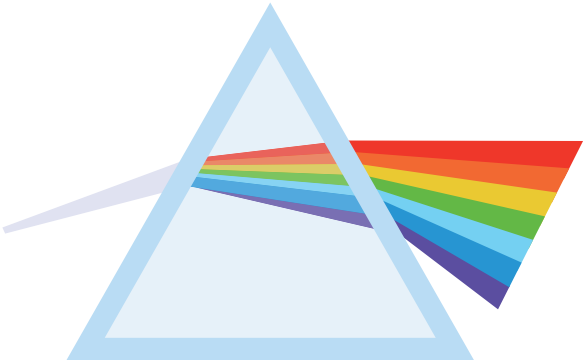
СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
SIBERIAN FEDERAL UNIVERSITY

М. С. Эльберг, Н. С. Цыганков

# ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

**Учебное пособие**



ИНСТИТУТ ИНЖЕНЕРНОЙ ФИЗИКИ
И РАДИОЭЛЕКТРОНИКИ

Министерство образования и науки Российской Федерации
Сибирский федеральный университет

М. С. Эльберг, Н. С. Цыганков

# ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

Учебное пособие

Красноярск
СФУ
2017

УДК 004.94(07)
ББК 22.181я73
Э530

Р е ц е н з е н т ы:

*А. Н. Втюрин*, доктор физико-математических наук, заместитель директора Института физики им. Л. В. Киренского СО РАН;

*В. В. Салмин*, доктор физико-математических наук, заведующий кафедрой МиБФ КрасГМУ

**Эльберг, М. С.**

Э530 Имитационное моделирование : учеб. пособие / М. С. Эльберг, Н. С. Цыганков. – Красноярск : Сиб. федер. ун-т, 2017. – 128 с.

ISBN 978-5-7638-3648-6

Рассмотрены базовые понятия, виды и инструментальные средства моделирования, этапы разработки компьютерных моделей сложных систем. Основное внимание уделено многоподходному инструменту моделирования объектов и процессов реального мира AnyLogic 7.3.6 и технологиям построения имитационных моделей в среде AnyLogic.

Предназначено для бакалавров, обучающихся по направлению подготовки 27.03.05 «Инноватика», и может быть рекомендовано для бакалавров направления 11.03.01 «Радиотехника», магистров направлений 27.04.05 «Инноватика» и 11.04.01 «Радиотехника», а также аспирантов, научных работников и инженеров, специализирующихся в области математического моделирования сложных систем.

**Электронный вариант издания см.:**
**http://catalog.sfu-kras.ru**

**УДК 004.94(07)**
**ББК 22.181я73**

*Выражаем благодарность авторам материалов, опубликованных на сайте Anylogic.ru, которые использовались в качестве примеров в пособии*

ISBN 978-5-7638-3648-6

© Сибирский федеральный университет, 2017

# ВВЕДЕНИЕ

Благодаря интенсивному развитию информатики и компьютерных технологий стало намного проще решать сложные задачи, требующие больших временных и финансовых затрат. Упростить их решение возможно с использованием моделирования.

Одним из наиболее распространенных и удобных способов моделирования сложных систем является имитационное компьютерное моделирование объектов и процессов реального мира.

Невозможно сразу моделировать какой-либо процесс, для этого необходимо специальное обучение способам, приемам и технологиям компьютерного имитационного моделирования (ОПК-3).

Специалист, приступая к решению задачи, должен знать основы динамических процессов, подходы и методы решения сложных процессов и систем, в том числе аналитических и имитационных, а также знать конкретные информационные системы моделирования и используемые в них языки программирования. Среди множества сред аналитического моделирования основными являются Maple, MathCAD, MATLAB + Simulink и др. (ОПК-3).

При обучении моделированию сложных систем могут быть использованы различные среды и методологии разработки аналитических и имитационных моделей сложных систем: MvStudium, MATLAB, Arena, GPSS, Extend, iThinkAnalyst, ProcessModel и др. (ОПК-3). Особое место среди сред разработки компьютерных моделей сложных систем принадлежит многоподходной среде моделирования имитационных моделей – AnyLogic. Разные средства спецификации и анализа результатов, имеющиеся в AnyLogic, позволяют строить модели (динамические, дискретно-событийные, агентные), имитирующие практически любой реальный процесс, а также конструировать и многие другие модели, выполнять анализ моделей на компьютере без проведения реальных экспериментов и самостоятельных сложных вычислений (ПК-8, ПК-10). Но для возможности оперировать этой программной средой и получать при моделировании верные результаты пользователь AnyLogic должен овладеть технологией работы в среде, понять ее функциональ-

ные особенности, в этих целях мы представляем учебное пособие по разработке компьютерных моделей сложных систем в среде AnyLogic [2].

Моделирование – метод решения задач, при использовании которого исследуемая система заменяется более простым объектом, описывающим реальную систему и называемым моделью.

Моделирование применяется в случаях, когда проведение экспериментов над реальной системой невозможно или нецелесообразно, например из-за высокой стоимости или длительности проведения эксперимента в реальном масштабе времени.

Руководствуясь жизненным опытом и научными знаниями, человек строит модели – от бумажных корабликов до картины мира. Чем они богаче и чем точнее мы можем ими оперировать, тем развитей наше сознание, наша «самая важная модель» соответствует реальности и находит способы ее изменения [1].

Моделирование – самое эффективное средство поддержки принятия решений, а по словам Ричарда Докинза – «один из самых интересных способов предсказывать будущее» [4].

Теоретические предпосылки этого утверждения формировались на протяжении веков. В основу математического моделирования легли математический анализ, теория вероятностей, численные методы, теория подобия. В XX в. появилась база практического приложения моделей: математическое программирование; теория массового обслуживания; теория алгоритмов; теория систем; кибернетика (ПК-8).

Другая, «фактологическая», основа моделирования – стремительно растущий потенциал знаний фундаментальных и прикладных наук.

В сочетании с современным технологическим прорывом эти основы создают необычайные возможности построения моделей, ограниченные лишь смелостью исследователя. Перечислим только злободневные глобальные темы, которые проходят непрерывную проверку моделированием: экономика, политика, экология.

Моделирование уверенно помогает понять, как устроен мир. Можно надеяться, что с его помощью мы когда-нибудь узнаем, как работает и наша «самая важная модель» [1].

Различают физическое и математическое моделирование. Примером физической модели является уменьшенная копия самолета, продуваемая в потоке воздуха. При использовании математического моделирования поведение системы описывается с помощью формул. Особым видом математических моделей являются имитационные модели (ПК-8).

Имитационная модель – это компьютерная программа, которая описывает структуру и воспроизводит поведение реальной системы во времени. Имитационная модель позволяет получать подробную статистику о различных аспектах функционирования системы в зависимости от входных данных (ОПК-3).

Имитационное моделирование – разработка компьютерных моделей и постановка экспериментов на них. Целью моделирования в конечном счете является принятие обоснованных, целесообразных управленческих решений; подготовка студентов к решению задач, связанных с процессами анализа, прогнозирования, моделирования в рамках профессионально ориентированных информационных систем сферы инноватики.

Задачи, решаемые в учебном пособии «Имитационное моделирование»:

- сформировать целостное представление о системе экономико-математических моделей и месте имитационных моделей, а также изучить процессы массового обслуживания;
- научить выполнять имитацию инновационного объекта в трех измерениях: материальном, денежном и информационном;
- произвести экономическое прогнозирование и предвидение развития экономических процессов;
- сформировать у студентов навыки, необходимые для выработки управленческих решений.

Компьютерное моделирование становится сегодня обязательным этапом в принятии ответственных решений во всех областях деятельности человека в связи с усложнением систем, в которых человек должен действовать и которыми он должен управлять. Знание принципов и возможностей имитационного моделирования, умение создавать и применять модели являются необходимыми требованиями к инженеру, менеджеру, бизнес-аналитику [4].

# Глава 1

# МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

## 1.1. Моделирование как научный метод

*Моделирование* является одним из способов решения практических задач. Зачастую решение проблемы нельзя найти путем проведения натурных экспериментов: строить новые объекты, разрушать или вносить изменения в уже имеющуюся инфраструктуру может быть слишком дорого, опасно или просто невозможно. В таких случаях целесообразно построить *модель* реальной системы, т. е. описать ее на языке моделирования. Данный процесс подразумевает переход на определенный уровень абстракции, опуская несущественные детали, с учетом только того, что считаем важным. Система в реальном мире всегда сложнее своей модели (рис. 1.1) [6].

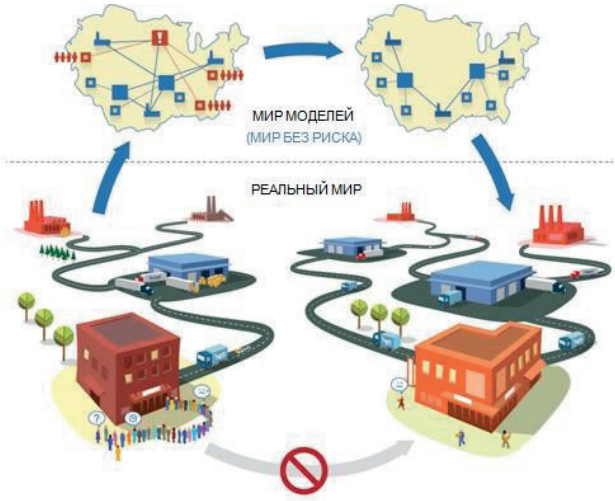



Рис. 1.1. Моделирование реальных систем

Все этапы разработки модели – проекция реального мира в мир моделей, выбор уровня абстракции и выбор языка моделирования менее стандартизированы, чем процесс использования моделей для решения задач. Моделирование до сих пор больше искусство, чем наука.

После создания модели – а иногда и в процессе разработки – мы начинаем исследовать структуру и понимать поведение системы, проверять, как она ведет себя при определенных условиях, сравнивать различные сценарии и оптимизировать ее. Когда оптимальное решение будет найдено, мы сможем применить его в реальном мире.

В сущности, моделирование является поиском решения задачи в защищенном от риска мире моделей, в котором мы можем ошибаться, отменять операции, возвращаться в прошлое и начинать все сначала [6].

Моделирование дает предположительную информацию о неком фрагменте реальности. После определенных проверок она может оказаться истинной или ложной и потребовать построения новых моделей [1].

В науке, наряду с наблюдением, измерением, экспериментом и сравнением, эта процедура выступает как один из общенаучных методов. Однако моделирование можно рассматривать как особый интегрирующий метод. Его эффективность и универсализм возрастают по мере развития информационных технологий. В силу разных причин объект может быть недоступен (слишком мал или велик, далеко расположен, дорог, прекратил существование, например в результате аварии). Исключительная польза моделирования заключается в том, что можно экспериментировать не с самой системой, а с ее аналогом – моделью.

Моделирование – процесс отражения свойств одного объекта (оригинала) в другом объекте (модели). Это могут быть объекты «как есть» в целом и (или) их отдельные сущности – процессы и явления. Явления – например, поведение животного, состояния погоды – рассматриваются как сложные процессы.

В основу моделирования заложена процедура формализации – перевод свойств объекта на язык понятий предметной области, алгоритмов и математики.

*Подобие модели объекту.* Объект и модель находятся в отношении сходства, т. е. модель по каким-то признакам должна быть подобна изучаемому объекту. Это явление называют ***изоморфизмом*** (от греч. *isos* – равный и *morphe* – форма). Различают три вида подобия.

*Первый вид подобия* – подобное масштабирование. Примеры такого подобия: модели автомобилей, самолетов, кораблей, сооружений и т. д.

*Второй вид подобия* – косвенное подобие (*математическая аналогия*). Удачный математический аналог из других областей знаний может сильно упростить построение модели и ее анализ. Так, очень многие физические процессы могут быть описаны уравнениями, общий вид которых $q = -\Theta \operatorname{grad} x$ (рис. 1.2).

**Перенос массы вещества – закон Фика:**

$$j = -D \cdot dC/dx,$$

где $D$ – коэффициент диффузии; $C$ – концентрация; $x$ – текущая координата.

**Перенос тепла – уравнение Фурье:**

$$q = -\lambda \cdot dT/dx,$$

где $q$ – тепловой поток; $T$ – температура; $\lambda$ – коэффициент теплопроводности.

**Перенос электричества – закон Ома:**

$$i = -c \cdot du/dx,$$

где $i$ – сила тока; $c$ – характерная проводимость; $u$ – потенциал.

Рис. 1.2. Тройная аналогия процессов переноса

Аналогичны законы Кулона и всемирного тяготения. Примером также может служить подобие электрических и механических явлений:

- колебание физического маятника:

$$T = 2\pi \sqrt{l/g};$$

- пружинного маятника:

$$T = 2\pi \sqrt{m/k};$$

- колебательного контура:

$$T = 2\pi \sqrt{L/C}.$$

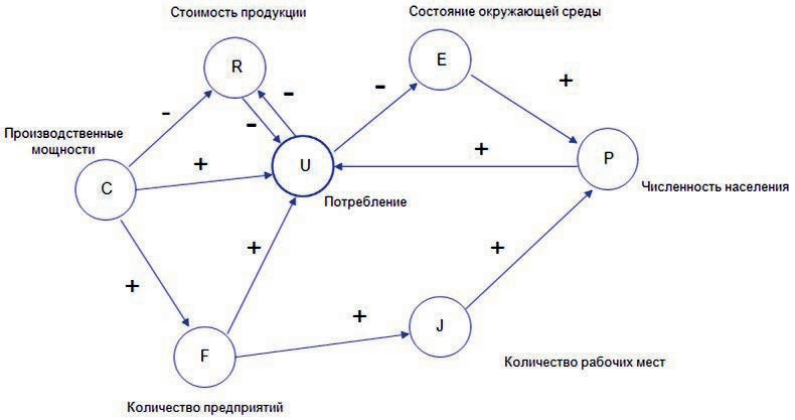



Рис. 1.3. Когнитивная модель потребления промышленной продукции (энергии, металлов и т. п.): «+» – положительные связи (влияния); «–» – отрицательные связи (влияния)

*Третий вид подобия* – условное подобие, или подобие по соглашению. Примерами являются когнитивные модели (рис. 1.3), географические карты, масштабированные чертежи сооружений, зданий, структурные схемы (модели системного анализа). При этом внешне сходство объекта и модели может не соблюдаться.

Таким образом, объект моделирования и модель могут быть любой природы – материальными или абстрактными. Например, макет самолета – это материальная модель. Схема производства – абстрактная модель. Уравнения физики – это описание абстракций разных явлений материального мира. Модели могут быть и абстракциями других моделей. Наследование (создание одних классов на базе других) в объектно-ориентированном программировании – наиболее характерный пример таких построений [1].

*Адекватность моделей.* Вид и свойства будущей модели определяются целями исследователя, использующего этот инструмент. В модели отражаются свойства объекта, соответствующие этим целям, которые определяют и уровни детализации модели. В первую очередь должны быть определены *существенные* свойства оригинала, характеризующие его как некую проблему, которую нужно решить с помощью модели [1].

При этом стоит помнить, что знать *все* свойства предмета вашего исследования нельзя. К тому же не будем забывать, метод – это инструмент, а универсальных инструментов не бывает. Означает ли это, что моделирование – ненадежный помощник? Нет. Во-первых, существует принцип множественности моделей. В соответствии с ним можно, а иногда – необходимо построить несколько моделей, позволяющих рассмотреть объект как проблему с различных позиций. К соответствующим решениям (моделям) можно идти, используя разные подходы. Например, создание модели поведения человека будет зависеть от разработки разных целей [1]:

1) добиться антропоморфной кинематики компьютерной модели тела человека;

2) получить модель характерных психических реакций человека;

3) смоделировать реакции различных социальных групп людей.

Во-вторых, существуют специальные процедуры проверки того, является ли модель точным представлением реальной системы, т. е. *адекватна* ли модель системе.

При верификации, т. е. проверке достоверности модели, определяется, правильно ли концептуальная модель (модельные допущения) преобразована в компьютерную программу [1].

Валидация – это процесс, позволяющий установить, является ли модель точным представлением системы для конкретных целей исследования. Определяющим моментом в этих процедурах является положение: «модель и ее результаты достоверны, если руководители проекта признают их правильными» [16]. В итоге, если модель «адекватна», ее можно использовать для принятия решений относительно системы, которую она представляет, как если бы они принимались на основании экспериментов с реальной системой.

В-третьих, итоговый результат (т. е. «хорошая» или «плохая» модель получится) зависит от личности разработчика. Моделирование как метод научного познания предполагает творческий подход к объекту и целям исследования.

В этом виде научного производства не обойтись без развитого воображения, умения анализировать и делать обобщения. Хорошие модели – это «мини-теории», и их создание требует нестандартного мышления [1].

