясно, какие факторы важны и как они могут влиять на отклики.

Имитационные эксперименты имеют ряд особенностей, благодаря которым выгодно отличаются от обычных физических экспериментов, проводимых в науке, промышленности или сельском хозяйстве и традиционно используемых в качестве примеров в литературе по планированию экспериментов.

При моделировании имеется возможность управлять такими факторами, как интенсивность прибытия клиентов, которые в реальности контролировать нельзя. Таким образом, можно исследовать гораздо больше видов случайностей, чем при проведении физических опытов с системой.

Другой аспект расширения возможностей управления моделирующими экспериментами связан с детерминированным характером генераторов случайных чисел. При моделировании можно контролировать сам источник изменчивости в отличие от физических экспериментов, результат которых зависит от реальных обстоятельств [6].

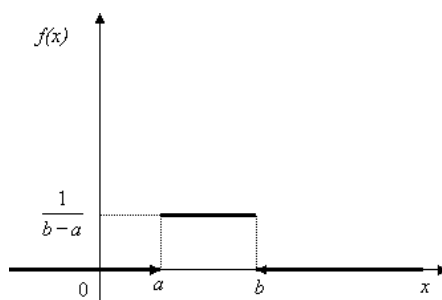
2.5. Выводы по главе

В данной главе рассмотрены основы теории вероятностей и математической статистики. Изучены базовые термины и понятия, используемые при построении моделей, а также рассмотрен такой важный аспект моделирования, как предварительное планирование эксперимента.

Таким образом, в первой и второй главе пособия изучены основные понятия модели и моделирования, рассмотрены существующие виды моделей, приведены базовые понятия теории вероятностей и математической статистики. В следующей главе перейдем к рассмотрению непосредственно моделирования вычислительных сетей.

2.6. Вопросы по главе*

1. Что такое эксперимент?
2. Какие законы распределения случайных величин вам известны?
3. Перечислите основные параметры равномерного распределения.
4. Самостоятельно исследуйте треугольное распределение случайных величин. В чем разница равномерного и треугольного распределения?
5. В чем состоит отличие дискретной и непрерывной случайной величины X ?
6. Что такое дисперсия случайной величины X ?
7. Что такое количественные и качественные факторы эксперимента?
8. Какой закон распределения чаще всего используют при описании работы систем массового обслуживания?
9. Что за график приведен ниже? Объясните его параметры.



10. Что описывает нижеприведенная формула? Поясните основные ее параметры.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}}.$$

* Данные вопросы могут быть использованы для учебной дискуссии.

3. МОДЕЛИРОВАНИЕ КОМПЬЮТЕРНЫХ СЕТЕЙ

Ни один проект крупной сети со сложной топологией в настоящее время не обходится без исчерпывающего моделирования будущей сети. Программы, выполняющие эту задачу, достаточно сложны и дороги. Целью моделирования является определение оптимальной топологии, адекватный выбор сетевого оборудования, определение рабочих характеристик сети и возможных этапов будущего развития. Необходимо помнить, что сеть, слишком точно оптимизированная для решений задач текущего момента, может потребовать серьезных переделок в будущем. При этом на модели можно опробовать влияние всплесков широковебательных запросов или реализовать режим коллапса (для Ethernet), что вряд ли можно применить в работающей сети.

В процессе моделирования выясняются следующие параметры:

- предельные пропускные способности различных фрагментов сети и зависимости потерь пакетов от загрузки отдельных станций и внешних каналов;
- время отклика основных серверов в разных режимах, в том числе таких, которые в реальной сети крайне нежелательны;
- влияние установки новых серверов на перераспределение информационных потоков;
- решение оптимизации топологии при возникновении узких мест в сети (размещение серверов, внешних шлюзов, организация опорных каналов и пр.);
- выбор того или иного типа сетевого оборудования (например, 10BaseTX или 100BaseFX) или режима его работы (например, cut-through, store-and-forward для мостов и переключателей и т.д.);
- выбор внутреннего протокола маршрутизации и его параметров (например, метрики);
- определение предельно допустимого числа пользователей того или иного сервера;
- оценка влияния мультимедийного трафика на работу локальной сети, например, при подготовке видеоконференций.

Результаты моделирования должны иметь точность 10–15 % (см. п. 1.1 гл. 1), т.к. этого достаточно для большинства целей и не требует слишком большого количества машинного времени. Для моделирования поведения реальной сети необходимо знать все ее рабочие параметры. Чем точнее будет воспроизведено поведение сети, тем больше машинного времени это потребует. Кроме того, необходимо сделать некоторые предположения относительно распределения загрузки для конкрет-

ных ЭВМ и других сетевых элементов, задержек в переключателях, мостах, времени обработки запросов в серверах. Здесь нужно учитывать и характер решаемых на ЭВМ задач.

В зависимости от целей и задач моделирования сетей выделяют моделирование аналитическое и имитационное.

3.1. Аналитическое моделирование

Использование аналитических методов связано с необходимостью построения математических моделей локальных вычислительных сетей (ЛВС) в строгих математических терминах. Аналитические модели вычислительных сетей носят обычно вероятностный характер и строятся на основе понятий аппарата теорий массового обслуживания, вероятностей и марковских процессов, а также методов диффузной аппроксимации. Могут также применяться дифференциальные и алгебраические уравнения [12].

При использовании этого математического аппарата часто удастся быстро получить аналитические модели для решения достаточно широкого круга задач исследования ЛВС. В то же время аналитические модели имеют ряд существенных недостатков, к числу которых следует отнести:

- значительные упрощения, свойственные большинству аналитических моделей (представление потоков заявок как простейших; предположение об экспоненциальном распределении длительностей обслуживания заявок; невозможность обслуживания заявок одновременно несколькими приборами, например процессором и оперативной памятью, и др.). Подобные упрощения, а зачастую искусственное приспособление аналитических моделей с целью использования хорошо разработанного математического аппарата для исследования реальных ЛВС, ставят иногда под сомнение результаты аналитического моделирования [19];

- громоздкость вычислений для сложных моделей, например использование для представления в модели процесса функционирования современной ЛВС по методу дифференциальных уравнений Колмогорова требует (для установившегося режима) решения сложной системы алгебраических уравнений [19];

- сложность аналитического описания вычислительных процессов ЛВС. Большинство известных аналитических моделей можно рассматривать лишь как попытку подхода к описанию процессов функционирования ЛВС;

- недостаточная развитость аналитического аппарата в ряде случаев не позволяет в аналитических моделях выбирать для исследования

наиболее важные характеристики (показатели эффективности) ЛВС. Особенно большие затруднения при аналитическом моделировании связаны с учетом в процессах функционирования ЛВС программных средств операционных систем и другого общего ПО [19].

Указанные особенности позволяют заключить, что аналитические методы имеют самостоятельное значение лишь при исследовании процессов функционирования ЛВС в первом приближении и в частных, достаточно специфичных задачах. В этих случаях возможности исследования аналитических моделей ЛВС существенно расширяют приближенные методы, например: методы диффузионной аппроксимации, методы операционного анализа и аналитические сетевые модели.

3.1.1. Аналитическое моделирование на основе систем массового обслуживания

При аналитическом моделировании исследование процессов или объектов заменяется построением их математических моделей и исследованием этих моделей. В основу метода положены идентичность формы уравнений и однозначность соотношений между переменными в уравнениях, описывающих оригинал и модель. Поскольку события, происходящие в локальных вычислительных сетях, носят случайный характер, то для их изучения наиболее подходящими являются вероятностные математические модели теории массового обслуживания [5].

Аналитическая модель сети представляет собой совокупность математических соотношений, связывающих между собой входные и выходные характеристики сети. При выводе таких соотношений приходится пренебрегать какими-то малосущественными деталями или обстоятельствами [8].

Телекоммуникационная сеть при некотором упрощении может быть представлена в виде совокупности процессоров (узлов), соединенных каналами связи. Сообщение, пришедшее в узел, ждет некоторое время до того, как оно будет обработано. При этом может образоваться очередь таких сообщений, ожидающих обработки. Время передачи или полное время задержки сообщения D равно:

$$D = T_p + S + W, \quad (3.1)$$

где T_p , S и W , соответственно, время распространения, время обслуживания и время ожидания. Одной из задач аналитического моделирования является определение среднего значения D . При больших нагрузках основной вклад дает ожидание обслуживания W . Для описания очередей в дальнейшем будет использована нотация Д. Дж. Кенделла:

$$A/B/C/K/m/z,$$

где A – процесс прибытия;

B – процесс обслуживания;

C – число серверов (узлов);

K – максимальный размер очереди (по умолчанию – ∞);

m – число клиентов (по умолчанию – ∞);

z – схема работы буфера (по умолчанию FIFO).

Буквы A и B представляют процессы прихода и обслуживания и обычно заменяются следующими буквами, характеризующими закон, соответствующий распределению событий:

D – постоянная вероятность;

M – марковское экспоненциальное распределение;

G – обобщенный закон распределения;

E_k – распределение Эрланга порядка k ;

H_k – гиперэкспоненциальное распределение порядка k [4].

Наиболее распространенными схемами работы буферов являются FIFO (First-In-First-Out), LIFO (Last-In-First-Out) и FIRO (First-In-Random-Out). Например, запись $M/M/2$ означает очередь, для которой времена прихода и обслуживания имеют экспоненциальное распределение, имеется два сервера, длина очереди и число клиентов могут быть сколь угодно большими, а буфер работает по схеме FIFO [19].

Среднее значение длины очереди Q при заданной средней входной частоте сообщений λ и среднем времени ожидания W определяется на основе теоремы Литла (1961) [19]:

$$\bar{Q} = \lambda \bar{W}. \quad (3.2)$$

Для варианта очереди $M/G/1$ входной процесс характеризуется распределением Пуассона со скоростью поступления сообщений λ . Вероятность поступления k сообщений на вход за время t равно:

$$P(k) = \frac{(\lambda t)^k}{k!} = e^{-\lambda t}, \quad k = 0, 1, 2, \dots \quad (3.3)$$

Пусть N – число клиентов в системе, Q – число клиентов в очереди и пусть вероятность того, что входящий клиент обнаружит j других клиентов, равна:

$$\Pi_j = P[n = j], \quad j = 0, 1, 2, \dots$$

$$\sum_{j=0}^{\infty} \Pi_j = 1.$$

$$\Pi_0 = 1 - r; r = lt. \quad (3.4)$$

Тогда среднее время ожидания $\bar{W}$:

$$\bar{W} = \frac{\bar{Q}}{\lambda} = \frac{pt}{2(1-p)} \left(1 + \frac{\sigma^2}{\tau^2} \right) \text{ (формула Поллажека – Хинчина)} \quad (3.5)$$

s – среднеквадратичное отклонение для распределения времени обслуживания.

Для варианта очереди $M/G/1$ $H(t) = P[X \leq t] = 1 - e^{-mt}$ (H – функция распределения времени обслуживания). Откуда следует $s^2 = t^2$ [19]

$$\bar{W} = \frac{p\tau}{1-p}. \quad (3.6)$$

Для варианта очереди $M/D/1$ время обслуживания постоянно, а среднее время ожидания составляет:

$$\bar{W} = \frac{p\tau}{2(1-p)}. \quad (3.7)$$

Рассмотрим вариант сети Ethernet на основе концентратора-переключателя с числом каналов N . При этом будет предполагаться, что сообщения на входе всех узлов имеют пуассоновское распределение со средней интенсивностью l_i , распределение сообщений по длине произвольно. Сообщения отправляются в том же порядке, в котором они прибыли. Трафик в сети предполагается симметричным. Очередь имеет модель $M/G/1$. Среднее время ожидания в этом случае равно:

$$\bar{W} = \hat{y} + \frac{\lambda \hat{y}^2}{1(1-p)},$$

где $\hat{y} = [1 + (N-2)pG] \bar{S}$;

$$\hat{y} = 2[1 + (N-2)pG + (N-2)(N-3)p^2G^2] \bar{S}^2. \quad (3.8)$$

$$p = \frac{\lambda \bar{S}}{1 - (N-2)G\lambda \bar{S}}. \quad (3.9)$$

$\lambda = \lambda_i$, а $G = 1/(N-1)$ равно вероятности того, что сообщение отправителя i направлено получателю j . Требование стабильности требует, чтобы $\lambda \bar{S} \leq (N-1)(2N-3)$. Для больших n это приводит к $\lambda \bar{S} \leq 1/2$.

Работа сети Ethernet характеризуется рядом параметров, к числу которых относятся вероятность захвата канала и эффективность [19].

Первый параметр определяется по выражению

$$P = \left(1 - \frac{1}{Q}\right)^{Q-1}, \quad (3.10)$$

где P – вероятность того, что ровно одна станция попытается передать кадр в течение такта и захватит канал;

Q – число станций, пытающихся захватить канал для передачи кадра данных.

Эффективность LAN Ethernet определяется следующим образом. Общее время работы сети Ethernet делится между интервалами передачи и интервалами конкуренции. Для передачи кадра данных требуется L/C секунд, где L – длина кадра в битах, C – скорость передачи данных в бит/сек. Среднее время T , необходимое на захват канала, равно:

$$T = W \cdot B, \quad (3.11)$$

где W – среднее число тактов, прошедших в интервале конкуренции, пока станция не захватит канал для передачи кадра данных;

B – длительность такта или время до обнаружения конфликта после начала передачи кадра.

Среднее число тактов W рассчитывается как:

$$W = \frac{1 - P}{P}. \quad (3.12)$$

С учетом введенных показателей, эффективность E работы локальной сети Ethernet определяется следующим образом:

$$E = \frac{L/C}{L/C + T}. \quad (3.13)$$

Для моделирования ЛВС наиболее часто используются следующие типы СМО:

1. Одноканальные СМО с ожиданием. Представляют собой один обслуживающий прибор с бесконечной очередью. Данная СМО является наиболее распространенной при моделировании. С той или иной долей приближения с ее помощью можно моделировать практически любой узел ЛВС.

2. Одноканальные СМО с потерями. Представляют собой один обслуживающий прибор с конечным числом мест в очереди. Если число

заявок превышает число мест в очереди, то лишние заявки теряются. Этот тип СМО может быть использован при моделировании каналов передачи в ЛВС.

3. Многоканальные СМО с ожиданием. Представляют собой несколько параллельно работающих обслуживающих приборов с общей бесконечной очередью. Данный тип СМО часто используется при моделировании групп абонентских терминалов ЛВС, работающих в диалоговом режиме.

4. Многоканальные СМО с потерями. Представляют собой несколько параллельно работающих обслуживающих приборов с общей очередью, число мест в которой ограничено. Эти СМО, как и одноканальные с потерями, часто используются для моделирования каналов связи в ЛВС.

5. Одноканальные СМО с групповым поступлением заявок. Представляют собой один обслуживающий прибор с бесконечной очередью. Перед обслуживанием заявки группируются в пакеты по определенному правилу.

6. Одноканальные СМО с групповым обслуживанием заявок. Представляют собой один обслуживающий прибор с бесконечной очередью.

Заявки обслуживаются пакетами, составляемыми по определенному правилу. Последние два типа СМО могут использоваться для моделирования таких узлов ЛВС, как центры (узлы) коммутации.

Локальная вычислительная сеть в целом может быть представлена в виде сети массового обслуживания. Различают *открытые*, *замкнутые* и *смешанные* сети.

Открытой называется сеть массового обслуживания, состоящая из M узлов, причем хотя бы в один из узлов сети поступает извне входящий поток заявок, и имеется сток заявок из сети. Для открытых сетей характерно то, что интенсивность поступления заявок в сеть не зависит от состояния сети, т.е. от числа заявок, уже поступивших в сеть. Открытые сети используются для моделирования ЛВС, работающих в неоперативном режиме. Каждая заявка поступает на вход соответствующего узла коммутации, где определяется место ее обработки. Затем заявка передается на «свой» сервер или по каналу связи – на «соседний» сервер, где обрабатывается, после чего возвращается к источнику и покидает сеть [19].

Замкнутой называется сеть массового обслуживания с множеством узлов M без источника и стока, в которой циркулирует постоянное число заявок. Замкнутые СМО используются для моделирования таких ЛВС, источниками информации для которых служат абонентские терминалы, работающие в диалоговом режиме. В этом случае каждая груп-

па абонентских терминалов представляется в виде многоканальной системы массового обслуживания с ожиданием и включается в состав устройств сети [19].

Различают *простой* и *сложный* режимы работы диалоговых абонентов. В простом режиме абоненты не производят никаких действий, кроме отправки заданий в ЛВС и обдумывания полученного ответа [19].

Абоненты с терминалов посылают запросы, которые по каналам связи поступают на узлы коммутации, а оттуда – на обработку на «свой» или «соседний» сервер. Дальнейшая обработка осуществляется так же, как в открытой сети [14].

При сложном режиме диалога работа абонентов представляется в виде совокупности операций некоего процесса, называемого технологическим процессом. Каждая операция технологического процесса моделируется соответствующей СМО. Часть операций предусматривает обращение к ЛВС, а часть операций может такого обращения не предусматривать [19].

Алгоритм работы самой ЛВС такой же, как для замкнутой сети.

Смешанной называется сеть массового обслуживания, в которой циркулирует несколько различных типов заявок (трафика), причем относительно одних типов заявок сеть замкнута, а относительно других типов заявок сеть открыта. С помощью смешанных СМО моделируются такие ЛВС, часть абонентов, которых работает в диалоговом, а часть – в неоперативном режиме. Для диалоговых абонентов также различают простой и сложный режим работы. Часто смешанные СМО моделируют ЛВС, в которых сервер дополнительно загружается задачами, решаемыми на фоне работы самой сети [19].

Алгоритм работы сети для диалоговых абонентов аналогичен алгоритму работы замкнутой сети, а алгоритм работы сети для неоперативных абонентов – алгоритму работы открытой сети.

Различают экспоненциальные и неэкспоненциальные модели ЛВС.

Экспоненциальные модели основаны на предположении о том, что потоки заявок, поступающие в ЛВС, являются пуассоновскими, а время обслуживания в узлах ЛВС имеет экспоненциальное распределение.

Для таких сетей получены точные методы для определения их характеристик; трудоемкость получения решения зависит в основном от размерности сети [19].

Однако в большинстве сетей (и локальных сетей в частности) потоки не являются пуассоновскими. Модели таких сетей называются *неэкспоненциальными*. При анализе неэкспоненциальных сетей в общем случае отсутствуют точные решения, поэтому наибольшее применение здесь находят приближенные методы.

Одним из таких методов является *метод диффузионной аппроксимации*. Использование диффузионной аппроксимации позволило, к настоящему времени получить приближенные аналитические зависимости для определения характеристик всех типов СМО, рассмотренных выше.

При этом не требуется точного знания функций распределения случайных величин, связанных с данной СМО (интервалов между поступлениями заявок временем обслуживания в приборах), а достаточно только знание первого (математического ожидания) и второго (дисперсии или квадрата коэффициента вариации – ККВ) моментов этих величин [19].

Применение диффузионной аппроксимации при анализе ЛВС основано на следующем:

- по каждому типу заявок вычисляется интенсивность поступления заявок данного типа в узлы сети так, как если бы данный поток заявок циркулировал в сети только один;
- по определенному правилу, зависящему от типа СМО и дисциплины обслуживания, складываются потоки заявок от всех источников;
- по определенному правилу определяется среднее время обслуживания в каждом узле ЛВС;
- полученные значения подставляются в соответствующую диффузионную формулу и определяются характеристики узлов ЛВС;
- определяются характеристики ЛВС в целом.

Постановка задачи анализа ЛВС при этом примет следующий вид. Дано:

- число узлов ЛВС;
- тип каждого узла ЛВС (тип СМО, моделирующей данный узел);
- дисциплина обслуживания в каждом узле ЛВС;
- общее число типов источников заявок, работающих в диалоговом режиме;
- общее число типов источников заявок, работающих в неоперативном режиме;
- для диалоговых источников в случае сложного режима работы число технологических процессов каждого типа, число операций в каждом технологическом процессе, среднее и ККВ времени выполнения каждой операции, матрица вероятностей передач между операциями, а также наличие или отсутствие на каждой операции обращения к ЛВС;
- для диалоговых источников в случае простого режима работы число источников (терминалов) каждого типа, среднее и ККВ времени реакции абонента на ответ сети;

- для неоперативных абонентов – средняя интенсивность поступления заявок и ККВ времени между поступлениями заявок; по каждому типу заявок (диалоговому и неоперативному) средняя интенсивность обслуживания в каждом узле ЛВС, ККВ времени обслуживания в узлах ЛВС и матрица вероятностей передач между узлами.

Требуется найти:

- среднее значение и дисперсию (или стандартное отклонение) времени задержки заявки каждого типа в ЛВС в целом;
- среднее значение и дисперсию (или стандартное отклонение) времени задержки в узлах ЛВС;
- загрузку узлов ЛВС;
- вероятность потери заявки в узле ЛВС (для узлов, моделируемых СМО с потерями).

Ограничения могут быть следующими:

- загрузка узлов не должна превышать 1;
- вероятность потери заявки не должна превышать 1;
- все характеристики должны быть положительны.

Иногда представляет интерес определение такого показателя, как максимальное время задержки заявки каждого типа в ЛВС. Максимальное время это такое время, превышение которого допустимо лишь для некоторого, наперед заданного процента заявок каждого типа. Для определения максимального времени используется методика, основанная на аппроксимации функции распределения времени задержки в сети эрланговским или гиперэкспоненциальным распределением, при этом необходимо задавать долю (процент) заявок, для которых рассчитывается максимальное время.

3.1.2. Сети Петри

Часто аналитики в задачах моделирования и анализа сложных параллельных и асинхронных систем обращаются к формальным системам, основанным на использовании математического аппарата сетей Петри. Формальная часть теории сетей Петри, основанная в начале 60-х годов немецким математиком Карлом Петри, в настоящее время содержит большое количество моделей, методов и средств анализа, имеющих обширное количество приложений практически во всех отраслях вычислительной техники [14].

Прикладная теория сетей Петри связана главным образом с применением сетей Петри к моделированию систем, их анализу и получающимся в результате этого глубоким проникновением в моделируемые системы [14].

Моделирование в сетях Петри осуществляется на событийном уровне. Определяются, какие действия происходят в системе, какие состояния предшествовали этим действиям и какие состояния примет система после выполнения действия. Выполнение событийной модели в сетях Петри описывает поведение системы. Анализ результатов выполнения может сказать о том, в каких состояниях пребывала или не пребывала система, какие состояния в принципе не достижимы. Однако такой анализ не дает числовых характеристик, определяющих состояние системы. Развитие теории сетей Петри привело к появлению так называемых «цветных» или «раскрашенных» сетей Петри. Понятие цветности в них тесно связано с понятиями переменных, типов данных, условий и других конструкций, более приближенных к языкам программирования.

Таким образом, структура сети Петри задается ориентированным двудольным мультиграфом, в котором одно множество вершин состоит из позиций, а другое множество – из переходов [14], причем множество вершин этого графа разбивается на два подмножества и не существует дуги, соединяющей две вершины из одного подмножества.

Итак, сеть Петри – это набор

$$N = (T, P, A), T \cap P = \emptyset, \quad (3.14)$$

где $T = \{t_1, t_2, \dots, t_n\}$ – подмножество вершин, называемых переходами;

$P = \{p_1, p_2, \dots, p_m\}$ – подмножество вершин, называемых позициями (местами);

$A \subseteq (T \times P) \cup (P \times T)$ – множество ориентированных дуг.

В сетях Петри вводятся объекты двух типов: динамические – изображаются метками (маркерами) внутри позиций и статические – им соответствуют вершины сети Петри.

Распределение маркеров по позициям называют маркировкой. Маркеры могут перемещаться в сети. Каждое изменение маркировки называют событием, причем каждое событие связано с определенным переходом. Считается, что события происходят мгновенно и одновременно при выполнении некоторых условий.

Каждому условию в сети Петри соответствует определенная позиция. Совершению события соответствует срабатывание (возбуждение или запуск) перехода, при котором маркеры из входных позиций этого перехода перемещаются в выходные позиции. Последовательность событий образует моделируемый процесс. Перемещаемые по сети маркеры часто называют фишками. Основные элементы сети Петри представлены в табл. 3.1.

Таблица 3.1

Элементы сетей Петри

Название элемента	Изображение элемента
Позиция (P)	
Переход (T)	
Дуга	

Переходы в сети Петри являются событиями, которые изменяют состояния в реальной системе. На рис. 3.1 приведен пример интерпретации сети Петри.

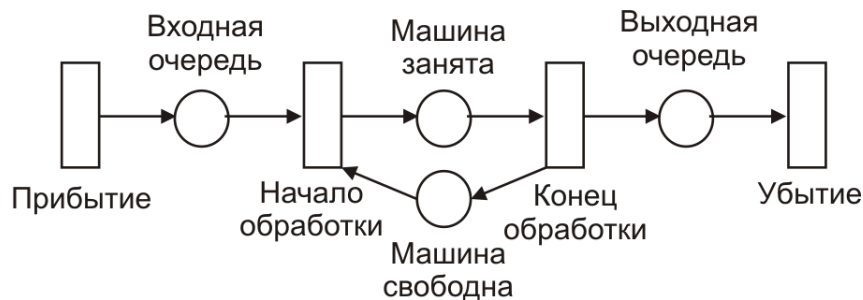


Рис. 3.1. Интерпретация сети Петри

Формальный аппарат сетей Петри предназначен для моделирования систем различного рода и отражает состояния исследуемой системы состоянием сети. Состояние сети Петри определяется ее маркировкой. Количество и распределение фишек сети определяют динамику исследуемой системы. Сеть Петри выполняется посредством запусков переходов в результате удаления фишек из его входных позиций и добавления их в выходные позиции перехода. Последовательность срабатываний переходов полностью определяет поведение сети. Таким образом, сеть Петри описывает структуру системы, ее состояние и поведение [14].

При введении ряда дополнительных правил и условий в алгоритмы моделирования получают различные разновидности сетей Петри. Это необходимо для определения модельного времени, которое позволит моделировать не только последовательность событий, но и их привязку ко времени. В настоящее время выделяют следующие разновидности сетей Петри:

- временная сеть Петри (переходы обладают весом, определяющим продолжительность срабатывания – задержку);
- стохастическая сеть Петри (задержки являются случайными величинами);
- функциональная сеть Петри (задержки определяются как функции некоторых аргументов, например: количества меток в каких-либо позициях, состояния некоторых переходов);
- цветная (раскрашенная) сеть Петри (метки могут быть различных типов, обозначаемых цветами, тип метки может быть использован как аргумент в функциональных сетях).

Основными свойствами сети Петри являются:

- ограниченность (число меток в любой позиции сети не может превысить некоторого значения K);
- безопасность (частный случай ограниченности);
- сохраняемость (постоянство загрузки ресурсов);
- достижимость (возможность перехода сети из одного заданного состояния, характеризуемого распределением меток, в другое);
- живость (возможностью срабатывания любого перехода при функционировании моделируемого объекта).

Среди достоинств аппарата сетей Петри можно указать следующие:

- позволяет моделировать асинхронность и недетерминизм параллельных независимых событий (в сети Петри могут одновременно и независимо друг от друга сработать несколько переходов), конфликтные взаимодействия между процессами;
- позволяет использовать единые методологические позиции для описания программного обеспечения, аппаратных средств и информационного обмена между системами;
- предоставляет возможность введения любой степени иерархической детализации описываемых программных и аппаратных подсистем модели;
- имеет большую анализирующую мощност, которая позволяет формальными средствами доказывать существование или отсутствие определенных состояний сети Петри.

Однако формальная модель сетей Петри, в силу своей универсальности, имеет ряд недостатков, затрудняющих практическое применение для моделирования сложных систем. К основным таким недостаткам можно отнести следующие:

- высокая трудоемкость анализа сетей большой размерности, а реальные бизнес-процессы предприятия моделируются именно сетями большой размерности;

- описательная мощность сетей Петри недостаточна для содержательного моделирования систем;
- обычные сети Петри не отражают требуемые временные характеристики моделируемой системы;
- фишка сети Петри не представляет собой никакой информации, кроме самого факта ее наличия, поэтому чрезвычайно сложно отразить преобразование информации при срабатывании переходов сети Петри;
- невозможность проведения логических преобразований и, как следствие, невозможность управления продвижением фишек по сети.

Недостатки сетей Петри не позволяют описывать сложные системы и в настоящее время используются для описания простейших операций. Также эти факторы явились причиной разработки подклассов и расширений сетей Петри, в которых вводятся определенные ограничения на структуру сети, что позволяет использовать более простые алгоритмы для ее анализа либо дополнительные элементы формальной системы, призванные увеличить ее описательную мощность.

Большого внимания заслуживают сети высокого уровня, такие как *раскрашенные сети Петри* (Color Petri Net), являющиеся модификацией сетей Петри и отличающиеся хорошо разработанным математическим аппаратом, широко применяемые для самых разнообразных практических целей. Основной причиной высокой эффективности этих формальных моделей является то, что они без потери возможностей формального анализа позволяют исследователю получить значительно более краткие и удобные описания, чем те, которые могут быть сделаны с помощью сетей низкого уровня. В сетях высокого уровня сложность моделей может быть разделена между структурой сети, надписями и описаниями. Это позволяет осуществлять описание значительно более сложных систем и анализировать процессы преобразования данных с помощью общепринятых математических выражений вместо сложного набора позиций, переходов и дуг. Раскрашенные сети Петри, в отличие от обычных сетей Петри, позволяют описывать структуру системы в виде иерархии диаграмм.

Но у данного аппарата моделирования также не устранен ряд недостатков, которые присущи сетям Петри. К таким недостаткам можно отнести:

- необходимость знания разработчиком специфического языка описания моделей [6];
- отсутствие использования принципов объектно-ориентированного подхода;

- низкую гибкость и трудоемкость описания систем в случае их декомпозиции до уровня некоторых элементарных операций.

Раскрашенные сети Петри до сих пор часто применяются для моделирования сетей.

3.2. Имитационное моделирование

Несмотря на большие достижения математического моделирования, многие реальные ситуации невозможно адекватно представить с помощью соответствующих математических моделей. В одних случаях этому мешает определенная «жесткость» математики как языка описания и представления событий и явлений. Кроме того, даже если есть возможность формализовать рассматриваемую жизненную ситуацию посредством построения математической модели, полученная на ее основе задача оптимизации может быть слишком сложной для современных алгоритмов решения задач этого класса.

Альтернативой математическому моделированию сложных систем может служить **имитационное моделирование (ИМ)**. Этот вид моделирования часто является наилучшим (если не единственным) способом исследования реальных систем.

Термин «*имитационное моделирование*» означает, что мы имеем дело с такими моделями, с помощью которых нельзя заранее вычислить или предсказать поведение системы, а для предсказания поведения системы необходим вычислительный эксперимент (имитация) на математической модели при заданных исходных данных [16].

Различие между математической и имитационной моделями заключается в том, что в последней вместо явного математического описания взаимоотношения между входными и выходными переменными реальная система разбивается на ряд достаточно малых (в функциональном отношении) элементов или модулей. Затем поведение исходной системы имитируется как поведение совокупности этих элементов, определенным образом связанных (путем установления соответствующих взаимосвязей между ними) в единое целое. Вычислительная реализация такой модели начинается с входного элемента, далее проходит по всем элементам, пока не будет достигнут выходной элемент модели.

Имитационные модели принято классифицировать по следующим наиболее распространенным признакам:

- по способу взаимодействия с пользователем;
- способу изменения модельного времени;
- цели эксперимента.

Классификация имитационных моделей схематично показана на рис. 3.2.

По способу взаимодействия с пользователем имитационные модели могут быть *автономными* и *интерактивными*. Автономные модели не требуют вмешательства исследователя после определения режима моделирования и задания исходных данных, взаимодействие пользователя с такими моделями сводится только к вводу исходной информации и управлению началом и окончанием работы моделей. Интерактивные модели предусматривают диалог с пользователем в том или ином режиме в соответствии со сценарием моделирования, позволяющие пользователю приостанавливать сеанс моделирования/изменять значения параметров модели, корректировать перечень регистрируемых данных и т.д.