## 1.2. Исходные понятия и определения

Теория основ математического и компьютерного моделирования предполагает содержательное и формальное определение категорий, дефиниций и понятий с целью построения математических моделей сложных систем [2].

Основными методологическими категориями теоретических основ моделирования являются понятия «*объект*», «*класс*», «*отношение* (*связь*)», «*система*», «*элемент*», «*структура*».

Определение понятия «*объект*» имеет различное толкование в зависимости от области рассмотрения. Если мы изучаем область имитационного моделирования, то в стратегии объектно-ориентированного подхода объект является первым важным понятием. Объект – это некоторая сущность в виртуальном пространстве, обладающая определенным состоянием и поведением, имеющая заданные значения свойств (атрибутов) и операций над ними.

Следующим важным понятием объектно-ориентированного подхода является «*класс*». Родственные по определенным характеристикам, поведению объекты объединяются в классы. В зависимости от характеристик одни и те же объекты могут быть в различных классах.

В одном из разделов современной математики «теории категорий» объект используется как термин для обозначения элементов произвольной категории, играющих роль множеств, групп, топологических пространств и т. п. Здесь также вводится понятие класса объектов и проводится изучение свойств отношений между математическими объектами, не зависящих от внутренней структуры объектов.

Понятие «*отношение*» определяет взаимное положение объектов, связи между объектами в виде иерархических, ассоциативных, алгоритмических, табличных и других структур.

Понятие «*система*» является основополагающим в теории математического моделирования. Существует несколько десятков различных определений понятия «система», используемых в зависимости от контекста, области знаний и целей исследования. Изучением систем занимаются такие научные дисциплины, как *системология*, *кибернетика*, *системный анализ*, *теория систем*, *системная динамика* и др. [2].

***Система*** – это 1) целое, созданное из частей и элементов целенаправленной деятельности и обладающее новыми свойствами, отсутствующими у элементов и частей, его образующих; 2) объективная часть мироздания, включающая схожие и совместимые элементы, образующие особое целое, которое взаимодействует с внешней средой; 3) объективное единство закономерно связанных друг с другом предметов, явлений, сведений, а также знаний о природе, обществе и т. п. Допустимы и многие другие определения. Общим в них является то, что система есть некоторое правильное сочетание наиболее важных, существенных свойств изучаемого объекта. Каждый объект, чтобы его можно было считать системой, должен обладать четырьмя основными свойствами или признаками (целостностью и делимостью, наличием устойчивых связей, организацией и эмерджентностью).

***Элемент*** – это простейшая неделимая часть системы, а ее свойства определяются конкретной задачей. Элемент всегда связан с самой системой. Элемент сложной системы может быть, в свою очередь, сложной системой в другой задаче.

*Подсистема* – компонент системы – объединение элементов, но по масштабу меньше, чем система в целом.

Система может включать большой перечень элементов, и ее целесообразно разделить на ряд подсистем [2].

Признаками системы являются множество составляющих ее элементов, единство главной цели для всех элементов, наличие связей между ними, целостность и единство элементов, наличие структуры и иерархичности, относительная самостоятельность и наличие управления этими элементами. Термин «организация» в одном из своих лексических значений означает также систему, но не любую, а в определенной мере упорядоченную, организованную.

Понятие «подсистема» выработано для анализа сложноорганизованных, саморазвивающихся систем, когда между элементами и системой имеются «промежуточные» комплексы, более сложные, чем элементы, но менее сложные, чем сама система. Они объединяют в себе разные части (элементы) системы, в своей совокупности способные к выполнению единой (частной) программы системы. Будучи элементом системы, подсистема, в свою очередь, оказывается системой по отношению к элементам, ее составляющим. Аналогич-

но обстоит дело с отношениями между понятиями «система» и «элемент»: они переходят друг в друга. Иначе говоря, система и элемент относительны. С этой точки зрения вся материя представляется как бесконечная система систем. «Системами» могут быть системы отношений, детерминаций и т. п. [2].

Наряду с представлением об элементах в понятие о любой системе входит и представление о ее структуре.

***Структура*** – это совокупность устойчивых отношений и связей между элементами. Сюда включается общая организация элементов, их пространственное расположение, связи между этапами развития и т. п.

По своей значимости для системы связи элементов (даже устойчивые) неодинаковы: одни малосущественны, другие существенны, закономерны. Структура прежде всего – это закономерные связи элементов. Среди закономерных наиболее значимы интегрирующие связи (или интегрирующие структуры). Они обусловливают интегрированность сторон объекта. В системе производственных отношений, например, имеются связи трех родов: относящиеся к формам собственности, к обмену деятельностью и к распределению. Все они существенны и закономерны. Но интегрирующую роль в этих отношениях играют отношения собственности (иначе – формы собственности). Интегрирующая структура является ведущей основой системы [2].

Существует ряд подходов к выделению систем по сложности и масштабу. Например, для систем управления удобно пользоваться классификацией по числу (количеству) элементов:

- малые ($10–10^3$ элементов);
- сложные ($10^4–10^7$ элементов);
- ультрасложные ($10^8–10^{30}$ элементов);
- суперсистемы ($10^{30}–10^{200}$ элементов).

Большая система – это всегда совокупность материальных и энергетических ресурсов, средств получения, передачи и обработки информации, людей, которые принимают решение на разных уровнях иерархии. В настоящее время для понятий «сложная система» и «большая система» используют такие определения:

- *сложная система* – упорядоченное множество структурно взаимосвязанных и функционально взаимодействующих *разнотип-*

*ных систем*, которые объединены структурно в целостный объект функционально разнородными взаимосвязями для достижения заданных целей в определенных условиях;

• *большая система* объединяет *разнотипные сложные системы.*

Тогда определение системы можно записать так: «система – это упорядоченное множество структурно взаимосвязанных и функционально взаимодействующих однотипных элементов любой природы, объединенных в целостный объект, состав и границы которого определяются целями системного исследования».

Характерные особенности больших систем:

• значительное количество элементов;

• взаимосвязь и взаимодействие между элементами;

• иерархичность структуры управления;

• наличие человека в контуре управления и необходимость принятия решений в условиях неопределенности.

Описание динамики системы или ее поведения составляет основу любой имитационной модели. В качестве исходных данных для решения этой задачи используются результаты, полученные на этапе разработки концептуальной модели системы. К ним относятся:

• определение принадлежности моделируемой системы одному из известных классов;

• описание рабочей нагрузки системы;

• выбор уровня детализации представления системы в модели и ее декомпозиция.

Все последующие действия исследователя по созданию модели могут быть отнесены к этапу ее формализации, который в общем случае предполагает:

• выбор метода отображения динамики системы (на основе событий, процессов или транзактов);

• формальное (математическое) описание случайных факторов, подлежащих учету в модели;

• выбор механизма изменения и масштаба модельного времени.

Рассмотрим устоявшиеся понятия в имитационном моделировании: «*процесс*», «*работа*», «*событие*», «*транзакт*».

***Работа*** (*активность*) − это единичное действие системы по обработке (преобразованию) входных данных. В зависимости от приро-

ды моделируемой системы под входными данными могут пониматься информационные данные или какие-либо материальные ресурсы.

Под ***процессом*** понимают логически связанный набор работ. Некоторые процессы могут рассматриваться как работы в процессе более высокого уровня. Любой процесс характеризуется совокупностью статических и динамических характеристик.

К статическим характеристикам относятся:

- длительность;
- результат;
- потребляемые ресурсы;
- условия запуска (активизации);
- условия остановки (прерывания).

Статические характеристики процесса не изменяются в ходе его реализации, однако при необходимости любая из них может быть представлена в модели как случайная величина, распределенная по заданному закону.

Динамической характеристикой процесса является его состояние (активен или находится в состоянии ожидания).

Моделирование в терминах процессов проводится в тех случаях, если система оценивается по каким-либо временным показателям либо с точки зрения потребляемых ресурсов.

Например, при оценке производительности вычислительной сети обработка заданий может быть представлена в модели как совокупность соответствующих процессов, использующих ресурсы сети (оперативную память, пространство на жестких дисках, процессорное время, принтеры и т. д.).

Если модель строится с целью изучения причинно-следственных связей, присущих системе, динамику системы целесообразно описывать в терминах событий.

***Событие*** представляет собой мгновенное изменение некоторого элемента системы или состояния системы в целом. Событие характеризуется:

- условиями (или законом) возникновения;
- типом, который определяет порядок обработки (дисциплину обслуживания) данного события;
- нулевой длительностью.

События подразделяют на две категории:

• события следования, которые управляют инициализацией процессов (или отдельных работ внутри процесса);

• события изменения состояний (элементов системы или системы в целом).

Механизм событий используется в качестве основы построения моделей, предназначенных для исследования причинно-следственных связей в системах при отсутствии временных ограничений. К таким задачам можно отнести, например, некоторые задачи по оценке надежности.

Еще один способ имитационного моделирования систем основан на использовании понятия транзакта, или сущности.

***Транзакт, или сущность,*** – это некоторое сообщение (заявка на обслуживание), которое поступает извне на вход системы и подлежит обработке.

В некоторых случаях, например при моделировании автоматизированных систем управления, удобно проследить функционирование системы относительно алгоритма обработки транзакта (сущности). В рамках одной имитационной модели могут рассматриваться транзакты (сущности) нескольких типов. Каждый транзакт (сущность) характеризуется соответствующим алгоритмом обработки и необходимыми для его реализации ресурсами системы. Прохождение транзакта (сущности) по системе можно в некоторых случаях рассматривать как последовательную активизацию процессов, реализующих его обработку («обслуживание заявки»).

Чтобы построить качественную компьютерную модель сложной системы необходимо уметь:

• определенным способом представить в модели динамику (движение) системы. Это может быть описано посредством событий, работ, процессов, транзактов;

• определить способ изменения модельного времени. Здесь выделяют моделирование с постоянным шагом и моделирование по особым состояниям.

В большинстве случаев конечной целью моделирования является оптимизация каких-либо параметров системы.

Виды имитационного эксперимента:

• исследование относительного влияния различных факторов на значения выходных характеристик системы;

• нахождение аналитической зависимости между интересующими исследователя выходными характеристиками и факторами;

• отыскание оптимальных значений параметров системы (так называемый «экстремальный эксперимент»);

• сравнение альтернатив для принятия решений;

• оптимизация системы для оценки и выработки оптимальной стратегии;

• анализ ситуаций и обучение в различных отраслях через виртуальные имитационные модели игр;

• визуализация и анимация деятельности разрабатываемого объекта.

Вид эксперимента влияет не только на выбор схемы его формализации, но также на построение плана эксперимента и выбор метода обработки его результатов.

С точки зрения организации взаимодействия исследователя с моделью (по способу взаимодействия с пользователем), в ходе эксперимента имитационные модели делятся на автоматические и диалоговые.

*Автоматическими* называются имитационные модели, взаимодействие пользователя с которыми сводится только к вводу исходной информации и управлению началом и окончанием работы моделей.

*Диалоговыми* называются имитационные модели, позволяющие исследователю активно управлять ходом моделирования, приостанавливать сеанс моделирования, изменять значения параметров модели, корректировать перечень регистрируемых данных и т. д. [2].

## 1.3. Классификация моделей

В общем случае все модели, независимо от областей и сфер их применения, бывают трех типов: познавательные, прагматические и инструментальные [3].

***Познавательная модель*** – форма организации и представления знаний, средство соединения новых и старых знаний. Познава-

тельная 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/или использования прагматических и/или познавательных моделей.

Познавательные модели 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. [3].

### 1.3.1. Классификация моделей по степени абстрагирования от оригинала

По степени абстрагирования от оригинала (рис. 1.4) модели могут быть разделены на материальные (физические) и идеальные. К ***материальным*** относятся такие способы, при которых исследование ведется на основе модели, воспроизводящей базовые геометрические, физические, динамические и функциональные характеристики изучаемого объекта. Основными разновидностями физических моделей являются [3]:

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

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

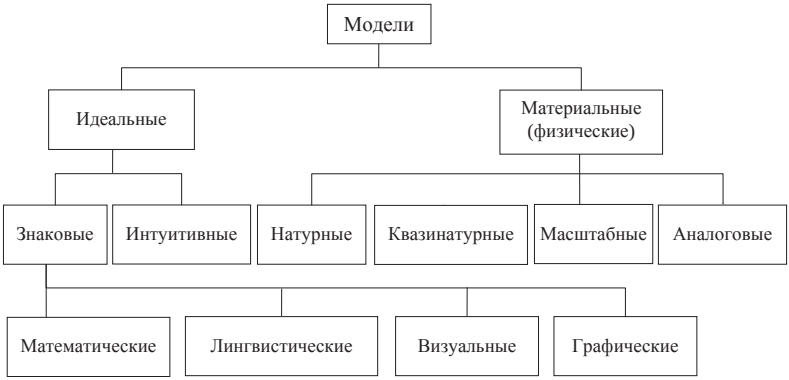



Рис. 1.4. Схема классификации моделей по степени абстрагирования от объекта-оригинала

***Квазинатурные*** модели (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.

***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, пневматиче-

ские системы, но наиболее широкое применение получили электрические и электронные аналоговые модели, в которых сила тока или напряжение являются 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 нуждающемся в ней. В этом смысле, например, 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. Виды знаковой модели: лингвистическая, визуальная, графическая и математическая.

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 представима 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 [3].

### 1.3.2. Классификация моделей по степени устойчивости

Все модели могут быть разделены на устойчивые и неустойчивые (рис. 1.5).

***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.

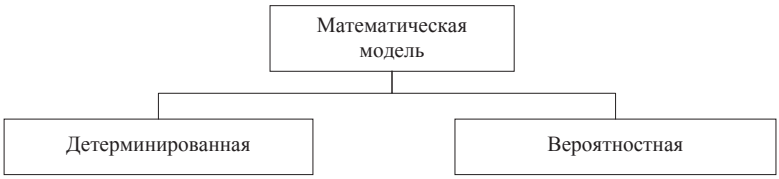



Рис. 1.5. Схема классификации математических моделей по устойчивости

В ***неустойчивой*** системе, находящейся первоначально в состоянии покоя, возникшее возмущение усиливается, вызывая увеличение значений соответствующих переменных или их колебания с возрастающей амплитудой [3].

### 1.3.3. 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. Например, 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*** [3].

### 1.3.4. Классификация моделей по отношению ко времени

По отношению к временному фактору модели делятся на динамические и статические (рис. 1.6).

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.

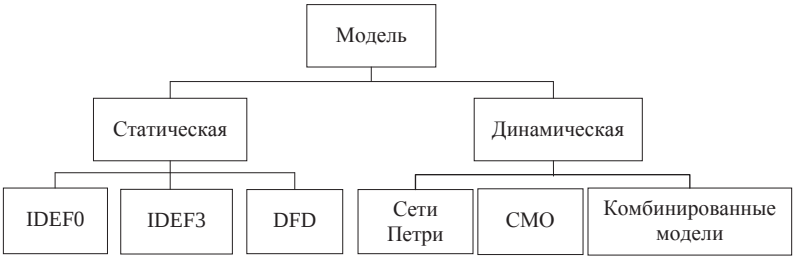



Рис. 1.6. Схема классификации математических моделей по отношению ко времени

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

## 1.4. Разновидности моделирования

Моделирование в широком смысле – это изучение объектов познания с помощью их моделей; построение и изучение моделей реально существующих предметов, процессов или явлений с целью

получения объяснений этих явлений, а также для предсказания явлений, интересующих исследователя [2].

Исследователь имеет дело с моделью, а не ее оригиналом. Классификацию видов моделирования можно проводить по разным признакам: по характеру моделей, по характеру моделируемых объектов, по сферам приложения моделирования. Мы выделим следующие виды моделирования:

• информационное (концептуальное) моделирование – процесс описания информации об объекте с помощью формализованных, неформализованных языков, образно-иллюстративных материалов и фиксированные в реальном материале эти представления и факты;

• эстетическое моделирование – процесс описания информации и объектов и явлений через ощущения и восприятия человека посредством живописи, декоративно-прикладного искусства и музыки;

• физическое моделирование – процесс разработки, конструирование натурных, физических, аналоговых или масштабных моделей объектов и исследование свойств и картины поведения объекта и реальных явлений на этих моделях;

• математическое (аналитическое и имитационное) моделирование;

• компьютерное моделирование.

В узком содержательном смысле под моделированием мы будем понимать ряд процессов:

• процесс описания или формализации объекта-оригинала, преследующий целью создание аналога (модели), адекватного объекту;

• процесс конструирования или проектирования объекта или его модели;

• изменение существующей модели в целях создания другой модели, более адекватной объекту-оригиналу;

• процесс проведения эксперимента на модели в целях прогнозирования и выяснения картины возможного поведения объекта-оригинала; для изучения различных характеристик и свойств объекта.

Рассмотрим более подробно виды математического и компьютерного моделирования. Математическое моделирование подразделяется на аналитическое и имитационное [2].

### 1.4.1. Математическое моделирование

Математические модели классифицируются:

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

Рассмотрим некоторые из них:

***По принадлежности к иерархическому уровню*** математические модели делятся на модели микроуровня, макроуровня, метауровня (рис. 1.7).

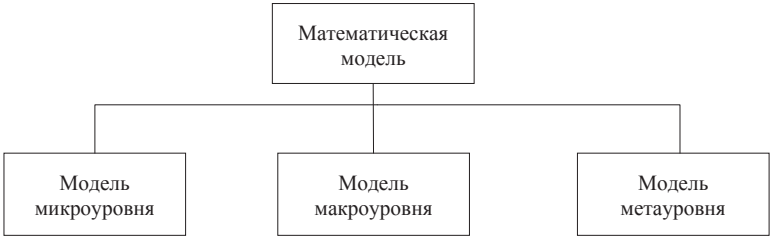



Рис. 1.7. Схема классификации математических моделей по принадлежности к иерархическому уровню

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).

***По характеру отображаемых свойств объекта*** модели можно классифицировать на структурные и функциональные (рис. 1.8).

Модель ***структурная***, если она представима структурой данных или структурами данных и отношениями между ними; например, структурной моделью может служить описание (табличное,

графовое, функциональное или др.) трофической структуры экосистемы. В свою очередь, структурная модель может быть иерархической или сетевой.

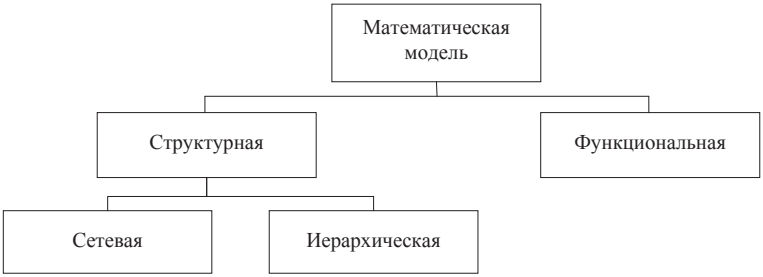



Рис. 1.8. Схема классификации математических моделей по характеру отображаемых свойств объекта

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. 1.9.

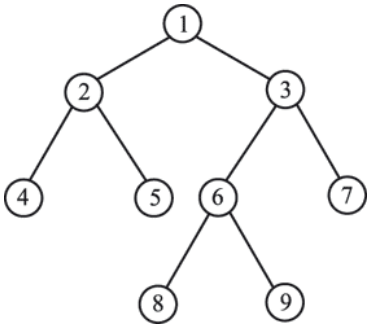



Рис. 1.9. Модель иерархической структуры

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. 1.10 и в табл. 1.1.

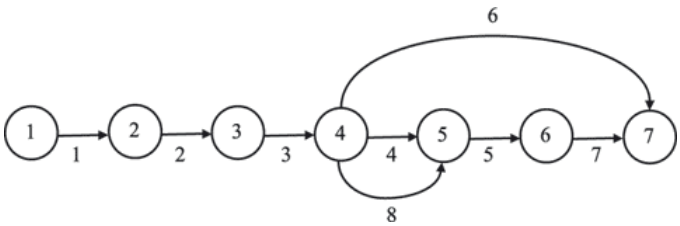



Рис. 1.10. Сетевой график строительства работ

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 |

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

***По способу представления свойств объекта*** (рис. 1.11) математические модели делятся на аналитические, алгоритмические и имитационные [3].

**Системный подход в математическом моделировании**

Материальные сущности – это собрание взаимосвязанных и взаимодействующих элементов, образующих системы разного уровня сложности [1].

*Топологическая* сложность определяется числом элементов и связей. *Функциональная* сложность характеризуется процессами (поведением) системы и ее элементов. По этим признакам можно

найти положение данного объекта в иерархии систем (вплоть до мировой) и сформировать предметную область моделирования.

На нижних уровнях главенствуют индивидуальные поведения, фиксированные физические связи, точные размеры, расстояния, скорости, времена. На верхних уровнях существенны глобальные причинные зависимости, тенденции, сценарии, динамика потоков, влияние обратных связей и окружающей среды, моделирование которой может быть выделено в отдельную и весьма непростую задачу.

В сложных системах состав элементов и типы связей могут существенно изменяться. Такие системы могут расти, стареть, умирать, перестраиваться и эволюционировать [1].

Систему как «организованно работающую целостность» характеризуют состояния и особенности их смены (рис. 1.12).

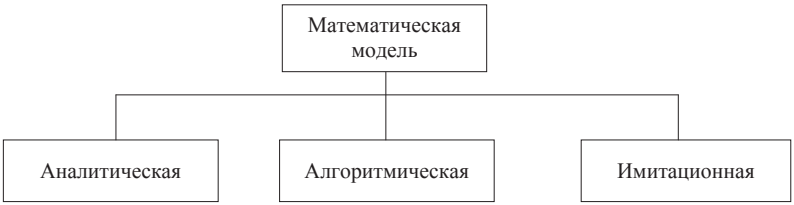



Рис. 1.11. Схема классификации математических моделей по способу представления свойств объекта

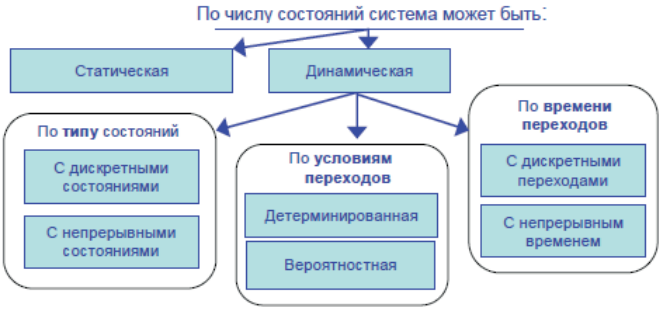



Рис. 1.12. Типы систем

При определении типа системы принимается решение, в рамках какой типовой математической схемы будет строиться модель (табл. 1.2) [3].

Таблица 1.2

Типовые математические схемы моделирования систем*

| Переходы состояния | Детерминированные | Вероятностные |
|---|---|---|
| Непрерывные | *D*-схемы | *Q*-схемы |
| Дискретные | *F*-схемы | *P*-схемы |

**D* (*dynamic*) – модели вида $dx/dt = f(x)$; *Q* (*queuing*) – модели систем массового обслуживания; *F* (*finite automata*) – конечные автоматы; *P* (*probabilistic automata*) – вероятностные автоматы.

### 1.4.2. Аналитическое моделирование

Под аналитическим моделированием мы будем понимать процесс формализации реального объекта и нахождение его решения в аналитических функциях. Модель, сформулированная на языке математики, физики, химии или другой науки с помощью системы специализированных символов с точными правилами сочетаемости называется аналитической моделью, чаще всего они представляются в виде формул, неравенств, линейных и нелинейных уравнений, в том числе дифференциальных и интегро-дифференциальных уравнений и их комбинаций [2].

Специалисты, занимающиеся математическим моделированием, исследование объекта или явления обычно начинают с поиска возможных аналитических решений упрощенной математической модели, используя различные приближения, т. е. на самом деле решают упрощенную задачу (модель), рис. 1.13.

Полученные аналитические решения для упрощенной модели удовлетворительно характеризует суть явлений. Аналитические решения позволяют понять и наглядно представить основные закономерности, особенно при изучении нового явления или процесса.

Однако возможности нахождения аналитического решения при исследовании непростых моделей ограничены, поэтому решения часто строятся в виде алгебраических итерационных формул.

Итерационные модели, представленные в виде алгебраических уравнений, можно решать приближенно, используя численные методы. Процедуру построения математической модели какого-либо реального явления или процесса и нахождения численного решения с помощью итерационных формул часто называют численным моделированием.

Рис. 1.13. Виды математического моделирования [2]

Теорию аналитического моделирования реальных процессов и технологии разработки компьютерных моделей можно изучать по книгам.

**Компьютерное моделирование** – это применение компьютерных технологий решения математических моделей на электронно-вычислительных машинах. Компьютерная модель (англ. *computer model*), или численная модель (англ. *computational model*), – это (1) компьютерная программа, работающая на отдельном компьютере, суперкомпьютере или множестве взаимодействующих компьютеров (вычислительных узлов), реализующая абстрактную модель некоторой системы; это (2) модель, выполненная с помощью компьютерных информационных, схематичных, электронных устройств и техноло-

гий и сетей; это (3) созданный за счет ресурсов компьютера виртуальный образ, качественно и количественно отражающий внутренние свойства и связи моделируемого объекта, иногда передающий и его внешние характеристики; это (4) модель, воспроизводящая моделируемый объект программными средствами на компьютере. Разработке компьютерной модели предшествуют мысленные, вербальные, структурные, математические и алгоритмические модели [2].

Компьютерные модели подразделяются на аналитические и имитационные. Компьютерные модели различаются по видам применения: обучающие, научно-исследовательские, научно-технические для исследования процессов и явлений, реальных объектов и промышленные, встроенные в производственный процесс или адекватно моделирующие производственные процессы на компьютерах. Имитационные модели не только отражают реальность с той или иной степенью точности, но и имитируют ее. Эксперимент с моделью либо многократно повторяется при разных исходных данных, чтобы изучить и оценить последствия каких-либо действий на реальную обстановку, либо проводится одновременно со многими другими похожими объектами, но поставленными в разные условия [2].

Имитационное моделирование при изучении сложных систем является практически основным доступным методом получения информации о поведении системы в условиях неопределенности.

Компьютерные модели сложных систем подразделяются условно на следующие виды [2]:

- структурно-функциональные, которые представляют собой условный образ объекта (технологические диаграммы, сетевые графики, структурные схемы, ГИС, табличный способ, анимационные и мультипликационные), описанный с помощью программных и компьютерных технологий;
- имитационные, представляющие собой программу или комплекс программ, позволяющий воспроизводить процессы функционирования объекта в разных условиях;
- комбинированные, с возможностями наблюдения и исследования объекта на динамических условных образах модели и имитационных моделях объекта.

Существует множество программных комплексов, которые позволяют проводить построение и исследование моделей (моделирование). Каждая программная среда имеет свой инструментарий и позволяет работать с определенными видами информационных моделей. Поэтому перед исследователем возникает нелегкий вопрос выбора наиболее удобной и эффективной среды для решения поставленной задачи. Надо сказать, что одну и ту же задачу можно решить, используя различные среды программирования и моделирования.

От выбора программной среды зависит алгоритм построения компьютерной модели, а также форма его представления. Например, это может быть блок-схема. Руководствуясь блок-схемой, задачу можно решить в разных средах. В среде программирования – это программа, записанная на алгоритмическом языке. В прикладных средах – это последовательность технологических приемов, приводящая к решению задачи [2].

Появление компьютеров позволило ускорить процесс нахождения решения математических моделей. Аналитические, численные и другие методы реализованы на ЭВМ. Было разработано множество компьютерных технологий моделирования. Это технологии моделирования на языках программирования, в системах компьютерной математики и схемотехнического моделирования. С помощью этих технологий создаются компьютерные вычислительные установки. Компьютерная имитация позволяет исследовать модель как в определенные моменты времени, так и в течение продолжительных периодов времени. Для нахождения решений (характеристик) при моделировании требуется его многократное воспроизведение с последующей обработкой, чаще всего с помощью компьютерных средств визуализации. В результате использования этих технологий мы получаем «компьютерное решение» рассматриваемой задачи [2].

**Имитационное моделирование**

В связи со стремительным развитием информационных и компьютерных технологий возможности моделирования реальных объектов расширились. Появились новые методы и технологии, позволяющие моделировать сложные объекты и процессы в промышленности, здравоохранении, в экономических и социальных си-

стемах, в науке и других сферах. Появление новых систем (пакетов) моделирования привело к созданию нового типа компьютерных моделей – «имитационных моделей» [2].

Под имитационным моделированием понимается разработка модели системы в виде программы для компьютера и проведение экспериментов с программой вместо проведения экспериментов с реальной системой или объектом.

Имитационное моделирование применяется, когда невозможно построить аналитическую модель системы, учитывающую причинные связи, последствия, нелинейности, стохастические переменные, когда необходимо имитировать поведение системы во времени, рассматривая различные возможные сценарии ее развития при изменении внешних и внутренних условий.

Таким образом, имитационное моделирование – это высокоуровневая информационная технология с применением компьютеров, чаще всего используемая при моделировании сложных систем. Существует достаточное количество работ, рассматривающих разные подходы и точки зрения к имитационному моделированию [2].

Появление новых современных программных продуктов существенно снижает требования к разработчику модели и открывает для специалистов широкого профиля, не обладающих навыками программирования, возможность разработки моделей, в том числе и для достаточно сложных систем. В то же время увеличивает требования к постановщику задач. Вышесказанное предопределяет чрезвычайно широкие возможности по применению методов имитационного моделирования при изучении социальных явлений, образовательной деятельности, при обучении управленческих кадров и т. д. специалистами в этих отраслях. Однако существует другая опасность при разработке и интерпретации результатов имитационных моделей сложных систем. Речь идет о создании плохих и уродливых моделей [2].

Модель представляет собой упрощенное отображение реальности – это менее детальное, менее сложное, менее подробное воспроизведение реально существующего объекта, системы или феномена, процесса. При этом, когда мы говорим о моделировании социальных феноменов, модель представляет не просто упрощение

реальности, а отображение реальности через призму определенного теоретического подхода или мнения эксперта.

Имитационное моделирование условно может быть представлено различными разновидностями или направлениями, соответственно имеющими свои методологии (рис. 1.14). Рассмотрим эти направления.

Рис. 1.14. Разновидности (направления) имитационного моделирования

*Статистическое (численное) моделирование* является разновидностью имитационного моделирования. Изначально оно появилось в теории случайных процессов и математической статистике как способ вычисления статистических характеристик случайных процессов путем многократного воспроизведения течения процесса с помощью модели этого процесса. Этот подход к исследованию реального процесса был назван методом статистических испытаний (методом Монте-Карло). Модели здесь строятся для явлений и систем объектов, входы и (или) функциональные соотношения между различными компонентами которых содержат элементы случайности или полностью случайных процессов, подчиняющиеся вероятностным законам.

Осуществление решения вероятностной модели реального объекта производится на ЭВМ. Машинная имитация позволяет ис-

следовать модель, как в определенные моменты времени, так и в течение продолжительных периодов времени. Для нахождения устойчивых решений (характеристик) при численном статистическом моделировании требуется его многократное воспроизведение с последующей статистической обработкой. Здесь проводится имитация воздействия многочисленных случайных факторов на различные элементы модели. Каждое воздействие на процесс в модели представляется в виде «розыгрыша» случайного явления с помощью процедуры, дающей случайный результат. Множество таких реализаций в ходе одного варианта имитации дает одну реализацию (историю) процесса. Затем вычисляются средние статистические характеристики по многим историям [2].

Статистическое моделирование в зависимости от области применения подразделяется на несколько направлений (рис. 1.15).

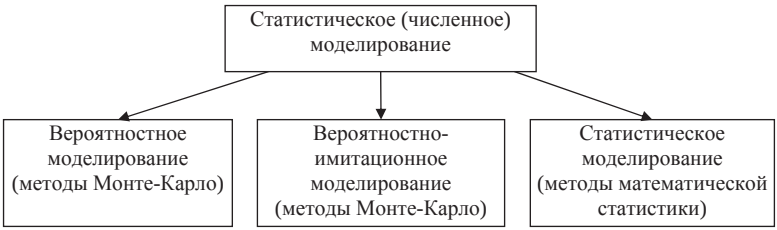



Рис. 1.15. Классификация видов статистического численного моделирования по типу решаемых задач

Вероятностное моделирование (методы Монте-Карло) – это направление развивается как способ решения математических задач – вычисление интегралов, решение систем линейных уравнений и дифференциальных уравнений.

Вероятностно-имитационное моделирование – применение теории вероятностей и методов Монте-Карло для построения имитационных моделей в молекулярной, статистической, квантовой, нейтронной физике, геофизике, газовой динамике, химической кинетике, в передаче и защите информации, в моделях массового обслуживания, финансовой математики, математической биологии и др. [2].

Статистическое моделирование – применение математической статистики для статистического оценивания и прогнозирования, корреляционно-регрессионного и многомерного статистического анализа, оптимизации систем, определения экстремума функций большого числа переменных и др. в различных отраслях производства и науки.