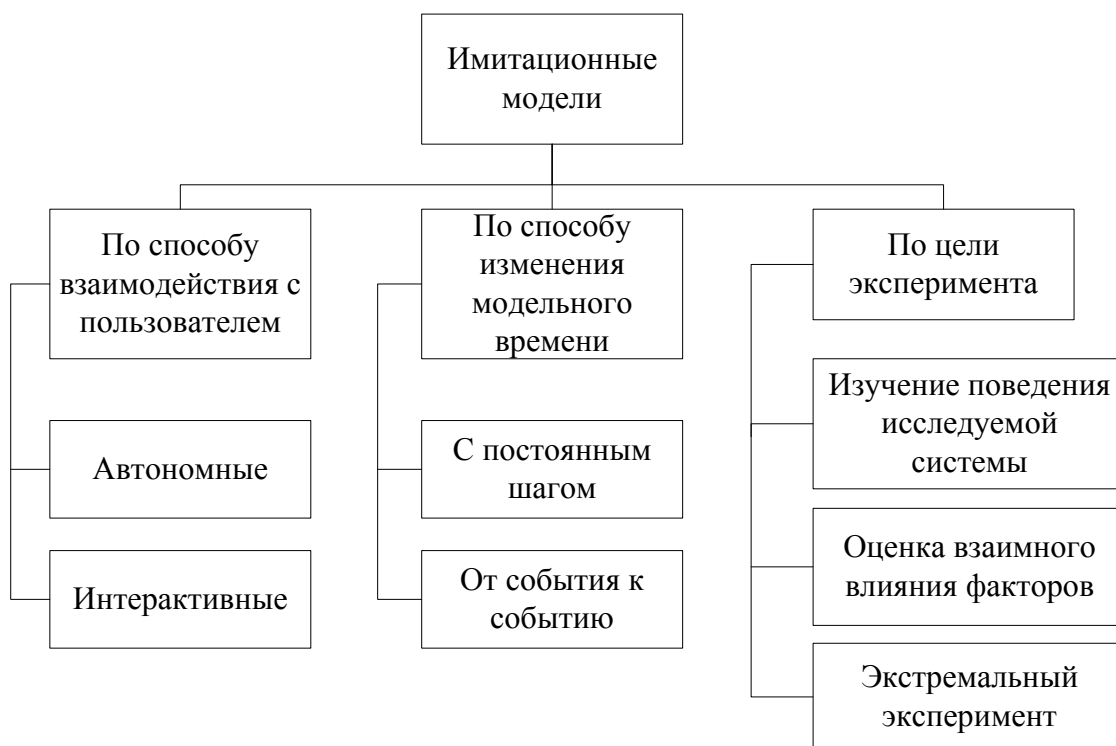


Рис. 3.2. Классификация имитационных моделей

Как было сказано ранее (п. 1.3 гл. 1) существует два механизма изменения модельного времени: *продвижение времени от события к событию* и *продвижение времени с постоянным шагом*.

Процесс построения имитационных моделей представляет собой последовательное выполнение этапов моделирования. Этапы имитационного моделирования, также как и любого другого вида моделирования, обобщены и представлены в п. 1.3. гл. 1 этого учебного пособия.

Названные выше этапы имитационного исследования редко выполняются в строго заданной последовательности, начиная с определения проблемы и заканчивая документированием. В ходе имитационного исследования могут быть сбои в прогонах модели, ошибочные допущения, от которых впоследствии приходится отказываться, т.е. на каждом этапе возможно возвращение назад, к предыдущим этапам. Именно такой итеративный процесс даёт возможность получить модель, которая позволяет принимать решения.

Вычислительные аспекты имитационных моделей обычно сравнительно несложные, но, как правило, очень трудоемкие. Поэтому реализация таких моделей подразумевает использование вычислительной техники.

Имитационные модели значительно гибче в представлении реальных систем, чем их математические «конкуренты». Причина такой гибкости заключается в том, что при имитационном моделировании исходная система рассматривается на элементарном уровне, в то время как математические модели стремятся описать системы на глобальном, как можно более общем уровне.

Но за гибкость имитационных моделей приходится платить высокими требованиями к потребляемым временным и вычислительным ресурсам. Поэтому реализация некоторых имитационных моделей даже на современных быстрых и высокопроизводительных компьютерах может быть очень медленной.

Таким образом, имитационное моделирование является мощным инструментом исследования поведения реальных систем. Методы имитационного моделирования позволяют собрать необходимую информацию о поведении системы путем создания ее компьютеризированной модели. Эта информация используется затем для проектирования системы.

Основное достоинство ИМ:

- возможность описания поведения компонент (элементов) процессов или систем на высоком уровне детализации;
- отсутствие ограничений между параметрами ИМ и состоянием внешней среды;
- возможность исследования динамики взаимодействия компонент во времени и пространстве параметров системы [6].

Эти достоинства обеспечивают имитационному методу широкое распространение.

Рекомендуется использовать *имитационное моделирование* 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 процессы сложны и трудоемки, *имитационное моделирование* дает более простой способ решения задачи.

3. Когда, кроме оценки влияния параметров (переменных) процесса или системы, желательно осуществить наблюдение за поведением компонент (элементов) процесса или системы в течение определенного периода.

4. Когда *имитационное моделирование* оказывается единственным способом исследования сложной системы из-за невозможности наблюдения явлений в реальных условиях (реакции термоядерного синтеза, исследования космического пространства).

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

6. При подготовке специалистов для новой техники, когда на *имитационных моделях* обеспечивается возможность приобретения навыков в эксплуатации новой техники.

7. Когда изучаются новые ситуации в поведении реальных процессов и систем. В этом случае имитация служит для проверки новых стратегий и правил проведения натурных экспериментов.

8. Когда особое значение имеет последовательность событий в проектируемых процессах и системах, и модель используется для предсказания узких мест в их функционировании [6].

Однако ИМ наряду с достоинствами имеет и недостатки:

- разработка хорошей ИМ часто обходится дороже создания аналитической модели и требует больших временных затрат;
- может оказаться, что ИМ неточна (что бывает часто), и исследователь не в состоянии измерить степень этой неточности;
- зачастую исследователи обращаются к ИМ, не представляя тех трудностей, с которыми они встретятся и совершают при этом ряд ошибок методологического характера [16].

И тем не менее, ИМ является одним из наиболее широко используемых методов при решении задач синтеза и анализа сложных процессов и систем.

Применительно к вычислительным сетям их имитационные модели воспроизводят: процессы генерации сообщений приложениями; разбиение сообщений на пакеты и кадры определенных протоколов; задержки, связанные с обработкой сообщений, пакетов и кадров внутри операци-

онной системы; процесс получения доступа компьютером к разделяемой сетевой среде; процесс обработки поступающих пакетов маршрутизатором и т.д. При имитационном моделировании сети не требуется приобретать дорогостоящее оборудование – его работы имитируются программами, достаточно точно воспроизводящими все основные особенности и параметры такого оборудования.

Преимуществом имитационных моделей является возможность подмены процесса смены событий в исследуемой системе в реальном масштабе времени на ускоренный процесс смены событий в темпе работы программы. В результате за несколько минут можно воспроизвести работу сети в течение нескольких дней, что дает возможность оценить работу сети в широком диапазоне варьируемых параметров.

Результатом работы имитационной модели являются собранные в ходе наблюдения за протекающими событиями статистические данные о наиболее важных характеристиках сети: временах реакции, коэффициентах использования каналов и узлов, вероятности потерь пакетов и т.п.

3.3. Выводы по главе

В данной главе рассмотрены основные понятия и виды моделирования вычислительных сетей, а именно два его основных типа: аналитическое и имитационное. Описаны достоинства и недостатки каждого типа моделирования, преимущества использования каждого типа моделирования для решения определенных задач.

Имитационное моделирование является более гибким в применении и предоставляет больше возможностей для моделирования вычислительных сетей, поэтому в дальнейшем будем рассматривать построение моделей в рамках имитационного моделирования.

В следующей главе рассмотрим наиболее распространенные программные средства моделирования вычислительных сетей.

3.4. Вопросы по главе*

1. Для чего необходимо моделирование сети?
2. Что такое аналитическое моделирование сети? Перечислите его достоинства и недостатки.
3. Что такое имитационное моделирование сети? Перечислите его достоинства и недостатки.
4. К какому типу моделирования сетей относятся СМО и сети Петри?

* Данные вопросы могут быть использованы для учебной дискуссии.

5. Перечислите основные структурные элементы СМО. Приведите примеры реальных устройств/компонент сети для каждого структурного элемента СМО.

6. Что в теории СМО означает запись $M/M/2$?

7. Что в СМО определяется по теореме Литла?

8. Какие типы СМО наиболее часто используются для моделирования ЛВС? Приведите примеры.

9. Что такое сеть Петри? Какие типы сетей Петри вам известны?

10. Перечислите основные элементы сети Петри и приведите примеры реальных устройств сети, соответствующим элементам сети Петри.

4. СРЕДСТВА МОДЕЛИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Существуют специальные, ориентированные на моделирование вычислительных сетей программные системы, в которых процесс создания модели упрощен. Такие программные системы сами генерируют модель сети на основе исходных данных о ее топологии и используемых протоколах, об интенсивностях потоков запросов между компьютерами сети, протяженности линий связи, о типах используемого оборудования и приложений. Программные системы моделирования могут быть узкоспециализированными и достаточно универсальными, позволяющими имитировать сети самых различных типов. Качество результатов моделирования в значительной степени зависит от точности исходных данных о сети, переданных в систему имитационного моделирования.

Программные системы моделирования сетей – инструмент, который может пригодиться любому администратору корпоративной сети, особенно при проектировании новой сети или внесении кардинальных изменений в уже существующую. Продукты данной категории позволяют проверить последствия внедрения тех или иных решений еще до оплаты приобретаемого оборудования. Конечно, большинство из этих программных пакетов стоят достаточно дорого, но и возможная экономия может быть тоже весьма ощутимой.

Программы имитационного моделирования сети используют в своей работе информацию о пространственном расположении сети, числе узлов, конфигурации связей, скоростях передачи данных, используемых протоколах и типе оборудования, а также о выполняемых в сети приложениях. Обычно имитационная модель строится не с нуля. Существуют готовые имитационные модели основных элементов сетей: наиболее распространенных типов маршрутизаторов, каналов связи, методов доступа, протоколов и т.п. Эти модели отдельных элементов сети создаются на основании различных данных: результатов тестовых испытаний реальных устройств, анализа принципов их работы, аналитических соотношений. В результате создается библиотека типовых элементов сети, которые можно настраивать с помощью заранее предусмотренных в моделях параметров. Системы имитационного моделирования обычно включают также набор средств для подготовки исходных данных об исследуемой сети – предварительной обработки данных о топологии сети и измеренном трафике. Эти средства могут быть полезны, если моделируемая сеть представляет собой вариант существующей сети и имеется возможность провести в ней измерения трафика и других параметров,

нужных для моделирования. Кроме того, система снабжается средствами для статистической обработки полученных результатов моделирования [6].

В табл. 4.1 приведены характеристики нескольких популярных систем имитационного моделирования различного класса – от простых программ, предназначенных для установки на персональном компьютере, до мощных систем, включающих библиотеки большинства имеющихся на рынке коммуникационных устройств и позволяющих в значительной степени автоматизировать исследование изучаемой сети.

Таблица 4.1

Сравнительные характеристики
систем имитационного моделирования

Компания и продукт	Стоимость (долл.)	Тип сети	Требуемые ресурсы
American HYTech, <i>Prophesy</i>	1495	ЛС*	8МбОП, 6 Мбдиск, DOS, Windows, OS/2
CACI Product, <i>COMNET III</i>	34500–39500	ЛС, ГС*	32 МбОП, 100 Мбдиск, Windows, Windows NT, OS/2, Unix
Opnet, <i>Modeler</i>	20000	ЛС, ГС	32 МбОП, 100 Мбдиск, Windows
Make System, <i>NetMaker XA</i>	40000–70000	ЛС, ГС	128 МбОП, 2000 Мбдиск, AIX, Sun OS, Sun Solaris
Scientific and Engineering Software, <i>SES/Strategizer</i>	9995	ЛС	2 МбОП, 8 Мбдиск, Windows
Network Analysis Center, <i>MIND</i>	9400–70000	ГС	8 МбОП, 65 Мбдиск, DOS, Windows
System& Networks, <i>BONES</i>	20000-40000	ЛС, ГС	32 МбОП, 80 Мбдиск, Sun OS, Sun Solaris

*ЛС – локальная сеть,

ГС – глобальная сеть.

Приведем более подробные характеристики нескольких популярных систем имитационного моделирования различного класса – от простых программ, предназначенных для установки на персональном компьютере, до мощных систем, включающих библиотеки большинства имеющихся на рынке коммуникационных устройств и позволяющих в значительной степени автоматизировать исследование изучаемой сети.

4.1. Система Prophesy

Prophesy относится к продуктам низкой ценовой категории (стоимость ее составляет \$1495), но при этом является полноценной системой имитационного моделирования.

Prophesy дает возможность моделировать состояние сетевой среды для оперативного ежедневного управления, позволяет проверить запланированные инвестиции, прежде чем тратить реальные деньги. С помощью моделирования решаются такие вопросы, как:

1. Каково будет время ответа компьютера, если не добавить дополнительный LAN сервер?
2. Сколько рабочих мест может обслуживаться в текущей сетевой среде?
3. Нужно ли добавить больше каналов связи?
4. Каково будет воздействие на систему, если один или более компонентов выйдут из строя, и другие [23].

Хотя Prophesy оптимизирован для моделирования информационных систем, этот продукт является достаточно гибким, чтобы моделировать практически любые другие системы, будь то производственная линия в производственном процессе или очередь на оказание услуги в розничной торговле.

Prophesy предоставляет мощную среду для моделирования. Простой в использовании графический подход дает возможность быстрого создания прототипов и концепции моделирования, обеспечив при этом возможность дополнительного моделирования более сложных функций.

Работает Prophesy под управлением MS/Windows 3.1 или выше. Минимальный объем оперативной памяти – 4 Мб, рекомендуемый – 8 Мб. Требуемое свободное дисковое пространство: 3 Мб. Но при этом, несмотря на небольшие системные требования, эта система предлагает своему пользователю широкий набор функций, таких как:

- полностью интегрированный визуальный, интерактивный интерфейс с возможностью построения графических моделей при помощи технологии «drag-and-drop» и диалоговых окон;
- графическое представление ресурсов и очередей в режиме реального времени;
- интерактивный анализ статистических результатов с возможностью графического представления результатов;
- до 100 классов объектов с несколькими возможностями представления ресурса;
- итоговый файл, экспортируемый в другие программы;
- возможность записи и воспроизведения моделирования;

- поддержка использования распределения Пуассона, нормального распределения, экспоненциального, эмпирического, а также равномерного распределения;

- функция затрат;

- мастер для ускорения создания модели и др.

Основными объектами моделирования в Prophesy являются:

1. Ресурсы (Resources). Это постоянные сущности с конечной емкостью, способные принимать сообщения в процессе моделирования, производить их обработку и доставку на другой ресурс. Каждый объект-ресурс может представлять собой единый ресурс или группу сотрудничающих ресурсов, представленных в виде одной иконки.

2. Процедуры (Procedures). Процедуры являются постоянными объектами, которые вызываются каким-либо ресурсом. Процедуры определяют последовательности событий (процессов), которые представляют собой процесс работы ресурса в хронологическом порядке.

3. Определения (Definitions). Это постоянные объекты, ассоциированные с ресурсами. Определения определяют характеристики связанного ресурса.

4. Сообщения (Messages). Сообщения создаются как динамические объекты в течение процесса моделирования в соответствии со спецификацией данной модели, и удаляются после использования.

Схема взаимодействия всех объектов показана на рис. 4.1.

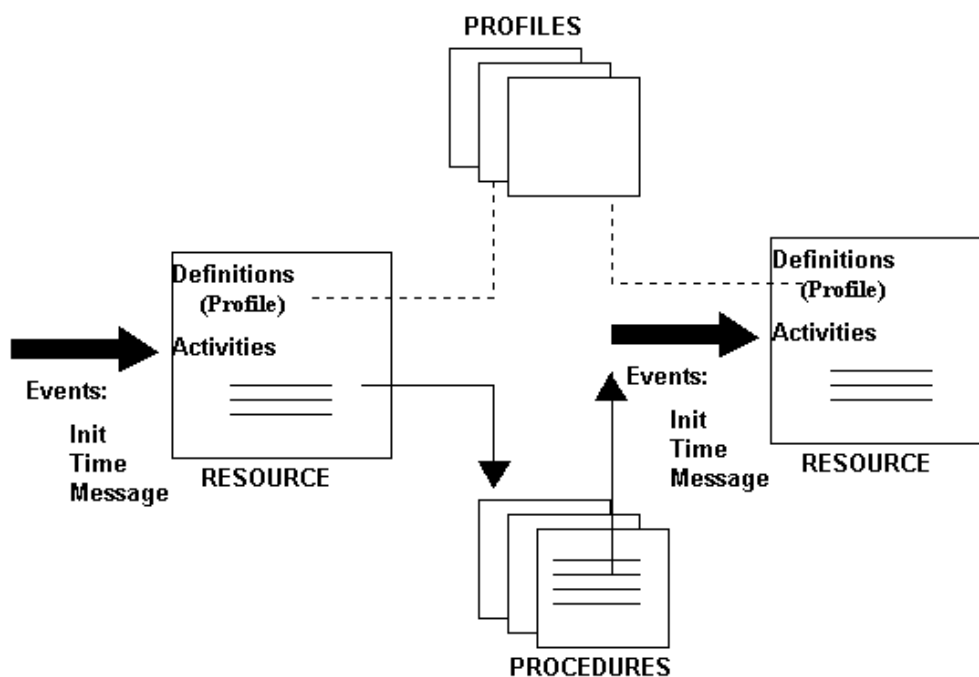


Рис. 4.1. Схема взаимодействия объектов

Некоторые особенности системы Prophesy рассмотрены более подробно в табл. 4.2.

Таблица 4.2

Особенности системы имитационного моделирования Prophesy

Общие характеристики	
Полностью интегрированный визуальный, интерактивный интерфейс	Пользователю не нужно выходить из программы для выполнения всех этапов моделирования: построения модели, проверки, запуска и анализа
Можно смоделировать любой рабочий процесс	Нет ограничения для использования при моделировании маршрутизаторов/мостов/ шлюзов и т.д.
Особенности пользовательского интерфейса	
Построение графической модели при помощи технологии «drag-and-drop», встроенных иконок и диалоговых окон	44 встроенные иконки. Выбор иконок посредством прокручивающейся панели. Кроме того, возможно использование иконок, определенных пользователем
Все определения могут быть добавлены посредством вспомогательных панелей	Не нужно учить какой-либо язык моделирования, чтобы построить модель. Существует возможность добавления определений напрямую

Работая с Prophesy, пользователю не требуется выходить из программы во время рабочей сессии, которая включает следующие этапы:

1. Построение модели. Здесь возможно создать ресурсы при помощи графического интерфейса «drag-and-drop». Ресурсам могут присваиваться характеристики путем открытия панели определений.

2. Проигрывание модели (Запуск). Выполнить простой запуск модели, либо использовать функцию Multiplay, позволяющую проиграть модель несколько раз с помощью одной команды. Кроме того, пользователь может взаимодействовать с моделируемой системой в процессе ее проигрывания, или подождать, пока процесс моделирования будет завершен. Скорость моделирования может быть скорректирована путем установления различных статистических частот обновления.

3. Анализ. Еще в режиме запуска все результаты моделирования динамически обновляются благодаря обновлению панелей утилизации ресурсов и очередей, а также обновляющейся статистической информации о задержках сообщений.

Приведем вид диалогового окна (рис. 4.2), сделанный во время фактического запуска (кроме небольшого фиолетового ящика, который

был добавлен программой PhotoFinish для того, чтобы изобразить анимированный поток сообщений) [23].

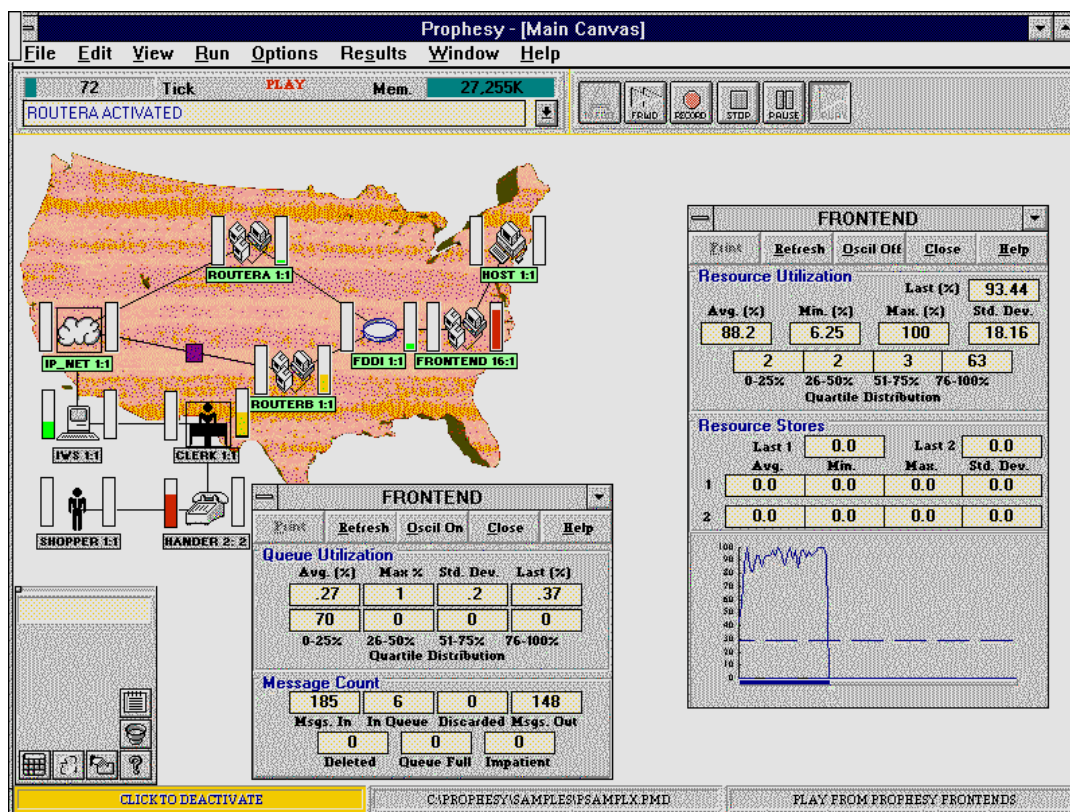


Рис. 4.2. Screenshot рабочего окна программы Prophesy

Программное средство Prophesy позволит моделировать сети. Имеет дружелюбный пользовательский интерфейс и доступно по цене.

4.2. Система Modeler

Modeler от компании OPNET позволяет ускорить процесс проектирования коммуникационных сетей, устройств, протоколов и приложений. Пользователи могут затем проанализировать смоделированные сети для сопоставления влияния различных конструкций системы на ее конечное поведение. Modeler включает в себя широкий набор протоколов и технологий, а также среду разработки, позволяющую производить моделирование всех типов сетей и технологий, в том числе VoIP, TCP, OSPFv3, MPLS, IPv6 и др.

Ключевыми особенностями этой системы являются:

- самое быстрое среди ведущих отраслевых решений средство моделирования дискретных событий;

- сотни протоколов и моделей с исходным кодом;
- объектно-ориентированное моделирование;
- 32- и 64-битное ядро моделирования;
- иерархически спроектированная среда моделирования;
- моделирование дискретных событий, смешанное и частично аналитическое моделирование;
- технология сетевых вычислений для поддержки распределенного моделирования;
- возможность интеграции внешних файлов, библиотек и других моделей;
- основанные на графическом интерфейсе пользователя, комплексные отладка и анализ [21, 24].

Создание сетевой модели становится интуитивно понятным благодаря использованию иерархически спроектированных редакторов, которые позволяют отображать структуру реальных сетей, сетевого оборудования и протоколов:

1. Редактор проекта (Project Editor). Графически представляет топологии коммуникационных сетей. Пользователи могут создавать узлы и объекты-ссылки, чтобы представить элементы, составляющие топологию сети, и быстро настроить их с помощью диалоговых окон. Проекты могут содержать множественные сценарии «что если» для облегчения сравнения и анализа.

2. Редактор узла (Node Editor). Отражает архитектуру сетевого устройства или системы, отображая обмен данными между функциональными элементами, называемыми «модулями». Модули обычно представляют сетевые протоколы или алгоритмы, и ассоциируются с моделями процессов (которые разрабатываются в редакторе процесса), чтобы достичь какого-либо определенного поведения.

3. Редактор процесса (Process Editor). Использует мощный аппарат для поддержки подробных спецификаций протоколов, ресурсов, приложений и алгоритмов.

Взаимодействие этих средств показано на рис. 4.3.

Ядро среды моделирования непрерывно совершенствовалось на протяжении более 20 лет, чтобы обеспечить все более быструю скорость моделирования, соизмеримую с растущим трафиком. Кроме того, Modeler включает в себя многочисленные функции для ускорения моделирования, в том числе:

- 64-битное ядро;
- параллельное моделирование;
- сетевые вычисления.

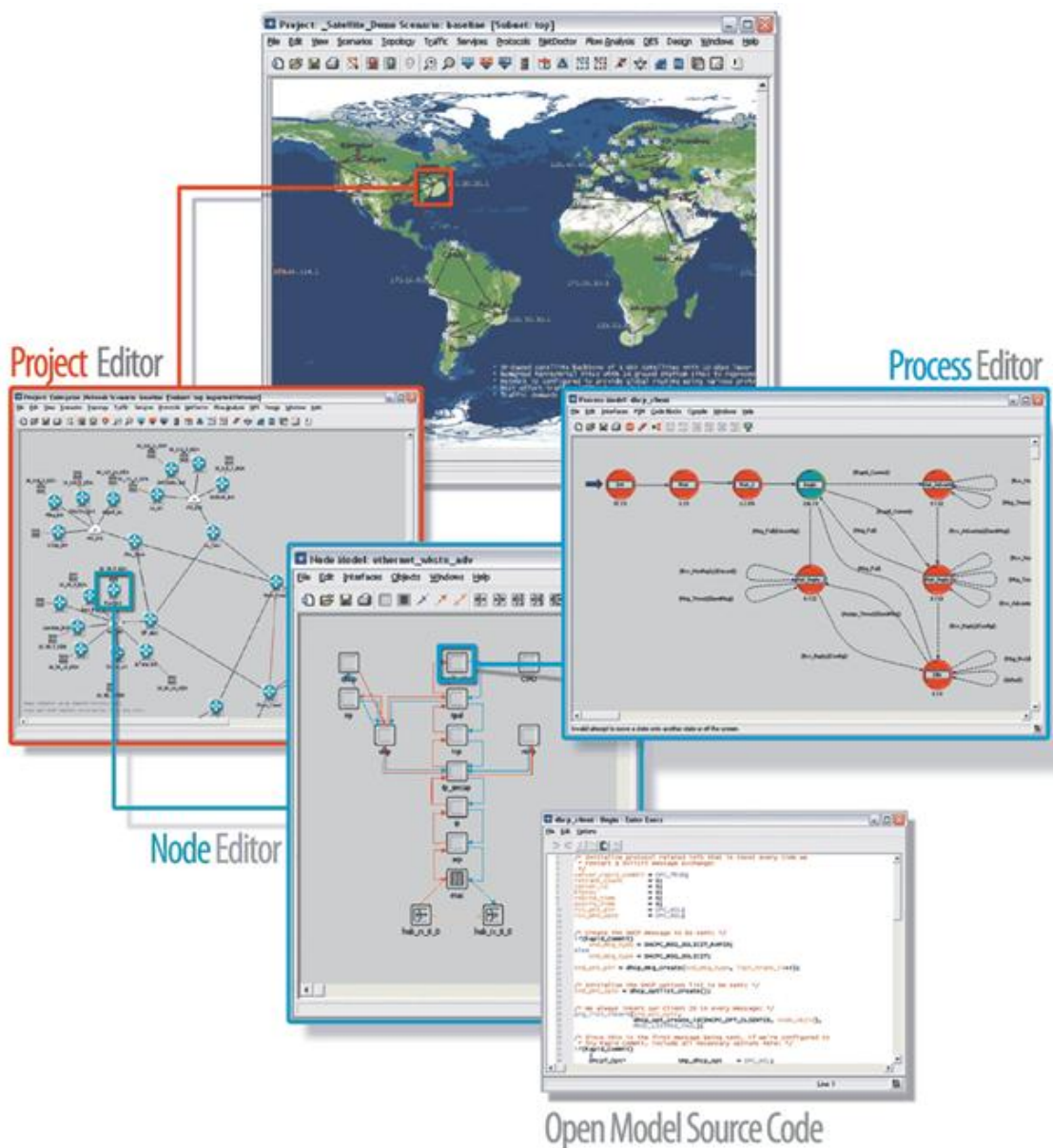


Рис. 4.3. Рабочие окна программы Modeler

Встроенное ядро моделирования параллельных дискретных событий использует многоядерные процессоры или многопроцессорные машины для ускорения времени моделирования. Пользователи могут выбрать, какие процессы моделирования выполнять параллельно для обеспечения оптимального использования ресурсов. В среде сетевых вычислений Modeler может распределять процессы моделирования на несколько машин для одновременного исполнения. Эта функция особенно полезна для параметрических исследований, в которых одно или более значения параметра модели варьируются, для того чтобы анализировать их влияние на общее поведение сети. Кроме того, еще одним

важным использованием параметрических исследований является проверка результатов моделирования.

Все распределенные процессы моделирования централизованно контролируются при помощи Менеджера Выполнения (Execution Manager). Важная информация системы, такая как использование процессора, памяти, и свершение событий и др. управляются из этого удобного аппарата.

Помимо моделирования дискретных событий, Modeler предлагает две другие технологии моделирования.

Первая технология – анализ потока (Flow Analysis) – предлагает полное аналитическое моделирование. Оно обеспечивает самое быстрое время выполнения по сравнению с другими подходами и подходит для поддержки больших сетей и объемов трафика. Анализ потока является полезным для итеративного моделирования, используемого при проектировании сети и анализе отказов.

Вторая технология моделирования – смешанное моделирование (Hybrid Simulation) – предлагает сочетание дискретного и аналитического моделирования движения для наиболее детализованного и быстрого моделирования.

Modeler предлагает сотни встроенных статистик для экспресс-анализа моделирования. Пользователи также могут создавать собственные статистические шаблоны или изменять существующие. Результаты моделирования представляются непосредственно в Modeler, который использует комплексную аналитическую среду. Диаграммы и графики автоматически генерируются на основе предварительно выбранных статистик. Необработанные данные настраиваются и могут быть экспортированы в таблицы и математические программы для более углубленного анализа.

Визуализация маршрутов в Modeler позволяет четко отображать передачу пакетов между любым источником и пунктом назначения, что упрощает сравнение протоколов маршрутизации. Анимированный просмотрщик позволяет пользователям повторять процесс моделирования в Редакторе Проектов для визуального анализа. Скорость анимации контролируется с помощью Контроллера Времени (Time Controller), предоставляющего пользователям свободно управлять паузами и усилением скорости процесса моделирования для более углубленного анализа (рис. 4.4) [24].

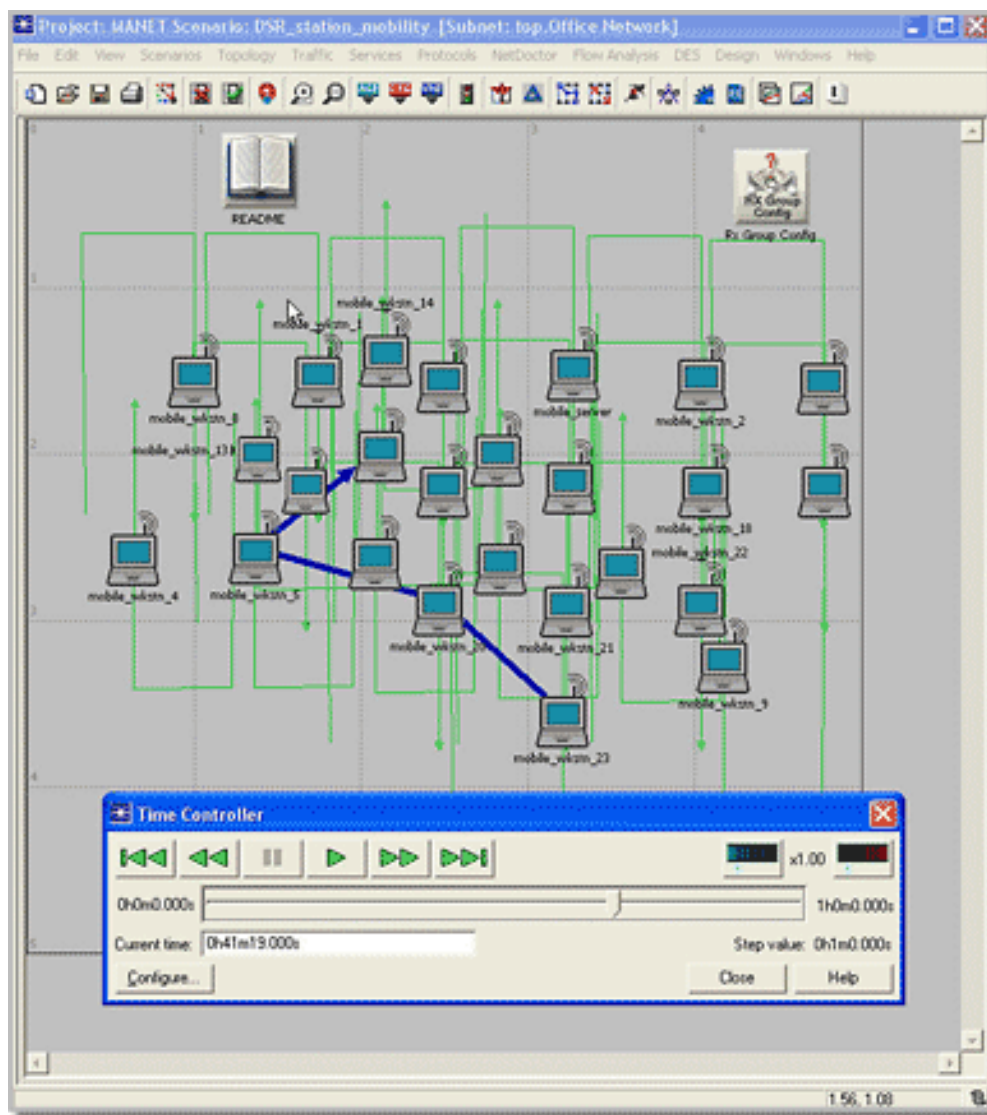


Рис. 4.4. Использование Time Controller в Modeler

Огромным и неоспоримым преимуществом компании OPNET является то, что она бесплатно предоставляет свое программное обеспечение университетам всего мира для целей научных исследований.

4.3. Система COMNETIII

Система имитационного моделирования сетей COMNETIII позволяет точно предсказывать производительность локальных, глобальных и корпоративных сетей. Система COMNETIII работает в среде Windows 95, WindowsNT и Unix.

COMNETIII предлагает использовать простой и интуитивно понятный способ конструирования модели сети, основанный на применении готовых базовых блоков, соответствующих хорошо знакомым сетевым

устройствам, таким как компьютеры, маршрутизаторы, коммутаторы, мультиплексоры и каналы связи.

Пользователь применяет технику drag-and-drop для графического изображения моделируемой сети из библиотечных элементов. Затем система COMNETIII выполняет детальное моделирование полученной сети, отображая результаты динамически в виде наглядной мультипликации результирующего трафика (рис. 4.5).

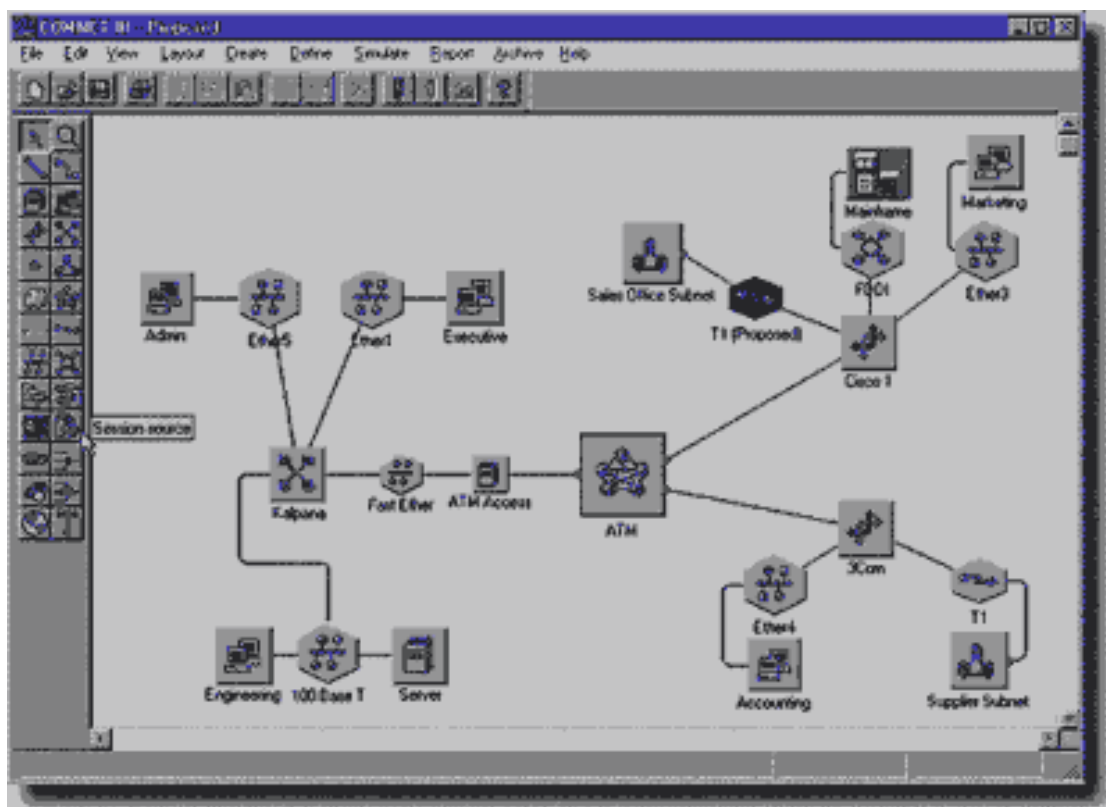


Рис. 4.5. Моделирование сети с помощью системы COMNETIII

После окончания моделирования пользователь получает в свое распоряжение следующие характеристики производительности сети:

1. Прогнозируемые задержки между конечными и промежуточными узлами сети, пропускные способности каналов, коэффициенты использования сегментов, буферов и процессоров.
2. Пики и спады трафика как функцию времени, а не как усредненные значения.
3. Источники задержек и узких мест сети [25].

В COMNETIII моделируется не только взаимодействие компьютеров по сети, но и процесс разделения процессора каждого компьютера между его приложениями. Работа приложения моделируется с помощью

команд нескольких типов, в том числе команд обработки данных, отправки и чтения сообщений, чтения и записи данных в файл, установления сессий и приостановки программы до получения сообщений.

Каналы связи моделируются путем задания их типа, а также двух параметров – пропускной способности и вносимой задержки распространения. Единицей передаваемых по каналу данных является кадр. Пакеты при передаче по каналам сегментируются на кадры. Каждый канал характеризуется: минимальным и максимальным размером кадра, накладными расходами на кадр и интенсивностью ошибок в кадрах.

COMNETIII позволяет при моделировании задавать форму отчета о результатах для каждого отдельного элемента модели. Отчет генерируется каждый раз при запуске определенной модели. Отчет представлен в стандартной текстовой форме, и его легко можно распечатать на любом принтере. Можно задать генерацию нескольких отчетов разного типа для каждого элемента сети [25].

Существуют другие способы получения статистических результатов прогона модели, кроме отчетов. В COMNETIII имеются кнопки Statistics, с помощью которых можно включить сбор статистики для каждого типа элемента модели – узлов, каналов, источников трафика, маршрутизаторов, коммутаторов и т.п. Монитор статистики каждого элемента можно установить для сбора только базовых статистических параметров (минимум, максимум, среднее значение и дисперсия) или же сбора данных во временном масштабе для построения графиков. Если результаты наблюдений сохранены в файле для последующего построения графиков и анализа, то возможно также построение гистограмм и процентных показателей. Возможно построение графиков и во время моделирования.

Построитель отчетов, монитор статистики, аппарат визуализации являются конкурентными и отличительными чертами этой разработки.

4.4. Система NetMaker XA

Вычислительное ядро моделирования, используемое в NetMaker XA от Make Systems, – по мнению многих [26] является одним из наиболее мощных на рынке, и это сыграло немаловажную роль в том, что к настоящему времени продукт зарекомендовал себя очень хорошо.

Главными недостатками NetMaker XA являются необходимость серьезного обучения пользователя и высокая стоимость. Если к цене базовой конфигурации изделия добавить стоимость дополнительных модулей, получится довольно значительная сумма (\$40 000 за базовую конфигурацию).

Основу продукта составляют модули Visualizer, Planner и Designer. Каждый из них выполняет какую-то одну функцию; чтобы смоделировать работу сети, необходимы все три.

1. Visualizer служит для получения информации о сети и ее просмотра. В его состав входят SNMP-модули автоматического распознавания, которые опрашивают сетевые устройства и создают соответствующие им объекты. Информацию об этих объектах можно затем редактировать с помощью Visualizer.

2. Planner – это библиотека устройств, которая помогает проанализировать, что получится при установке в сети нового устройства (например, дополнительного маршрутизатора). Make Systems предоставляет встраиваемые модули (plug-in), содержащие объекты с данными о продуктах различных производителей. В таких объектах содержится полное описание различных моделей устройств (от числа сетевых интерфейсов до типа процессора); вся информация заверяется производителем. С помощью Planner пользователь может самостоятельно строить свои собственные объекты для описания сетевых устройств и каналов связи, не включенных в библиотеку.

3. Designer нужен для построения схем сетей. Данное средство позволяет легко и быстро создавать модели и анализировать альтернативы. Если пользоваться им совместно с Planner, можно получать информацию о том, как будет работать сеть заданной конфигурации [26].

Явным недостатком данной системы является то, что установить ее можно только на SPARCstation от Sun Microsystems и стоимость.

4.5. Система SES/Strategizer

SES/Strategizer от Scientific and Engineering Software занимает средний ценовой сегмент рынка средств моделирования сетей (цена 9995 долл.), но его возможности не позволяют моделировать возможность роста сети.

SES/Strategizer просчитывает модели очень быстро. Можно также собирать тонкие статистические данные о каком-то одном конкретном элементе модели, например следить за степенью загрузки центрального процессора с разбивкой по процессам, пользователям и моделям поведения.

Один из серьезных недостатков программы – необходимость перезапуска модели при каждом внесении каких-либо изменений. Другие продукты позволяют вставлять в модель различные переменные (например, учитывающие рост сети); в результате можно опробовать несколько вариантов в ходе одного прогона программы [27].

Как и многие пакеты, SES/Strategizer позволяет без труда задавать и модифицировать значения параметров, таких как пропускная способность. Кроме того, продукт выдает запрос на подтверждение («Применить» или «Отмена»), если пользователь пытается закрыть диалоговое окно, щелкнув мышью на крестике в правом верхнем углу. Такая функция не предусмотрена в других продуктах, что неудобно, поскольку с ними никогда нельзя быть уверенным, какое действие будет предпринято по умолчанию.

SES/Strategizer имеет ряд недостатков: для просмотра результатов моделирования на том же ПК, где работает сама программа, требуется запустить Microsoft Excel; данные должны браться из создаваемых SES/Strategizer файлов, где для разделения числовых полей используются знаки табуляции. Если Excel не установлен, пользователь получает сообщение об ошибке, указывающее на совершенно другую причину сбоя [27].

4.6. Система Arena

Система имитационного моделирования *Arena* – основной программный продукт Systems Modeling. Корпорация Systems Modeling была основана в 1982 г. Деннисом Педгеном, автором SIMAN – первого промышленно-ориентированного общецелевого языка имитационного моделирования. В настоящее время область деятельности Systems Modeling включает в себя имитационное моделирование и разработку технологического программного обеспечения [6, 22].

Система *Arena* позволяет моделировать виды деятельности, представленные на рис. 4.6. В том числе *Arena* позволяет моделировать сети и системы телекоммуникаций.

Arena снабжена удобным объектно-ориентированным интерфейсом и обладает удивительными возможностями по адаптации к всевозможным предметным областям. Система не требует написания программного кода и исключительно проста в использовании, но требует значительного времени для освоения и достаточно глубоких знаний теории вероятностей, математической статистики, теории систем массового обслуживания и сетей Петри.

Для отображения результатов моделирования используется анимационная система *Cinema animation*. Интерфейс *Arena* включает в себя всевозможные средства для работы с данными, в том числе электронные таблицы, базы данных, ODBC, OLE, поддержку формата DXF.

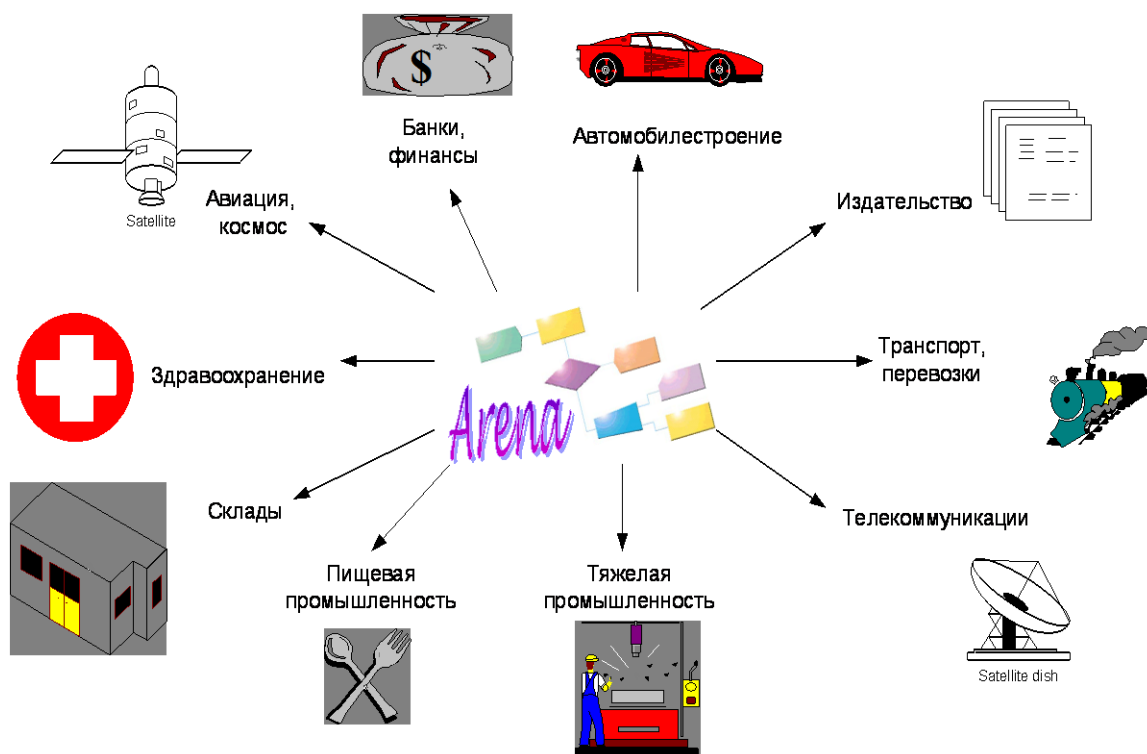


Рис. 4.6. Области применения Arena

Система имитационного моделирования Arena включает:

- двухмерный графический редактор;
- трехмерный графический редактор (пакет 3D player);
- редакторы временных шаблонов и расписаний;
- редактор символов и библиотека графических заготовок;
- связь с библиотекой графических заготовок и буфером обмена Microsoft.

Arena позволяет использовать дискретное, непрерывное, а также совмещенное дискретно-непрерывное моделирование.

Данный программный продукт поддерживает возможность взаимодействия с пакетом VBA Visual Basic for Applications корпорации Microsoft; объектной моделью ActiveX для внешнего управления, доступом через ADO/ODBC к базам данных (Oracle, Access, Excel, SQL); поддерживает импорт файлов из пакета AutoCad (в формате dxf); импорт данных из пакета Visio; импорт данных из пакета Blue Pumpkin Workforce, коммуникация между отдельными процессами.

При моделировании процессов и систем в Arena, используются три строительные панели:

1. *Basic Process Panel* – модули панели являются фундаментом для создаваемых моделей. Объекты данной панели инструментов состоят из

графических модулей: Create, Dispose, Process, Deeside, Separate, Batch, Assign, Record и модулей данных: Entity, Resource, Queue, Variable, Schedule, и Set.

2. *Advanced Process Panel* – панель усовершенствованных процессов; позволяет моделировать более сложные процессы. Если в Basic Process Panel один модуль может быть наделен несколькими внутренними свойствами, то в Advanced Process эти свойства вынесены как самостоятельные графические модули. Панель состоит из 13 графических модулей (Flowchart Modules): Delay, Dropoff, Hold, Match, Pickup, ReadWrite, Release, Remove, Seize, Search, Signal, Store, Unstore и семи модулей данных (Data Modules): Advanced Set Module, Expression, Failure, File, StateSet, Statistic и Storage.

3. *Advanced Transfer Panel* – панель процессов передачи, перемещения; содержит 17 графических модулей: Enter Module, Leave Module, Pick Station, Route, Station; Access, Convey, Exit, Start, Stop; Activate, Allocate, Free, Halt, Move, Request, Transport и пять модулей данных: Sequence, Conveyor, Segment, Transporter, Distance.

В рамках дисциплины «Моделирование сетей ЭВМ и СТК» студентам в качестве средства имитационного моделирования будет предложено именно это средство, т.к.:

1. В основе программный пакет (ПП) Arena заложены математические аппараты СМО и раскрашенных сетей Петри, а комбинация, как известно, берет достоинства обоих математических аппаратов, и возникает синергетический эффект от такого объединения.

2. В ПП Arena первоначально разработчиками предлагается использовать этот пакет для моделирования, анализа сетей и СТК.

3. В НИ ТПУ в рамках дисциплин: «Компьютерное моделирование», «Имитационное моделирование» и «Моделирование и анализ сложных систем» на протяжении более 4 лет использовался этот ПП. Есть разработки и опыт моделирования систем.

4. Компания-разработчик предлагает бесплатную академическую лицензию ПП для учебных заведений.

4.7. Выводы по главе

В данной главе рассмотрены средства моделирования вычислительных сетей, отмечены их основные функциональные возможности, особенности и предназначения каждого из рассматриваемых продуктов.

Для дальнейшего моделирования был выбран программный продукт Rockwell Software Arena, обладающий рядом преимуществ перед

другими продуктами (прост в изучении и реализации поставленной задачи).

В следующей главе реализуется имитационная модель компьютерного кластера для распределено-параллельных вычислений с помощью программного продукта Arena 9.0.

4.8. Вопросы по главе*

1. Какие программные средства моделирования сетей вам известны?
2. Какие программные средства, из известных вам, позволяют моделировать не только сети?
3. Что является основными элементами моделирования в системе Prophesy?
4. Какая система моделирования позволяет описывать широкий спектр сетей и технологий, в том числе VoIP, TCP, OSPFv3, MPLS, IPv6?
5. В какой системе возможно дискретное, аналитическое и смешанное моделирование?
6. Какие системы моделирования обладают мощными функциями работы со статистикой и отчетами?
7. Какие характеристики производительности сети моделирует COMNETIII?
8. Возможно ли моделировать работу сети в системе NetMaker XA без Planner? А без Designer?
9. Перечислите достоинства и недостатки системы моделирования SES/Strategizer.
10. Возможно ли моделировать сети и телекоммуникационные системы в ПП ARENA? Аргументируйте свой ответ.

* Данные вопросы могут быть использованы для учебной дискуссии.

5. ПРИМЕР МОДЕЛИРОВАНИЯ СЕТИ

В качестве примера моделирования для студентов был выбран суперкомпьютерный кластер (СКК). Данная тематика в последние годы является, безусловно, актуальной и востребованной. Но в общедоступной литературе не встречались работы по моделированию функционирования самого СКК.

5.1. Введение в предметную область

Большие объемы накопленной информации требуют огромных мощностей вычислительных ресурсов не только для ее хранения, но и для обработки хранимых данных. Одним из возможных вариантов решения данной задачи является использование кластеров для распределено-параллельных вычислений.

Под кластером будем понимать группу компьютеров, объединенных высокоскоростными каналами связи и представляющую, с точки зрения конечного пользователя, единый аппаратный ресурс.

Вычислительные суперкомпьютерные кластеры высоко востребованы для разных отраслей, таких как машиностроение, судостроение, энергетическая отрасль, наука и др.

Использование суперкомпьютерного кластера позволяет:

- моделировать аэрогазодинамические процессы, химические реакции, физические процессы;
- проверять статическую и динамическую прочность элементов и механических систем;
- испытывать гидравлику, термодинамику, теплопроводность;
- решать сложные математические уравнения;
- проводить расчеты геологических пластов;
- решать многие другие задачи.

Результатом использования СКК является снижение количества требуемых натурных и дорогостоящих испытаний, предсказание поведения системы во внештатных условиях, что позволяет существенно снизить затраты на производство тестовых образцов [28, 29].

В свою очередь, часто возникает сложная задача моделирования работы самого компьютерного кластера для распределенно-параллельных вычислений. Решение данной задачи позволит следующее:

- упростить и оптимизировать процесс распределения поступающих данных между членами кластера – компьютерами;

- выделить параметры кластера, наиболее важные для успешного функционирования, и проанализировать имеющиеся между ними связи;
- рассмотреть процессы в различных масштабах времени;
- разработать процессы, позволяющие бороться с тупиками и неопределенностью, вызванными случайностью и непостоянством систем;
- обнаружить скрытые резервы и ресурсы моделируемой системы и оптимально их использовать.

Таким образом, в качестве примера моделирования вычислительной сети рассмотрим процесс функционирования компьютерного кластера для оптимального распределения заданий между компьютерами, входящими в его состав.

5.2. Постановка задачи

Пусть на суперкомпьютерный кластер приходят задания в виде пакетов конечного размера от 10 МБ до 100 МБ, но чаще всего 60 Мбайт. Каждое задание требует определенного алгоритма обработки (всего 5 алгоритмов) на кластере. Алгоритмы сложности назначаются заданию случайным образом.

В кластере имеется центральная ЭВМ, занимающаяся подготовкой и распределением заданий, а также сбором результатов с 5 компьютеров, на которых происходит собственно обработка. Машины могут обрабатывать задания размером до 20 Мбайт.

В зависимости от размера задания, оно распределяется между несколькими машинами. Кластер построен по топологии звезда. Каналы связи, соединяющие рабочие ЭВМ с центральной, дуплексные; имеют одинаковую проводимость 1 Мбайт в секунду. Задания поступают на центральную ЭВМ и распределяются между ЭВМ. При этом, если в кластере остаются свободные ресурсы (компьютеры), их можно загрузить следующим заданием.

5.3. Описание модели суперкомпьютерного кластера

В качестве среды для разработки модели выберем программный продукт (ПП) Arena (разработчик Rockwell Software, США), позволяющий создавать имитационные модели разнородных процессов и систем в терминах предметной области для дальнейшей работы с ними.

На рис. 5.2 представлена разработанная в Arena модель компьютерного кластера для распределено-параллельных вычислений. Рассмотрим ее подробнее.

Логику разработанной модели условно разобьем на 2 части: *поступление заданий и их обработка*. Задание, в терминах математического аппарата систем массового обслуживания и программного пакета Arena, будем называть *сущностью*.

Модуль Create 1 (рис. 5.1) генерирует поступление сущностей в систему генератором случайных чисел. Максимальное число заданий – 100.

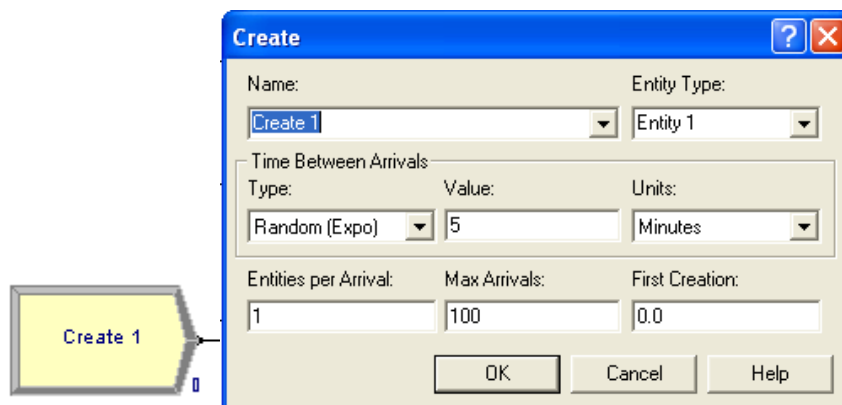


Рис. 5.1. Модуль Create 1

Задания поступают в модуль Decide 3 (рис. 5.2), откуда случайным образом с равной вероятностью направляются в модули Assign 1, Assign 2, Assign 3, Assign 5 или Assign 6.

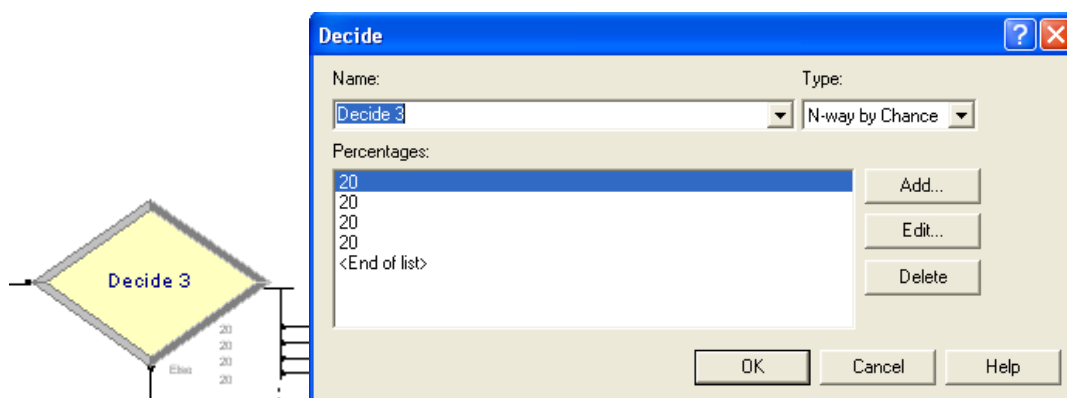


Рис. 5.3. Модуль Decide 3

После прохождения одного из этих модулей (рис. 5.4) каждой сущности (заданию) присваивается время обработки, которое равно значению атрибута Attribute 2. Время обработки, Attribute 2, будет влиять на время обработки поступившего задания и моделирует алгоритм обработки задания компьютерами кластера, т.е. влияет на загрузку узлов кластера.

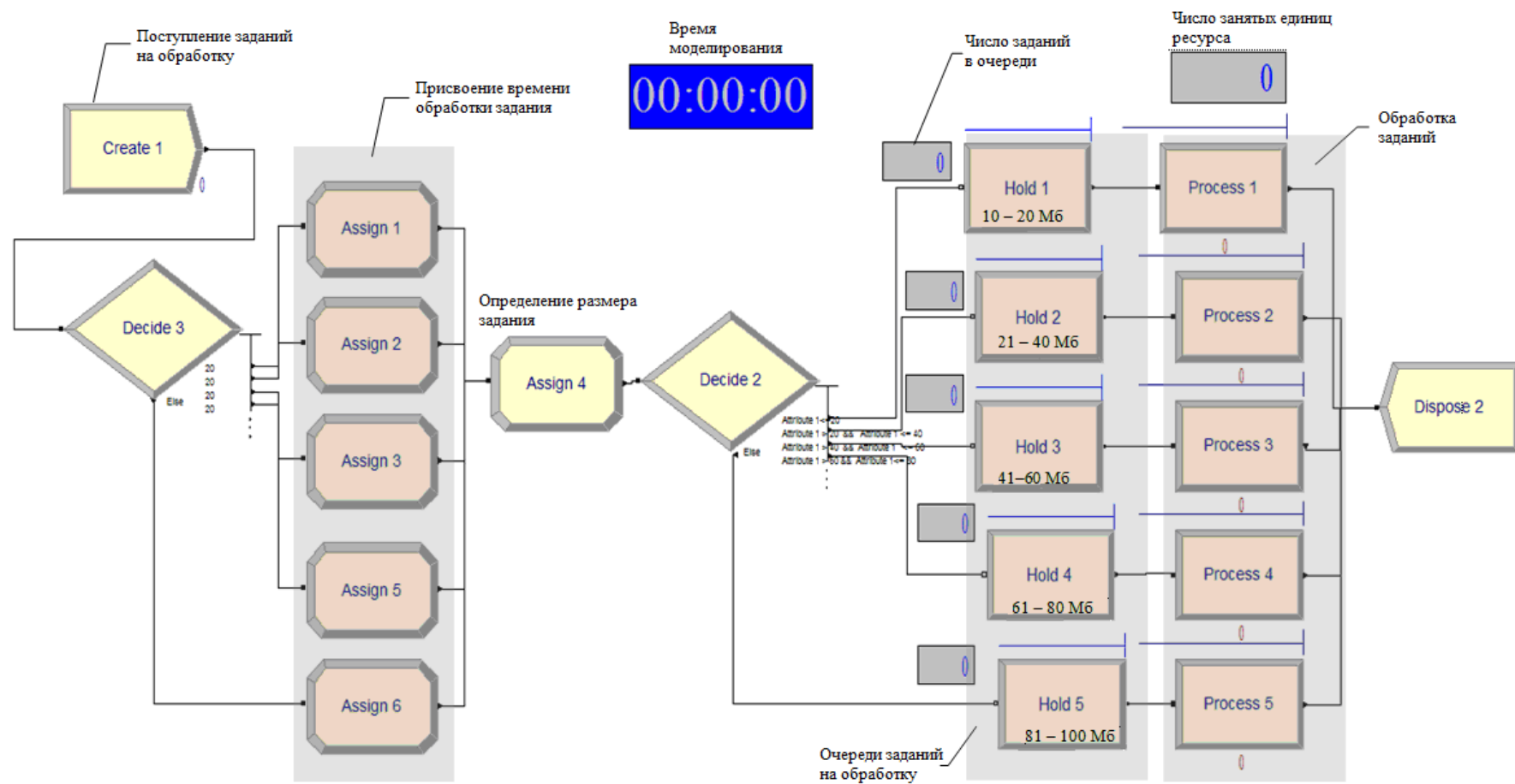


Рис. 5.2. Логика модели компьютерного кластера для распределено-параллельных вычислений

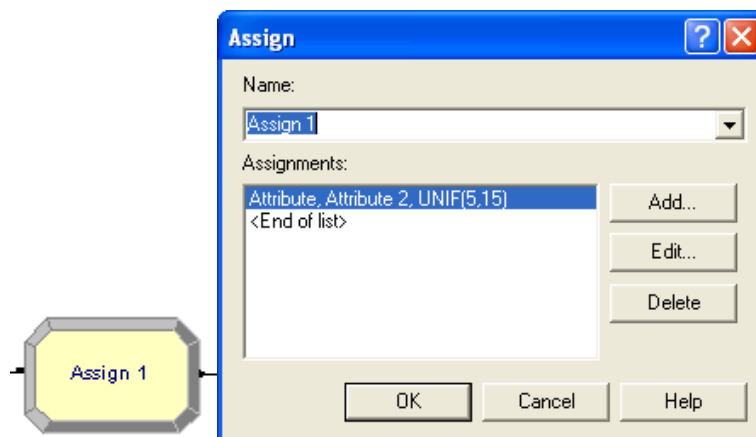


Рис. 5.4. Модуль Assign 1

Затем задания проходят через модуль Assign 4 (рис. 5.5), где задается значение атрибута Attribute 1. Attribute 1 – размер задания – изменяется от 10 Мб до 100 Мб.