В экономических и социальных науках чаще всего используется статистическое моделирование. Здесь при разработке модели решаются задачи формализации экспертного знания и различных теоретических концепций, при разработке модели максимально исчерпывающе описываются положения теоретической концепции или результаты экспертного анализа (например, в форме мозгового штурма, социологического экспертного опроса и пр.). Сама по себе подобная формализация является важным результатом, и моделирование в этом свете может рассматриваться как своеобразный формальный язык и выступать в качестве определенного аналога математики для естественных наук. Результатом применения статистического моделирования является возможность проведения прогноза. Прогнозирование можно считать одним из наиболее ценных приложений имитационного моделирования [2].

**Области применения имитационного моделирования**

Имитационное моделирование доказало свою успешность во многих областях применения. Появление новых методов моделирования и рост вычислительной мощности компьютеров позволяет утверждать, что количество этих областей будет только расти [6].

На рис. 1.16 вы можете видеть распределение областей применения имитационного моделирования соответственно используемым в моделях уровням абстракции.

В нижней части рисунка располагаются модели физического уровня, в которых объекты реального мира моделируются максимально подробно. На этом уровне мы учитываем физическое взаимодействие, размеры, скорости, расстояния. Антиблокировочная система тормозов автомобиля, эвакуация болельщиков со стадиона, движение на регулируемом перекрестке, взаимодействие солдат на поле боя – все эти примеры требуют низкого уровня абстракции при их моделировании.

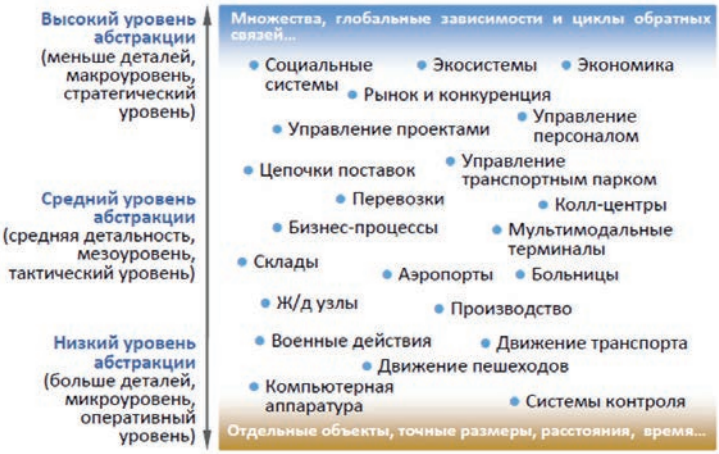



Рис. 1.16. Применение имитационного моделирования

Модели, расположенные в верхней части схемы, более абстрактны и чаще всего оперируют обобщенными понятиями, такими как совокупность потребителей или статистика уровня занятости, а не отдельными объектами. Так как взаимодействие между объектами происходит на высоком уровне, такие модели помогают понять взаимосвязи в системе без необходимости моделировать промежуточные шаги, например изучить влияние вложений в рекламу на продажи продукта компании.

Другие модели имеют средний уровень абстракции. Например, при моделировании отделения скорой помощи необходимо учитывать реальные размеры помещения, чтобы узнать, как долго пациент будет идти от приемной до рентгеновского кабинета. При этом, предположив, что помещение не переполнено, можно исключить из рассмотрения физическое взаимодействие между людьми.

Разрабатывая модель бизнес-процесса или работы центра обработки звонков, моделируют последовательность и длительность операций, а не место, в котором они происходят. В модели грузоперевозок учитывают скорость грузовика или поезда, но в модели цепочки поставок на более высоком уровне просто считают, что доставка заказа занимает от семи до десяти дней.

От правильности выбора уровня абстракции зависит успешность проекта моделирования. После того, как Вы решите, что включать в модель, а что оставить за пределами уровня абстракции, выбрать метод моделирования будет уже не так сложно.

Нормально и даже ожидаемо, что в процессе разработки модели Вам порой придется пересматривать выбранный уровень абстракции. В большинстве случаев Вы начинаете с высокого уровня, а позже добавляете необходимые детали [6].

**Методы имитационного моделирования**

В имитационном моделировании под методом понимается некая основа, которую мы используем, чтобы «перевести» систему из реального мира в мир моделей. Метод предполагает определенный язык, «положения и условия» для разработки модели. На данный момент существует три метода (рис. 1.17) [6]:

- системная динамика;
- дискретно-событийное моделирование;
- агентное моделирование.

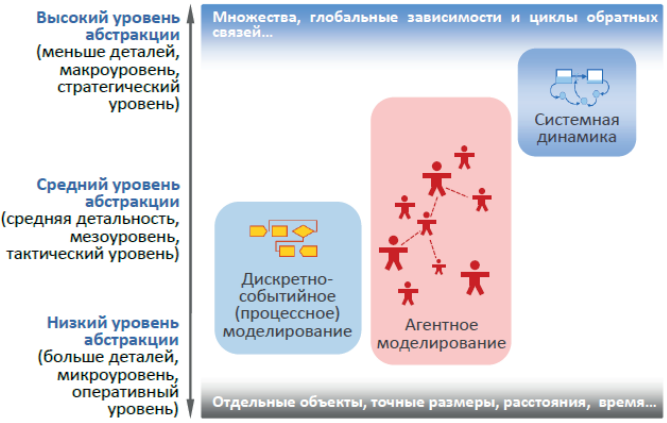



Рис. 1.17. Уровни абстракции видов моделирования [6]

Каждый метод применяется в некотором диапазоне уровней абстракции. Системная динамика предполагает очень высокий уровень абстракции и, как правило, используется для стратегического

моделирования. Дискретно-событийное моделирование поддерживает средний и низкий уровни абстракции. Между ними находятся агентные модели, которые могут быть как очень детализированными, когда агенты представляют физические объекты, так и предельно абстрактными, когда с помощью агентов моделируются конкурирующие компании или правительства государств.

Прежде чем выбрать метод моделирования, следует тщательно исследовать моделируемую систему и цели моделирования. На схеме ниже показано, что конкретная задача, стоящая перед разработчиком, во многом определяет подход к моделированию супермаркета. Разработчик может построить диаграмму процессов, в которой участвуют покупатели-заявки и кассиры-ресурсы, или агентную модель, в которой на покупателей-агентов влияет реклама и общение между собой и с сотрудниками-агентами компании, или диаграмму потоков и накопителей, в которой продажи связаны с рекламой, качеством сервиса, ценами и лояльностью клиентов. Иногда отдельные части системы проще моделировать с помощью методов, отличных от основного. В таких ситуациях лучше всего строить многоподходные модели (рис. 1.18).

Рис. 1.18. Схема многоподходного моделирования [6]

*Динамические системы.* Под *динамической системой* будем понимать любой объект, процесс или явление, для которого однозначно определено понятие состояния как совокупности некоторых величин и задан закон, который описывает изменение начального состояния с течением времени, двигающийся в пространстве и изменяющийся во времени. Динамическими объектами могут быть механические, производственные, физические, химические, биологические объекты, вычислительные процессы и др. [2].

Динамические системы описываются различными способами: дифференциальными уравнениями, дискретными отображениями, марковскими цепями, графическими образами и др. Они классифицируются в зависимости от вида оператора отображения и структуры фазового пространства. Различают линейные и нелинейные, непрерывные и дискретные операторы, соответственно определяются линейные и нелинейные системы, с *дискретным* временем и с *непрерывным* временем.

В основе методологии *моделирования динамических систем* и построения объектно-ориентированных моделей в технических системах лежит агрегативный подход, который был заложен в 1960–70-х годах гениальным советским ученым Н. П. Бусленко, здесь сложная система представлялась в виде агрегата (черного ящика), имеющего множество входных и выходных сигналов и воздействующих управляющих сигналов. Математически агрегат задается совокупностью множеств $T$, $X$, Г, $Y$, $Z$ и случайными операторами $H$ и $G$, где $T$ – множество моментов времени, $X$, Г, $Y$ – множества входных, управляющих выходных сигналов агрегата, $H$ и $G$ – операторы переходов и выхода. Этот подход широко используется при исследовании сложных индивидуальных управленческих систем, к которым относятся АСУ. Агрегативные системы позволяют описать широкий круг объектов исследования с отображением системного характера этих сложных объектов, с возможностью расчленения сложной системы на конечное число подсистем, с сохранением связей между ними и взаимодействия частей. В теории автоматического управления основным объектом изучения являются системы управления сложными динамическими (техническими) объектами и ее элементами. Математические модели системы автоматического управления и ее элементов представляются в виде уравнений динамики (движения), которые записываются либо в форме дифференциаль-

ных, интегральных и разностных уравнений, либо в виде уравнений «вход – выход» (в общем случае матричных уравнений) в пространстве состояний, благодаря которому они нашли широкое применение в инженерной практике. Описание динамических систем и элементов в пространстве состояний позволяет легко перейти к уравнениям для моделирования на ЭВМ, а также провести моделирование систем автоматического управления в виде структурных схем с помощью аппарата передаточных функций и динамических звеньев. Для моделирования динамических систем используются так называемые среды схемотехнического моделирования: VISSIM, SIMULINK+MATLAB, PowerSim, Multisim, LabView, Easy5, MvStudium и др. [2].

*Дискретно-событийное моделирование* обязано своим рождением Дж. Гордону, который в начале 1960-х спроектировал и реализовал на IBM систему дискретно-событийного программирования GPSS (Global Purpose Simulation System). Основной объект в этой системе – пассивный транзакт (заявка на обслуживание), который может определенным образом представлять собой работников, клиентов, покупателей, детали, сырье, документы, сигналы и т. п. «Перемещаясь» по модели, транзакты становятся в очереди к одноканальным и многоканальным устройствам, захватывают и освобождают эти устройства, расщепляются, уничтожаются и т. д. Таким образом, дискретно-событийную модель можно рассматривать как глобальную схему обслуживания заявок. Аналитические результаты для большого количества частных случаев таких моделей рассматриваются в теории массового обслуживания.

Дискретно-событийное моделирование зародилось примерно тогда же, когда появилась системная динамика. В 1961 году инженер компании IBM Джеффри Гордон представил программу GPSS, которая считается первой реализацией метода моделирования на основе дискретных событий. В наши дни существует множество различных программных инструментов для дискретно-событийного моделирования, в том числе и современная версия GPSS [6].

Дискретно-событийное моделирование предполагает представление моделируемой системы в виде процесса, т. е. последовательности операций, выполняемых с агентами.

Модель задается графически в виде диаграммы процесса, блоки которой представляют собой отдельные операции. Как правило,

диаграмма процесса начинается с блока «источник», генерирующего агентов. Этот блок передает агентов в последующие блоки диаграммы, задающие операции моделируемого процесса. Завершается диаграмма процесса обычно блоком, уничтожающим этих агентов.

Под агентами, называющимися в GPSS транзакциями, а в некоторых других моделирующих программах – заявками, могут пониматься клиенты, пациенты, телефонные звонки, документы, компоненты изделий или сами изделия, поддоны, автомобили, проекты, идеи и т. д.

Под ресурсами могут пониматься персонал, врачи, рабочие, оборудование и транспорт.

Типовыми операциями дискретно-событийной модели являются задержка (моделирующая выполнение определенной операции, например обработку звонка или детали), обслуживание агента ресурсом, ветвление процесса и т. д. Поскольку агенты конкурируют за обладание ресурсами, необходимыми для выполнения операций, то это может приводить к задержкам, и практически во всех дискретно-событийных моделях присутствуют очереди.

Как времена прибытия агентов, так и времена их обслуживания обычно являются случайными величинами, и их значения генерируются функциями распределения вероятностей. Поэтому и сами дискретно-событийные модели являются стохастическими, и для получения репрезентативного результата модель должна проработать определенное время, или же нужно выполнить определенное количество прогонов модели.

Типовыми результатами дискретно-событийной модели являются [6]:

- занятость ресурсов;
- время, проведенное агентом в системе или определенной ее части;
- длины очередей;
- время ожидания;
- пропускная способность и узкие места системы.

Этот подход используется для описания функционирования системы (процесса) из одного состояния в другое дискретным образом в виде события. Подход к построению имитационных моделей, предлагающий аппроксимировать реальные процессы такими событиями,

и называется дискретно-событийным моделированием (*discrete event modeling*). Этот вид моделирования чаще всего используется для производственных процессов, где динамика системы может быть представлена как последовательность операций (процессное моделирование). Данный подход широко применяется в теории массового обслуживания, которая изучает широкий класс случайных процессов в системах распространения информации, информационно-коммуникативных системах (компьютеры, интернет, связь и т. д.) и в различных отраслях массового обслуживания (железнодорожный, автомобильный транспорт, аэропорты, поликлиники, санаторные и лечебные учреждения, любые торговые предприятия, сферы обслуживания и др.).

*Системная динамика* – парадигма моделирования, где для исследуемой системы строятся графические диаграммы причинных связей и глобальных влияний одних параметров на другие во времени, а затем созданная на основе этих диаграмм модель имитируется на компьютере [6].

Системная динамика – это подход к имитационному моделированию, своими методами и инструментами позволяющий понять структуру и динамику сложных систем. Также системная динамика – это метод моделирования, использующийся для создания точных компьютерных моделей сложных систем для дальнейшего их использования с целью проектирования более эффективной организации и политики взаимоотношений с данными системами. Вместе эти инструменты позволяют нам создавать микромиры-симуляторы, где пространство и время могут быть сжаты и замедлены так, чтобы мы могли изучить последствия наших решений, быстро освоить методы и понять структуру сложных систем, спроектировать тактики и стратегии для большего успеха.

Метод системной динамики был изобретен в 1950-х годах Дж. Форрестером из Массачусетского технологического института (MIT). Используя свой научный и инженерный опыт, Форрестер искал способ применения законов физики, в частности законов электрических цепей, к исследованиям и описанию динамики процессов социальных и экономических систем [6].

Системная динамика чаще всего используется для разработки долгосрочных стратегических моделей и предполагает высокий уровень агрегации объектов: модели системной динамики рассматрива-

ют людей, товары, ресурсы и другие отдельные элементы в количественных терминах.

Системная динамика предоставляет методы изучения динамических систем. Предполагается, что Вы:

- моделируете систему как закрытую структуру, которая сама определяет собственное поведение;
- обнаруживаете циклы обратной связи, уравновешивающего или усиливающего типа. Циклы обратной связи занимают центральное место в системной динамике;
- задаете накопители и потоки, которые на них влияют.

Накопители характеризуют состояние системы. Они содержат память системы. Модель работает только с совокупностью объектов: отдельные элементы, содержащиеся в накопителе, не различимы. Потоки представляют интенсивность, с которой меняются эти состояния системы.

Если Вам сложно разделить понятия потока и накопителя, представьте, что мы ими измеряем. Накопители обычно используются, чтобы обозначить совокупность людей, уровни запасов, денежные средства или знания, тогда как потоки измеряют количество чего-либо за период времени, например количество клиентов в месяц или долларов в год [6].

Цель данной главы – научить разрабатывать модели системной динамики в AnyLogic. Существующий метод основан Дж. Форрестером в 1950-х годах и используется для анализа сложных систем с нелинейными обратными связями. Такой вид моделирования более всех других парадигм помогает понять суть происходящего выявления причинно-следственных связей между объектами и явлениями. С помощью системной динамики строят модели общества, мировой динамики, бизнес-процессов, развития города, модели производства, динамики популяции, экологии и развития эпидемии и другие. Системная динамика – это подход имитационного моделирования, своими методами и инструментами позволяющий понять структуру и развитие сложных систем. Также системная динамика – это метод моделирования, использующийся для создания точных компьютерных моделей сложных систем для дальнейшего использования с целью проектирования более эффективной организации и политики

взаимоотношений с данной системой. Системная динамика главным образом используется в долгосрочных, стратегических моделях и принимает высокий уровень абстракции. Люди, продукты, события и другие дискретные элементы представлены в моделях системной динамики не как отдельные элементы, а как система в целом [6].

*Агентное моделирование* (agent-basedmodel (ABM)) – разновидность имитационного моделирования, современный метод, позволяющий исследовать работу децентрализованных агентов и то, как такое поведение определяет деятельность всей системы в целом.

В отличие от системной динамики аналитик определяет поведение агентов на индивидуальном уровне, а глобальное поведение возникает как результат деятельности множества агентов (моделирование «снизу вверх»).

Агентное моделирование – относительно новый метод моделирования. Поначалу оно являлось преимущественно предметом теоретических дискуссий в академических кругах, а начиная с 2000-х годов разработчики имитационных моделей стали использовать его на практике [6].

Переход к агентному моделированию был вызван:

- желанием глубже изучить системы, которые сложно описать традиционными методами моделирования;
- развитием технологии агентного моделирования (объектно-ориентированное моделирование, диаграммы состояний);
- быстрым ростом мощности процессоров и объема оперативной памяти компьютеров. Агентные модели более требовательны к ресурсам, чем модели системной динамики или дискретно-событийные модели.