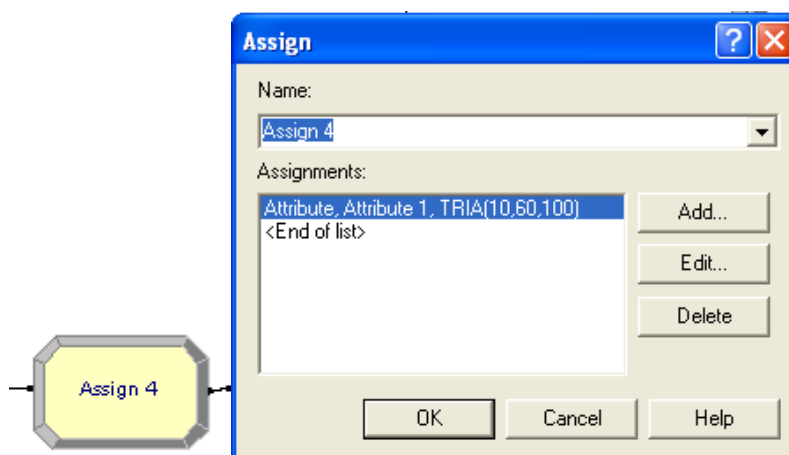


Рис. 5.5. Модуль Assign 4

В результате на компьютерный кластер (модуль Decide 2) поступают задания, имеющие определенный конечный размер (Attribute 1) и требующие определенного времени обработки (Attribute 2).

В модуле Decide 2 (рис. 5.6) в зависимости от длины задания происходит распределение заданий по пяти очередям. В очередь к модулю Hold 1 поступают задания, длина которых от 10 до 20 Мб, к модулю Hold 2 – с длиной 21–40 Мб, к Hold 3 – с длиной 41–60 Мб, к Hold 4 – с длиной 61–80 Мб, и к модулю Hold 5 – с длиной 81–100 Мб.

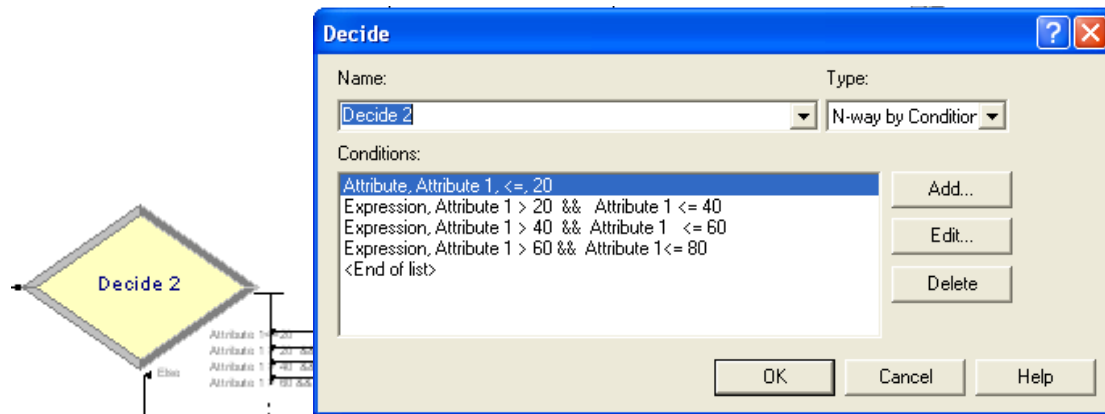


Рис. 5.6. Модуль Decide 2

Каждый модуль Hold (рис. 5.7) выпускает сущности в зависимости от назначаемого условия (количество свободных единиц ресурса и назначаемый приоритет модуля).

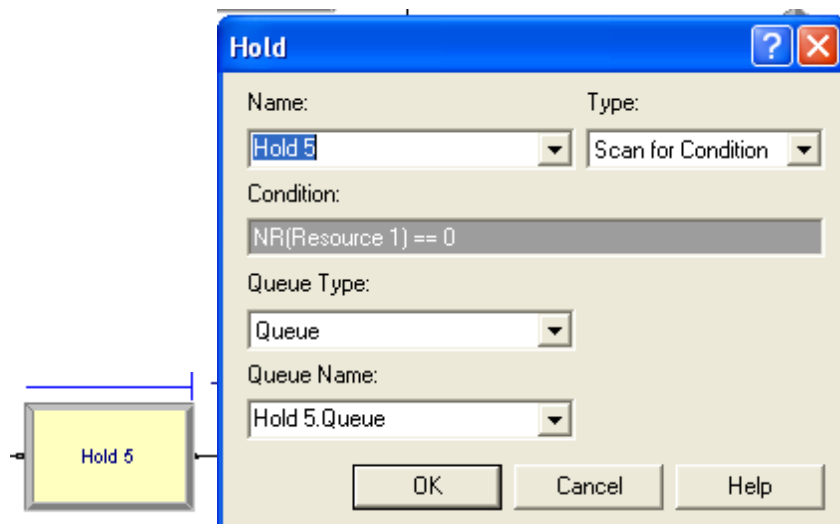


Рис. 5.7. Модуль Hold 5

Вычислительные машины, на которых происходит обработка заданий, представлены в виде одного ресурса (Resource 1), заданной емкостью – 5. Данный ресурс смоделирован в пяти различных модулях Process, различной емкостью: 1, 2, 3, 4, 5. В зависимости от размера задания поступают на обработку в различные процессы и захватывают различное количество ячеек ресурса.

Сущности, выходящие из очереди Hold1.Queue, поступают в модуль Process 1 (рис. 5.8) и захватывают одну единицу ресурса Resource1. Соответственно, сущности из очереди Hold2.Queue захватывают две

единицы ресурса (рис. 5.9), из Hold3.Queue – три единицы, из Hold4.Queue – четыре, и из Hold5.Queue – все пять единиц. В каждом случае длительность обработки задания равна значению атрибута Attribute 2 сущности (время обработки задания).

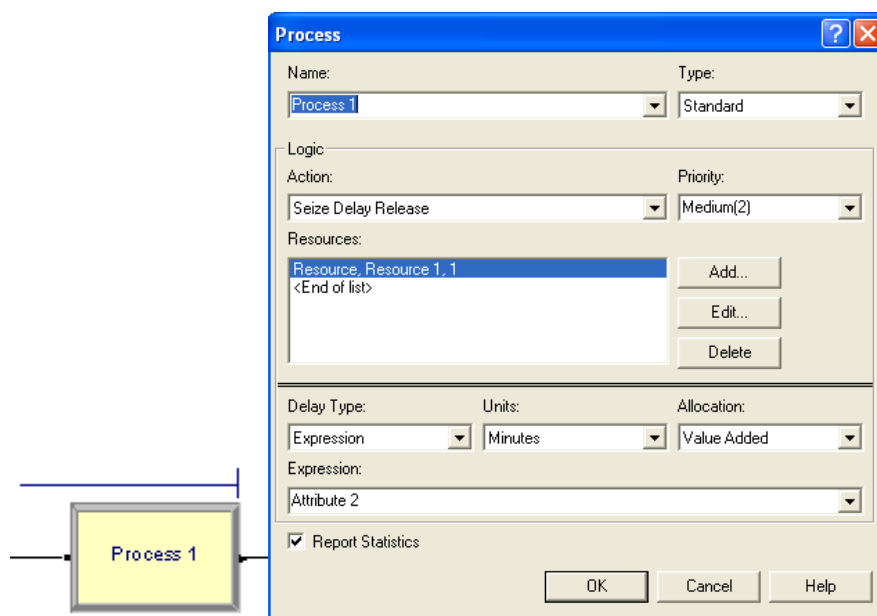


Рис. 5.8. Модуль Process 1

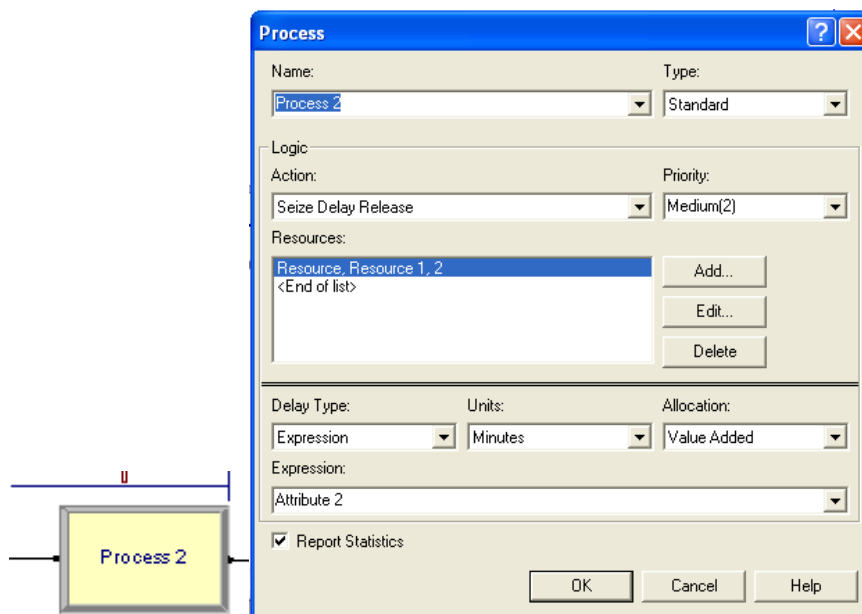


Рис. 5.9. Модуль Process 2

После обработки на вычислительных машинах задания поступают в модуль Dispose 1 и удаляются из системы (рис. 5.10).

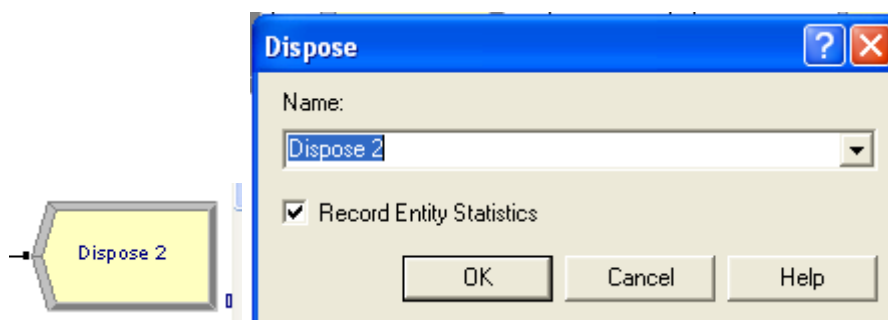


Рис. 5.10. Модуль Dispose 1

5.4. Анализ вариантов распределения приоритетов обработки заданий

Используя разработанную модель суперкомпьютерного кластера, проанализируем эффективность его работы, изменяя последовательность выборки заданий из очередей. Для этого рассмотрим работу суперкомпьютера с определенным количеством поступивших заданий.

Смоделируем обработку 100 заданий и посмотрим, как изменяются такие параметры, как время обработки заданий, загрузка машин и среднее число заданий в очереди при изменении последовательности обработки заданий. При наличии пяти узлов в кластере количество вариантов последовательностей обработки равно $5!$, т.е. 120.

Рассмотрим примеры возможных вариантов последовательностей обработки заданий:

- сначала будут обрабатываться задания с наименьшим Attribute 1, т.е. задания размером от 0 до 20 Мб, затем по возрастанию Attribute 1 – от 20 до 40 Мб и т.д. Эту последовательность обозначим как 1_2_3_4_5;
- возможна обратная ситуация, когда будем обрабатывать задания по уменьшению значения Attribute 1, сначала обработав самые большие по размеру задания 80–100 Мб, затем 60–80 Мб и т.д. Эту последовательность мы обозначим 5_4_3_2_1;
- изменяя таким образом последовательность обработки заданий, промоделируем все 120 возможных вариантов обработки.

На рис. 5.11 приведены последовательности из серий обработки заданий, представляющие наибольший интерес для нашего моделирования. Экспериментальные результаты этих серий показали различные результаты.

Наименьшее (лучшее для данного исследования) время обработки заданий, равное 6,28 с, получено в следующих последовательностях: 1_4_3_2_5, 1_4_3_5_2, 1_5_4_3_2, 4_1_3_2_5, 4_3_2_1_5, 4_3_2_5_1, 4_5_3_2_1. Худший результат, со временем равным 15:03:27, был получен в последовательностях: 2_4_5_3_1, 2_4_5_1_3, 2_3_4_1_5, 2_3_5_1_4.

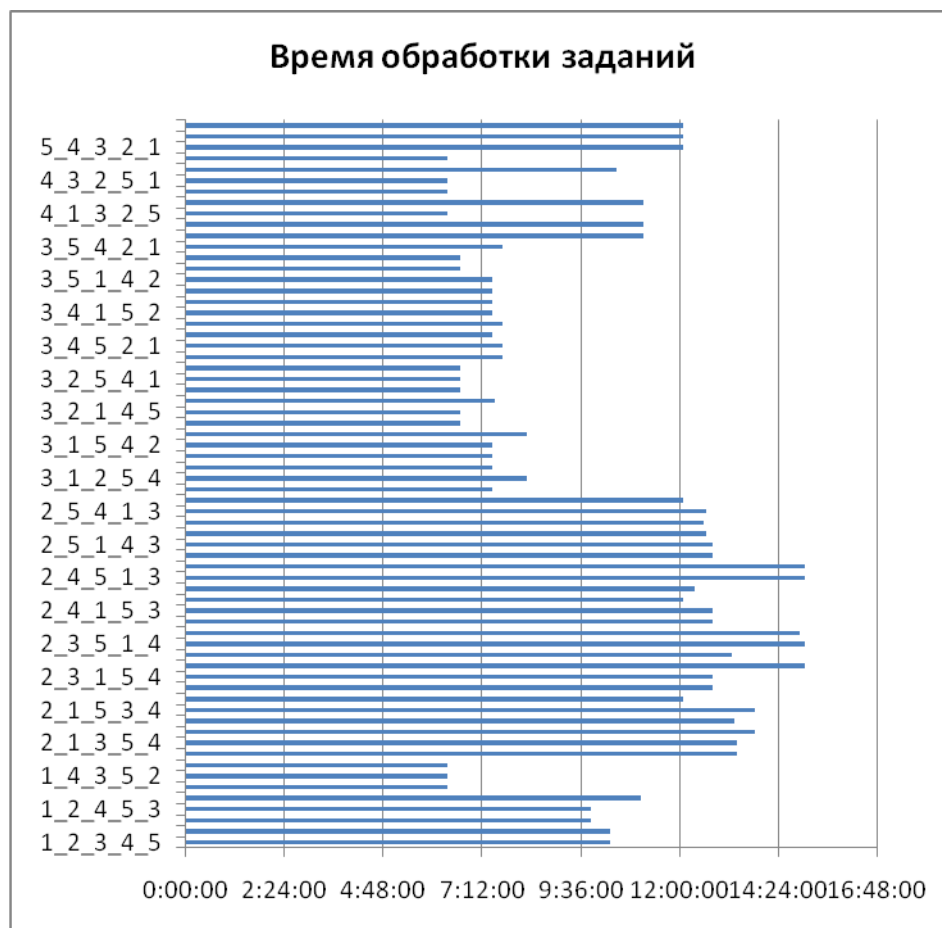


Рис. 5.11. Результаты эксперимента по обработке заданий

На рис. 5.12 приведен график загрузки компьютеров кластера. Для данного исследования наилучшим результатом является максимальная загрузка компьютеров, т.е. распределение поступающих заданий организовано таким образом, что работа кластера проходит без простоя компьютеров. Наилучшими экспериментальными данными (максимальная загрузка) обладают следующие последовательности: 1_4_3_2_5, 1_4_3_5_2, 1_5_4_3_2, 4_1_3_2_5, 4_3_2_1_5, 4_3_2_5_1, 4_5_3_2_1. Худший результат был получен в последовательностях: 2_4_5_3_1, 2_4_5_1_3, 2_3_4_1_5, 2_3_5_1_4.

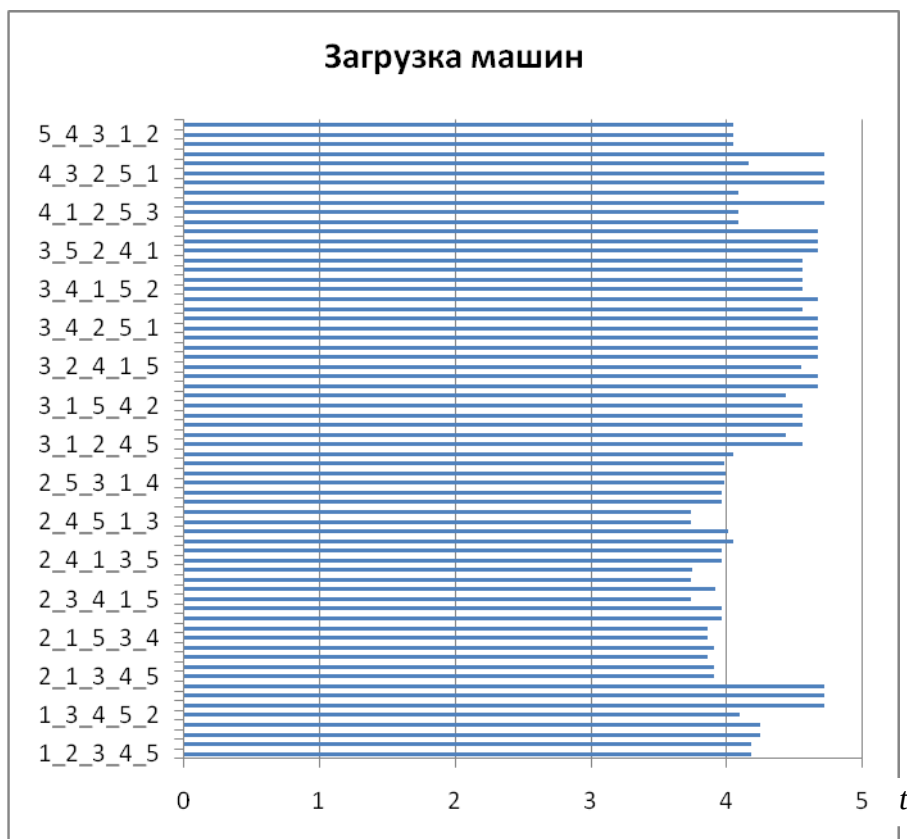


Рис. 5.12. Результаты эксперимента по загрузке компьютеров кластера

На рис. 5.13–5.15 приведены графики, отображающие среднее количество заданий в очереди к компьютерам моделируемого кластера. В качестве примеров выбраны графики очереди 2 ко второму компьютеру (рис. 5.13), очереди 3 (рис. 5.14), очереди 4 (рис. 5.15). Наилучшим результатом моделирования является минимизация количества заданий, ожидающих обработки. Таким образом, наилучшими результатами являются:

- для компьютера 1: 1_3_4_5_2, 1_4_3_2_5, 1_4_3_5_2, 1_5_4_3_2, 2_3_4_1_5, 2_3_5_1_4, 2_4_3_1_5, 2_4_5_1_3, 4_5_3_2_1;
- для компьютера 2: 2_1_5_4_3;
- для компьютера 3: 1_2_4_3_5, 1_2_4_5_3;
- для компьютера 4: 4_5_2_3_1, 4_5_3_2_1;
- для компьютера 5: 2_4_5_3_1.

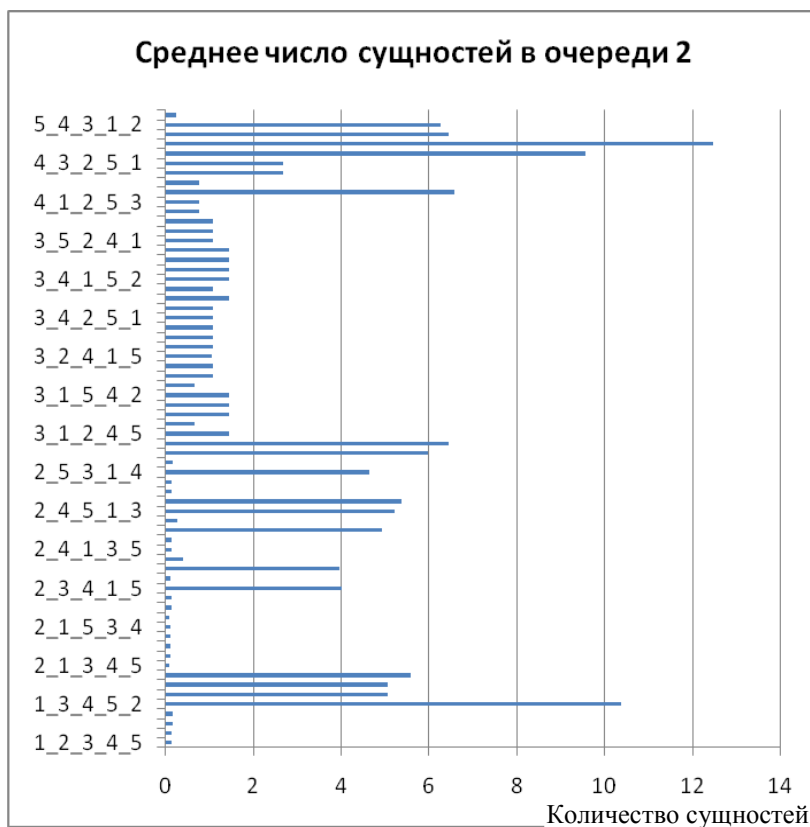


Рис. 5.13. Результаты эксперимента
по среднему числу заданий в очереди 2



Рис. 5.14. Результаты эксперимента
по среднему числу заданий в очереди 3

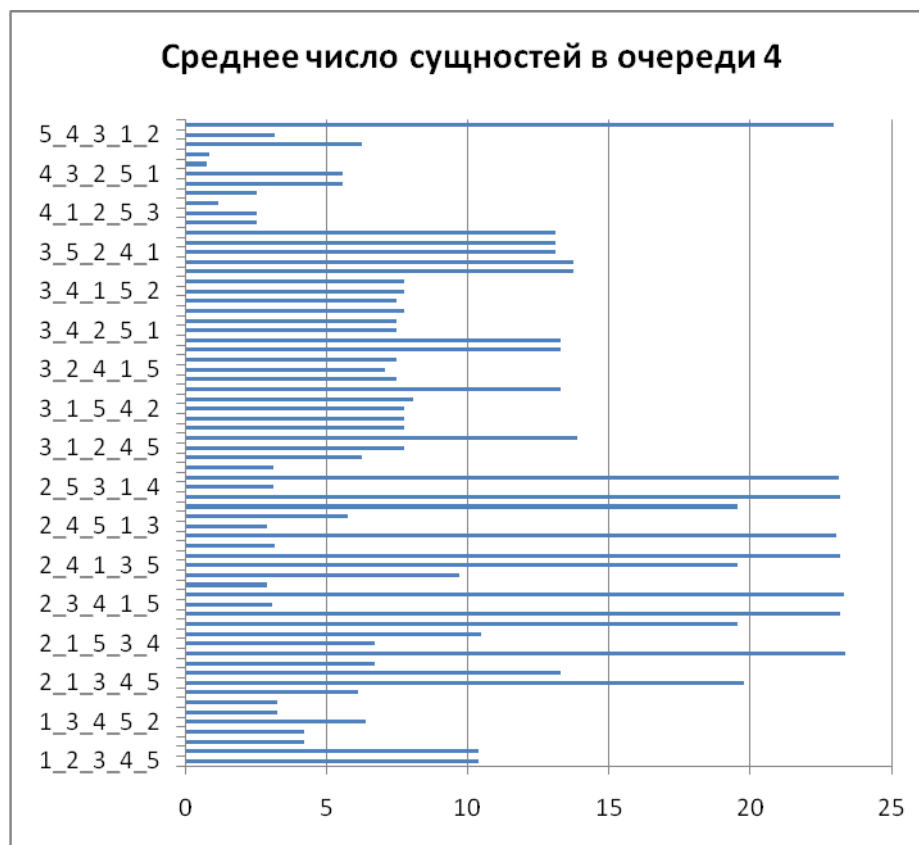


Рис. 5.15. Результаты эксперимента
по среднему числу заданий в очереди 4

Создадим сводную таблицу экспериментальных данных, которая позволит определить оптимальные результаты распределения заданий на обработку (табл. 5.1).

В таблице приведена выборка данных из проведенных 120 экспериментов, в которых были получены лучшие комбинации хотя бы по 1 параметру: время обработки заданий, среднее число заданий в очередях, загрузка компьютеров. Последний столбец таблицы отображает количество наилучших параметров по комбинациям.

Проанализировав полученную таблицу, можно сделать вывод, что наибольшая эффективность работы СКК получается при комбинации, когда первыми обрабатываются задания размером 60–80 Мб, затем 80–100 Мб, 40–60 Мб, 20–40 Мб и последними 0–10 Мб (4_5_3_2_1).

Таблица 5.1

Сравнительные данные по результатам моделирования

Последовательность	Время, с	Среднее число сущностей					Загрузка машин, с	Количество совпадений
		очередь 1	очередь 2	очередь 3	очередь 4	очередь 5		
1_2_4_3_5	9:50:43	0,0418	0,1566	0,2138	4,2137	17,3516	4,2454	1
1_2_4_5_3	9:50:43	0,0418	0,1566	0,2138	4,2137	17,3516	4,2454	1
1_4_3_2_5	6:22:29	0,0359	5,0602	5,6218	3,2509	16,592	4,7193	2
1_4_3_5_2	6:22:29	0,0359	5,0602	5,6218	3,2509	16,592	4,7193	2
1_5_4_3_2	6:22:29	0,0207	5,5708	7,4727	6,1213	11,6352	4,7193	2
2_1_5_4_3	12:04:45	0,0138	0,0838	21,3764	10,4509	1,1969	3,8569	1
2_3_4_1_5	15:03:27	0,0000	4,0222	13,8152	3,0450	7,7039	3,7348	1
2_3_5_1_4	15:03:27	0,0000	3,9644	25,0443	2,9005	3,9653	3,7348	1
2_4_3_1_5	12:04:45	0,0000	4,9231	20,3758	3,1399	7,5969	4,0431	1
2_4_5_1_3	15:03:27	0,0000	5,2127	25,3230	2,9005	3,0675	3,7348	1
2_4_5_3_1	15:03:27	0,0429	5,3694	25,3566	5,7620	0,2370	3,7348	1
2_5_3_1_4	12:39:34	0,0000	4,6468	20,5508	3,0902	7,6191	3,9791	1
2_5_4_1_3	12:39:34	0,0000	5,9767	24,3638	3,0902	3,2681	3,9791	1
4_1_3_2_5	6:22:29	0,0487	6,5768	7,3073	1,1409	16,592	4,7193	2
4_3_2_1_5	6:22:29	1,6565	2,6841	3,6266	5,5577	16,592	4,7193	2
4_3_2_5_1	6:22:29	1,6565	2,6841	3,6266	5,5577	16,592	4,7193	2
4_5_3_2_1	6:22:29	0,0584	12,4622	15,4654	0,8454	7,1423	4,7193	3

5.5. Выводы по главе

В данной главе был рассмотрен пример моделирования СКК. Для чего были изучены основы функционирования СКК, определены основные параметры СКК.

Для разработки имитационной модели компьютерного кластера был использован ПП Arena 9.0, позволяющий проанализировать процесс распределения поступающих задач между членами кластера.

После разработки модели СКК проведен анализ вариантов выборки задач из очередей с различным приоритетом. На основании полученных результатов показано, что наибольшая эффективность работы компьютерного кластера для распределено-параллельных вычислений будет обеспечена при комбинации, когда первыми обрабатываются задания размером 60–80 Мб, затем 80–100 Мб, 40–60 Мб, 20–40 Мб и последними 0–10 Мб (4_5_3_2_1).

В следующей главе будет разработан и представлен практический материал по дисциплине, который может быть предложен студентам в виде цикла лабораторных работ, заданий на курсовое проектирование и семинара. Все эти виды деятельности подразумевают использование активных методов обучения, таких как командная работа, учебная дискуссия, индивидуальные траектории обучения и пр.

6. ПРАКТИЧЕСКАЯ ЧАСТЬ ДИСЦИПЛИНЫ

При выполнении практической части работ по дисциплине «Моделирование сетей ЭВМ и СТК» студентам будет предложено несколько видов практических работ:

1. Лабораторный практикум, состоящий из 7 лабораторных работ (п. 6.1 этой главы).
2. Курсовое проектирование с использованием методов групповой работы (п. 6.2 этой главы).
3. Семинар (п. 6.3 этой главы) или миниконференция.
4. Мастер-класс по моделированию, проводимый на первом лабораторном занятии преподавателем или приглашенным экспертом.
5. Методы case-study, используемые преподавателем при чтении лекционного курса и выполнении практических заданий.
6. Дискуссия, проводимая преподавателем в течение всего лекционного курса, а также во время семинара.

6.1. Лабораторный практикум

Выполнение лабораторных работ в рамках данного учебного пособия и дисциплины, читаемой в Национальном исследовательском Томском политехническом университете, будет предложено студентам с использованием программного пакета ARENA 9.0 (см. п. 4.6 гл. 4 и Приложение 1. Описание основных модулей программного пакета ARENA). Это обусловлено тем, что:

1. Имеется огромный задел по курсам: «Имитационное моделирование», «Компьютерное моделирование» и «Моделирование и анализ сложных систем», лабораторный практикум, по которым проводился с использованием именно этого программного пакета.
2. Программный пакет ARENA 9.0 предназначен, в том числе, для моделирования сетей и систем телекоммуникаций (см. п. 4.6 гл. 4).
3. Разработчики предлагают бесплатную лицензию для академических целей.

Всё это никоим образом не ограничивает преподавателей в использовании других средств моделирования сетей. Задания, приведенные ниже, могут быть реализованы с использованием и других средств моделирования сетей.

При определении номеров вариантов заданий, предлагается использовать табл. 6.1, при формировании которой выстраивались индивидуальные траектории для каждого студента (активные методики обучения – метод индивидуального обучения).

Таблица 6.1

Траектории вариантов для студентов

№	Л/р 1, номер варианта	Л/р 2, номер варианта	Л/р 3, номер варианта	Л/р 4, номер варианта	Л/р 5, номер варианта	Л/р 6, номер варианта	Л/р 7, номер варианта
Студент 1	1	2	3	4	5	6	6
Студент 2	2	3	4	5	6	7	7
Студент 3	3	4	5	6	7	8	8
Студент 4	4	5	6	7	8	9	9
Студент 5	5	6	7	8	9	10	10
Студент 6	6	7	8	9	10	1	1
Студент 7	7	8	9	10	1	2	2
Студент 8	8	9	10	1	2	3	3
Студент 9	9	10	1	2	3	4	4
Студент 10	10	1	2	3	4	5	5

6.1.1. Лабораторная работа № 1.**Модули Create, Resource, Process и Schedule**

Цель работы: Разработать имитационную модель согласно варианту задания. Научиться работать с модулями Create, Resource, Process и Schedule.

Порядок выполнения работы:

1. В ПП ARENA разработать модель согласно Вашему варианту.
2. Скопировав файл *.doe из лабораторной работы № 1 в новый файл, провести изменения, воспользовавшись модулем Schedule.
3. Подготовить отчет.

Вариант 1

В ЭВМ, работающую в системе управления технологическим процессом, через каждые 8 с поступают данные от датчиков. Обработка данных осуществляется с использованием *оперативной памяти* [длится 14 с, требует 2 условных единицы оперативной памяти (которая освобождается после обработки)] и *жесткого диска* [длится 17 с, требует 3 условные единицы жесткого диска]. Всего в системе присутствует 5 единиц оперативной памяти и 6 условных единиц жесткого диска.

Смоделируйте работу системы по обработке 100 заданий.

Измените начальные условия поступления данных: «поступают одновременно 3 набора данных от датчиков». Смоделируйте работу системы в течение 3 часов.

Пункт 2. Используя полученную и скопированную в новый файл модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Данные в вычислительную машину поступают каждые 20 минут. Количество поступающих данных определяется следующим множеством {1, 5, 3, 3, 4, 5, 6, 3, 3}.

Вариант 2

Система обработки данных, поступающих один раз в 4 с, состоит из трех специализированных устройств, которые последовательно обрабатывают поступающие пакеты данных. Устройства используют для обработки данных соответственно 1, 3 и 2 единицы *оперативной памяти* (которая освобождается после обработки), первое и третье устройство также требуют работы 4 и 5 единиц *записывающего устройства*. Всего в системе 5 единиц оперативной памяти и 600 единиц записывающего устройства. Обработка на первом устройстве занимает 10 с, на втором – 3 с, а на третьем – 8 с.

Смоделируйте работу системы по обработке 200 пакетов данных.

Измените начальные условия поступления данных: «поступают одновременно 2 пакета данных». Смоделируйте работу системы в течение 1 часа.

Пункт 2. Используя полученную и скопированную в новый файл модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Данные в вычислительную машину поступают каждые 3 минуты. Количество поступающих пакетов данных определяется следующим множеством {2, 1, 3, 2, 1, 1, 1, 2, 6, 3, 3, 2, 5, 3, 6, 7, 4, 2, 1, 5}.

Вариант 3

Банк данных состоит из двух машин, которые подсоединены к хранилищам, содержимое которых синхронизовано. На каждую машину каждые 5 с поступает запрос. Обработка запроса состоит из двух этапов: поиска запрошенных данных (9 с) и их отправки (11 с). Поиск занимает 3 условные единицы оперативной памяти (которая освобождается) каждой из машин (всего на машинах установлено по 7 единиц оперативной памяти), а также эксклюзивного доступа к хранилищу. Запрос данных требует 5 условных единиц оперативной памяти, но не требует эксклюзивного доступа к хранилищу.

Смоделируйте обработку 100 запросов.

Измените начальные условия поступления запросов: «поступают одновременно 3 запроса». Смоделируйте работу системы в течение часа.

Пункт 2. Используя полученную и скопированную в новый файл модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Запросы в каждую машину поступают в 0 (100 запросов) и 0:30 часов (75 запросов).

Вариант 4

Сетевое хранилище состоит из двух жестких дисков и одного накопителя на магнитной ленте. На вход каждого устройства каждые 3 с поступают пакеты данных объемом 5 условных единиц, которые необходимо записать. Жесткие диски обладают объемом 500 условных единиц, а накопитель на магнитной ленте – 200 условных единиц. Процесс записи на жесткий диск длится 4 с и требует 5 единиц питания, а на магнитную ленту – 2 с и 3 единиц питания (всего в системе присутствует 10 единиц питания).

Смоделируйте сохранение 70 пакетов данных.

Измените начальные условия поступления пакетов данных: «поступают одновременно 2 пакета данных». Смоделируйте работу системы в течение 4 часов.

Пункт 2. Используя полученную и скопированную модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Пакеты данных в каждый накопитель поступают в 0 (10 пакетов) часов, 1 (3 пакета) час, 2 (7 пакетов) часа, 3 (5 пакетов) часа и 4 часа (7 пакетов).

Вариант 5

В каждую из двух ЭВМ вычислительного центра каждые 5 с поступает пакет данных, которые необходимо обработать и передать на запись в общее внешнее хранилище. Обработка данных длится 9 с и требует 5 условных единиц *оперативной памяти* (которая освобождается после обработки). Запись во внешнее хранилище происходит в режиме эксклюзивного доступа и длится 10 с. На каждой ЭВМ присутствует 7 единиц оперативной памяти.

Смоделируйте обработку 300 пакетов данных.

Измените начальные условия поступления данных: «поступают одновременно 2 пакета данных». Смоделируйте работу системы в течение 3 часов.

Пункт 2. Используя полученную и скопированную модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Данные в вычислительную машину поступают каждые 15 минут. Количество поступающих пакетов данных определяется следующим множеством {10, 7, 5, 9, 3, 12, 6, 1, 6, 8, 3, 5}.

Вариант 6

В ЭВМ, работающую в системе управления технологическим процессом, через каждые 8 с поступает блок данных от датчиков. Обработка блока данных состоит из трех этапов: запись, сравнение и выдача управляющего воздействия. Длительность записи блока данных занимает 2 с и 3 единицы ресурса ЭВМ (всего ЭВМ состоит из 4 единиц ресурса). Длительность сравнения занимает 3 с и требует 2 единиц ресурса ЭВМ, по результатам сравнения генерируется управляющее воздействие. Длительность выдачи управляющего воздействия 1 с и требует 1 единицу ресурса.

Смоделируйте работу системы по обработке 350 блоков данных.

Измените начальные условия поступления данных: «поступают одновременно 2 пакета данных». Смоделируйте обработку в течение 1 часа.

Пункт 2. Используя полученную и скопированную в новый файл модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Данные в вычислительную машину поступают каждые 3 минуты. Количество поступающих пакетов данных определяется следующим множеством {2, 1, 3, 2, 1, 1, 1, 2, 6, 3, 3, 2, 5, 3, 6, 7, 4, 2, 1, 5}.

Вариант 7

В вычислительном центре имеются две ЭВМ. Задания на обработку поступают каждые 2 мин в пункт приема. Здесь в течение 12 мин они регистрируются и сортируются оператором, после чего каждое задание поступает на одну из ЭВМ. Примерно в 70 % заданий в результате их первой обработки на ЭВМ обнаруживаются ошибки ввода, которые сразу же в течение 3 мин исправляются оператором. На время корректировки ввода задание не освобождает соответствующей ЭВМ, и после корректировки начинается его повторная обработка. Возможность ошибки при повторной обработке исключается, т.е. повторная обработка всегда является окончательной. Продолжительность работы ЭВМ при обработке задания в каждом случае составляет 10. В центре имеется лишь одно рабочее место оператора.

Смоделировать процесс функционирования вычислительного центра при условии, что обработать необходимо 100 заданий.

Измените начальные условия поступления заданий: «поступают одновременно 2 задания». Смоделируйте обработку в течение 1 суток.

Пункт 2. Используя полученную и скопированную в новый файл модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Задания в вычислительный центр поступают каждый четный час. Количество поступающих заданий определяется следующим множеством {6, 3, 3, 2, 5, 3, 6, 7, 4, 2, 1, 5}.

Вариант 8

Банк данных состоит из двух ЭВМ, которые подсоединены к хранилищу. На каждую ЭВМ каждые 10 с поступает запрос. Обработка запроса состоит из двух этапов: поиска запрошенных данных и их отправки. Длительность поиска 2 с и требует 3 единицы ресурса ЭВМ, а длительность отправки 5 с и требует 2 единицы ресурса ЭВМ.

Смоделируйте обработку 500 запросов.

Измените начальные условия поступления запросов: «поступают одновременно 3 запроса». Смоделируйте работу системы в течение часа.

Пункт 2. Используя полученную и скопированную в новый файл модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Запросы в каждую машину поступают в 0 (30 запросов) и 0:30 часов (50 запросов).

Вариант 9

Система обработки данных состоит из двух специализированных устройств и общей оперативной памяти (2 единицы ресурса), которые последовательно обрабатывают поступающие пакеты данных.

Блок данных поступает каждые 7 с. Сначала первое устройство производит анализ данных, длительность которого 10 с и запись данных, которая требует 1 единицу ресурса памяти и 7 с. Затем второе устройство производит его непосредственную обработку, которая занимает 2 с, после чего также происходит запись данных, которая требует 2 единицы ресурса памяти и 9 с.

Смоделируйте обработку 100 блоков данных.

Измените начальные условия поступления данных: «поступают одновременно 2 пакета данных». Смоделируйте работу системы в течение 3 часов.

Пункт 2. Используя полученную и скопированную в новый файл модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Данные в вычислительную машину поступают каждые 30 минут. Количество поступающих пакетов данных определяется следующим множеством {20, 15, 35, 10, 27, 14}.

Вариант 10

Специализированная ЭВМ получает на вход задания трех типов: А, В и С. Задания типа А поступают раз в 15 с, типа В – раз в 10 с, типа С – раз в 7 с. Все задания предварительно записываются, процесс записи занимает 2 с. Затем задания поступают на обработку, причем каждый тип задания на выделенную ЭВМ – это обработка первого уровня, она занимает 10 с. После обработки на первом уровне задачи распределяются на две ЭВМ второго уровня случайным образом, что соответствует вероятности 50 %. Каждая ЭВМ имеет 1 ресурс.

Смоделируйте обработку 200 заданий.

Измените начальные условия поступления заданий: «поступают одновременно 2 задания». Смоделируйте работу системы в течение 2 часов.

Пункт 2. Используя полученную и скопированную в новый файл модель, проведите следующие изменения, воспользовавшись модулем Schedule.

Запросы в каждую машину поступают в 0 (100 заданий), 0:30 часов (75 заданий), 1:00 (25 заданий), 1:30 (50 заданий) часа и 2:00 (60 заданий).

6.1.2. Лабораторная работа № 2.

Генераторы случайных чисел и законы распределения

Цель работы: Разработать имитационную модель согласно варианту задания. Научиться задавать параметры случайных процессов в модулях Create и Process.

Порядок выполнения работы:

1. В ПП ARENA разработать модель согласно Вашему варианту.
2. Подготовить отчет.

Вариант 1

В ЭВМ, работающую в системе управления технологическим процессом, через каждые (8 ± 1) с поступает блок данных от датчиков. Обработка блока данных состоит из трех этапов: запись, сравнение и выдача управляющего воздействия. Длительность записи блока данных определяется нормальным распределением с базовым значением 2 с и максимальным отклонением 1 с, в результате записи 17 % данных теряется. Длительность сравнения занимает (3 ± 1) с, но чаще всего 2 с, по результатам сравнения генерируется управляющее воздействие. Длительность выдачи управляющего воздействия определяется экспоненциальным распределением с наиболее вероятным значением 1 с.

Смоделируйте работу системы по обработке 350 блоков данных.

Вариант 2

В вычислительном центре имеются три ЭВМ. Задания на обработку поступают с интервалом (20 ± 5) мин в пункт приема. Здесь в течение (12 ± 3) мин они регистрируются и сортируются оператором, после чего каждое задание поступает на одну из свободных ЭВМ. Примерно в 70 % заданий в результате их первой обработки на ЭВМ обнаруживаются ошибки ввода, которые сразу же в течение (3 ± 2) мин исправляются оператором. На время корректировки ввода задание не освобождает соответствующей ЭВМ, и после корректировки начинается его повторная обработка. Возможность ошибки при повторной обработке исключается, т.е. повторная обработка всегда является окончательной. Продолжительность работы ЭВМ при обработке задания в каждом случае составляет (10 ± 5) мин). В центре имеется лишь одно рабочее место оператора.

Смоделировать процесс функционирования вычислительного центра при условии, что обработать необходимо 100 заданий.

Вариант 3

Центр обработки запросов состоит из двух ЭВМ.

На первую ЭВМ запросы поступают в соответствии с экспоненциальным распределением с наиболее вероятным значением 1 с, а на вторую в соответствии с нормальным распределением с базовым значением 3 с и максимальным отклонением 1 с. Обработка запроса любой из ЭВМ длится (2 ± 1) с.

Смоделируйте работу системы в течение получаса.

Вариант 4

Банк данных состоит из двух машин, которые подсоединены к хранилищу. На каждую машину каждые (10 ± 2) с поступает запрос. Обработка запроса состоит из двух этапов: поиска запрошенных данных и их отправки. Длительность поиска определяется нормальным распределением с базовым значением 2 с и максимальным отклонением 1 с, а длительность отправки данных определяется экспоненциальным распределением с наиболее вероятным значением 5 с.

Смоделируйте обработку 500 запросов.

Вариант 5

На вход каждой из трех ЭВМ системы управления технологическим процессом каждые (5 ± 1) с поступает блок данных, который необходимо проанализировать и сохранить во внешнее хранилище.

Процесс анализа блока данных занимает (2 ± 1) с на первой ЭВМ, на второй и третьей – наиболее вероятно 2 с (в соответствии с экспоненци-

альным распределением). Запись во внешнее хранилище занимает 1 с с максимальным отклонением 0,5 с (в соответствии с нормальным распределением).

Смоделируйте работу системы в течение 4 часов.

Вариант 6

Система обработки данных состоит из двух специализированных устройств, которые последовательно обрабатывают поступающие пакеты данных.

Блок данных поступает каждые (7 ± 1) с. Сначала первое устройство производит анализ данных, длительность которого определяется экспоненциальным распределением с наиболее вероятным значением 1 с. Затем второе устройство производит его непосредственную обработку, которая занимает 2 с с максимальным отклонением 1 с в соответствии с нормальным распределением.

Смоделируйте работу системы в течение суток.

Вариант 7

В центре есть две ЭВМ. Запросы случайным образом поступают на обработку одной из ЭВМ через (10 ± 3) с, первичная обработка запроса занимает 2 с, выдача ответа требует (18 ± 4) с, но чаще всего 20 с, передача по каналу связи занимает в среднем 3 с. Временные характеристики второй ЭВМ аналогичны первой.

Смоделировать прохождение 400 запросов.

Вариант 8

Специализированная вычислительная система состоит из трех процессоров и общей оперативной памяти. Задания поступают на обработку через интервалы времени (5 ± 2) мин. После трансляции первым процессором в течение (3 ± 2) мин, но чаще всего 4,50 % заданий идут на второй процессор на редактирование, а 50 % сразу на решение на третий процессор. Редактирование происходит во втором процессоре и занимает $(2,5 \pm 0,5)$ мин. Отредактированные задания поступают в третий процессор на решение, требующее $(1,5 \pm 0,4)$ мин.

Смоделировать работу вычислительной системы в течение 30 ч.

Вариант 9

Обработка заданий состоит из двух этапов: расчет и тестирование. Задания поступают двух видов. Время поступления заданий первого вида – случайная величина, равномерно распределенная на интервале от 1 до 6 мин, второго вида – от (10 ± 2) мин. На расчет заданий первого ви-

да тратится от 4 до 10 мин, но чаще всего 6 мин, второго вида – от 6 до 9 мин. Тестирование любого задания равномерно распределено на интервале (1–3) мин.

Расчет и тестирование выполняют два процессора, причем задания первого вида обслуживаются только первым процессором, а задания второго типа – только вторым.

Вариант 10

В каждую из двух машин вычислительного центра каждые 5 с поступает пакет данных, который необходимо обработать и передать на запись в общее внешнее хранилище. Обработка данных длится (9 ± 4) с, но чаще всего 6 с. Запись во внешнее хранилище происходит в режиме эксклюзивного доступа и длится 10 с.

Смоделируйте обработку 300 пакетов данных.

6.1.3. Лабораторная работа № 3.

Модуль Decide

Цель работы: Разработать имитационную модель согласно варианту задания. Научиться работать с модулем Decide.

Порядок выполнения работы:

1. В ПП ARENA разработать модель согласно Вашему варианту.
2. Подготовить отчет.

Вариант 1

Специализированная ЭВМ получает на вход задания трех типов: А, В и С. Задания типа А поступают случайно раз в 15 с, типа В – раз в (10 ± 4) с, но чаще всего в 8 с, типа С – раз в (7 ± 2) с. Все задания предварительно записываются, процесс записи занимает 2 с. Затем задания поступают на обработку, причем каждый тип задания на выделенную ЭВМ, – это обработка первого уровня, она занимает 10 с. После обработки на первом уровне задачи распределяются на две ЭВМ второго уровня случайным образом, что соответствует вероятности 50 %.

Вариант 2

Распределенный банк данных системы сбора информации организован на базе ЭВМ, соединенных дуплексным каналом связи. Поступающий запрос обрабатывается на первой ЭВМ и с вероятностью 50 % необходимая информация обнаруживается на месте. В противном случае необходима посылка запроса во вторую ЭВМ.

Запросы поступают через (10 ± 3) с, первичная обработка запроса занимает 2 с, выдача ответа требует (18 ± 2) с, передача по каналу связи занимает 3 с. Временные характеристики второй ЭВМ аналогичны первой.

Смоделировать прохождение 400 запросов.

Вариант 3

На предприятии выпускаются микропроцессоры с периодом $(3 \pm 0,5)$ часа. Затем готовые изделия тестируются. Тестирование состоит из нескольких этапов: запись тестируемого серийного номера микропроцессора (занимает примерно 5 минут), затем все микропроцессоры с четными номерами проходят тестирование корпуса (занимает (25 ± 5) минут), а с нечетными номерами – тестирование элементов электроники (занимает (25 ± 5) минут). Тестирование в процессе работы проходят все микропроцессоры (занимает $(1,5 \pm 0,5)$ часа).

В общем процессе тестирования выявляется брак в 5 % случаев, каждый раз бракованное изделие посылается на доработку.

Вариант 4

ЭВМ обеспечивает процесс записи информации на диск и имеет в своем распоряжении два записывающих устройства. Задания на запись поступают в ЭВМ каждые (5 ± 4) мин. С вероятностью 0,5 ЭВМ опрашивает одно из записывающих устройств и передает задание на запись, если то свободно. В противном случае ЭВМ запрашивает другое устройство и таким образом продолжает сеансы связи, пока одно из устройств не освободится. Каждый сеанс опроса длится ровно 1 мин. ЭВМ допускает накопление у себя до пяти заданий, после чего вновь прибывшие задания получают отказ. Записывающие устройства выполняют задание на запись за (12 ± 8) мин.

Смоделировать работу системы в течение 10 час.

Вариант 5

Вычислительная система состоит из трех ЭВМ. С интервалом (3 ± 1) мин в систему поступают задания, которые с вероятностями: $P = 0,4$ идут на первую ЭВМ, с $P = 0,3$ адресуются второй ЭВМ, а все остальные идут на обработку на третью ЭВМ. Перед каждой ЭВМ имеется очередь заданий, длина которой не ограничена. После обработки задания на первой ЭВМ оно с вероятностью $P = 0,3$ поступает в очередь ко второй ЭВМ и с вероятностью $P = 0,7$ – в очередь к третьей ЭВМ. После обработки на второй или третьей ЭВМ задание считается выполненным.

Продолжительность обработки заданий на разных ЭВМ характеризуется интервалами времени: $T_1 = 4 \pm 1$ мин, $T_2 = 3 \pm 1$ мин, $T_3 = 5 \pm 2$ мин.

Смоделировать процесс обработки 200 заданий.

Вариант 6

Время поступления сообщений распределено равномерно на интервале (10 ± 2) с, причем 60 % сообщений поступает сначала на вторую ЭВМ.

Если сообщение поступает на первую ЭВМ, то затем оно поступает на вторую ЭВМ. Если поступает сначала на вторую, то потом на первую.

Время обработки сообщений на ЭВМ распределено равномерно на интервале (10 ± 4) с на первой ЭВМ, (9 ± 3) с на второй ЭВМ.

Промоделировать 1 час работы.

Вариант 7

Система обработки и хранения данных предприятия каждые 7 с получает на вход пакеты данных трех типов.

Данные каждого типа распределяются на выделенную ЭВМ. Обработка данных любой ЭВМ занимает (3 ± 2) с. В случае успешной обработки (85 % случаев) данные передаются во внешнее хранилище.

Смоделируйте работу системы в течение 8 часов.

Вариант 8

Специализированная вычислительная машина получает на вход задания двух типов. На первом уровне задания первого типа распределяются на первую машину, а второго – на вторую. После обработки на первом уровне задачи распределяются на две машины второго уровня случайным образом, что соответствует вероятности 50 %.

Обработка задания на машинах первого уровня занимает 7 с, а на машинах второго уровня – (4 ± 1) с.

Смоделируйте обработку 300 заданий.

Вариант 9

Распределенное хранилище данных организовано на базе ЭВМ, соединенных дуплексным каналом связи. В систему попадают запросы двух типов. Запросы первого типа распределяются на первую ЭВМ, а второго – на вторую.

Поступающий запрос обрабатывается на одной ЭВМ и с вероятностью 70 % необходимая информация обнаруживается на месте. В противном случае необходима посылка запроса на другую ЭВМ.

Запросы поступают каждые (8 ± 2) с, проверка наличия данных занимает 1 с, выдача ответа требует (4 ± 1) с, передача по каналу связи занимает 0,5 с. Временные характеристики второй ЭВМ аналогичны первой.

Смоделировать прохождение 700 запросов.

Вариант 10

В вычислительную машину, работающую в системе управления технологическим процессом, через каждые (8 ± 1) с поступает блок данных от датчиков одного из двух типов. Обработка блока данных состоит из двух этапов: анализ и непосредственно обработка.

После поступления блока данных он случайным образом распределяется на одно из двух устройств для анализа. Анализ занимает 1 с. Затем задания первого типа попадают на первое устройство обработки, а второго типа – на второе устройство обработки. Устройства обрабатывают задания в течение (3 ± 1) с.

Смоделируйте работу системы по обработке 100 заданий.

6.1.4. Лабораторная работа № 4.

Модули Batch, Separate и Match

Цель работы: Разработать имитационную модель согласно варианту задания. Научиться группировать сущности во временные/постоянные группы или по типам. Также уметь разъединять ранее созданные временные группировки и создавать дубликаты сущностей.

Порядок выполнения работы:

1. В ПП ARENA разработать модель согласно Вашему варианту.
2. Подготовить отчет.

Вариант 1

Сообщения поступают в вычислительную систему от датчиков типа А и Б. Сообщения типа А поступают (9 ± 4) с, а типа Б каждые 2 с. Перед тем как поступить на обработку, сообщения группируются в задания по 4. Задание проходит предварительную обработку, которая осуществляется двумя процессорами, задание случайным образом попадает на обработку к первому или второму процессору. Предварительная обработка занимает 10 с, в результате которой 24 % заданий бракуются и требуют повторной предварительной обработки. Успешно обработанные задания поступают в распределитель, где задания разбиваются на исходные сообщения. Распределитель собирает 2 сообщения типа А и 3 сообщения типа Б, когда сообщения обоих типов собраны, они записываются в хранилище по одному. Причем каждое сообщение записывается в основное и резервное хранилище.

Смоделируйте работу системы в течение 4 часов.

Вариант 2

С почтового сервера на электронный адрес менеджера отдела закупок поступают сообщения через (20 ± 5) мин. Треть из них обрабатывается в течение 60 мин и отправляется сообщение-ответ. Вторая треть сообщений – спам, который требует удаления в течение 1 мин. Третья треть сообщений предварительно обрабатывается за 30 мин перед комплектацией, которая требует наличия одного сообщения от клиента и двух сообщений-подтверждений: от поставщика и со склада. После того, как поступили все три сообщения, начинается процесс, который занимает (60 ± 2) мин обработки для сообщения клиента и (60 ± 8) мин для двух других, причем процессы обработки идут одновременно. При наличии на выходе одновременно всех трех сообщений, заказ считается выполненным.

Смоделировать работу в течение 100 ч.

Вариант 3

Специализированная вычислительная система состоит из трех процессоров и общей оперативной памяти. Задания, поступающие на обработку через интервалы времени (5 ± 2) мин, занимают объем оперативной памяти размером в страницу. После трансляции первым процессором в течение (5 ± 1) мин их объем увеличивается до двух страниц, и они поступают в оперативную память. Затем после редактирования во втором процессоре, которое занимает $(2,5 \pm 0,5)$ мин на страницу, объем возрастает до трех страниц. Отредактированные задания через оперативную память поступают в третий процессор на решение, требующее $(1,5 \pm 0,4)$ мин на страницу, и покидают систему, минуя оперативную память.

Смоделировать работу вычислительной системы в течение 50 ч.

Вариант 4

В хранилище данных на распределитель каждые 2 секунды поступает блок данных одного из двух типов: типа А или типа Б. Распределитель собирает 5 блоков типа А и 3 блока типа Б, после чего происходит проверка целостности собранных блоков данных, которая занимает (2 ± 1) с. Затем блоки данных записываются в хранилище по одному. Причем каждый блок данных записывается в основное и резервное хранилище.

Блоки данных поступают каждые 10 с. Смоделируйте работу системы в течение 8 часов.

Вариант 5

В вычислительный центр каждые 3 с поступают пакеты данных, которые необходимо проанализировать. Однако алгоритмы анализа оперирует не одним пакетом, а парой последовательных пакетов. С целью увеличения производительности пары пакетов группируются по 5 штук, а лишь затем отправляются на анализ, который занимает 1 с.

Кроме того, обнаружилось, что при анализе пакета данных периодически происходят ошибки. Для отладки алгоритма необходимо параллельно с анализом блока данных производить его запись в отдельное внешнее хранилище. Процесс записи лога во внешнее хранилище длится 0,5 с.

Вариант 6

Банк данных системы сбора информации организован на базе ЭВМ, которая обрабатывает поступающие каждые 3 с запросы.

Обработка запроса состоит из двух этапов: проверки наличия данных (занимает 0,5 с) и выдаче ответа (занимает 2 с).

ЭВМ оперирует не одним запросом, а парой последовательных запросов. В случае, когда запрашиваемая информация найдена, прежде чем отправить информацию клиенту, пары запросов группируются по 5 штук, а затем ответы на запросы отправляются клиенту по одному. Параллельно с отправкой результата клиенту его необходимо записать в резервное хранилище. Процесс записи во внешнее хранилище длится 0,5 с.

Смоделируйте прохождение 400 запросов.

Вариант 7

Специализированная вычислительная система имеет в своем составе две ЭВМ и внешнее хранилище. Каждые (8 ± 1) на каждую из ЭВМ поступает задание, результат обработки которого записывается в хранилище.

Для эффективной работы конвейера процессора ЭВМ оперирует тройками последовательных заданий. После обработки заданий процессором они группируются по 5 троек, которые подвергаются постобработке. Затем результаты обработки отправляются в хранилище по одному.

Кроме того, обнаружилось, что при обработке задания первой ЭВМ периодически происходят ошибки. Для отладки алгоритма необходимо параллельно с обработкой задания производить его запись в отдельное внешнее хранилище. Процесс записи лога во внешнее хранилище длится 0,5 с.

Время обработки задания процессором составляет $(2 \pm 0,5)$ с, время передачи результата обработки от ЭВМ до хранилища составляет 1 с, а время записи в хранилище – 0,5 с.

Смоделируйте обработку 300 заданий.

Вариант 8

Специализированная вычислительная система состоит из трех процессоров и общей оперативной памяти. Задания, поступающие на обработку через интервалы времени (5 ± 2) мин, занимают объем оперативной памяти размером в страницу. После трансляции первым процессором в течение (5 ± 1) мин их объем увеличивается до двух страниц, и они поступают в оперативную память. Затем после редактирования во втором процессоре, которое занимает $(2,5 \pm 0,5)$ мин на страницу, объем возрастает до трех страниц. Отредактированные задания через оперативную память поступают в третий процессор на решение, требующее $(1,5 \pm 0,4)$ мин на страницу, и покидают систему, минуя оперативную память.

Смоделировать работу вычислительной системы в течение 20 ч.

Вариант 9

Сообщения поступают на сервер каждые 2 с. После чего они группируются по 10 для дальнейшей обработки и передаче по каналу связи. Обработка выполняется процессором и занимает 20 с для каждого пакета из 10 сообщений. Передача пакета сообщений по каналу связи занимает (27 ± 9) с. После передачи пакет разгруппировывается на первоначальные сообщения. В процессе разгруппировки 2 % сообщений считаются утерянными.

Смоделируйте 8 часов работы. Подсчитайте общее количество переданных и потерянных сообщений.

Вариант 10

Сервером производится обработка информации. Процесс обработки имеет блочную структуру, а именно: происходит сбор пакета данных из отдельных сообщений, которые поступают на сервер постоянно каждую секунду. Сообщения поступают трех видов: пользовательские, технические и управляющие.

Процесс компоновки пакета имеет следующую последовательность:

1. Производится компоновка пользовательского и технического сообщения, в результате получается неразъемная пара. Этот процесс занимает от 0,5 до 1,5 с, но чаще всего 1 с.

2. Затем к неразъемной паре присоединяется управляющее сообщение, процесс занимает $(1 \pm 0,5)$ с.

После компоновки и обработки 10 % пакетов невозможно передать по каналу связи из-за проблем на втором этапе. Поэтому от неразъемной пары отделяется управляющее сообщение, и процесс повторяется.

6.1.5. Лабораторная работа № 5.

Приоритеты

Цель работы: Разработать имитационную модель согласно варианту задания. Научиться устанавливать приоритеты обработки сущностей. Необходимо уметь различать приоритет РЕСУРСА и приоритет ОЧЕРЕДИ.

Порядок выполнения работы:

1. В ПП ARENA разработать модель согласно Вашему варианту.
2. Подготовить отчет.

Вариант 1

Специализированная ЭВМ обрабатывает запросы двух типов. У второго типа запросов более высокий приоритет при предварительной обработке, которая занимает около 4 с. У запросов второго типа более высокий приоритет при основной обработке, которая занимает 3 с для запросов первого типа и (5 ± 2) с, но чаще всего 4 с, – у запросов второго типа.

Запросы первого типа поступают каждые 2 секунды, а второго – каждые 10 секунд. Смоделируйте работу системы в течение 2 часов.

Вариант 2

В специализированной вычислительной системе периодически выполняются три вида заданий, которые характеризуются уровнями приоритета: нулевым, первым и вторым. Каждый новый запуск задания оператор производит при помощи дисплея, работая на нем (50 ± 30) с. После запуска задания оно требует для своего выполнения (100 ± 50) с времени работы процессора, причем задания выполняются в соответствии с уровнем приоритета. Результаты обработки задания выводятся на печать без прерываний в течение (30 ± 10) с, после чего производится их анализ в течение (60 ± 20) с, и задание запускается снова. Можно считать, что при работе дисплея и при выводе результатов на печать процессор не используется.

Смоделировать процесс работы системы при условии, что задание второго уровня приоритета выполняется 100 раз.

Вариант 3

Вычислительный центр проводит обработку заданий трех классов А, В и С.

Исходя из наличия оперативной памяти ЭВМ задания классов А и В могут решаться одновременно, а задания класса С монополизируют ЭВМ.

Задания класса А поступают через (20 ± 5) мин, класса В – через (20 ± 10) мин и класса С – через (30 ± 10) мин и требуют для выполнения: класс А – (20 ± 5) мин, класс В – (21 ± 3) мин и класс С – (28 ± 5) мин. Задачи класса С загружаются в ЭВМ, если она полностью свободна. Задачи классов А и В могут дозагружаться к решаемой задаче. Смоделировать работу ЭВМ за 30 ч.

Вариант 4

К миниЭВМ подключено четыре терминала, с которых осуществляется решение задач. По команде с терминала выполняют операции редактирования, трансляции, планирования и решения. Причем, если хоть один терминал выполняет планирование, остальные вынуждены простаивать из-за нехватки оперативной памяти. Если два терминала выдают требование на решение, то оставшиеся два простаивают. Если работают три терминала, выдающих задания на трансляцию, то оставшийся терминал блокируется. Интенсивности поступления задач различных типов равны. Задачи одного типа от одного терминала поступают через экспоненциально распределенные интервалы времени со средним значением 160 с. Выполнение любой операции длится 10 с.

Вариант 5

В распределенный банк данных каждые 5 с приходят запросы с приоритетом от 1 до 5. На этапе проверки наличия данных, который занимает 0,5 с, обработка запросов происходит в соответствии с приоритетом. На следующем этапе получения данных, который занимает (2 ± 1) с, обработка происходит в порядке очереди.

Смоделируйте обработку 500 запросов.

Вариант 6

В ЭВМ, работающую в системе управления технологическим процессом, каждые (7 ± 1) с поступает блок данных от одного из двух типов датчиков. Затем информация обрабатывается в течение 2 с. Информация от датчиков первого типа имеет более высокий приоритет обработки. После обработки блока данных ЭВМ отправляет его во внешнее хранилище. Запись в хранилище длится 1 с и происходит с приоритетом, обратным приоритету обработки.

Смоделируйте обработку 250 блоков данных.

Вариант 7

В узел коммутации сообщений, состоящего из входного буфера, процессора и одной выходной линии, поступают сообщения с двух направлений. С первой линии поступают сообщения, которые требуют более срочной обработки.

Сообщения поступают каждые 3 с, обработка сообщения занимает 1 с, а передача сообщения от входного буфера до процессора занимает 0,5 с.

Смоделируйте работу узла коммутации в течение часа.