Агентное моделирование предлагает разработчику моделей альтернативный взгляд на поведение системы.

Вы можете не знать ни поведения системы в целом, ни ее главных переменных и зависимостей между ними или не видеть четкой схемы процессов, но при этом понимать, как ведут себя отдельные элементы системы. В таком случае Вы можете начать создание модели с идентификации моделируемых объектов (агентов) и задания их поведения. Иногда Вам может понадобиться объединить агентов в сеть и позволить им взаимодействовать друг с другом либо же по-

местить агентов в среду, которая имеет свою собственную динамику. Таким образом, глобальное поведение системы формируется из многих десятков (тысяч, миллионов) параллельно протекающих процессов [6].

На данный момент не существует стандартного языка агентного моделирования. Структура агентной модели может быть определена как графически, так и с помощью сценариев. Поведение агента может быть задано различными способами. Если у агента есть состояние, от которого зависят его действия и реакции, то его поведение лучше всего задавать с помощью диаграммы состояний. Иногда поведение агента задается действиями, выполняемыми при наступлении определенных событий.

Иногда внутренняя динамика агента лучше всего задается с помощью дискретных событий или системной динамики. Также и динамика среды, в которой живут агенты, может моделироваться с помощью традиционных методологий. По этой причине многие агентные модели совмещают в себе несколько подходов к моделированию.

Агентами могут быть самые разные объекты (рис. 1.19): транспортные средства, оборудование, проекты, организации, земельные участки, люди и т. д.

Рис. 1.19. Агентные объекты [6]

Ученые до сих пор спорят, какими именно свойствами должен обладать объект, чтобы называться агентом: способностью действовать и реагировать на действия других, ориентироваться в пространстве, обучаться, взаимодействовать и общаться, обладать «интеллектом» и т. п. На практике в агентных моделях вам могут встретиться агенты любых типов: одни общаются друг с другом, а другие находятся в полной изоляции; одни живут в пространстве, а другие – нет; одни обучаются и приспосабливаются, а другие никогда не меняют своего поведения [6].

Приведем несколько полезных фактов об агентах, чтобы многообразие теорий не вводило Вас в заблуждение:

• Агенты не являются клеточными автоматами и не обязательно обитают в дискретном пространстве (как в игре «Жизнь»). Во многих агентных моделях пространство вообще отсутствует. Когда пространство все же необходимо, оно чаще всего является непрерывным (это может быть карта мира или план здания).

• Агенты – не обязательно люди. Агентом может быть все, что угодно: транспортное средство, оборудование, проект, организация или даже идея.

• Агентом может быть объект, кажущийся абсолютно пассивным. Например, в модели нефтепровода Вы можете представить сегмент трубы как агента, задав для него графики техобслуживания, вероятности происхождения аварий, логику проведения ремонтных работ, затраты и т. д.

• Агентов в модели может быть как много, так и мало. При этом агенты могут быть как одного типа, так и разных.

• Существуют агентные модели, в которых агенты вообще не взаимодействуют друг с другом. Например, в моделях потребления алкоголя, развития ожирения или хронических заболеваний индивидуальная динамика агента зависит только от его личных параметров и, в некоторых случаях, от среды [6].

*Ингредиенты агентного моделирования*

Агент – это некая сущность, обладающая активностью, автономным поведением, может принимать решения в соответствии с некоторым набором правил, взаимодействовать с окружением, а также самостоятельно изменяться. Агентами могут быть люди (потребители,

жители, работники, пациенты, доктора, клиенты, солдаты и др.); транспорт, оборудование (автомобили, краны, самолёты, вагоны, станки); нематериальные вещи (проекты, продукты, инновации, идеи, инвестиции); организации (компании, политические партии, страны) [2].

Среда – некоторое пространство, в котором находятся агенты, характеризуемая своими состояниями и факторами, агенты находятся в определенном месте этого пространства, с возможностью ориентирования и передвижения в данном пространстве.

Правила взаимодействия – законы взаимодействия агентов в окружающей среде, с процедурами принятия решения и выбора стратегии при очередном шаге взаимодействия.

Задача имитационного моделирования при агентном подходе заключается в определении характеристик состояния агентов и среды, изучения поведения агентов при различных ситуациях взаимодействия – и изменяющихся состояниях среды.

Через изучение поведения множества агентов в некотором пространстве согласно некоторым правилам взаимодействия возможно прогнозирование поведения системы в целом.

Цель создания агентных моделей – получить представление об этих глобальных правилах, общем поведении системы исходя из предположений об индивидуальном, частном поведении ее отдельных активных объектов и взаимодействии этих объектов в системе.

Агентный подход позволяет исследовать задачи коллективного взаимодействия, эффективно решать задачи прогнозирования. Агентные системы позволяют исследовать процессы самоорганизации, дают возможность естественного описания сложных систем, обладают высокой гибкостью [2].

*Методология когнитивного моделирования*, предназначенная для анализа и принятия решений в *плохо определенных ситуациях*, была предложена американским исследователем Р. Аксельродом. Изначально когнитивный анализ сформировался в рамках социальной психологии, а именно – когнитивизма, занимающегося изучением процессов восприятия и познания. Применение разработок социальной психологии в теории управления привело к формированию особой отрасли знаний – когнитологии, концентрирующейся на исследовании проблем управления и принятия решений.

Под когнитивными технологиями понимается широкий спектр технологий рационализации и формализации интеллектуальных систем создания и функционирования знаний, экспертизы, коммуникации и принятия решения. Когнитивные информационные технологии представляют собой совокупность методов, приемов, действий, процессов, осуществляемых в определенной последовательности, инструментальных средств (ПК), позволяющих преобразовать входную информацию в варианты управленческого решения.

В триаде «теория – натурный эксперимент – машинный имитационный эксперимент» последний блок является быстроразвивающимся научным методом, который применяется практически во всех высокотехнологических отраслях для моделирования сверхсложных систем. Когнитивная наука в широком смысле слова – совокупность наук о приобретении, хранении, преобразовании и использовании знания.

В настоящее время когнитивный подход является направлением исследования больших систем, к которым относятся социально-экономические, политические, экологические системы. Методология когнитивного моделирования развивается в направлении совершенствования аппарата анализа, моделирования и поиска решений в слабоформализуемых и плохо структурированных ситуациях при отсутствии или неполной информации о процессах, происходящих в таких ситуациях и условиях быстрых перемен [2].

*Ситуационное моделирование.* Ситуационное моделирование (*situational simulation*), ситуационное управление (*management situations*) – направление исследований и принятия решений, развиваемое с 60-х годов прошлого столетия.

Ситуационное управление – это метод управления сложными техническими и организационными системами, основанный на идеях теории искусственного интеллекта: представление знаний об объекте управления и способах управления им на уровне логико-лингвистических моделей, использование обучения и обобщения в качестве основных процедур при построении процедур управления по текущим ситуациям, применение дедуктивных систем для построения многошаговых решений.

Ситуационное моделирование – метод исследования ситуаций, включающий в себя построение модели реальной ситуации и проведение с ней различного рода мысленных экспериментов: прогнозирования направлений ее развития и (или) «проигрывание» на ней предполагаемых решений по управлению ситуацией с целью выбора оптимального. В целях исследования ситуаций различного вида разрабатывают компьютерные аналитические и имитационные модели для прикладных применений, например, различные компьютерные игровые модели для общества, бизнеса, производства подразделяются на деловые, экономические, военные, образовательные. С помощью таких моделей можно разрешать конфликтные ситуации, оказывать психологическую помощь, проигрывать поведение объекта и системы в различных ситуациях [2].

Особо ценное практическое значение приобретает создание многовариантных ситуационных моделей как перспективной, так и ретроспективной направленности.

Многовариантное ситуационное моделирование предоставляет его субъекту «поисковое поле», на котором происходит многовариантный выбор: например, избрание того или иного тактического приема, использование той или иной типовой версии. Многовариантное моделирование постоянно вызывает потребность выбирать, сравнивать, искать альтернативы, находить лучшие решения и средства их реализации. Основными достоинствами многовариантного моделирования являются его многофакторность и многофункциональность, гибкость и продуктивность. Разработка такого рода компьютерной имитационной модели для принятия решения в многовариантном ситуационном поле является одним из важных направлений имитационных исследований.

Логическая последовательность процедуры ситуационного моделирования может быть сведена к следующим его этапам:

а) постановка проблемы (определение задач моделирования);

б) построение модели ситуации (моделирование ситуации), т. е. заполнение структурных блоков конкретным содержанием;

в) абстрагирование от несущественных для исследования обстоятельств;

г) диагностика ситуации;

д) учет динамических факторов;

е) определение ряда возможных альтернативных решений по управлению ситуацией;

ж) «проигрывание» решений на модели и выбор оптимального.

Заметим, что далеко не всегда процесс ситуационного моделирования включает в себя вышеназванные этапы.

Для формализации и описания ситуаций моделей используют различные подходы:

- логика предикатов;
- дискретные ситуационные сети, представляющие семантическую сеть, в которой ситуация описывается ориентированным графом;
- универсальный семантический код, в котором используется конструкция SAO, где S – субъект совершает A – действие над объектом O; RX – коды, которые представляют собой язык бинарных отношений. Одним из направлений ситуационного моделирования является разработка компьютерных и игровых имитационных моделей – своеобразных тренажеров, позволяющих проводить обучение по производственным, транспортным, психологическим, педагогическим, управленческим и другим ситуациям. В основе данных тренажеров лежит имитационная модель, которая позволяет обучающимся применять определенный набор инструментов воздействия и управления и воспроизводит ответную реакцию социальной системы. Тренажеры такого рода (виртуальные игровые имитационные модели) уже получили широкое распространение в зарубежной практике [2].

Результатом развития направления ситуационного моделирования является ситуационное управление как научный подход к анализу и решению задач ситуационного отображения информации. Разработаны системы ситуационного отображения информации, которые позволяют решать задачи управления динамическими объектами (центры управления космическими станциями, цифровые системы анализа и изображений, предназначенные для интерпретации различных политических и военных ситуаций, системы опознавания различных объектов и др.). Эти системы позволяют строить изображения ситуаций, возникающих в предметной области, на

основе которых можно принимать управляющие решения в рамках поставленной задачи [2].

Важность развития этого направления показывает и тот факт, что в каждом регионе созданы ситуационные центры для обобщения и моделирования различных видов ситуаций.

Ситуационный центр теоретически представляется в виде совокупности интеллектуально организованных рабочих мест с автоматизированными операциями:

- поступления и пополнения информации;
- процедурами построения имитационных и тренажерных моделей;
- анализа ситуаций;
- прогона моделей;
- графического представления проигранных сценариев;
- экспертных систем прогнозирования.

В соответствии с поставленными задачами ситуационного центра возникают направления развития ситуационного моделирования. В зависимости от класса и масштаба задач ситуационные центры подразделяются на стратегические и оперативные.

Стратегические ситуационные центры решают сложные, масштабные, ответственные задачи, направленные на структурную и функциональную перестройку и ориентированы на объекты класса: регион, отрасль, крупное предприятие (холдинг), ведомство и т. п. Руководители, имеющие стратегические ситуационные центры, владеют большей информацией и возможностями ситуационного прогноза.

Оперативные ситуационные центры решают задачи свертки оперативной информации в ситуационную модель, дающие лицу, принимающему решение (ЛПР), возможность маневрирования в пределах своей компетентности (области интересов) или бизнеса. Эти центры настроены на объекты класса: предприятие (компания), проект, крупная акция, процесс и др. Здесь строятся имитационные модели компании или отрасли, позволяющие осуществлять оперативный контроль и анализ состояния компании, заметно улучшить процесс управления компанией. Для ситуационного моделирования применимы методологии когнитивного моделирования.

**Основные направления и перспективы развития имитационного моделирования**

Имитационное моделирование в настоящее время является одним из основных видов моделирования и исследования для экономических, производственных, экологических, демо- и этнографических систем, систем массового обслуживания и др. Оно заключается в создании модели-имитатора работы сложных (чаще всего при наличии стохастических факторов) систем и процессов при неполных знаниях о ряде процессов в моделируемых объектах. Одними из наиболее эффективных методов исследования указанных объектов и систем являются методы имитационного и комплексного моделирования. Имитационное моделирование (ИМ) – это метод исследования, который основан на том, что анализируемая динамическая система заменяется имитатором и с ним проводятся эксперименты для получения информации об изучаемой системе. Роль имитатора выполняет компьютерная программа или комплекс программ для ЭВМ, которая называется имитационной моделью [2].

Анализ перечисленных ранее особенностей формального описания и исследования сложных объектов и систем показывает, что при моделировании и управлении данными объектами и системами следует базироваться на концепциях и принципах, положенных в основу современных технологий системного (комплексного) моделирования. Более того, как показывает анализ, при решении актуальных в современных условиях проблем структурно-функционального синтеза облика гибридных интеллектуальных систем управления (ГИСУ) СОТО (сложно организационных технически объектов) целесообразно рассматриваемые технологии системного моделирования, традиционно связанные с количественными вычислениями, дополнить интеллектуальными информационными технологиями, ориентированными на символьную обработку информации. К указанным информационным технологиям принято относить [2]:

- технологии мультиагентного моделирования;
- технологии экспертных систем (Expert Systems) или систем, основанных на знаниях (Knowledge-Based Systems);
- технологии нечёткой логики (Fuzzy Logic);

• технологии искусственных нейронных сетей (Artificial Neural Networks);

• технологии вывода, основанного на прецедентах CBR (Case Based Reasoning);

• технологии естественно-языковых систем и онтологии;

• технологии ассоциативной памяти;

• технологии когнитивного картирования и операционного кодирования;

• технологии эволюционного моделирования.

Несмотря на рост производительности современных компьютеров, их мощности не хватает для моделирования задач, связанных с самолето- и автомобилестроением, логистикой, сборочным производством и т. п., когда имитационные прогоны моделей могут длиться часами. Одним из вариантов решения этой проблемы является использование параллельного и распределенного дискретно-событийного моделирования. В этой области есть даже стандарт HLA – High Level Architecture.

Вопросам практического использования результатов имитационного и комплексного моделирования сложных процессов и систем уделяется особое внимание на многих конференциях, в частности на конференциях ИММОД (имитационного моделирования).

Если говорить о направлениях построения и использования имитационных моделей, то они следующие [2]:

1. Модели, предназначенные для выявления функциональных соотношений, т. е. для определения природы и закономерностей между двумя или несколькими факторами с одной стороны и откликом системы с другой стороны.

2. Модели, предназначенные для прогноза, т. е. для оценки поведения системы при некотором сочетании рабочих условий, в том числе и по времени.

3. Модели для экспертной оценки предлагаемой структуры или конфигурации системы по предлагаемым некоторым критериям или совокупность аксиом, сформулированных с помощью экспертов. Эта совокупность является как бы основой (ядром) разработанной экспертной системы – имитационной модели.

4. Модели, позволяющие сопоставлять конкурирующие системы, рассчитанные на выполнение определенной функции или сравнение различных предлагаемых рабочих методик, принципов. Эти модели разрабатываются для принятия решений в виде утилиты, запускаются регулярно при принятии оперативных решений.

5. Модели для оптимизации в целях определения сочетания действующих факторов и их величин, при котором обеспечивается наилучший отклик системы в целом. Могут использоваться как инструмент для оценки и сравнения вариантов предполагаемых изменений или выработки оптимальной стратегии. Эти модели широко применяются при решении задач управления (планирования, проектирования) и часто называются интерактивными оптимизационными моделями (системами).

6. Модели виртуальных игр, предназначенные для обучения студентов, управленческого персонала, сотрудников. Разработка моделей виртуальной реальности – инструментов для компьютерных имитационных игр для различных отраслей, военных, экономических, государственных и межгосударственных отношений.

7. Модели для анализа чувствительности, т. е. для выявления тех факторов, которые в наибольшей степени влияют на общее поведение системы.

8. Модели, встроенные в производственный процесс, в технические и автоматизированные установки, запускающиеся автоматически при выполнении соответствующих операций.

9. Модели, созданные для динамической визуализации (демонстрации) проектируемого объекта в целях аргументации разрабатываемого проекта для руководства или потенциальных заказчиков (воспроизводящие виртуальную иллюзию процесса развития системы).

Говоря о моделировании как исследовательском методе и имитационном (компьютерном) моделировании как его разновидности, необходимо остановиться на основных областях применения, в которых могут решаться отраслевые задачи при помощи данного метода, о его преимуществах и особенно – об ограничениях.

Активнее всего имитационным моделированием интересуются в металлургии, нефтегазовой отрасли, производстве строймате-

риалов, пищевых продуктов, в различных областях массового обслуживания (аэропорты, транспорт, медицина, торговые предприятия, сферы обслуживания и др.).