Вариант 8

В вычислительную систему периодически поступают три вида заданий, которые имеют приоритет от 1 до 3. Задания первого типа поступают каждые 7 с, второго – каждые 10 с, а третьего – каждые 15 с. Обработка задания занимает 3 с и происходит в соответствии с приоритетом. Как только результат выполнения задания получен, он в общем порядке записывается в резервное хранилище на базе накопителя на магнитной ленте (запись занимает 1 с).

Смоделируйте процесс работы системы в течение 8 часов.

Вариант 9

Вычислительным центром на обработку принимаются три класса заданий А, В и С. Исходя из наличия оперативной памяти ЭВМ, задания классов А и В могут решаться одновременно, а задания класса С монополизируют ЭВМ.

Задания класса А поступают через (20 ± 5) мин, класса В – через (20 ± 10) мин, класса С – через (30 ± 10) мин и требуют для выполнения: класс А – (20 ± 5) мин, класс В – (21 ± 3) мин, класс С – (28 ± 5) мин.

Задачи класса С загружаются в ЭВМ, если она полностью свободна. Задачи классов А и В могут выполняться одновременно.

Смоделировать работу ЭВМ за 20 часов.

Вариант 10

Обработка заданий состоит из двух этапов: расчет и тестирование. Задания поступают двух видов. Время поступления заданий первого вида – случайная величина, равномерно распределенная на интервале от 1 до 6 мин, второго вида – от 1 до 2 мин. На расчет заданий первого вида тратится от 4 до 10 мин, второго вида – от 6 до 9 мин. Тестирование любого задания равномерно распределено на интервале (1–3) мин.

Расчет и тестирование выполняет один процессор, причем задания первого вида при обслуживании имеют более высокий приоритет.

6.1.6. Лабораторная работа № 6. Модули Assign и Hold, построитель выражений (Build Expression)

Цель работы: Разработать имитационную модель согласно варианту задания. Научиться работать с модулями Assign и Hold, а также строить различные выражения, используя стандартные операторы и атрибуты сущностей, в ПП ARENA.

Порядок выполнения работы:

1. В ПП ARENA разработать модель согласно Вашему варианту.
2. Подготовить отчет.

Таблица 6.2

Стандартные операторы и атрибуты сущности

STATE(Resource 1)==IDLE_RES	Состояние Ресурса 1 – свободен
STATE(Resource 1)==BUSY_RES	Состояние Ресурса 1 – занят
NQ(Hold1.Queue)	Количество сущностей в очереди Hold 1
Process1.WIP	Количество сущностей в процессе
TNOW	Текущее время моделирования
Entity.Create.Time	Атрибут – время создания сущности
Entity.Serial.Number	Атрибут – серийный номер сущности
Entity.Type	Атрибут – тип сущности

Вариант 1

Вычислительным центром обрабатываются задания, поступающие через (300 ± 100) с, задания длиной (500 ± 200) байт. Скорость ввода, вывода и обработки заданий 100 байт/мин. Задания проходят последовательно ввод, обработку и вывод, буферизируясь перед каждой операцией. После вывода 5 % заданий оказываются выполненными неправильно вследствие сбоев и возвращаются на ввод. Для ускорения обработки задания в очередях располагаются по возрастанию их длины, т.е. короткие сообщения обслуживают в первую очередь. Задания, выполненные неверно, возвращаются на ввод и во всех очередях обслуживаются первыми.

Смоделировать работу вычислительного центра в течение 3 ч.

Вариант 2

Для ускорения прохождения «коротких» заданий на ЭВМ выбран пакетный режим работы с квантованием времени процессора. Это значит, что всем заданиям пакета по очереди представляется процессор на одинаковое время 10 с (круговой циклический алгоритм разделения времени). Если в течение этого времени заканчивается выполнение за-

дания, оно покидает систему и освобождает процессор. Если же очередного кванта времени не хватает для завершения задания, оно помещается в конец очереди – пакета. Последнее задание пакета выполняется без прерываний. Пакет считается готовым к вводу в ЭВМ, если в нем содержится 5 заданий. Новый пакет вводится в ЭВМ после окончания обработки предыдущего. Задания поступают в систему с интервалом времени (60 ± 30) с и характеризуется временем работы процессора (50 ± 45) с.

Смоделировать процесс обработки 200 заданий.

Вариант 3

Распределенный банк данных организован на базе трех удаленных друг от друга вычислительных центров А, В и С. Все центры связаны между собой каналами передачи информации, работающими в дуплексном режиме независимо друг от друга. В каждый из центров с интервалом времени (12 ± 5) мин поступают заявки на проведение информационного поиска.

Если ЭВМ центра, получившего заявку от пользователя, свободна, в течение (2 ± 1) мин производится ее предварительная обработка, в результате которой формируются запросы для центров А, В и С. В центре, получившем заявку от пользователя, начинается поиск информации по запросу, а на другие центры по соответствующим каналам передаются за 1 мин тексты запросов, после чего там также может начаться поиск информации, который продолжается: в центре А – (5 ± 2) мин, в центре В – (10 ± 2) мин, в центре С – (15 ± 2) мин. Тексты ответов передаются за 2 мин по соответствующим каналам в центр, получивший заявку на поиск. Заявка считается выполненной, если получены ответы от всех трех центров. Каналы при своей работе не используют ресурсы ЭВМ центров.

Смоделировать процесс функционирования распределенного банка данных при условии, что всего обслуживается 100 заявок.

Вариант 4

Специализированное вычислительное устройство, работающее в режиме реального времени, имеет в своем составе два процессора, соединенные с общей оперативной памятью. В режиме нормальной эксплуатации задания выполняются на первом процессоре, а второй является резервным. Первый процессор характеризуется низкой надежностью и работает безотказно лишь в течение (150 ± 20) мин. Если отказ происходит во время решения задания, в течение 2 мин производится включение второго процессора, который продолжает решение прерванного задания, а также решает и последующие задания до восстановления первого процессора. Это восстановление происходит за (20 ± 10) мин, по-

сле начинается решение очередного задания на первом процессоре, а резервный выключается. Задания поступают каждые (10 ± 5) мин и решаются за (5 ± 2) мин.

Смоделировать процесс работы устройства в течение 5 ч.

Вариант 5

Магистраль передачи данных состоит из двух каналов (основного и резервного) и общего накопителя. При нормальной работе сообщения передаются по основному каналу за (7 ± 3) с. В основном канале происходят сбои через интервалы времени (200 ± 35) с. Если сбой происходит во время передачи, то за 2 с запускается запасной канал, который передает прерванное сообщение с самого начала. Восстановление основного канала занимает (23 ± 7) с. После восстановления резервный канал выключается и основной канал продолжает работу с очередного сообщения. Сообщения поступают через (9 ± 4) с и остаются в накопителе до окончания передачи. В случае сбоя передаваемое сообщение передается повторно по запасному каналу.

Смоделировать работу магистрали передачи данных в течение 1 ч.

Вариант 6

ЭВМ обслуживает три терминала по круговому циклическому алгоритму, предоставляя каждому терминалу 30 с. Если в течение этого времени задание обрабатывается, то обслуживание завершается; если нет, то остаток задачи становится в специальную очередь, которая использует свободные циклы терминалов, т.е. задача обслуживается, если на каком-либо терминале нет заявок. Заявки на терминалы поступают через (30 ± 5) с и имеют длину (300 ± 50) знаков. Скорость обработки заданий ЭВМ равна 10 знаков/с.

Смоделировать 2 ч работы ЭВМ.

Вариант 7

В узел коммутации сообщений, состоящий из входного буфера, процессора, двух исходящих буферов и двух выходных линий, поступают сообщения с двух направлений. Сообщения с одного направления поступают во входной буфер, обрабатываются в процессоре, буферизируются в выходном буфере первой линии и передаются по выходной линии. Сообщения со второго направления обрабатываются аналогично, но передаются по второй выходной линии. Применяемый метод контроля потоков требует одновременного присутствия в системе не более трех сообщений на каждом направлении. Сообщения поступают через интервалы (15 ± 7) мс. Время обработки в процессоре равно 17 мс на со-

общение, время передачи по выходной линии равно (15 ± 5) мс. Если сообщение поступает при наличии трех сообщений в направлении, то оно получает отказ.

Смоделировать работу узла коммутации в течение 10 с.

Вариант 8

Для обеспечения надежности АСУ ТП в ней используется две ЭВМ. Первая ЭВМ выполняет обработку данных о технологическом процессе и выработку управляющих сигналов, а вторая находится в «горячем резерве». Данные в ЭВМ поступают через (10 ± 2) с, обрабатываются в течение 3 с, затем посылается управляющий сигнал, поддерживающий заданный темп процесса. Если к моменту посылки следующего набора данных не получен управляющий сигнал, то интенсивность выполнения технологического процесса уменьшается вдвое и данные посылаются через (20 ± 4) с. Основная ЭВМ каждые 30 с посылает резервной ЭВМ сигнал о работоспособности. Отсутствие сигнала означает необходимость включения резервной ЭВМ вместо основной. Характеристики обеих ЭВМ одинаковы. Подключение резервной ЭВМ занимает 5 с, после чего она заменяет основную до восстановления, а процесс возвращается к нормальному темпу. Отказы ЭВМ происходят через (300 ± 30) с. Восстановление занимает 100 с. Резервная ЭВМ абсолютно надежна.

Смоделировать 1 ч работы системы.

Вариант 9

Информационно-поисковая библиографическая система построена на базе двух ЭВМ и имеет один терминал для ввода и вывода информации.

Первая ЭВМ обеспечивает поиск литературы по научно-техническим проблемам (вероятность обращения к ней – 0,7), а вторая – по медицинским (вероятность обращения к ней – 0,3). Пользователи обращаются к услугам системы каждые (5 ± 2) мин. Если в очереди к терминалу ожидают 10 пользователей, то пользователи, прибывшие вновь, получают отказ в обслуживании.

Поиск информации на первой ЭВМ продолжается (6 ± 4) мин, а на второй (3 ± 2) мин. Для установления связи с нужной ЭВМ и передачи текста запроса пользователи тратят (2 ± 1) мин. Вывод результатов поиска происходит за 1 мин.

Смоделировать процесс работы системы за 2 ч.

Вариант 10

В ЭВМ, работающую в системе управления технологическим процессом, через каждые (3 ± 1) с поступает информация от датчиков и измерительных устройств. До обработки на ЭВМ информационные сообщения накапливаются в буферной памяти емкостью в одно сообщение. Продолжительность обработки сообщений на ЭВМ – (5 ± 2) с. Динамика технологического процесса такова, что имеет смысл обрабатывать сообщения, ожидавшие в буферной памяти не более 12 с. Остальные сообщения считаются потерянными.

Смоделировать процесс поступления в ЭВМ 200 сообщений.

6.1.7. Лабораторная работа № 7.

Статистика

Цель работы: Используя имитационную модель из 6-й лабораторной работы, научиться работать с данными и статистикой из стандартных отчетов ПП ARENA.

Порядок выполнения работы:

1. Взять разработанную Вами модель из 6-й лабораторной работы. Найти и/или рассчитать задания согласно Вашему варианту, а также провести дополнительные обязательные исследования по оптимизации модели. Докажите, что полученная модель – оптимальна по параметрам.

2. Подготовить отчет с использованием screen shorts из стандартных отчетов ARENA.

Вариант 1

Найдите:

1. Среднее количество заданий в очереди:
 - на ввод;
 - обработку;
 - вывод.
2. Количество неверно выполненных заданий.
3. Загруженность операторов
 - при вводе;
 - выводе.

Вариант 2

Найдите:

1. Среднее время формирования пакета из 5 заданий.
2. Загруженность процессора.
3. Среднюю длину задания.

Вариант 3

Найдите:

1. Среднее время ожидания ответа.
2. Общее время загрузки всех каналов связи.
3. Загруженность ЭВМ в центрах А, В и С:
 - при поиске информации;
 - при предварительной обработке.

Вариант 4

Найдите:

1. Среднее количество заданий, ожидающих обработки первым и вторым процессором.
2. Количество сбоев в системе.
3. Загруженность первого и второго процессора.

Вариант 5

Найдите:

1. Среднее количество сообщений, ожидающих передачи по основному каналу связи.
2. Количество сбоев в системе.
3. Загруженность основного и резервного канала.

Вариант 6

Найдите:

1. Среднее количество заданий в специальной очереди и среднее время ожидания в этой очереди.
2. Общее количество обработанных знаков.
3. Загруженность терминалов.

Вариант 7

Найдите:

1. Среднее количество сообщений в очереди на обработку процессором.
2. Количество сообщений, получивших отказ.
3. Загруженность процессора и выходной линии.

Вариант 8

Найдите:

1. Среднее количество данных на обработку к первой ЭВМ.
2. Количество сбоев в системе.
3. Загруженность первой и второй ЭВМ.
4. Общее количество обработанных данных.

Вариант 9

Найдите:

1. Среднее время ожидания обработки на первой и второй ЭВМ.
2. Количество пользователей, получивших отказ на обслуживание.
3. Загруженность первой и второй ЭВМ.
4. Общее количество пользователей.

Вариант 10

Найдите:

1. Среднее количество сообщений в очереди к ЭВМ.
2. Количество потерянных сообщений.
3. Загруженность ЭВМ.

6.2. Курсовое проектирование

При выполнении студентами курсового проекта предлагается использовать один из методов активного обучения, а именно: работа в команде/группе. Количество человек в команде зависит от объема выполняемого задания. В рамках данной дисциплины предполагается работа в командах по 2 человека.

Вариант 1

Вычислительная система включает три ЭВМ. В систему в среднем через 30 с поступают задания, которые попадают в очередь на обработку к первой ЭВМ, где они обрабатываются около 30 с. После этого задание поступает одновременно во вторую и третью ЭВМ.

Вторая ЭВМ может обработать задание за (14 ± 5) с, а третья – за (16 ± 1) с. Окончание обработки задания на любой ЭВМ означает снятие ее с решения с той и другой машины. В свободное время вторая и третья ЭВМ заняты обработкой фоновых задач.

Исследования разработанной модели проводить в следующих направлениях:

1. Смоделировать 4 часа работы системы.

Определить:

- необходимую емкость накопителей перед всеми ЭВМ (максимальные и средние значения);
- коэффициенты загрузки ЭВМ;
- время обслуживания основной и фоновой задачи (максимальные, минимальные и средние значения).

Определить производительность второй и третьей ЭВМ на решении фоновых задач при условии, что одна фоновая задача решается 2 мин.

2. Проанализировать зависимость приведенных выше характеристик от периода поступления на обработку основных и фоновых задач.

3. Самостоятельно провести исследования каких-либо других характеристик функционирования данной модели и их зависимостей между собой.

Вариант 2

В сеть включено 5 компьютеров. Извне (время поступления сообщений (5 ± 1) мин) в сеть поступает по адресам каждого компьютера информация по трем каналам связи. Каждый канал имеет скорость передачи 5 Кб/мин. Интенсивность поломок каждого канала составляет (40–80) мин. При поломке канала передачи информация передается полностью по другому каналу, который свободен и функционирует. При поломке всех трех каналов информация остается и передается по первому восстановленному каналу. Время восстановления каждого канала 4–8 минут.

Компьютеры имеют неравные возможности по отражению информации в печатном виде. Поэтому, после обработки информации (2–6 минут) компьютером, обработанная информация разделяются на три класса по виду печати в зависимости от объема данных:

- 1 класс – объем данных в интервале (5–10) Кб;
- 2 класс – объем данных в интервале (10–15) Кб;
- 3 класс – объем данных в интервале (15–20) Кб.

В зависимости от класса, данные отправляются по сети для печати на соответствующий компьютер, где минимальная очередь на печать.

Первый и пятый компьютер могут распечатать информацию 1 и 2 класса.

Второй и четвертый компьютер могут распечатать информацию 2 и 3 класса.

Третий компьютер может распечатать информацию 1 и 3 класса.

При поломке принтера на компьютере, куда посылалась информация, данные посылаются обратно и производится поиск подобного свободного принтера, способного распечатать информацию. При отсутствии таковых данные посылаются в буфер, где хранятся до восстановления принтера.

Вероятность поломки принтера – 10 %.

Время пересылки данных по сети – 1–2 мин.

Время восстановления принтера – 5–15 мин.

Время печати – 5–10 мин.

Объем, поступаемых в систему данных, – (5–20) Кб. Блоками по 5 Кб.

Оценить работу системы в течение двух суток по параметрам:

- загруженность каждого канала передачи, загруженность каждого компьютера;
- количество раз отказа каналов передачи информации;
- количество переданной информации по каждому каналу;
- количество раз отказа принтеров;
- загруженность принтеров.

Самостоятельно провести исследования каких-либо других характеристик функционирования данной модели и их зависимостей между собой.

Вариант 3

В компьютерной фирме ежедневно осуществляется сборка и продажа компьютеров. Работают на сборке 3 человека. Мастер может собрать и проверить от 10 до 15 компьютеров за день. Каждый из двух его помощников – от 5 до 10 компьютеров за день.

Комплектующие распределяются по следующим категориям: процессоры, материнские платы, видеоплаты, звуковые платы, оперативная память, жёсткие диски. При проверке собранных, не все компьютеры оказываются в рабочем состоянии (в среднем один из 10 собранных компьютеров не работает). Если компьютер собран мастером, то он сам выполняет диагностику и определяет, какая деталь неисправна (неисправной может быть любая или одновременно несколько деталей). Если же неисправен компьютер, собранный одним из помощников, то они отдают его мастеру на диагностику.

Частота попадания неисправных комплектующих распределяется следующим образом:

- жёсткие диски – 1–3 из 55,
- материнские платы – 1 из 20,

- звуковые платы – 1 из 30,
 - процессоры – 2–3 из 100,
 - видеоплаты – 1 из 40.
1. Определить (за день):
 - сколько всего собирается компьютеров;
 - сколько собирается неисправных компьютеров (и ремонтируется соответственно);
 - сколько неисправных деталей (и к каким категориям относятся эти детали) обнаруживается в ходе проверок;
 - сколько необходимо фирме комплектующих всех категорий;
 - сколько компьютеров собирает каждый сборщик.
 2. Исследовать приведенные выше характеристики в зависимости от изменения количества собираемых компьютеров каждым сборщиком, от изменения процента попадания неисправных деталей.
 3. Самостоятельно провести исследования каких-либо других характеристик функционирования данной модели и их зависимостей между собой.

Вариант 4

Магистраль передачи данных состоит из двух каналов (основного и резервного) и общего накопителя. При нормальной работе сообщения передаются по основному каналу за (7 ± 3) с. В основном канале происходят сбои через интервалы времени (200 ± 35) с. Если сбой происходит во время передачи, то за 2 с запускается запасной канал, который передает прерванное сообщение с самого начала.

Восстановление основного канала занимает (23 ± 7) с. После восстановления резервный канал выключается и основной канал продолжает работу с очередного сообщения.

Сообщения поступают через (9 ± 4) с и остаются в накопителе (буфере) до окончания передачи. В случае сбоя передаваемое сообщение передается повторно по запасному каналу.

Исследования разработанной модели проводить в следующих направлениях:

1. Смоделировать работу магистрали передачи данных в течение 10 часов. Определить при имеющихся в задании исходных данных:
 - загрузку основного канала;
 - загрузку резервного канала;
 - частоту отказов основного канала;
 - число прерванных сообщений;
 - среднее время передачи сообщений.

Провести десять экспериментов и найти средние значения требуемых параметров.

2. Провести сбор статистики для параметров:

- загрузка основного канала;
- загрузка резервного канала;
- среднее время передачи сообщений;
- максимальная длина очереди сообщений в буфере.

3. Самостоятельно провести исследования каких-либо других характеристик функционирования данной модели и их зависимостей между собой.

Вариант 5

В целях разгрузки основного канала, фирма – провайдер сети Интернет осуществляет подключение своих абонентов к внешнему миру через Прокси-сервер.

Прокси-сервер предназначен для того, чтобы в течение некоторого времени сохранять на своих винчестерах всю информацию, которая запрашивалась и была получена каким-либо абонентом (т.е. прокси-сервер записывает на свои магнитные носители целиком те WEB-страницы, которые раскрывались клиентом, и все полученные файлы). При поступлении от абонента заявки на получение пакета данных (при наборе пользователем какого-либо адреса в сети Интернет или при срабатывании какой-либо ссылки), сначала проверяется вся информация, лежащая на прокси-сервере. При отсутствии на нём нужных файлов данная ссылка по общему каналу связи со всемирной сетью поступает на конкретный запрашиваемый сервер.

Пусть в нашем случае мы имеем три телефонные линии для модемной связи с провайдером. Каждая линия постоянно находится в активном состоянии, т.е. клиент может в любой момент послать заявку.

Допустим, что все линии модемной связи абонентов с провайдером имеют одинаковую скорость передачи – 33600 бит/сек.

По мере поступления на прокси-сервер заявки от каждого клиента становятся в очередь.

Сканирование дискового пространства прокси-сервера занимает примерно 2–8 сек.

При обнаружении нужного пакета данных на прокси-сервере, данные по модемной линии передаются клиенту. Вероятность, что запрашиваемый пакет данных находится на прокси-сервере – 30 %. При отсутствии нужной информации заявка поступает в Интернет по общему каналу.

Пусть скорость обмена информацией в сети на данный момент составляет всего 1,5 Кбайт/сек.

Клиент в ответ на свою заявку получает по сети нужный ему пакет (от 10 до 100 Кбайт) или в случае сбоя – отказ. Вероятность возникновения отказа – 10 %. Необходимо учесть тот факт, что клиент не может послать новую заявку, не получив запрашиваемых данных или признак отказа по предыдущей заявке.

Пакеты данных сначала по общему каналу поступают на сервер провайдера, а потом по телефонной линии (модемная связь) к кому-либо из клиентов.

Исследования разработанной модели проводить в следующих направлениях:

Смоделировать работу магистрали передачи данных в течение 10 часов.

1. Определить при имеющихся в задании исходных данных:
 - количество переданных пакетов по общему каналу передачи;
 - объём принятой информации каждым из клиентов.
2. В зависимости от вероятности нахождения запрашиваемого пакета данных на прокси-сервере определить:
 - загруженность общего канала;
 - загруженность каждой станции приёма информации.
3. Самостоятельно провести исследования каких-либо других характеристик функционирования данной модели и их зависимостей между собой.

В результате выполнения проекта студент должен подготовить отчет, содержащий:

- титульный лист;
- содержание;
- введение;
- постановку задачи;
- краткую теоретическую часть;
- общее описание логики модели и саму модель;
- описание каждого модуля и сам модуль с его параметрами;
- отдельный раздел, посвященный проведению экспериментов и сбору статистики;
- заключение, содержащее выводы по проделанной работе;
- список использованных источников.

6.3. Семинар

Одним из основных методов использования инновационных форм и средств обучения являются семинары и/или конференции. В рамках данного курса предлагается использовать данный метод активного обучения.

После проведения первой лекции студентам на выбор предлагаются тематики работ в рамках изучаемой дисциплины, которые они должны исследовать и в середине семестра в рамках семинара сделать доклад в виде презентации, а также предоставить реферат работы преподавателю.

По результатам презентации преподаватель совместно со студентами, участниками семинара, оценивают каждое выступление согласно критериям оценочной таблицы. Студент, набравший большее количество баллов, поощряется, например, во время сдачи экзамена, он получает на один теоретический вопрос меньше, чем все остальные студенты.

Таблица 6.3

Пример оценочного листа для семинара

Студент-докладчик _____					
Глубина проработки материала	Широта проработки материала	Качество презентации	Умение вызвать интерес у слушателя	Наличие примеров	Ответы на вопросы

Оценка производится по 10-балльной шкале (0 – неудовлетворительно, 10 – отлично).

Общая сумма баллов _____

В качестве заданий/примерных тем для семинара могут выступать следующие:

1. Типы СМО.
2. Типы сетей Петри.
3. Другие (не аналитическое и имитационное) виды моделирования сетей.
4. Программные средства моделирования сетей на основе сетей Петри.
5. Программные средства моделирования сетей на основе СМО.
6. Другие программные средства моделирования сетей (не на основе СМО и сети Петри).

7. Пример моделирования сети на основе сети Петри.
8. Пример моделирования сети на основе СМО.
9. Законы распределения и генераторы случайных чисел.
10. Параметры моделирования сети и их отображение в статистике.

6.4. Выводы по главе

В шестой главе представлен практический материал дисциплины «Моделирование сетей ЭВМ и СТК». Практический материал состоит из заданий к лабораторному практикуму и методических рекомендаций по используемому программному средству (см. Приложение), курсового проектирования и семинара. Данные виды деятельности позволят студентам закрепить вышеизложенный теоретический материал.

ЗАКЛЮЧЕНИЕ

В последние годы в сфере российского образования произошли большие перемены: переход на двухуровневую систему подготовки «бакалавр-магистр», введение федерального государственного стандарта нового поколения, требование использования инновационных технологий обучения студентов. Таким образом, разработка учебного пособия по дисциплине «Моделирование сетей ЭВМ и СТК» для магистерской программы направления 230100 «Информатика и вычислительная техника» специализации «Сети ЭВМ и телекоммуникации» является неоспоримой необходимостью.

Первая и вторая глава пособия разработаны, в большей части, для студентов, которые окончили направления бакалавриата, не связанные с информатикой и вычислительной техникой, либо будут повторением для студентов, окончивших это направление. Так как эти главы посвящены основам теории моделирования, дана классификация моделей, этапы построения моделей, а также основы теории вероятности и математической статистики.

Третья глава описывает виды моделирования сетей: аналитическое и имитационное. Особое внимание уделено моделированию на основе систем массового обслуживания и сетей Петри.

В четвертой главе рассмотрен рынок средств моделирования сетей. Пятая глава приводит пример моделирования суперкомпьютерного кластера с использованием программного пакета Arena. Автор понимает, что в настоящее время в высших учебных заведениях при обучении студентов используют различные программные средства моделирования. Но использование пакета Arena является перспективным и апробировано на ряде зарубежных предприятий различных отраслей экономики. Тем более, что основа всех средств моделирования одна (сети Петри и СМО).

Шестая глава пособия содержит практический материал к выполнению лабораторных работ, курсового проекта и семинара. В приложении даны методические рекомендации по использованию предлагаемого программного пакета Arena.

Приложение

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

ПО РАБОТЕ С ПП ARENA

1. Начало работы с программным пакетом Arena 9.0

1.1. Создание модели с помощью ПП Arena 9.0

Для того чтобы создать новую модель, необходимо открыть ПП Arena 9.0 через Пуск → Rockwell Software→Arena 9.0→Arena 9.0.1. После запуска Arena автоматически открывается новый файл. Модули помещаются на панель методом «drug & drop», соединяются с помощью коннектора . Если модуль остается «горячим» (т.е. выделенным), то при помещении нового модуля на рабочую область (окно блок-схемы) эти модули автоматически соединяются друг с другом.

Среда моделирования Arena представлена на рис. П.1.

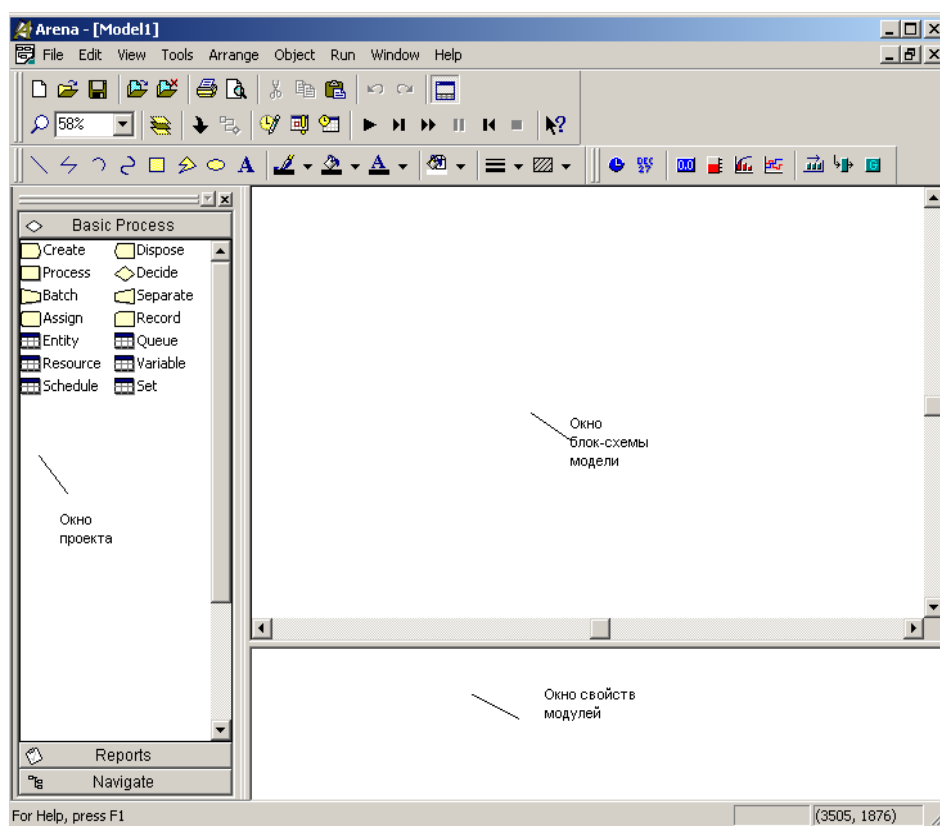


Рис. П.1. Среда моделирования Arena

Окно приложения разделено на три области:

1. *Окно рабочего поля модели*, в котором описывается логика модели с использованием схемных (графических) модулей. Окно рабочего поля представляет графику модели, включая блок-схему процесса, анимацию и другие элементы.

2. *Окно свойств модулей*, в котором отображаются свойства всех модулей (как модулей данных, так и схемных), имеющих и используемых в модели.

3. *Окно проекта* – это навигатор системы, в котором отображается рабочая панель со всеми модулями и другие доступные и открытые панели.

Окно проекта включает в себя несколько панелей:

1. Basic Process Panel (панель основных процессов) – содержит модули, которые используются для моделирования основной логики системы.

2. Advanced Process Panel (панель усовершенствованных процессов) – содержит дополнительные модули для создания моделей со сложной логикой процесса.

3. Advanced Transfer Panel (панель перемещения) – содержит специально разработанные блоки для моделирования процесса перемещения объектов с помощью транспортера или конвейера.

4 Reports (панель отчетов) – панель сообщений; содержит сообщения, которые отображают результаты имитационного моделирования.

5. Navigate (панель навигации) – панель управления; позволяет отображать все виды модели, включая управление через иерархические подмодели.

Таким образом, для того чтобы разрабатывать имитационные модели с использованием ПП Arena, необходимо изучить три основные панели: Basic Process Panel, Advanced Process Panel и Advanced Transfer Panel.

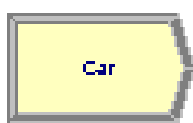
Каждая из этих панелей состоит из двух типов модулей: схемных модулей (Flowchart Modules) и модулей данных (Data Modules).

Рассмотрим более подробно состав каждой панели, свойства и назначение каждого модуля.

2. Basic Process Panel (панель основных процессов)

2.1. Схемные модули

Модуль Create



Этот модуль является отправной точкой для сущностей в имитационной модели. Сущности – это индивидуальные элементы, обрабатываемые в системе. Создание сущностей модулем происходит по рас-

писанию или же основываясь на значении времени между прибытиями сущности в модель. Покидая модуль, сущности начинают обрабатываться в системе. Тип создаваемых сущностей определяется в этом модуле.

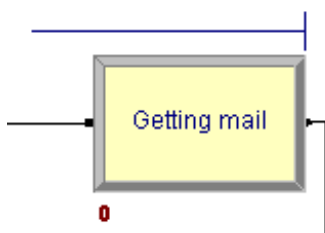
Применение: прибытие различных документов в сфере бизнеса (например: заказы, чеки, документация); прибытие клиентов в сфере обслуживания (например: в ресторан, в магазин); начало изготовления продукции на производственной линии.

Таблица П.1

Параметры модуля Create

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Entity Type	Название типа сущности, который будет создаваться модулем
Type	Способ формирования потока прибытия. Type может иметь значения: <i>Random</i> (используется экспоненциальное распределение со средним значением, определенным пользователем); <i>Schedule</i> (определяется модулем Schedule); <i>Constant</i> (будет использоваться постоянное значение, определенное пользователем) или <i>Expression</i> (поток прибытия будет формироваться по определенному выражению)
Value	Определяет среднее значение времени между прибытиями сущностей
Schedule Name	Имя расписания, которое определяет характер прибытия сущности в систему
Expression	Этот параметр задает тип распределения или любое выражение, определяющее время между прибытиями сущностей в модель
Units	Единицы измерения времени между прибытиями (день, час, минута, секунда)
Entities per arrival	Количество сущностей, входящих в систему за одно прибытие
Max arrivals	Максимальное число сущностей, которое может создать этот модуль (ресурс генератора)
First Creation	Время, через которое прибует первая сущность в модель, от начала моделирования

Модуль Process



Этот модуль является основным модулем процесса обработки сущностей в имитационной модели. В модуле имеются опции использования ресурсов, т.е., как и при любой обработке, захватываются какие-то ресурсы. Кроме стандартного модуля Process, можно использовать подмодель, придавая ей особую, определенную пользователем, иерархическую

логическую схему. В модуле можно также задавать добавочные стоимостные и временные характеристики процесса обработки сущности.

Наиболее частое применение модуля Process: проверка документов; выполнение заказов; обслуживание клиентов; обработка деталей.

Таблица П.2

Параметры модуля Process

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Type	Определяет логическую схему модуля. <i>Standard</i> означает, что логическая схема находится внутри модуля и зависит от параметра Action. <i>Submodel</i> показывает, что логическая схема будет находиться ниже в иерархической модели. Подмодель может содержать любое количество логических модулей
Action	Тип обработки, происходящей внутри модуля, может быть четырех типов: <i>Delay</i> просто показывает, что процесс занимает какое-то время и не отражает использование ресурсов; <i>Seize Delay</i> указывает на то, что в этом модуле были размещены ресурсы и будет происходить их захват и задержка, ресурсы будут захватываться (т.е. будут заняты обработкой сущности), а их освобождение будет происходить позднее с помощью какого-то другого модуля; <i>Seize Delay Release</i> указывает на то, что ресурсы были захвачены, а затем (через время) освободились; <i>Delay Release</i> означает, что ресурсы до этого были захвачены сущностью, а в таком модуле сущность задержится и освободит ресурс. Все эти параметры доступны только тогда, когда Type = Standard
Priority	Значение приоритета модулей, использующих один и тот же ресурс где угодно в модели. Это свойство недоступно, если Action = Delay (или Delay Release) или когда Type = Submodel
Resources	Определяет ресурсы или группы ресурсов, которые будут обрабатывать сущности в этом модуле
Delay Type	Тип распределения или процедура, определяющая параметры задержки
Units	Единицы измерения времени задержки (день, час, минута, секунда)
Allocation	Определяет стоимостные характеристики обработки. Value Added – означает учитывать стоимостные характеристики, а Non-Value Added – не учитывать
Minimum	Поле, определяющее минимальное значение для равномерного и треугольного распределения
Maximum	Поле, определяющее максимальное значение для равномерного и треугольного распределения
Value	Поле, определяющее среднее значение для нормального и треугольного распределения или значения для постоянной временной задержки
Std Dev	Параметр, определяющий стандартное отклонение для распределения
Expression	Поле, в котором задается выражение, определяющее значение временной задержки, если Delay Type = Expression

Более подробно остановимся на параметре Priority (приоритет) модуля Process. Говоря об этом параметре, мы должны ввести понятие «приоритет ресурса» и «приоритет очереди». Рассмотрим пример и объясним, что такое «приоритет ресурса».

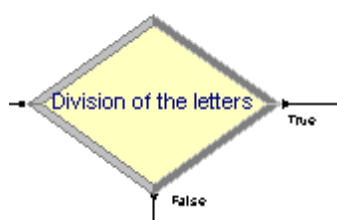
На прием к доктору приходят пациенты двух типов: взрослые и дети. Доктор (наш ресурс) – один. Он ведет прием и детей, и взрослых, но детей доктор принимает около 30 минут, а взрослых около 20 минут, причем у детей приоритет выше, чем у взрослых.

Каким образом мы можем реализовать это с помощью модуля Process? Во-первых, параметр Action этого модуля должен быть установлен Seize Delay Release для назначения ресурса, т.е. когда сущность «пациент» зайдет в модуль, то она захватит ресурс «доктор» на определенное время. Во-вторых, у нас по условию время обслуживания пациентов различное; таким образом, мы процесс обслуживания пациентов доктором смоделируем в виде двух блоков Process с разными временными задержками (в 30 и 20 минут), но одним и тем же ресурсом «доктор».

В-третьих, чтобы установить приоритет у детей выше, мы в параметре Priority в том процессе, где время обслуживания 30 минут, т.е. обслуживание детей, установим приоритет – High, а во втором процессе – Low или Medium. Таким образом, когда у нас будут приходить сущности «дети», они будут иметь наивысший приоритет в обслуживании.

Рассмотрение понятия «приоритет очереди» будет приведено ниже (см. модуль данных очередь Queue).

Модуль Decide



Этот модуль позволяет описать и задать логику модели, учитывая принятие решений. Он включает опции принятия решений, основанных на условии By Condition (например, если тип сущности Car) или основанных на вероятности By Chance (например, 75 % – true, а 25 % – false). Условия могут быть основаны на значении атрибута

Attribute, значении переменной Variable, типе сущности Entity Type или основанные на выражении Expression.

Если поставленное условие выполняется, то сущности будут покидать модуль через ветку True, иначе – по ветке False.

Данный модуль позволяет выполнять проверку не только одного условия, но и нескольких. Это достигается с помощью свойства Type→N-way by Chance/by Condition. В зависимости от условия сущность идет по

нужной ветке. Таким образом, по ветке True у модуля может быть любое количество выходов (по ветке False – всегда один выход).

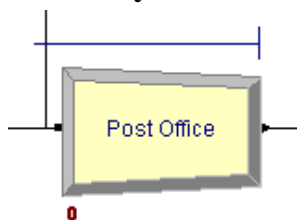
Применение: разделение дел на срочные дела и несрочные; перенаправление недоделанных или сделанных неправильно работ на доработку.

Таблица П.3

Параметры модуля Decide

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Type	Тип принятия решения: <i>By Chance</i> – выбор направления основывается на вероятности и <i>By Condition</i> – проверка на выполнение конкретно заданного условия
Percent True	Значение, определяющее процент сущностей, который пойдет по направлению True
If	Тип условия, которое будет проверяться на выполнение
Named	Имя переменной, атрибута или типа сущности, который будет проверяться при входе сущности в модуль
Is	Математический знак условия, например больше, меньше, равно и т.д.
Value	Значение, с которым будет сравниваться атрибут или переменная пришедшей сущности. Если тип условия – Expression, то в выражении должен стоять знак условия, например Color <> Red

Модуль Batch



Этот модуль отвечает за механизм группировки сущностей в имитационной модели. Группировка может быть постоянной или временной. Временно сгруппированные комплекты сущностей позднее могут быть разъединены с помощью модуля Separate. Комплекты могут состоять из любого числа входящих сущностей, определенного пользователем, или же сущности могут объединяться в комплект в зависимости от атрибута сущности. Временные и стоимостные характеристики выходящей сущности, представляющей комплект, будут равны сумме характеристик вошедших в группу сущностей.

Сущности прибывают в модуль, становятся в очередь и остаются там до тех пор, пока в модуле не будет набрано заданное количество сущностей. Когда соберется нужное число сущностей, создается сущность, представляющая комплект.

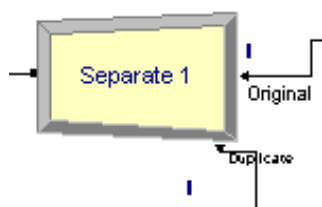
Применение: собрать необходимое количество данных, прежде чем начинать их обработку; собрать ранее разделенные копии одной формы; соединить пациента и его больничную карту приема к врачу.

Таблица П.4

Параметры модуля Batch

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Type	Способ группировки сущностей может быть: <i>Temporary</i> (временная) и <i>Permanent</i> (постоянная)
Batch Size	Число сущностей, образующих один комплект
Rule	Определяет, по какому признаку будут группироваться. Если Rule = Any Entity, – это значит, что первые 3 (если Batch Size = 3) сущности будут сгруппированы. Если Rule = By Attribute, то будет объединяться заданное количество сущностей с определенным атрибутом. Например, если Attribute Name = Color, то все сущности, имеющие одинаковое значение атрибута Color, будут сгруппированы
Attribute Name	Имя атрибута, по значению которого будут группироваться сущности

Модуль Separate



Этот модуль может использоваться в двух возможных вариантах:

1. Для создания копий входящих сущностей. Если модуль создает копии сущностей, то пользователь может задать количество дубликатов сущности. У дублированной сущности значения атрибута, а также анимационная картинка такие же, как и оригинала. Оригинальная сущность также покидает модуль.

2. Для разделения ранее сгруппированных сущностей. Правило для разделения стоимостных и временных характеристик копий сущностей и разделенных сущностей определяется пользователем. Когда временно сгруппированные сущности прибывают в модуль, они раскладываются на составные сущности. Сущности покидают модуль в той же последовательности, в которой они добавлялись в комплект.

Применение: разъединение ранее сгруппированных комплектов документов; для параллельной обработки счетов и документов по одному заказу.

Таблица П.5

Параметры модуля Separate

Параметры	Описание
Name	Уникальное имя модуля
# of Duplic	Количество создаваемых копий входящей сущности
Type	Способ разделение входящей в модуль сущности. <i>Duplicate Original</i> – просто делает дубликаты входящей сущности. <i>Split Existing Batch</i> проводит разгруппировку
Allocation Rule	Метод разделения стоимости и времени, если выбран Type=Split Existing Batch. Retain Original Entity Values сохраняет оригинальные значения сущностей. Take All Representative Values – все сущности принимают одинаковое значение. Take Specific Representative Values – сущности принимают специфическое значение

Модуль Assign



Этот модуль предназначен для задания нового значения переменной, атрибуту сущности, типу сущности, анимационной картинке сущности или другой переменной в системе.

В одном модуле можно сделать только любое количество назначений: сменить тип сущности, ее картинку, задать любое количество переменных и т.д.

Пример применения модуля Assign: установление приоритета для клиентов; присвоение номера вышедшему приказу.

Таблица П.6

Параметры модуля Assign

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Type	Тип назначения, которое будет осуществляться. Other может включать в себя встроенные переменные, такие как вместимость ресурса или конечное время моделирования
Variable Name	Имя переменной, которая будет изменяться в этом модуле
Attribute Name	Имя атрибута, который будет изменяться в этом модуле
Entity Type	Новый тип сущности, присваиваемый сущности в этом модуле

Entity Picture	Новая анимационная картинка для сущности, прошедшей этот модуль
New Value	Присваиваемое новое значение для атрибута, переменной

Модуль Record



Этот модуль предназначен для сбора статистики в имитационной модели. Модуль может собирать различные типы статистики, включая время между выходами сущностей из модуля, статистику сущности (время цикла, стоимость), статистику за период времени (период времени от заданной точки до текущего момента). Также доступен количественный тип статистики. Частое применение модуля: подсчитать, какое количество заказов было выполнено с опозданием; подсчитать количество работы, совершаемое за один час.

Таблица П.7

Параметры Модуль Record

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Type	Определяет тип статистики, которая будет собираться. <i>Count</i> будет увеличивать или уменьшать статистику на заданное значение. <i>Entity Statistics</i> будет собирать общую статистику о сущности, например: время цикла, стоимостные характеристики и т.д. <i>Time Interval</i> будет считать разницу между значением атрибута и текущим временем моделирования. <i>Time Between</i> будет отслеживать время между входом сущностей в модуль. <i>Expression</i> будет просто фиксировать значение, определяемое выражением
Attribute Name	Имя атрибута, значение которого будет использоваться для интервальной статистики
Value	Значение, которое будет добавляться к статистике, когда в модуль будет прибывать сущность

Модуль Dispose



Этот модуль является выходной точкой из имитационной модели. Статистика о сущности может собираться до того момента, пока она не выйдет из системы. Применение: окончание бизнес-процесса; клиенты покидают отдел.

Таблица П.8

Параметры модуля Dispose

Параметры	Описание
-----------	----------

Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Record Entity Statistics	Определяет, будет ли вестись статистика о выходе сущности из системы

2.2. Модули данных

Все модули данных в навигаторе панелей имеют одинаковый вид, т.к. они не отображаются физически в блок-схеме модели, в связи с этим их изображение не приводится. Также мы не будем рассматривать стоимостные параметры модулей, т.к. они не влияют на логику модели.

Модуль Entity

Этот модуль определяет тип сущности и ее анимационную картинку в имитационном процессе, также определяет стоимостную информацию. Для каждого источника должен быть определен тип сущности, который он генерирует.

Применение модуля Entity: документы (факсы, письма, отчеты и т.д.); люди в моделях больницы или магазина.

Таблица П.9

Параметры модуля Entity

Параметры	Описание
Entity Type	Название типа сущности
Initial Picture	Графическое представление сущности в начале имитационного процесса. Это значение может быть впоследствии изменено с помощью модуля Assign. Просмотреть анимационные картинки можно так: Edit/ Entity picture

Модуль Queue

Этот модуль данных предназначен для изменения правила расстановки сущностей в очереди, т.е. задается правило обслуживания сущности в процессе. По умолчанию тип очереди First in First out.

Применение: стопка документов, ожидающих освобождения ресурса; место для собирания частей, ожидающих упаковки (группировки).

Таблица П.10

Параметры модуля Queue

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Attribute Name	Имя атрибута, значение которого будет учитываться, если тип = Lowest Attribute Value или Highest Attribute Value
Type	Правило расстановки сущностей в очереди: <i>First in First out</i> –

	первый вошел, первый вышел; <i>Last in first out</i> – последний пришел, первый вышел; <i>Lowest Attribute Value</i> – первый выйдет из очереди тот, значение атрибута у которого низшее; <i>Highest Attribute Value</i> – первый выйдет из очереди тот, значение атрибута у которого наивысшее
--	---

Более подробно хотелось бы остановиться на параметре *Type*, т.к. именно с помощью него можно определить, что такое «приоритет очереди» и как его необходимо задавать. Рассмотрим несколько измененный наш пример.

На прием к доктору приходят пациенты двух типов: взрослые и дети. Доктор (наш ресурс) – один. Он ведет прием и детей, и взрослых, причем время приема одинаково (около 30 минут), но у детей приоритет при обслуживании выше, чем у взрослых.

Каким образом мы это можем реализовать? Во-первых, в модуле *Process* задается ресурс «доктор»; с помощью параметра *Action*, который устанавливаем *Seize Delay Release* для назначения ресурса. Таким образом, когда сущность «пациент» зайдет в модуль процесс, то она захватит ресурс «доктор» на определенное время (около 30 минут). Во-вторых, у нас по условию время обслуживания пациентов одинаковое, таким образом, мы процесс обслуживания пациентов доктором смоделируем в виде одного блока *Process*, с временной задержкой в 30 минут. Но здесь возникает вопрос: каким образом задать приоритет? В данном случае, мы рассматриваем ситуацию, когда ресурс задан в одном блоке, т.е. нет смысла менять параметр *Priority* модуля *Process*. В этом случае, возникает ситуация, когда приоритет не ресурса, а приоритет очереди. И задается он в модуле *Queue*. Необходимо выбрать, у какого типа сущности он выше. Это производится с помощью параметра *Type*: *Lowest Attribute Value* – первый выйдет из очереди тот, значение атрибута у которого низшее или *Highest Attribute Value* – первый выйдет из очереди тот, значение атрибута у которого наивысшее. Таким образом, когда у нас будут приходить сущности «дети», они будут иметь наивысший приоритет в обслуживании.

Модуль Resource

Этот модуль предназначен для определения ресурсов и их свойств в имитационном процессе; кроме того, модуль включает в себя стоимостную информацию о ресурсах и вместимость ресурсов. Ресурсы могут иметь фиксированную вместимость или же основанную на расписании. У ресурсов с фиксированной вместимостью в течение имитационного процесса вместимость изменяться не может. Ресурс должен быть связан с каким-либо процессом.

Применение: люди (клерки, продавцы, бухгалтеры, рабочие и т.д.); оборудование (телефонная линия, станок, компьютер).

Таблица П.11

Параметры модуля Resource

Параметры	Описание
Name	Имя ресурса
Type	Метод, определяющий вместимость ресурса. <i>Fixed Capacity</i> – фиксированная вместимость ресурса. <i>Based on Schedule</i> – вместимость ресурса определяется модулем Schedule
Capacity	Число ресурсов, находящихся в системе
Schedule Name	Имя Schedule модуля, который определяет вместимость ресурса, если Type = Based on Schedule
Busy / Hour	Почасовая стоимость обработки сущности ресурсом. Время учитывается только тогда, когда ресурс занят обработкой и прекращает учитываться, когда ресурс освобождается
Idle / Hour	Стоимость ресурса, когда он незанят
Per Use	Стоимость обработки ресурсом одной сущности (не зависит от времени)

Модуль Schedule

Этот модуль может использоваться вместе с модулем Resource для определения вместимости ресурса и с модулем Create – для задания расписания прибытия сущностей. Применение: расписание работы персонала с перерывами на обед; значение покупателей, прибывающих в супермаркет.

Таблица П.12

Параметры модуля Schedule

Параметры	Описание
Name	Название расписания
Type	Тип расписания, который может быть <i>Capacity</i> (расписание для ресурсов), <i>Arrival</i> (для модуля Create) или <i>Other</i> (разнообразные временные задержки или факторы)
Time Units	Масштаб оси времени в графике расписания

Модуль Set

Этот модуль данных, который описывает группу ресурсов, использующихся в модуле Process. В группе могут находиться несколько ресурсов. Модуль Set автоматически создает ресурсы, вместимость которых по умолчанию равна 1, и без всякой стоимостной информации.

Следовательно, если для ресурсов, входящих в группу, не нужно стоимостной информации и вместимость более 1, то можно обойтись созданием только модуля Set.

Возможно применение модуля для организации работы группы работников, например по очереди.

Таблица П.13

Параметры модуля Set

Параметры	Описание
Name	Название группы
Members	Перечисляет ресурсы, входящие в группу. Порядок перечисления ресурсов важен, когда в модуле Process используется правило выбора Cyclical или Preferred Order
Resource Name	Названия ресурсов, входящих в группу

Модуль Variable

Этот модуль данных определяет значение переменных. Переменные, относящиеся к модулю Decide или Assign, могут использоваться в выражениях. Если переменная не описана в этом модуле, то ее первоначальное значение равно 0.

Применение: число документов обрабатываемых в час; присвоение серийного номера для идентификации продукции.

Таблица П.14

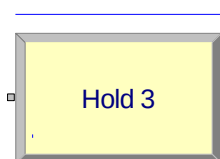
Параметры модуля Variable

Параметры	Описание
Name	Имя переменной
Initial Value	Первоначальное значение переменной. Это значение впоследствии может меняться модулем Assign
Rows	Число строк в размерной переменной
Columns	Число столбцов в размерной переменной
Clear Option	Определяет время, когда значение переменной сбрасывается в начальное значение. <i>Statistics</i> – сбрасывает переменную в начальное значение в любой момент, когда статистика была рассчитана. <i>System</i> – сбрасывает переменную в начальное значение в любой момент, когда система была рассчитана. <i>None</i> – никогда не сбрасывает переменную в начальное значение, исключая предшествующую первой репликации
Statistics	Определяет, будет ли вестись статистика по этой переменной

3. Advanced Process Panel (панель усовершенствованных процессов)

3.1. Схемные модули

Модуль Hold



Модуль Hold удерживает (захватывает) сущности. Процесс удержания может продолжаться до бесконечности или до выполнения условия.

Применение модуля: складироваться детали; пассажиры ожидают транспорт на остановке.

Таблица П.15

Параметры модуля Hold

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Type	Тип удержания сущности: <i>Infinite Hold</i> (удерживает до бесконечности, в этом случае у блока нет выхода), <i>Scan of Condition</i> (ожидает срабатывания определенного условия), <i>Wait of Signal</i> (ожидает сигнала, который вырабатывается только модулем Signal)

Если у модуля тип *Infinite Hold*, то забрать сущность из блока можно другими специальными модулями: *Remove*, *Signal* или *Pickup*. Соответственно, сущность выйдет по ветке именно из этих модулей, а не из *Hold*.

Поля *Queue Type* и *Queue name* присутствуют среди параметров модуля *Hold* всегда, задаются чаще всего автоматически (менять не рекомендуется).

Если тип имеет значение *Wait for signal*, то появляются поля *Wait for value* и *Limit* (ожидание конкретного значения сигнала и предел количества сущностей для освобождения из модуля *Hold*).

Если тип принимает значение *Scan of Condition*, то в этом случае становится доступным поле *Condition*, т.е. задержка напрямую зависит от выражения, заданную в этом поле.

Модуль Signal



Этот модуль посылает значение сигнала каждому модулю *Hold* в модели, в котором установлен тип *Wait for signal*, и освобождает заданное число сущностей.

Когда сущность прибывает в модуль Signal, то вырабатывается сигнал и посылается код сигнала в систему. В это время сущности в модуле Hold, который ожидает этого же сигнала, удаляются из очереди Hold и выходят из модуля.

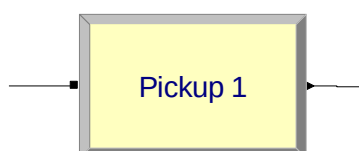
Применение: прием преподавателем экзамена у определенного количества студентов; ожидание людьми определенного автобуса.

Таблица П.16

Параметры модуля Signal

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Signal value	Значение посылаемого сигнала для модуля Hold
Limit	Число сущностей, которые будут освобождены из модуля Hold, когда сигнал будет получен

Модуль Pickup



Этот модуль предназначен для удаления определенного количества последовательно стоящих сущностей из определенной очереди. Сущности, которые удаляются из очереди, добавляются в конец сущности, вошедшей в блок Pickup. Чаще всего используется для удаления сущностей из модуля Hold, при условии, что тип Infinity Hold (без выхода). В модуле Pickup задается имя очереди, из которой будут забираться сущности, и определяется количество забираемых сущностей. Все сущности (вместе с исходной) выйдут из модуля Pickup в виде временной группировки.

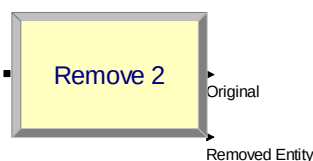
Применение: развоз товаров по магазинам со склада; посадка пассажиров в автобус на автобусной остановке.

Таблица П.17

Параметры модуля Pickup

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Quantity	Количество сущностей, которые должны быть удалены из очереди
Queue Name	Имя очереди, из которой будут удаляться сущности
Starting Rank	Позиция сущностей в очереди, с которой начинается удаление

Модуль Remove



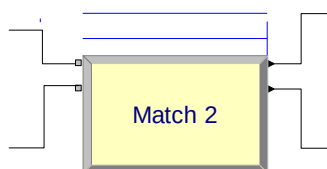
Модуль предназначен для удаления сущностей из любой очереди при условии, что эти сущности задерживаются бесконечно (Infinity). Отличие этого модуля от других заключается в том, что он может забрать только одну сущность из очереди. И у этого модуля 2 выхода: original и removed entity. По ветке original выходит та сущность, которая зашла (активировала) в этот модуль, а по ветке removed entity выходит та сущность, которая была забрана из очереди другого модуля (чаще всего модуля Hold).

Таблица П.18

Параметры модуля Remove

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Queue name	Название очереди, из которой будет произведено удаление
Rank of entity	Глубина удаления (количество сущностей для удаления)

Модуль Match



Этот модуль предназначен для синхронизации движения двух или более сущностей, расположенных в различных, несвязанных очередях. Количество очередей может варьироваться от 2 до 5. Сущность ждет в очереди до тех пор, пока в остальных очередях не появятся другие сущности либо с таким же значением атрибута, как и у исходной сущности.

Применение: сборка частей детали для дальнейшей обработки; соби́рание различных, но строго определенных продуктов по заказу клиента; синхронизация выхода покупателя с выходом заполненного заказа

Таблица П.19

Параметры модуля Match

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Number to Match	Количество очередей для синхронизации сущностей

Type	Метод сравнения входящих сущностей для синхронизации. Значения: <i>Any Entities</i> – в каждой очереди должно быть по одной любой сущности, для того чтобы они вышли. <i>Based on Attribute</i> – в каждой очереди должна быть хотя бы одна сущность с таким же атрибутом для выхода
Attribute Name	Название атрибута, по которому сущности должны сравниваться. Используется, только если установлен тип Based on Attribute

Модуль Dropoff



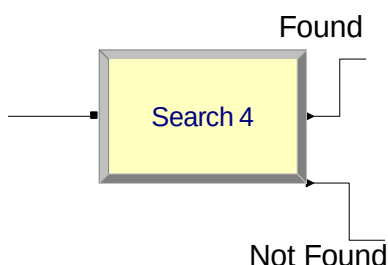
Модуль Dropoff перемещает определенный набор сущностей из группы сущностей и посылает их в другой модуль, связанный с ним графическим соединением. В этот модуль приходит временная группировка, из которой мы можем выделить требуемое количество сущностей, они пойдут по ветке Members, оставшаяся группа (в виде одной сущности) пойдет по ветке Original.

Таблица П.20

Параметры модуля Dropoff

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Quantity	Число сущностей, которое будет выбрано из всех пришедших в группу сущностей
Starting Rank	Начальное значение выбрасываемой сущности
Member Attributes	Метод определения того, как назначить значение атрибута представленной сущности для выброса оригинальных сущностей
Attribute Name	Название атрибутов сущности, которые обозначены для выброса оригинальной сущности из группы

Модуль Search



Этот модуль необходим для поиска определенного элемента в очереди, в пакете либо в каком-то выражении. Он имеет два выхода: True, если элемент найден, и False, если элемент не найден.

Применение: поиск среди коробок самой легкой.

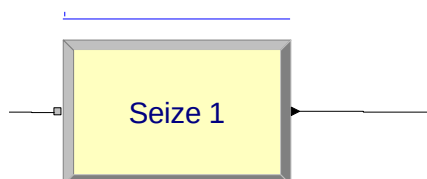
Таблица П.21

Параметры модуля Search

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме

Type	Тип поиска: среди сущностей, объединенных в очередь; сущности, объединенные в пакет или поиск выражения
Queue Name	Имя очереди, в которой будет осуществляться поиск
Starting Value	Начальный класс в очереди или в пакете или начальное значение для переменной в выражении
Ending Value	Конечный класс в очереди или в пакете или конечное значение для переменной в выражении
Search condition	Условия, включающие индекс поиска выражений, или содержащие атрибут при поиске пакетов или сущностей в очереди

Модуль Seize



Модуль Seize предоставляет сущности один или несколько ресурсов. Он может быть использован для того, чтобы захватывать отдельный ресурс, ресурс из набора ресурсов или ресурс, определённый альтернативным методом, таким как атрибут или выражение.

Когда сущность поступает в этот модуль, она ждёт в очереди, пока определённые в этом модуле ресурсы не будут доступны. Также здесь определяется тип распределения ресурсов для поступивших сущностей.

Замечания

1. Сущности, которые захватываются с более высокой величиной приоритета, имеют более высокий приоритет, чем сущности, которые захватываются с более низкой величиной. Приоритетные выражения, оцененные как отрицательные величины, рассматриваются как нулевой приоритет. Если несколько сущностей с равными приоритетами пытаются захватить один и тот же ресурс, то его получает сущность с наибольшим временем ожидания.

2. Возможно определить набор состояний (State set) для ресурса и назначить состояние ресурса в определённых ситуациях, используя область состояния ресурса (Resource State Field). Затем можно собрать статистику: сколько времени приходится на каждое состояние ресурса.

Таблица П.22

Параметры модуля Seize

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Allocation	Определяет категорию, по которой будет распределена стоимость использования ресурса для сущности, проходящей через модуль Seize
Priority	Приоритет сущности, ожидающей в этом модуле ресурс. Определяется в случае, когда одна или несколько сущностей

	из других модулей ожидают тот же ресурс (1 – высокий, 2 – средний, 3 – низкий, др.)
Type	Тип ресурса, который должен быть захвачен. Определяет конкретный ресурс или выбирает набор ресурсов. Имя ресурса также может быть определено атрибутом или выражением (Resource, Set, Attribute, Expression)
Resource name	Имя ресурса, который должен быть захвачен
Selection rule	Метод выбора среди доступных ресурсов в наборе

Модуль Delay



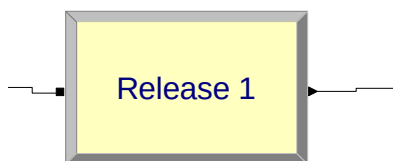
Модуль Delay задерживает сущность на определённое количество времени. По прибытии сущности в модуль выражение времени задержки оценивается и сущность остаётся в модуле на результирующее время. Затем время выделяется и, в зависимости от Allocation, либо добавляется к значению сущности, либо не добавляется, либо передаётся, либо ждет другое время. Также стоимости складываются, вычисляются и выделяются.

Таблица П.23

Параметры модуля Delay

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Allocation	Тип категории, в которой сущности могут быть подвергнуты задержке времени и добавлению стоимости
Delay Time	Определяет значение задержки времени для сущности
Units	Указывает единицу измерения задержки времени

Модуль Release



Модуль Release используется для того, чтобы освобождать ресурсы, которые прежде были захвачены сущностью. Этот модуль может быть использован для того, чтобы освобождать индивидуальные ресурсы или ресурсы в пределах набора. Для каждого ресурса, который нужно освободить, определяется имя и количество. Когда сущность поступает в модуль, она теряет управление определённым ресурсом. Любые сущности, ожидающие в очередях этот ресурс, получают его немедленно.

Замечания:

1. Если есть сущность, ожидающая в очередях для захвата определённого ресурса, то, когда ресурс освобождается, он автоматически распределяется в ждущую сущность. Эта ждущая сущность будет обработана, как только сущность, которая освободила ресурс, переместится.

2. Системная переменная NR (имя ресурса) возвращает номер последнего занятого ресурса. Когда сущность поступает в модуль Release, NR уменьшается на количество освобождённых ресурсов, если ресурс не будет немедленно захвачен другой сущностью.

3. Если освобождается больше количество ресурсов, чем было ранее захвачено, то происходит ошибка.

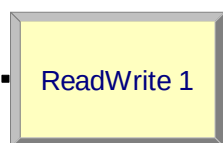
4. Освобождение множества ресурсов выполняется в порядке их появления в модуле Release.

Таблица П.24

Параметры модуля Release

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Resources	Имя освобождаемых модулем ресурсов

Модуль ReadWrite



Модуль ReadWrite используется для чтения данных из входного файла или с клавиатуры и задания данных в список переменных или атрибутов (или других выражений). Этот модуль также используется, чтобы записать данные на выходное устройство, например на экран или в файл.

Когда объект приходит в модуль ReadWrite, то файл изучается для того, чтобы увидеть, открыт он или нет. Если нет, файл автоматически открывается. Величины атрибутов, переменные или выражения будут прочитаны или записаны в зависимости от того, какой формат определен.

Таблица П.25

Параметры модуля ReadWrite

Параметры	Описание
Name	Уникальный модульный идентификатор. Это имя отображается в модульной форме
Type	Метод использования (чтение или запись). Данные могут быть записаны в файл или на экран. Данные могут быть считаны из файла или с клавиатуры
Arena File Format	Имя файла, чтобы идентифицировать файл в пределах модуля File

Overriding File Format	Формат для записи или чтения данных. Этот формат аннулирует любой формат, определенный в структурной области модуля File. FORTRAN или C может использоваться, чтобы описать тип и позицию каждой области
Variable Type	Тип информации, что будет прочитана или записана
Attribute Name	Определяет символьное имя атрибута для записи или чтения
Variable Name	Определяет символьное имя переменной для записи или чтения
Other	Определяет выражение для чтения или записи других типов информации

3.2. Модули данных

Модуль Advanced Set

Этот модуль определяет наборы (очереди, хранилищ или другие наборы) с соответствующими его составляющими. Набор определяет группу схожих элементов, к которым можно обращаться через имя и индекс. К элементам, входящим в набор, можно обращаться как к членам этого набора.