Анализ результатов конференций ИММОД показывает, что сводный перечень основных областей применения выглядит следующим образом [2]:

- предоставление услуг связи, сети передачи данных;
- управление подвижными (космическими) объектами, воздушным и автотранспортным движением;
- разработка ситуационных имитационных моделей полета военно-воздушного транспорта;
- организация промышленного производства (ГПС, обувное, мелкосерийное, сборка персональных компьютеров, разработка программного обеспечения);
- проектирование рыбообрабатывающих комплексов на судах промыслового флота;
- информационное противоборство, блочное шифрование;
- динамика популяции зверей и животных;
- региональные экономические системы;
- лечебно-эвакуационные мероприятия в авиадивизии;
- налоговое и пенсионное законодательство, обращение граждан в органы государственного управления;
- подготовка специалистов по управлению железнодорожным транспортом и магистральными трубопроводами;
- исследования в области экосистем (водных, морских, лесных, степных и др.);
- исследования демографических и этнических систем.

Из области применения имитационного моделирования можно сформировать основные тематические направления имитационных исследований:

- теоретические основы и методология имитационного и комплексного моделирования;
- методы оценивания качества моделей и полимодельных комплексов;
- методы и системы распределенного моделирования;
- моделирование глобальных процессов;

• разработка средств автоматизации и визуализации имитационного моделирования;

• системная динамика (с обязательным наличием имитационной составляющей);

• практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования;

• виртуальное и цифровое производство – промышленное моделирование;

• имитационное моделирование в обучении и образовании. Применение имитационного моделирования достаточно медленно развивается из-за сложности рассматриваемых объектов и трудоемкости создания имитационных моделей, из-за отсутствия профессиональных кадров в области имитационного моделирования и по ряду других причин. Особенно сложно обстоит дело в областях социальных и гуманитарных наук, где слабое применение методов имитационного моделирования может быть объяснено причинами, изложенными выше, нелинейностью и многофакторностью социальных процессов, сложностью взаимосвязей, а также необоснованными ожиданиями, возлагаемыми на модель, которые не могут оправдаться в реальности. Тем не менее, формализацией социальных систем в настоящее время занимаются, определяются математические подходы и получены некоторые результаты.

Одна из причин неуспеха прогнозирования, планирования и проектирования социальных систем в настоящее время (в том числе крупных и сложных социально-экономических объектов, регионов) состоит в том, что [2]:

• отсутствуют компетентные в отрасли специалисты по системному анализу и имитационным исследованиям;

• отсутствуют соответствующие структуры в виде центров, институтов, занимающихся имитационными исследованиями;

• отсутствуют возможности самостоятельного владения инструментами имитационных исследований у большинства органов управления в силу их стоимости и необходимости наличия высокооплачиваемых компетентных специалистов со стороны.

Предлагаемые подходы к моделированию и проектированию будущего (широкомасштабные региональные и всероссийские проекты, финансируемые большими компаниями и государством) для уровня регионов и крупных компаний [2]:

• информирование научной и производственной общественности о причинах, полезности и перспективах применения имитационного моделирования систем и технологий проектирования будущего;

• для регионов – создание соответствующих структур – центров, институтов имитационных, ситуационных, когнитивных исследований;

• широкое распространение систем проектирования и имитационного моделирования будущего и практики их применения;

• эффективное обучение применению технологий проектирования и моделирования будущего в университетах;

• воспроизводство и «расширенное производство» специалистов, владеющих методологиями и технологиями имитационного моделирования и проектирования будущего, в том числе создание новых образовательных программ по имитационному моделированию и проектированию будущего.

В итоге сформулируем основные преимущества имитационного моделирования:

1. Имитационные модели позволяют анализировать системы и находить решения в тех случаях, когда такие методы, как аналитические вычисления и линейное программирование не справляются с задачей.

2. После того, как Вы определитесь с уровнем абстракции, разрабатывать имитационную модель будет гораздо проще, чем аналитическую, поскольку процесс создания модели будет инкрементальным и модульным.

3. Структура имитационной модели естественным образом отображает структуру моделируемой системы.

4. Имитационная модель позволяет вам отслеживать все объекты системы, учтенные в выбранном уровне абстракции, добавлять метрики и проводить статистический анализ.

5. Одним из главных преимуществ имитационного моделирования является возможность проигрывать модель во времени и ани-

мировать ее поведение. Анимация будет неоспоримым преимуществом при демонстрации модели и может оказаться полезной для верификации модели и нахождения ошибок.

6. Имитационные модели намного убедительнее электронных таблиц. Если Вы используете имитационное моделирование, то при презентации проекта у Вас будет явное преимущество перед теми, у кого на руках только цифры и решение, полученное из «черного ящика».

### 1.4.3. Информационные системы моделирования и проектирования

Имитационная модель представляется в виде компьютерной программы, компьютерной установки, которая описывает структуру и воспроизводит поведение реальной системы во времени. Имитационная модель позволяет получать подробную статистику о различных аспектах функционирования системы в зависимости от входных данных.

Имитационные модели могут создаваться в виде программ технологией прямого программирования, а также с помощью систем компьютерного моделирования, представленных на рис. 1.20.

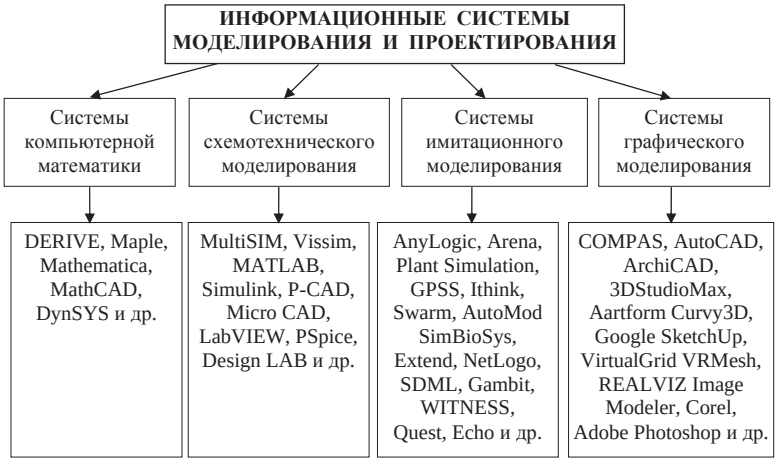



Рис. 1.20. Условная классификация информационных систем по типу решаемых задач [2]

В системах компьютерной математики, технического и имитационного моделирования предусмотрены возможности создания статистических (монтекарловских) имитационных моделей (генераторы случайных чисел, генераторы случайных величин, распределений и т. д.) и компьютерных имитационных моделей сложных систем.

В настоящее время идет стремительное развитие направления разработки инструментальных средств имитационного моделирования (ИСИМ), целенаправленно поддерживающих те или иные методологии и направления имитационного моделирования сложных систем (ИМСС):

• AnyLogic – программное обеспечение для имитационного моделирования сложных систем и процессов, позволяющее поддерживать направление агентного моделирования, дискретно-событийного моделирования и разработки моделей системной динамики (осуществляется российской компанией (англ. XJ Technologies) «Экс Джей Текнолоджис»);

• GPSS (англ. General Purpose Simulation System – общецелевая система моделирования) – язык объектно-ориентированного программирования, используемого для имитационного моделирования систем массового обслуживания, различных информационных процессов и разработки имитационных моделей в сети Интернет;

• Arena – разрабатываемое компанией Systems Modeling Corporation программное обеспечение для имитационного моделирования, позволяющее создавать подвижные компьютерные модели, используя которые можно адекватно представить очень многие реальные системы;

• PlantSimulation – программная среда имитационного моделирования систем и процессов, предназначенная для оптимизации материалопотоков, загрузки ресурсов, логистики и метода управления для всех уровней планирования от целого производства и сети производств до отдельных линий и участков;

• SimBioSys: C++ – оболочки агентно-базового эволюционного моделирования в биологических и общественных науках;

• система моделирования SWARM и ее расширения MAML (Multi-Agent Modelling Language) для моделирования искусственного мира;

• пакеты Ascape (Agent Landscape) и RePast (Recursive Porous Agent Simulation Toolkit), написанные на платформе языка Java для поддержки агентно-базового моделирования;

• NetLogo и MIMOSE (Micro- and Multilevel Modelling Software) – информационные системы, предназначенные для создания имитационных моделей и технологий моделирования в общественных науках;

• SPSS, Statistica, PilGrim, Z-Tree – системы статистического моделирования для исследования экономических, педагогических и психологических явлений и процессов.

Перечень программного обеспечения и инструментальных средств имитационного моделирования можно посмотреть на сайте http://dic.academic.ru/dic.nsf, а также сайте Национального общества имитационного моделирования: www.simulation.su.

В отрасли имитационного моделирования реальных объектов условно выделились четыре основных направления: моделирование динамических систем, дискретно-событийное моделирование, агентное моделирование и системная динамика. В табл. 1.3 приведены языки и средства автоматизации имитационного моделирования, которые однозначно, а некоторые условно можно отнести к соответствующим подходам (методологиям) имитационного моделирования [2].

Таблица 1.3

Инструментальные средства имитационного моделирования

| Динамические системы | Системная динамика | Дискретное событийное моделирование | Агентное моделирование |
|---|---|---|---|
| Dynamo, PowerSim, MIMIC, АРТОН MIDAS, PACTOLUS, CSSL, СЛАМ, GASP, НЕДИС, МИКС, MATLAB+Simulink, Multisim VisSim, LabView, Easy5, MvStudium и др. | AnyLogic, Arena, SimBioSys, eM-Plant, Tecnomatix, Plant Simulation, SimuLab, VenSim, PowerSim, Pilgrim, Dynamo, Stella, Ithink и др. | AnyLogic, Arena, Extend, PowerSim Studio, Witness, ProModel, Pilgrim, Taylor Simulation, GPSS, SimScript, Quest, SIMULA, SIMUL8, Modelling, SimProcess, AutoMod, Enterprise Dynamics, FlexSim и др. | AnyLogic, Swarm+MAML, SimAgent, SimBioSys, C++, Java, AgentSpeak, Oz, TeleScript, RePast, NetLogo, Ascape, Mason и др. |

Как видно из таблицы, систем имитационного моделирования достаточно много, однако не все перечисленные программные продукты доступны для использования. Многие программные продукты по автоматизации имитационного моделирования, представленные в табл. 1.3, не используются в России.

Это связано с тем, что отсутствуют представительства разработчиков этих систем имитационного моделирования, или большинство из представленных инструментальных средств являются коммерческими и недоступны по причине дороговизны продукта как для университетов, так и для коммерческих IT-компаний, занимающихся имитационными исследованиями. Наиболее широко в России используются следующие зарубежные системы имитационного моделирования: Arena (www.interface.ru), GPSSWorld (www.gpss.ru), платформа ARIS (www.softwareag.com/ru), VISSIM, VISUM (www.ptvvision.ru), QuestDelmiaSolution (www.3ds.com) и др.; свободно распространяются и студенческие продукты имитационного моделирования Simplex3, PlantSimulation и др.

Системная динамика и дискретно-событийное моделирование – традиционные устоявшиеся подходы, агентное моделирование – относительно новый подход. Подход динамического моделирования позволяет увидеть поведение модели во времени при движении в прошлое (для получения исторического результата) и в будущее (для выявления возможных исходов). При изменении параметров модели можно наблюдать причины успеха или неудачи, находить оптимальные решения. Математически системная динамика и динамические системы оперируют в основном с непрерывными во времени процессами, тогда как дискретно-событийное и агентное моделирование – в основном с дискретными.

Для разработки моделей сложных систем по данным подходам используются среды имитационного моделирования, разработанные в России: AnyLogic (www.anylogic.ru), Pilgrim (www.mfpa.ru), RandModelDesigner (www.mvstudium.com), расширенный редактор GPSS (www.elina-computer.ru) и др.

В табл. 1.4 приведены некоторые CASE-средства анализа и проектирования программных систем. Здесь указаны производители этих средств, нотации и диаграммы, которые они поддерживают.

Таблица 1.4

Средства проектирования

| Система проектирования | Производитель ПО | Нотация DFD | Поддержка | |
|---|---|---|---|---|
| | | | методологии | диаграмм |
| Системы анализа и проектирования | | | | |
| BPWin | Logic Works | Гейна – Сарсона | IDEF0/SADT IDEF3/SADT | Функциональной декомпозиции |
| ProKit *Work bench | MDIS | Гейна – Сарсона | Собственная STRADIS | Потоков данных в нотации Гейна – Сарсона, сущность-связь, структурные карты Константайна |
| CASE. Аналитик | Эйтекс | Гейна – Сарсона | Структурного системного анализа Гейна – Сарсона | Функциональной декомпозиции, потоков данных, управляющих потоков, структуры данных, сущность-связь |
| CASE /4/0 | MicroTO OL | Йодана (расшир.) | Анализа и проектирования систем реального времени Уорда – Меллера | Функциональной декомпозиции, потоков данных, переходов состояний, карты Джексона |
| Design/ IDEF | Meta Software | – | IDEF0, IDEF1 IDEF1X, IDEF/CPN | Сущность-связь |
| Visible Analyst Work-bench | Visible Systems | Гейна – Сарсона, Йодана | Информационного моделирования Мартина | Функциональной декомпозиции, сущность-связь, потоков данных в нотации Йодана и Гейна – Сарсона, структурные карты Константайна |
| Rational Rose | Rational Software | UML, также поддерживается нотация Буча и OMT-2 | Собственная методология «Rational Objectory Process», UML | Вариантов использования, взаимодействия объектов, последовательности взаимодействий, переходов состояний, классов, пакетов, компонентов, размещения |
| UML 2.x | Object Management Group | Собственная | UML | Классов, компонентов, композитной/составной структуры, кооперации (UML 2.0), объектов, пакетов, профилей (UML 2.2), деятельности, состояний, прецедентов, коммуникации (UML 2.0) / кооперации (UML 3.x), обзора взаимодействия (UML 2.0), последовательности, синхронизации (UML 2.0) |

Окончание табл. 1.4

| Система проектирования | Производитель ПО | Нотация DFD | Поддержка | |
|---|---|---|---|---|
| | | | методологии | диаграмм |
| ARIS Toolset | Software AG | Нотация ARIS eEPC | UML | Сущность-связь, описания процессов, информационных потоков |
| Системы проектирования баз данных и файлов | | | | |
| Designer/ 2000 | Oracle | Гейна – Сарсона | Собственная CASE*Method | Сущность-связь, потоков данных, иерархии функций, взаимодействия модулей, структурные карты Джексона |
| ERWin | Logic Works | Гейна – Сарсона | IDEF3/SADT | Сущность-связь |
| EasyCASE | Evergreen CASE Tools | Гейна – Сарсона, Йодана | Структурного системного анализа Гейна – Сарсона | Сущность-связь, потоков данных, переходов состояний, структурные карты в нотации Константайна |
| Vantage Team Builer | CAY-ENNE | Йодана | Структурного анализа и проектирования Чена и Йодана | Сущность-связь в нотации Чена, потоков данных в нотации Йодана, переходов состояний, структурные карты Константайна |
| Chen Toolkit | Chen &Associates | Чена | Чена | Сущность-связь, потоков данных, переходов состояний |
| S-Designor | Sybase/ Powersoft | – | Информационного моделирования | Сущность-связь |
| SILVER-RUN | Computer Systems Advisers | Произвольная | DATARUN | Потоков данных BPM, сущность-связь ERX и RDM |

Эти системы поддерживают те или иные методологии анализа и проектирования информационных систем, которые определяют руководящие указания для оценки и выбора проекта программного продукта, последовательность шагов выполнения работы, правила распределения и назначения операций и методов [2].

Практически любая современная крупная программная система разрабатывается с применением CASE-технологий, по крайней мере, на этапах анализа и моделирования, что связано с большой

сложностью данной проблематики и со стремлением повысить эффективность работ.

Целью систем анализа и проектирования является определение системных требований и свойств, создание проекта и архитектуры информационной системы, а также детальная «калька» проекта, включающая алгоритмы и определения структур данных.

Функциональная модель – описание системы с помощью IDEF0. Данная модель предназначена для характеристики существующих бизнес-процессов, использует как естественный, так и графический языки. Для передачи информации о конкретной системе источником графического языка является сама методология IDEF0.

Методология IDEF0 предназначена для построения иерархической системы диаграмм – единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего происходит функциональная декомпозиция – система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и т. д. до достижения нужной степени подробности.

Каждая IDEF0-диаграмма содержит блоки и дуги. Блоки – функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.

Функциональные блоки (работы) на диаграммах изображаются прямоугольниками, означающими поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты.

Каждая сторона блока имеет особое назначение. Левая сторона блока предназначена для входов, верхняя – для управления, правая – для выходов, нижняя – для механизмов. Такое обозначение отражает определенные системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, что и как выполняет функция.

Остановимся на малых интегрированных средствах проектирования информационных систем. Типичный представитель малых интегрированных средств моделирования – комплект программных

продуктов PlatinumTechnology (CA/Platinum/LogicWorks), основанный на популярных пакетах BPwin и Erwin [2].

***BPWin.*** Компания LogicWorks, разработчик BPwin, сейчас входящий в ComputerAssociates, работает на рынке технологий моделирования уже более 20 лет. Для проведения анализа и реорганизации сложных систем и процессов LogicWorks предлагает CASE-средство верхнего уровня – BPwin, который поддерживает три методологии:

- IDEF0 (функциональная модель);
- IDEF3 (Work Flow Diagram) – только диаграммы процессов;
- DFD (Data Flow Diagram) – диаграммы потоков данных.

Функциональная модель предназначена для описания существующих систем и процессов (так называемая модель AS-IS) и идеального положения вещей – того, к чему нужно стремиться (модель TO-BE).

Интеграция выполняется как путем слияния нескольких моделей, так и посредством переключения на различные методологии в процессе разработки отдельных диаграмм информационной модели. Предусмотрено расширение возможностей анализа систем как в самом пакете BPwin (функционально-стоимостный анализ), так и с помощью экспорта данных в другие пакеты. BPwin автоматизирует задачи, связанные с построением моделей развития, обеспечивая семантическую строгость, необходимую для гарантирования правильности и непротиворечивости результатов.

***ERwin.*** Поддерживает несколько разновидностей методологии информационного моделирования, основанной на ER-диаграммах (сущность – связь). Интеграция моделей BPwin с моделями ERwin выполняется путем обмена данными через функции экспорта/импорта.

Основной из трех методологий является IDEF0, она относится к семейству IDEF, которое было введено в 1973 году Россом под названием SADT (Structured Analysisand Design Technique). Хотя основной акцент использования малых интегрированных систем делается на применение к бизнес-процессам на предприятиях, эти технологии употребимы для декомпозиции и проектирования широкого класса систем, в том числе информационных систем, в частности для проектирования сайтов, порталов, систем и программных комплексов компьютерных установок для моделирования реальных объектов. Для информационных систем применение IDEF0 имеет

своей целью определение требований и указание функций для последующей разработки системы, отвечающей поставленным требованиям и реализующей выделенные функции. Применительно к уже существующим системам IDEF0 может быть использована для анализа функций, выполняемых системой, и отображения механизмов, посредством которых эти функции выполняются.

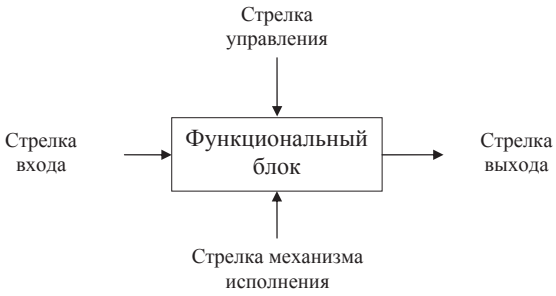



Рис. 1.21. Соединение стрелок со сторонами функционального блока

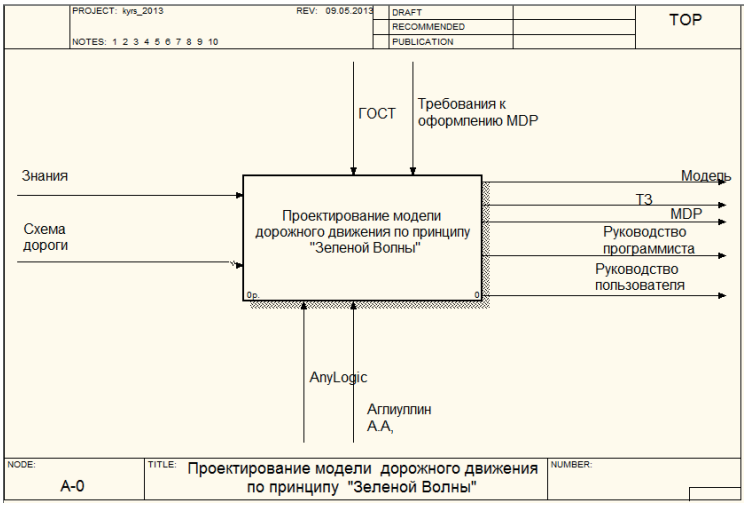



Рис. 1.22. Функциональная модель дорожного движения

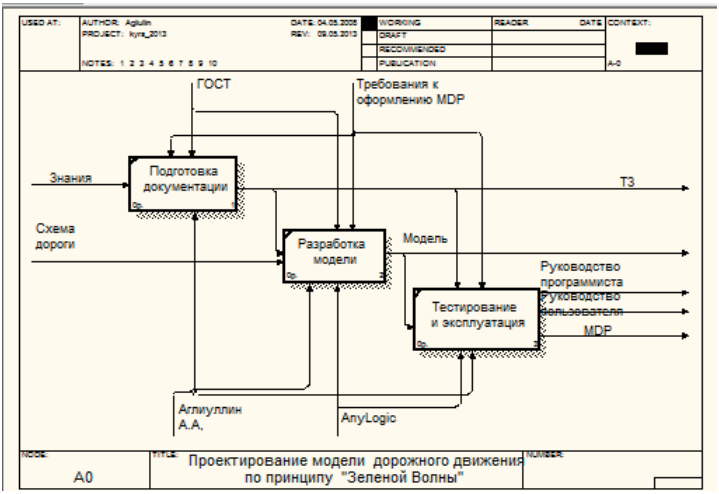



Рис. 1.23. Модель декомпозиции

Первая диаграмма в иерархии диаграмм IDEF0 всегда изображает функционирование информационной системы в целом (рис. 1.21).

Такая диаграмма называется контекстной. В контекст входит описание цели моделирования, области (описания того, что будет рассматриваться как компонент системы, а что – как внешнее воздействие) и точки зрения (позиции, с которой будет строиться модель). Функциональная модель дорожного движения по принципу «Зеленой волны» представлена на рис. 1.22, а ее декомпозиция – на рис. 1.23.

## Контрольные вопросы и задания

1. Чем, в сущности, является моделирование?

2. Что называется изоморфизмом (от греч. *isos* – равный и *morphe* – форма)?

3. Что может являться объектом для моделирования и какие виды моделей могут быть реализованы?

4. Если мы рассматриваем область имитационного моделирования, то чем является объект в стратегии объектно-ориентированного подхода?

5. Что такое система?

6. Перечислите ряд подходов к выделению систем по сложности и масштабу.

7. Как могут быть разделены модели по степени абстрагирования от оригинала?

8. Какая модель является замкнутой?

9. Специалисты, занимающиеся математическим моделированием, исследование объекта или явления обычно начинают с поиска возможных аналитических решений упрощенной математической модели, используя различные приближения. Какими приближениями они пользуются?

10. Что понимается под имитационным моделированием?

11. Что понимается под динамической системой?

12. Что является типовыми результатами дискретно-событийной модели?

13. Ситуационное управление – это метод управления?

14. Что содержит каждая IDEF0-диаграмма?

15. Каким путем выполняется интеграция моделей BPwin с моделями ERwin?

# Глава 2

# СИСТЕМНАЯ ДИНАМИКА

## 2.1. Методология системной динамики

Системная динамика – парадигма моделирования, где для исследуемой системы строятся графические диаграммы причинных связей и глобальных влияний одних параметров на другие во времени, а затем созданная на основе этих диаграмм модель имитируется на компьютере. Такой вид моделирования помогает понять суть происходящего обнаружения причинно-следственных связей между объектами и явлениями. Системная динамика применяется для решения производственных, организационных и социально-экономических задач [2].

Системная динамика как метод имитационного моделирования включает в себя:

- структуризацию объекта;
- построение системной диаграммы объекта, где указываются связи между элементами;
- определение переменных для каждого элемента и темпов их роста;
- принятие гипотез о зависимости каждого темпа роста от переменных и формальное описание этих гипотез;
- процесс оценки введенных параметров с помощью имеющейся статистики.

Для построения и исследования моделей с помощью метода системной динамики в 1958 году был разработан специальный язык программирования DYNAMO (DYNAmic MOdels).

Модель Форрестера включает следующие элементы [2]:

- *уровни (ресурсы, резервуары)* характеризуют накопленные значения величин внутри системы. Это могут быть заготовки, комплектующие и готовая продукция, товары на складе, товары в пути, банковская наличность, страховые межоперационные запасы, про-

изводственные площади, численность работающих, финансовые ресурсы и т. д. Уровни применимы не только к физическим величинам. Например, уровень осведомленности существенен при принятии решения. Уровни удовлетворения, оптимизма и негативных ожиданий влияют на экономическое поведение. Уровни представляют собой значения переменных, накопленные в результате разности между входящими и исходящими потоками. Каждый уровень описывается его переменной величиной, зависящей от разности входящих и исходящих потоков. Например, производительность транспортной системы может быть выражена количеством товаров в пути (уровень) и константой (запаздывание на время транспортировки). Более сложный пример: решение о найме рабочих может быть связано с уровнями имеющейся рабочей силы, среднего темпа поступления заказов, числа работников, проходящих курс обучения, числа вновь принятых работников, задолженности по невыполненным заказам, запасов, наличия оборудования и материалов. На диаграммах в модели изображаются прямоугольниками;

• *темпы потоков* – скорости изменения уровней, перемещающие содержимое одного уровня к другому. Примерами могут быть потоки материалов, заказов, денежных средств, рабочей силы, оборудования, информации. Изображаются сплошными стрелками;

• *функции решений (винтили)*, которые регулируют темпы потока между уровнями. Функция решения может иметь форму простого уравнения, определяющего реакцию потока на состояние одного или двух уровней, и обеспечивает заданный темп потока в связи. Изображается двумя треугольниками в виде бабочки;

• *линии задержки (запаздывания)* – служат для имитации задержки потоков. Характеризуются параметрами среднего запаздывания и типом неустановившейся реакции. Второй параметр характеризует отклик элемента на изменение входного сигнала. Разные типы линий задержки имеют различный динамический отклик;

• *каналы информации*, соединяющие функции решений (винтили) с уровнями. Изображаются штриховыми стрелками;

• *вспомогательные переменные* – располагаются в каналах информации между уровнями и функциями решений и определяют некоторую функцию и могут иметь размерность уровней либо

темпов; в процессе имитации вычисляются совместно с группой переменных, которые имеют такую же размерность. Их обозначают кружком большего размера;

• постоянные параметры модели обозначают кружочком малого размера.

На рис. 2.1 приведена модель работы отделения банка. Здесь хорошо представлены резервуары (уровни), темпы потоков, вспомогательные переменные и постоянные параметры.

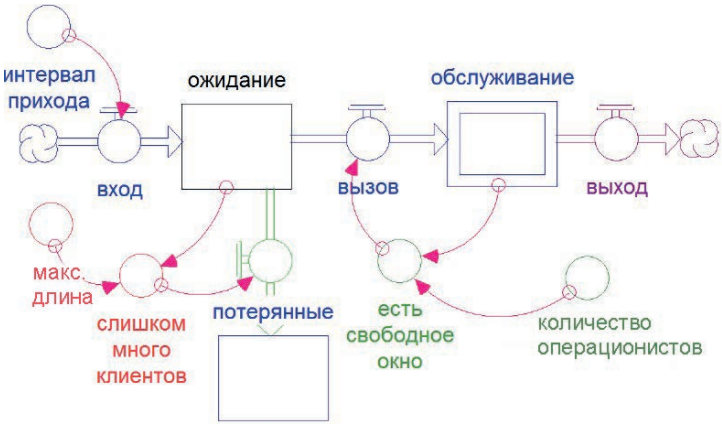



Рис. 2.1. Схема модели системной динамики работы отделения банка [2]

На рис. 2.2 представлена базовая структура модели Форрестера, которая показывает только одну сеть с элементарной схемой информационных связей между уровнями и темпами. Чтобы отразить деятельность всего промышленного предприятия, необходимо построить несколько взаимосвязанных сетей. Для предприятий можно выделить шесть типов сетей, представляющих различные типы переменных: заказы, материалы, денежные средства, рабочую силу и оборудование, соединенных воедино с помощью сети информации. Любая из этих сетей может быть разбита еще на несколько отдельных частей.

Базовая структура модели дополняется системой уравнений, которые связывают характеристики уровней этой структуры. В ос-

новном эта система состоит из уравнений двух типов: уравнения уровней и уравнения темпов. При построении уравнений уровней временная ось разбивается на интервалы времени $\Delta t_{ij}$ между $i$-м и $j$-м моментами времени [2].

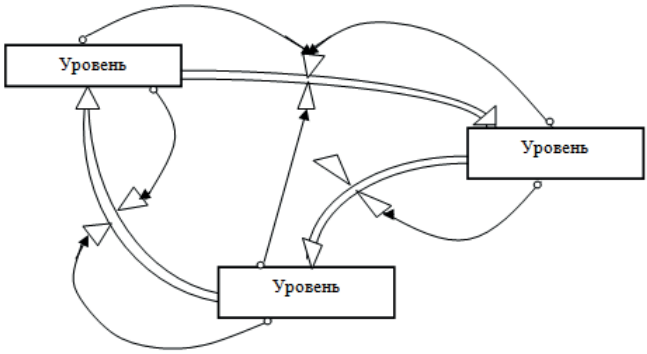



Рис. 2.2. Базовая структура модели Форрестера: ⋈ – уравнения темпов, ⟹ – канал потока, → – источники информации

Новые значения уровней рассчитываются на конец интервала, и по ним определяются новые темпы (решения) для следующего интервала $\Delta\, t_i\, (j + 1)$.

Новый уровень = Текущий уровень + Входящий поток – Исходящий поток

$$z_i = z_{i-1} + \Delta t\, (\Delta x_i - \Delta y_i), \quad i = 1, ..., n. \tag{2.1}$$

В системной динамике такое равенство называется ф*ункциональным уравнением уровня*.

А для вычисления прироста $\Delta x$ и сокращения $\Delta y$ уровней $z$ вводят *функциональные уравнения темпов.* Уравнение темпа на выходе можно записать в виде

$$\Delta y_i / \Delta t = Z_{i-1} / T, \tag{2.2}$$

где $Z_{i-1}$ – величина уровня, отражающего запаздывание в момент времени $i$; $T$ – константа (среднее время), необходимое для преодоления запаздывания.

К числу достоинств модели относятся возможность отражать практически любую причинно-следственную связь; простая математическая форма; использование терминологии, синонимичной языку экономики и производства. Из уравнений уровней (2.1) для основных фазовых переменных (так называемых системных уровней) можно записать дифференциальные уравнения типа

$$\frac{dz}{dt} = y_i^+ - y_i^-, \tag{2.3}$$

где $y_i^+$ – положительный темп скорости переменной $y$, включающий в себя все факторы, вызывающие рост переменной $y$, $i$ – индекс резервуара (уровня); $y_i^-$ – отрицательный темп скорости, включающий в себя все факторы, вызывающие убывание переменной $y$.

Предполагается, что эти темпы расщепляются на произведение функций, зависящих только от «факторов» – комбинаций основных переменных, т. е., в свою очередь, самих являющихся функциями системных уровней:

$$y^{\pm} = g(y_1, y_2, \ldots, y_n) = f(F_1, F_2, \ldots, F_k) = f_1(F_1)\, f_2(F_2) \ldots f_k(F_k), \tag{2.4}$$

где $F_j = g_j(y_{i1} \ldots y_{im})$ – факторы, причем $m = m(j) < n$, $k < n$, $n$ – число уровней, т. е. факторов меньше, чем основных переменных, и каждый фактор зависит не от всех системных уровней, а только от какой-то их части. Это позволяет упростить задачу моделирования.

Этапы построения модели Форрестера [2]:

1) построение базовой структуры модели в виде специализированного графа;

2) параметризация графа и построение соответствующей системы уравнений;

3) описание полученной модели на одном из языков и систем программирования и проведение экспериментов.

Рассмотрим задачу диффузии по Басу. Пусть имеется магазин, в который приходят покупатели (клиенты), и задача состоит в построении модели системной динамики того, как происходит процесс продажи, и за счет чего происходит изменение продажи.