Наборы очередей могут быть определены при помощи модуля Seize.

Таблица П.26

Параметры модуля Advanced Set

Параметры	Описание
Name	Уникальный идентификатор
Set Type	Тип набора. Может быть Queue, Store, Other (другой)
Members	Задаются конкретные составляющие (очереди, хранилища), входящие в набор

Модуль Expression

Модуль Expression позволяет определять выражения и задавать им значения. К выражению обращаются при помощи имени. Выражения могут быть заданы как одномерный или двумерный массив.

Таблица П.27

Параметры модуля Expression

Параметры	Описание
Name	Уникальное имя выражения
Row	Максимальное количество строк в определяемом выражении
Column	Максимальное количество столбцов в определяемом выражении. Данное свойство задается, только когда задано свойство Row

Expression Value	Значение, которое соответствует выражению
------------------	---

Этот модуль необходим для того, чтобы задавать какие-то часто используемые выражения, чтобы разгрузить модель, например в модулях Decide, Hold, Pickup.

Модуль Statistic

Модуль Statistic используется для того, чтобы определить дополнительную статистику, которая должна собираться в течение времени моделирования, а также чтобы определить файлы выходных данных.

Таблица П.28

Параметры модуля Statistic

Параметры	Описание
Name	Уникальное имя модуля
Type	Тип статистики. Тип может быть time-persistent, tallies (observational data), count-based, outputs, and frequency-based

В зависимости от выбранного типа статистики появляются дополнительные поля.

1. Если выбран тип Tally: Tally Name – определяет символьное имя для типа статистики Tally, Tally Output File – имя выходного файла.

2. Если выбран тип Counter: Counter Name – определяет символьное имя для типа статистики Counter; Limit определяет лимит счетчика; Counter Output File – имя выходного файла.

Модуль Storage

Модуль Storage определяет имя Хранилища. Хранилище автоматически создается любым модулем, который на него ссылается.

Модуль File

Модуль File должен быть включен всякий раз, когда обращаются к внешнему файлу, используя ReadWrite модуль. Этот модуль выделяет системный файл, называет и определяет метод доступа, форматирование и эксплуатационные характеристики файла.

Таблица П.29

Параметры модуля File

Параметры	Описание
Operating System File Name	Операционное системное имя, путь к файлу, откуда читаем или записываем. Символьная строка
Structure	Тип файловой структуры. Неформатированный, свободный формат, WorksSheet, специфические C- или FORTRAN-форматы
End of File Action	Тип действия, которое произойдет, когда будет достигнут конец файла. Ошибка, выход, на начало, игнорировать
Comment Character	Символ, указывающий отображение комментирующей записи. Одиночный символ

Модуль StateSet

Модуль используется для того, чтобы определить состояние ресурса или набора ресурсов. Состояния могут быть связаны с автосостоянием или могут быть заданы новые состояния для ресурса. Модуль Resource в базовой панели Process ссылается на StateSet, который данный ресурс будет использовать.

Таблица П.30

Параметры модуля StateSet

Параметры	Описание
StateSet Name	Название набора состояний, которые могут быть назначены ресурсу в течение модельного времени
State Name	Имя пользователя определившего состояние
Auto State or Failure	Используется, чтобы связать State Name с автосостоянием или с заданным пользователем, именем отказа

Модуль Failure

Модуль Failure разработан для использования с ресурсами, а именно: для имитации отказов ресурса. Может использоваться для ресурсов с однократной способностью или для ресурсов многократной способности, когда индивидуальные единицы ресурса заняты в одно и то же время.

Таблица П.31

Параметры модуля Failure

Параметры	Описание
Name-	Имя отказа
Count	Определяет число ресурсов, реализуемых для отказов
Time	Определяет время для отказов
Up Time	Определяет время между отказами (число)
Up Time Units	Задаем формат времени (секунда, минута, час, день)
Down Time	Определяем продолжительность отказа (число)



Down Time Units	Задаем формат времени (секунда, минута, час, день)
-----------------	--

4. Advanced Transfer Panel (панель перемещения)

4.1. Схемные модули

Модуль Station

Модуль Station определяет станцию или набор станций для физической или логической обработки, некая логическая («отправная») точка в модели.

Таблица П.32

Параметры модуля Station

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Station Type	Тип станции
Station Name	Имя станции
Set Name	Уникальное имя набора станций
Save Attribute	Название атрибута, куда будут сохраняться значения атрибутов сущностей
Station Set Members	Перечисляется набор станций

Модуль Route



Модуль Route позволяет принять указанную сущность на заданную станцию, при этом позволяет имитировать время, которое будет затрачено сущностью на дистанцию к заданной станции.

Таблица П.33

Параметры модуля Route

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Route Time	Время прохода через этот модуль
Units	Единицы измерения времени задержки (день, час, минута, секунда)

Destination Type	Тип станции назначения, на которую должна прибыть сущность (Station, Sequential, Attribute, Expression)
------------------	---

Модуль PickStation



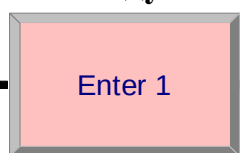
Модуль PickStation позволяет сущностям выбирать определенную станцию из множества существующих (маршрутизатор).

Таблица П.34

Параметры модуля PickStation

Параметры	Описание
Name	Уникальное имя блока
Test Condition	Определяется тип выбора станции (минимум или максимум по полям): <i>Number In Queue</i> (количество в очереди); <i>Number En Route to Station</i> (количество маршрутизированных станций); <i>Number of Resources Busy</i> (количество занятых ресурсов) и <i>Expression</i> (выражение)
Route Time	Время в пути (до станции)
Units	Единицы измерения времени пути (день, час, минута, секунда)
Save Attribute	Имя атрибута, который хранит имя станции
Transfer Type	Определяет, каким образом сущности будут транспортироваться до следующей станции (Route, Transport, Convey or Connect)

Модуль Enter



Модуль Enter определяет станцию (или станции), соответствующую физическим или логическим позициям, где происходит обработка. Если модуль Enter определяет конкретную станцию, он эффективно определяет многочисленные обработки позиций. Станция (или каждая станция в пределах решаемого комплекта) соотносится к области деятельности, которая используется, чтобы сообщить о времени и издержках, повышенных сущностями, на этих станциях. Эта сущность имени AreaTs также называется станцией.

Сущность может переместиться из предыдущего модуля в модуль Enter, причем двумя способами: отправлением на станцию, связанную с модулем дистанционно или через реальное графическое соединение.

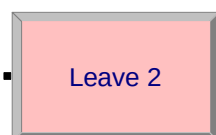
Когда сущность прибывает в модуль Enter, «разгружая», может произойти задержка и любое действие с передачей.

Таблица П.35

Параметры модуля Enter

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Station Type	Определяет индивидуальную станцию или комплект станций, чтобы определить точку входа в этот модуль. Если выбран комплект (set), это указывает, что этот модуль входит в подмодель станции
Station Name	Имя станции активно в том случае, когда выбран тип Type Station
Parent Activity Area	Имя места отправления
Delay	Время задержки сущности по прибытии на данную станцию
Allocation	Тип категории, к которому будет добавляться время сущности и цена
Transfer In	Если выбран ресурс (транспортёр или конвейер), чтобы доставить сущность к станции, используется для «отпускания», «освобождения», или «выхода»

Модуль Leave



Этот модуль используется для передачи сущности к станции или другому модулю.

Когда сущность прибывает в модуль Leave, она ожидает прибытия транспорта, когда прибывает транспорт, тратится время на загрузку и в конечном итоге сущность отправляется в пункт модуля назначения.

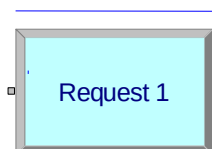
Таблица П.36

Параметры модуля Leave

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Allocation	Тип категории, к которому будет добавляться время сущности
Delay	Время задержки сущности по прибытии на данную станцию
Unit	Величина задержки: день, час, минута, секунда
Transfer Out	Тип, содержащий запрос на транспорт

Далее будут подробно рассмотрены модули транспортера.

Модуль Request



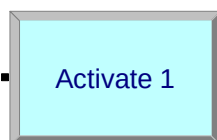
Модуль Request вызывает (запрашивает) транспортер по прибытии в него сущности. Когда сущность достигает модуля Request, она размещается на транспортере, когда он доступен. Сущность остается в модуле Request, пока транспортер не достиг станции. Только тогда сущность перемещается из модуля Request для дальнейшего движения по модели.

Таблица П.37

Параметры модуля Request

Параметры	Описание
Name	Уникальное имя модуля
Transporter Name	Название (имя) транспорта
Velocity	Скорость, с которой транспортер перемещает (единица длины в единицу времени). Единица времени определена в поле Units
Units	Определяет единицы времени для Velocity (т.е. в минуту, в час и т.д.)
Queue Type	Определяет тип очереди при загрузке транспорта
Queue Name	Эта область видима, только если тип очереди – очередь, и это определяет имя символа очереди

Модуль Activate



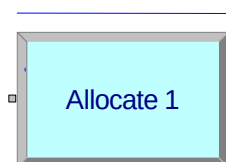
Модуль Activate активирует или увеличивает вместимость предварительно приостановленного транспорта или транспорта, который был первоначально бездействующим (как определено в модуле Transporter).

Таблица П.38

Параметры модуля Activate

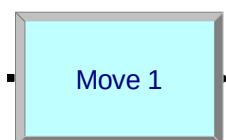
Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Transporter Name	Имя транспорта, с которым работает модуль
Unit Number	Определяет, насколько увеличится вместимость

Модуль Allocate



Модуль Allocate аналогичен модулю Request. Различие только в том, что модуль Allocate не позволяет задавать скорость и единицы измерения скорости транспорта.

Модуль Move



Модуль Move продвигает транспортер от одной станции к другой, которая является пунктом назначения. Контролируемая сущность ожидает в текущем мо-

дуле, пока транспортер прибывает в назначенный пункт. После этого сущность может перемещаться в другой модуль модели.

Время задержки перемещения транспортера из одного пункта (модуля Station) в другой основано на скорости транспортера, которая определяется в модуле Transporter, и расстоянии между пунктами, определенном в модуле Distance.

Сущность не может быть перемещена транспортером, если он не вызван с помощью модулей Request или Allocate. Сущность будет оставаться в модуле Move, пока транспортер не достигнет своего пункта назначения. Если определена скорость движения, это изменение временно и утилизируется только для определенного транспортера, который перемещается.

Таблица П.39

Параметры модуля Move

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Transporter Name	Имя транспортера для перемещения
Unit Number	Определяет транспортер из множества транспортеров
Destination Type	Тип места назначения транспортера
Station Name	Имя места назначения (станции), в которое транспортер переместится
Velocity	Скорость, с которой транспортер переместится в пункт назначения, в единицах времени. Единицы времени определяются в поле Units
Units	Определяет единицы времени (секунды, минуты, часы, дни)

Модуль Transport



Модуль Transport по прибытии в него сущности запускает транспортер и перемещает его от одной станции к другой. Время задержки на перемещение и передачу сущности от одной станции к другой основывается на скорости транспортера и расстоянии между станциями.

Когда сущность входит в модуль Transport, то атрибут станции (Entity.Station) подставляется в станцию назначения, затем сущность передается в станцию назначения. Если станция назначения входит как Sequential, то следующая станция определяется посредством «Запроса сущности» и Jobstep с множеством (специально определенных атрибутов Entity.Sequence and Entity.Jobstep, respectively).

Модуль Transport является эквивалентом модуля Move, с той разницей, что Transport передает сущности дистанционно.

Таблица П.40

Параметры модуля Transport

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Transporter Name	Определяет имя транспорта для передачи
Unit Number	Определяет, какой из транспортеров из множества транспортеров подлежит перемещению
Destination Type	Определяет тип места назначения сущности
Station Name	Определяет имя места назначения (станции), в которое сущность будет перемещаться
Velocity	Скорость, с которой транспортер перемещается к станции назначения
Units	Это поле определяет единицы измерения времени для скорости

Модуль Free



Модуль Free освобождает транспортер для дальнейшего его использования.

Таблица П.41

Параметры модуля Free

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Transporter Name	Имя транспорта, который освободится

Модуль Halt



Модуль Halt изменяет состояние (статус) транспорта на неактивное. Если транспортер занят, в то время как сущность вошла в модуль Halt, то его статус определяется как занят и неактивен до тех пор, пока сущность, которая управляет транспортером, не освободится. Если во время вхождения сущности в модуль Halt транспортер является свободным, то статус транспорта изменяется на неактивный немедленно. Никакая сущность не может получить управление над остановленным транспортером, пока он снова не будет активизирован.

Таблица П.42

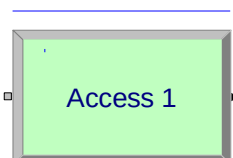
Параметры модуля Halt

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме

Transporter Name	Имя транспортера, который требуется остановить
Unit Number	Определяет, какие из модулей транспортера из набора транспортера следует останавливать

Далее будут подробно рассмотрены модули конвейера.

Модуль Access



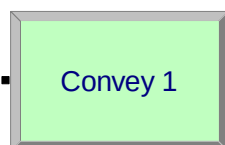
Этот модуль вызывает конвейер, распределяет ячейки конвейера для перемещения сущности от станции к станции. Получив контроль над ячейками конвейера, сущность может переместиться к другой станции конвейера. Этот модуль является эквивалентом модуля Request.

Таблица П.43

Параметры модуля Access

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Cell	Количество ячеек, необходимых для перемещения конвейера
Conveyor Name	Имя конвейера-исполнителя
Queue Name	Имя очереди, в которую поступают сущности конвейера, если конвейер занят

Модуль Convey



Модуль Convey перемещает сущности по конвейеру от одной станции к другой. Время задержки сущности в пути определяется полем Velocity модуля Conveyor и расстоянием между станциями, определенным в модуле Segment. Этот модуль является эквивалентом модуля Transport.

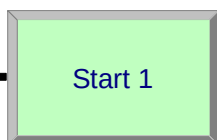
Таблица П.44

Параметры модуля Convey

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Cell	Количество ячеек, необходимых для перемещения конвейера
Conveyor Name	Имя конвейера, который будет использоваться
Destination Type	Определяет метод для определения пункта назначения сущности: <i>Station Name</i> – имя станции; <i>Attribute Name</i> – имя атрибута, который хранит имя станции; <i>Sequential</i> – следующая станция, которая определяется атрибутами сущности

	<i>Entity.Sequence</i> и <i>Entity.JobStep</i> , и <i>Expression</i> – выражение, которое определяет станцию
--	--

Модуль Start



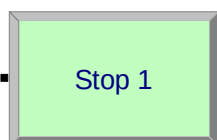
Модуль Start изменяет статус конвейера от бездействующего до активного, т.е. активизирует (вызывает) конвейер. Конвейер может быть остановлен или от модуля Stop, или окончания создания сущности в начале моделирования. Скорость конвейера может изменяться постоянно после начала работы конвейера. Является эквивалентом модуля Move.

Таблица П.45

Параметры модуля Start

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Conveyor Name	Имя конвейера, который требуется активировать
Velocity	Скорость, с которой конвейер переместится в пункт назначения, в единицах времени. Единицы времени определяются в поле Units
Units	Определяет единицы времени (секунды, минуты, часы, дни)

Модуль Stop



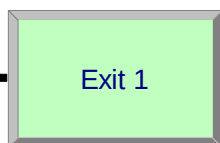
Модуль Stop устанавливает действующий статус конвейера в неактивный. Конвейер может быть активирован для любого модуля Start или по причине активации в начале моделирования. Когда сущность входит в модуль Stop, конвейер мгновенно останавливается, принимая во внимание тип конвейера или номер сущности, вошедшей в конвейер. Является эквивалентом модуля Halt для транспорта.

Таблица П.46

Параметры модуля Stop

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Conveyor Name	Имя конвейера для остановки

Модуль Exit



Модуль Exit выпускает сущности из определенного конвейера и освобождает его для дальнейшей перевозки сущностей. Является эквивалентом модуля Free транспортера.

Таблица П.47

Параметры модуля Exit

Параметры	Описание
Name	Уникальное имя модуля, которое будет отражено в блок-схеме
Conveyor Name	Имя конвейера, который освободится
# of Cells	Число последовательных сущностей для выпуска

4.2. Модули данных

Модуль Transporter

Модуль Transporter предназначен для определения транспортера в модели. Чаще всего модуль связан со схемным модулем Request, который вызывает транспортер, и модулем Move, который передвигает транспортер по схеме.

Таблица П.48

Параметры модуля Transporter

Параметры	Описание
Name	Уникальное имя транспортера
Capacity	Количество транспортеров в наборе
Distance set	Определяет имя дистанции (пути), по которому будет двигаться транспортер
Velocity	Определяет начальную скорость транспортера
Units	Единицы измерения скорости
Initial Position	Определяет начальную станцию, с которой транспортер начнет свое движение

Модуль Distance

Модуль Distance предназначен для определения пути, по которому будет двигаться транспортер.

Таблица П.49

Параметры модуля Distance

Параметры	Описание
Name	Уникальное имя дистанции
Beginning Station	Начальная станция дистанции
Ending Station	Конечная станция дистанции

Distance	Длина дистанции
----------	-----------------

Модуль Sequence

Этот модуль используется для определения последовательности для сущностей в модели при их движении. Последовательность состоит из списка станций, которые сущность должна посетить.

Таблица П.50

Параметры модуля Sequence

Параметры	Описание
Name	Название последовательности
Station Name	Название станции
Step Name	Название станции, которая может быть в последовательности
Next Step	Шаг последовательности

Модуль Conveyor

Модуль Conveyor позволяет перемещать сущности между станциями, является аналогом модуля Transporter.

Таблица 51

Параметры модуля Conveyor

Параметры	Описание
Name	Название конвейера
Segment Name	Имя сегмента, по которому будет двигаться конвейер
Type	Существует 2 типа конвейера: накапливающий и не накапливающий
Velocity	Определяет начальную скорость транспортера
Units	Единицы измерения скорости

Модуль Segment

Модуль Segment определяет путь, по которому будет двигаться конвейер.

Таблица 52

Параметры модуля Segment

Параметры	Описание
Name	Имя сегмента
Beginning Station	Начальная станция
Next Station	Следующая станция в сегменте (может задаваться набором)
Length	Расстояние до предыдущей станции

5. Панель отчетов

С помощью панели отчетов можно просмотреть результаты имитации. На панели отчетов представлены несколько видов отчетов: Отчет «Краткий обзор категорий» и отчеты по четырем категориям, такие как Сущности, Процессы, Очереди и Ресурсы.

1. *Отчет Category Overview категорий (Краткий обзор категорий).* Отчет Category Overview отражает итоговую информацию о сущностях, процессах, очередях и ресурсах. Также показывает информацию о заданных пользователем переменных и информацию, собранную модулем Record.

2. *Отчет о сущностях.* Отчет о сущностях разделен на несколько частей.

2.1. Cycle Time: в этой части отчета показано среднее, максимальное и минимальное время существования сущности. Время существования сущности считается с момента её прибытия в систему и до того момента, когда сущность попадает в модуль Dispose. Ниже представляется гистограмма среднего времени цикла для каждого типа сущности.

2.2. NVA Cost: в этой части показано среднее, максимальное и минимальное значение недобавочной стоимости сущностей по каждому типу. Недобавочная стоимость рассчитывается на основании значения NVA Time.

2.3. Total Cost: в этой части показано среднее, максимальное и минимальное значение общей стоимости сущностей по каждому типу. Общая стоимость вычисляется путем сложения стоимости ожидания, добавочной стоимости и недобавочной стоимости для каждой сущности.

2.4. VA Cost: в этой части показано среднее, максимальное и минимальное значение добавочной стоимости сущностей по каждому типу. Добавочная стоимость рассчитывается на основании VA Time.

2.5. Wait Cost: в этой части показано среднее, максимальное и минимальное значение стоимости ожидания сущностей по каждому типу. Стоимость ожидания подсчитывается, исходя из времени ожидания, стоимости ресурса и стоимости нахождения сущности в системе.

2.6. Wait Time: в этой части показано среднее, максимальное и минимальное значение времени ожидания сущностей по каждому типу. Время ожидания – это период времени с момента поступления сущности в очередь (либо в модуле Process ожидает ресурс, либо в модуле Batch ожидает группировки) и до момента выхода из нее (начнет обрабатываться либо будет сгруппирована).

2.7. WIP (Work In Process): в этой части показано среднее, максимальное и минимальное значение времени ожидания сущностей в процессах.

3. *Отчет о процессах*. Отчет о процессах разделен на такие же части, как и отчет по сущностям, только с уклоном на процессы.

4. *Отчет о ресурсах* содержит информацию о загрузенности и простое ресурсов.

5. *Отчет по очередям* содержит информацию о среднем, минимальном и максимальном времени нахождения сущности в очереди и максимальных, средних и минимальных очередях.

6. Панель навигации

С помощью панели навигации можно быстро передвигаться по различным уровням модели, быстро менять виды. Можно задать быстрые клавиши для изменения вида. Виды подмоделей создаются автоматически, но также возможно добавить новые виды с помощью команды Add View. Можно передвигаться не только по различным уровням модели, но также быстро получать нужный масштаб какой-либо части модели.

7. Построитель выражений

ПП Arena позволяет строить сложные выражения. Это достигается с помощью Expression Builder. Построитель выражений имеет внешний вид, показанный на рис. П.2.

Построитель выражений имеет 3 секции:

1. Окно типов выражений. Рассмотрим более подробно *окно типов выражений*, которое состоит из четырех разделов:

1.1. Random Distributions (Вероятностные распределения). В ПП Arena 9.0 заложены 13 типов стандартных распределений:

normal (нормальное): *Mean, StdDev*;

exponential (экспоненциальное): *Mean*;

uniform (равномерное): *Min, Max*;

poisson (пуассоновское): *Mean*;

gamma (гамма): *Beta, Alpha*;

beta (бета): *Beta, Alpha*;

triangular (треугольное): *Min, Mode, Max*;

continious (непрерывное): *CumP₁, Val₁, CumP_n, Val_n*;

discrete (дискретное): *CumP₁, Val₁, CumP_n, Val_n*

erlang (распределение Эрланга): *ExpoMean, k*;

johnson (распределение Джонсона): *Gamma, Delta, Lambda, Xi*;

lognormal (логнормальное): *LogMean, LogStd*;

weibull (распределение Вейбулла): *Beta, Alpha*.

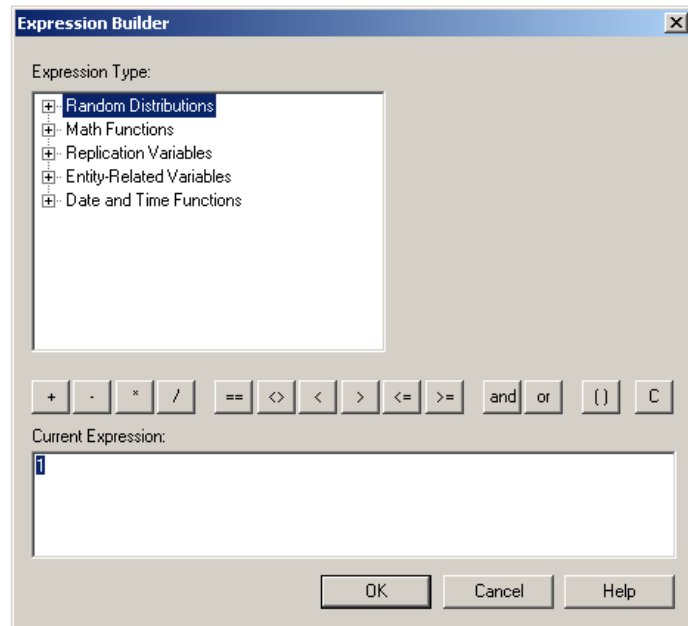


Рис. П.2. Внешний вид построителя выражений

Остановимся более подробно на двух видах распределений, которые наиболее часто используются при моделировании сложных систем. Это равномерное (UNIF или Uniform) и треугольное распределения, приведенные на рис. П.3, *а* и *б* соответственно. Равномерное распределение показывает, что вероятность возникновения события P_1 одинакова (равновероятна) на интервале от Min до Max, например клиенты приходят раз в 5–9 минут. Треугольное распределение показывает, что наиболее вероятно (Most Likely) появление события в какое-то определенное время, например клиенты приходят раз в 5–9 минут, но чаще всего раз в 7 минут.

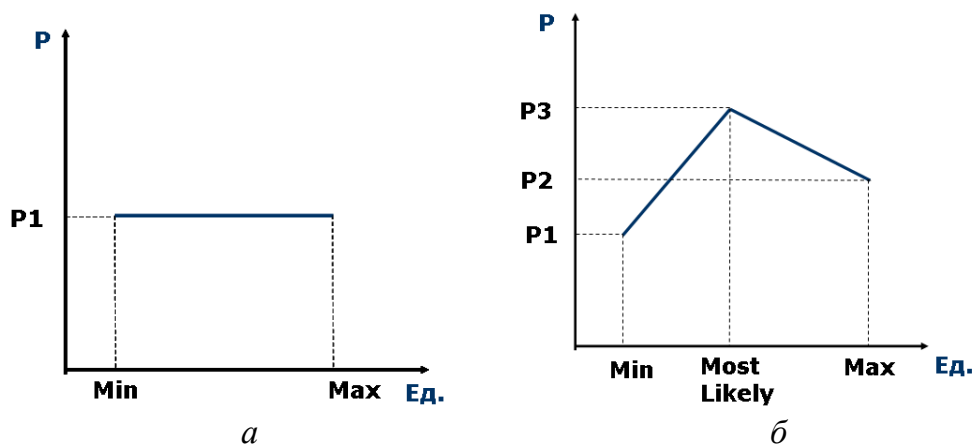


Рис. 3. Графики зависимостей распределений:
а – равномерное распределение; *б* – треугольное распределение

1.2. Math Functions (Математические функции), к которым относятся 11 алгебраических операторов:

- абсолютное значение;
- округление до ближайшего целого;
- целая часть от нецелочисленного значения;
- минимальное значение;
- максимальное значение;
- натуральный логарифм;
- корень квадратный и т.д.

и 9 геометрических функций:

- синус;
- косинус;
- тангенс;
- арксинус и т.д.

• Replication Variables (переменные, связанные с репликациями модели);

- Maximum Replications (максимальное количество повторений);
- Current Replication Number (текущее количество повторений).

1.3. Entity-Related Variables (переменные, связанные с сущностью):

• Attributes (Атрибуты). К наиболее интересным атрибутам следует отнести: Entity.Type (тип сущности), Entity.SerialNumber (серийный номер сущности), Entity.Picture (анимационная картинка сущности), Entity.CreateTime (Время создания сущности), User-Defined Attribute Value (атрибуты, заданные пользователем);

- Group Member Variables (Групповые переменные).

1.4. Date and Time Functions (временные функции). Наиболее интересный и часто используемый оператор из этой группы – это TNOW (Current Simulation Time – текущее время моделирования).

2. Панель операторов, используемых в выражениях (сложение, вычитание, и т.д.; элементы сравнения, логические операторы и т.д.).

3. Окно записи выражения.

СПИСОК ЛИТЕРАТУРЫ

1. Бусленко Н.П. Моделирование сложных систем. – М.: Наука, 1978.
2. Шеннон Р.Ю. Имитационное моделирование систем – искусство и наука: пер. с англ. – М.: Мир, 1978.
3. Бешенков С.А. Моделирование и формализация: методическое пособие. – М.: Лаборатория базовых знаний, 2002.
4. Максимей И.В. Имитационное моделирование на ЭВМ. – М.: Радио и связь, 1988.
5. Клейнен Дж. Статистические методы в имитационном моделировании. – М.: Статистика, 1978.
6. Лоу А.М., Кельтон В.Д. Имитационное моделирование. Классика CS. – 3-е изд. – СПб.: Питер; Киев: Издательская группа BNV, 2004.
7. Веб-сайт зам. зав. кафедры «Телекоммуникационные сети и системы» МФТИ Семенова Ю.А. [Электронный ресурс]. – Режим доступа: <http://book.iter.ru/>
8. Советов Б.Я. Моделирование систем: учеб. для вузов. – 3-е изд., перераб. и доп. – М.: Высш. шк., 2001.
9. Докукин В.П. Основы математического моделирования: конспект лекций; Санкт-Петербургский ГГИ. – М.: Дело, 2000.
10. Максименков, А.В., Селезнев М.Л. Основы проектирования информационно-вычислительных систем и сетей ЭВМ. – М.: Радио и связь, 1991.
11. Карпов Ю. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. - СПб.: БХВ — Петербург, 2005. - 400 с.
12. Локальные вычислительные сети. Кн. 3. Организация функционирования, эффективность, оптимизация: справочник / под ред. С.В. Назарова. – М.: Финансы и статистика, 1995.
13. Минько А.А. Статистический анализ в Microsoft Office Excel. Профессиональная работа [Электронный ресурс]. – Режим доступа: <http://www.dialektika.com/books/5-8459-0692-X.html>
14. Питерсон Дж. Теория сетей Петри и моделирование систем: пер. с англ. – М.: Мир, 1984.
15. Малышкин В.Э. Основы параллельных вычислений. Часть 1: учеб. пособие. – Новосибирск: Изд-во НГТУ, 1998.
16. Максимей И.В. Имитационное моделирование на ЭВМ. – М.: Радио и связь, 1988.
17. Суворов И.Ф. Краткий курс высшей математики. – М.: Высш. шк., 1961.

18. Корн Г., Корн Т. Справочник по математике (для научных работников и инженеров). – М.: Наука, 1974.
19. Хемди А. Таха Введение в исследование операций = Operations Research: An Introduction. – 7-е изд. – М.: Вильямс, 2007.
20. Abstraction Software LLC. Makers of Prophecy – a low cost, Windows based network and workflow simulation and modeling system [Электронный ресурс]. – Режим доступа: <http://www.abstraction.com/prophecy.htm>
21. Network Modeling and Network Simulation – Opnet. R&D process for analyzing and designing communication networks, devices, protocols, and applications [Электронный ресурс]. – Режим доступа: http://www.opnet.com/solutions/network_rd/modeler.html
22. Arena Simulation Software by Rockwell Automation [Электронный ресурс]. – Режим доступа: <http://www.arenasimulation.com/>
23. Prophecy Software [Электронный ресурс]. – Режим доступа: <http://www.abstraction.com/prophecy.htm>
24. Modeler web page. [Электронный ресурс]. – Режим доступа: http://www.opnet.com/solutions/network_rd/modeler.html#
25. comnetIII web page [Электронный ресурс]. – Режим доступа: <http://feamane.org/comms/testtools/comnet3/comnet3.html>
26. Официальный сайт NetMaker ХА фирмы Make Systems [Электронный ресурс]. – Режим доступа: www.makesystems.com
27. Официальный сайт SES/Strategizer фирмы Scientific and Engineering Software [Электронный ресурс]. – Режим доступа: www.ses.com
28. Высокопроизводительные вычисления // Официальный сайт компании Trinity [Электронный ресурс]. – Режим доступа: <http://www.trinitygroup.ru/> (дата обращения: 09.06.2010).
29. Giambene G. Queuing theory and telecommunications: networks and applications. – Springer Sciense+Media Inc., 2005.

Учебное издание

ЗАМЯТИНА Оксана Михайловна

МОДЕЛИРОВАНИЕ СЕТЕЙ

Учебное пособие

Научный редактор
доктор технических наук, профессор *В.А. Силич*

Редактор *Л.А. Егорова*

Компьютерная верстка *О.М. Замятина*

Дизайн обложки *Ж.А. Бухарина*

Подписано к печати Формат 60×84/16.

Бумага «Снегурочка». Печать ХероХ.

Усл. печ. л. 9,77. Уч.-изд. л. 8,84.

Заказ . Тираж экз.



Национальный исследовательский
Томский политехнический университет
Система менеджмента качества



Издательства Томского политехнического университета сертифицирована
NATIONAL QUALITY ASSURANCE по стандарту BS EN ISO 9001:2008

ИЗДАТЕЛЬСТВО  **ТПУ.** 634050, г. Томск, пр. Ленина, 30.
Тел./факс: 8(3822)56-35-35, www.tpu.ru