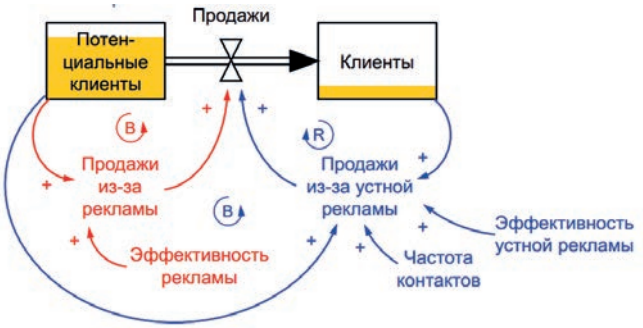



Рис. 2.3. Модель системной динамики

На рис. 2.3 представлена модель системной динамики. Резервуарами являются потенциальные клиенты и клиенты, темпы прироста клиентов могут изменяться за счет рекламы и устной рекламы.

С другой стороны, эта графовая модель представляет собой систему дифференциальных уравнений.

Модель Форрестера – простая, ясная и полезная модель – иллюстрирует интересный графовый подход к моделированию сложных нелинейных систем. Задуманная как учебный пример применения метода системной динамики, она стала образцом для последующих работ, привлекла внимание к проблеме мировой динамики, дала толчок к проведению других исследований, что привело к появлению целого направления, получившего название глобального моделирования. Модели такого рода применяются для прогнозирования экологии, демографии, финансового анализа, экономики и социологии.

Программными средствами, поддерживающими подход системной динамики (в частности, модель Форрестера), являются: AnyLogic, Arena, SimBioSys, eM-Plant, Tecnomatix Plant Simulation, SimuLab, VenSim, PowerSim, Stella, Ithink, ModelMaker и др. Для построения моделей в них используется графическое представление зависимостей переменных в виде так называемых «stock and flow diagrams».

Модели системной динамики хорошо реализуются в среде AnyLogic, так как AnyLogic является инструментом имитационного моделирования нового поколения, который основан на результатах, полученных в теории моделирования и в информационных техноло-

гиях за последнее десятилетие. AnyLogic поддерживает разработку и моделирование систем обратной связи (диаграммы потоков и накопителей, правила решений, включая массивы переменных). Итак, с помощью AnyLogic можно:

- определять потоковые переменные одну за другой;
- использовать автозаполнение при работе с формулами;
- создавать копии переменных для лучшей читаемости модели;
- использовать табличные функции со ступенчатой, линейной, сплайновой интерполяцией;
- определять поведение функции за пределами допустимой области;
- определять поддиапазоны и подразмерности;
- объявлять переменные-массивы с заданной размерностью; задать различные уравнения для различных наборов элементов массива;
- использовать как специальные инструменты системной динамики, так и возможности языка Java.

Для разработки сложных аналитических и имитационных моделей используют среды визуального программирования и моделирования MATLAB, MvStudium, Anylogic, Arena и др. Программный пакет AnyLogic позволяет анализировать и моделировать системы динамического типа. Разные средства анимации, спецификации и анализа результатов, имеющиеся в Anylogic, позволяют строить модели, имитирующие или анимирующие практически любой физический процесс (а также строить и многие другие модели), выполнять анализ моделей на компьютере без проведения реальных экспериментов и самостоятельных сложных вычислений.

Пакет AnyLogic является средой визуального программирования и моделирования, поэтому освоение методов моделирования в этой системе широко доступно непрофессионалам. Эта система может быть освоена также специалистами нематематического профиля.

### 2.1.1. Системный анализ и этапы имитационного моделирования сложных систем

Большинство изучаемых и подлежащих моделированию объектов являются сложными системами. Характерные признаки слож-

ной системы – невозможность рассмотрения отдельно каждого элемента (без установления связей с другими элементами и внешней средой), неопределенность, проявляющаяся в большом числе возможных состояний системы, неопределенность достоверности исходной информации, разнообразие вариантов путей достижения конечной цели функционирования системы, адаптивность (приспосабливаемость системы к возмущающим факторам воздействия внешней среды). Эти особенности вызывают необходимость использования методологии системного анализа при создании имитационной модели сложного объекта. Для анализа сложных объектов и процессов рассматривают системные направления, включающие в себя следующие термины: системный подход, системные исследования, системный анализ.

*Системный подход.* Этот термин начал применяться в первых работах, в которых элементы общей теории систем использовались для практических приложений. Заимствованные при этом понятия теории систем вводились не строго, не исследовался вопрос, каким классом систем лучше отобразить объект, какие свойства и закономерности этого класса следует учитывать при конкретных исследованиях и т. п. Иными словами, термин «системный подход» практически использовался вместо терминов «комплексный подход», «комплексные исследования». Под системным подходом понимается направление методологии научного познания, в основе которого лежит рассмотрение объекта как системы: целостного комплекса взаимосвязанных элементов; совокупности взаимодействующих объектов; совокупности сущностей и отношений (А. Д. Холл, Р. И. Фейджин, поздний Берталанфи).

В настоящее время системный подход рассматривается как начальная фаза современного системного анализа, как стадия (имеющая несколько этапов) первоначального качественного анализа проблемы решения и постановки задач исследования.

*Системные исследования.* В работах под этим названием понятия теории систем используются более конструктивно: определяется класс систем, вводится понятие структуры, а иногда и правила ее формирования и т. п. Это был следующий шаг в системных направлениях. В поисках конструктивных рекомендаций появились

системные направления с разными названиями: системотехника, системология и др. Для их обобщения стал применяться термин «системные исследования». Часто в работах использовался аппарат исследования операций, который к тому времени был больше развит, чем методы конкретных системных исследований.

*Системный анализ.* В настоящее время системный анализ является наиболее конструктивным направлением анализа систем. Здесь предлагается методология проведения исследования, делается попытка выделить этапы исследования и предложить методику выполнения этих этапов в конкретных условиях. Особое внимание при анализе систем уделяется определению целей системы, вопросам формализации представления целей.

Системный анализ в широком смысле – это методология (совокупность методических приемов) постановки и решения задач построения и исследования систем, тесно связанная с математическим моделированием. В более узком смысле системный анализ – методология формализации сложных (трудно формализуемых, плохо структурированных) задач. Системный анализ возник как обобщение приемов, накопленных в задачах исследования операций и управления в технике, экономике, военном деле. Соответствующие модели и методы заимствовались из математической статистики, математического программирования, теории игр, теории массового обслуживания, теории автоматического управления. Системный анализ – это целенаправленная творческая деятельность человека, на основе которой обеспечивается представление исследуемого объекта в виде системы. Системный анализ характеризуется упорядоченным составом методических приемов исследования.

Системный анализ – направление анализа, содержащее методику разделения процессов на этапы и подэтапы, систем на подсистемы, целей на подцели и т. д. В системном анализе выработана определенная последовательность действий (этапов) при постановке и решении задач, которую будем называть алгоритмом (методикой) системного анализа. Эта методика помогает более осмысленно и грамотно ставить и решать прикладные задачи. Если на каком-то этапе возникают затруднения, то нужно вернуться на один из предыдущих этапов и изменить (модифицировать) его. Если и это не помогает, то это значит,

что задача оказалась слишком сложной и ее нужно разбить на несколько более простых подзадач, т. е. провести декомпозицию. Каждую из полученных подзадач решают по той же методике.

Метод системного анализа с успехом применяется к решению самых разных проблем практически в любой области – от проблем корпоративного управления и принятия управленческих решений до моделей в области социологии, экономики, физики, информатики, биологии и т. п. Более того, в последние годы этот метод используется как один из основных подходов к анализу и построению структуры диссертационных исследований в любых отраслях.

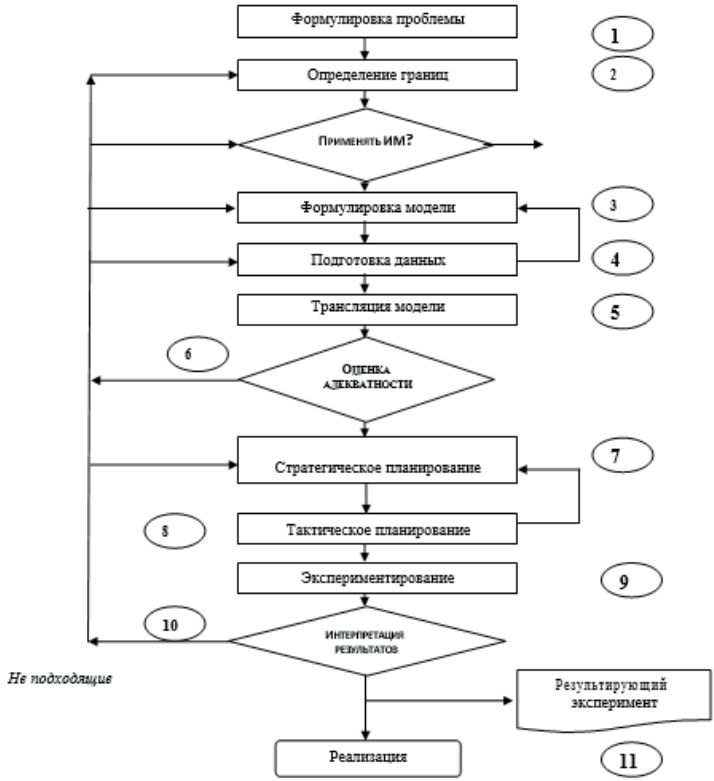



Рис. 2.4. Этапы разработки компьютерной модели сложной системы [2]

Остановимся на этих ранних этапах основного содержания деятельности системного аналитика. Для всех следующих этапов имитационного моделирования эта работа важна. Именно здесь же специалист по имитационному моделированию может проявить себя как системный аналитик, который владеет таким искусством, как моделирование.

На рис. 2.4 представлена схема проведения имитационного исследования сложной системы, предложенная Р. Шенноном.

### 2.1.2. Проектирование и разработка имитационных моделей сложных объектов

Основные стадии и этапы в стадиях для разработки компьютерных моделей сложных систем как разновидности программного обеспечения или информационной системы могут быть кратко представлены следующим образом [2]:

1. Предпроектная стадия – стадия формирования требований к автоматизированной системе:
   - этап разработки концепции автоматизированной системы;
   - этап разработки и утверждения технического задания.
2. Стадия проектирования и разработки программного обеспечения:
   - этап разработки эскизного и технического проекта автоматизированной системы;
   - этап проектирования программного обеспечения;
   - этап проектирования интерфейса;
   - этап реализации программного обеспечения (создание программного кода);
   - этап создания и оформления документации.
3. Стадия внедрения.
4. Период сопровождения или пользовательский период.

Эти стадии создания автоматизированных систем или программного продукта применимы также для разработки компьютерных установок для моделирования.

Современный подход к проектированию информационных систем предполагает создание модели исходной информационной

системы, описывающей все необходимые аспекты её функционирования. Применение моделей позволяет сократить сроки проектирования, улучшить качество проекта за счёт устранения большого числа ошибок в решении стратегических вопросов уже на ранних стадиях работы.

При создании такой модели обычно применяется функциональная методология. Она предполагает рассмотрение системы в виде набора функций, преобразующих входной поток информации в выходной.

**Формулировка проблемы**

Здесь проводится описание исследуемой проблемы и определение целей исследования; постановка задачи, формулировка и установление иерархии целей и подзадач; изучение поведения системы в целом.

Результатом этого этапа должно быть документированное содержательное описание объекта моделирования. Иначе говоря, построение полной информационной модели объекта или системы [2].

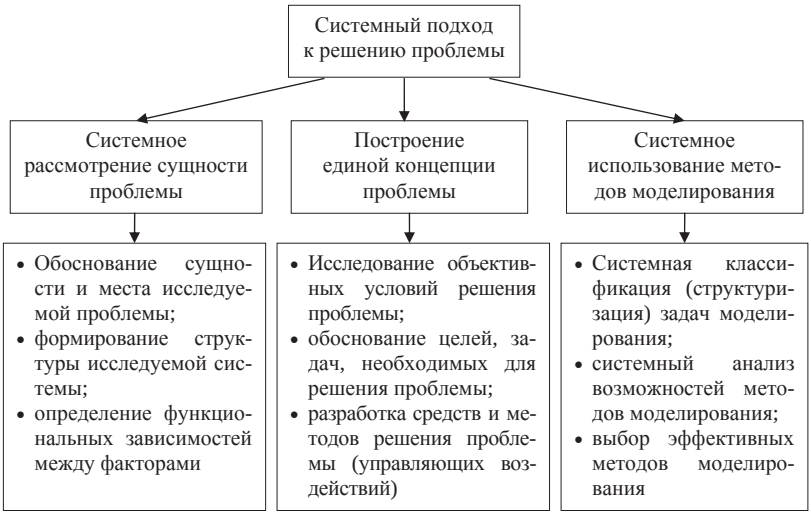



Рис. 2.5. Содержание системного подхода к решению проблем [2]

Анализ проблемы начинается с детального изучения всех аспектов его функционирования. Здесь очень важно понимание деталей, поэтому надо взаимодействовать с экспертами либо быть хорошим специалистом в конкретной предметной области. Данная система весьма тесно связана с другими системами, поэтому важно правильно и четко определить задачи, но при этом задача моделирования разбивается на частные задачи.

Разработаны алгоритмы системного подхода к решению проблемы, они представлены в виде ряда этапов (см. рис. 2.5).

Первый и самый решающий шаг при создании абсолютно любой модели состоит в обосновании ее целевого назначения. Возможно применение метода декомпозиции целей, который предполагает разделение целого на части: задач – на подзадачи, целей – на подцели и т. д. Если использовать такой подход на практике, то он приводит к иерархическим древовидным структурам (т. е. построению дерева целей).

Остановимся на более употребляемых категориях целей в имитационном исследовании: предсказание, сопоставление альтернатив, оценка, оптимизация и др. Эксперименты по моделированию проводятся с разнообразными целями:

- прогноз – критика поведения системы при некоем предполагаемом сочетании рабочих условий;
- сравнение альтернатив – сравнение соперничающих систем, рассчитанных на выполнение конкретной функции либо же на сравнение 1-го и более предлагаемых рабочих принципов либо методик;
- выявление многофункциональных соотношений – определение зависимости между двумя или более действующими факторами, с одной стороны, и откликом этой системы, с другой стороны;
- анализ чувствительности – обнаружение из чуть большего числа работающих факторов тех, которые в значительной степени воздействуют на все поведение системы;
- оценка – определение того, как буквально система предлагаемой структуры станет подходить неким конкретным аспектам;
- оптимизация – конкретное определение сочетания работающих величин и их причин, обеспечивающих наилучший отклик всей системы в целом;

• демонстрация – показ возможностей модели и имитационных исследований системы.

**Определение границ**

Логико-математическое описание моделируемой системы в соответствии с формулировкой проблемы. Определение границ системы и внешней среды, т. е. выделение системы из окружающей среды. Определение входных параметров и выходных характеристик системы [2].

В каждой модели существует некоторая комбинация таких составляющих, как переменные, параметры, компоненты, функциональные зависимости, ограничения, целевые функции (аспекты).

При описании моделируемой системы и процессов определяются основные *параметры и переменные* модели. *Параметрами* являются величины, которые исследователь может выбирать произвольно, в отличие от переменных модели, которые могут принимать значения, определяемые видом заданной функции.

*Компонентами* системы считают составные части, которые образуют систему. Иногда компонентами считают также элементы системы или ее подсистемы. Система определяется как группа или совокупность объектов, объединенных некоторой формой регулярного взаимодействия или взаимозависимости для выполнения заданной функции.

*Функциональные зависимости* описывают поведение переменных и параметров в пределах компоненты или же выражают соотношения между компонентами системы. Эти соотношения по природе являются либо детерминистскими, либо стохастическими.

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

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

**Формулировка и разработка модели**

Включает в себя разработку концептуальной модели и формализацию построенной концептуальной модели. На этой стадии результатом деятельности разработчика компьютерной модели является создание полной концептуальной модели [2].

Концептуальная (содержательная) модель – это абстрактная модель, определяющая структуру моделируемой системы, свойства ее элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования. Построение концептуальной модели включает в себя декомпозицию системы, определение и выделение основных компонент, элементов и подсистем (построение модели состава). Концептуальная модель – это логико-математическое описание смоделированной системы в соответствии с формулировкой проблемы.

Главным содержанием этого шага является переход от настоящей системы к логической схеме ее функционирования, формулировка всеобщего плана модели. В этом шаге приводится алгоритмизация функционирования ее составляющей и отображение объекта в терминах математических понятий.

Итогом работы на данном шаге является избранный метод формализации моделируемой системы, документированное концептуальное отображение. Если формируют малые модели, то этот шаг совмещается с шагом составления содержательного описания моделируемой системы, на котором уточняется способ имитационного опыта. Построение концептуальной модели наступает с определения действия наружной среды, на базе цели моделирования устанавливаются рубежи моделируемой системы, выдвигаются гипотезы и укрепляются все дозволения (предположения), какие нужны для построения имитационной модели, обсуждается степень детализации моделируемых процессов.

Найти систему можно как совокупность взаимосвязанных частей. Дефиниция системы зависит от того, кто определяет систему, и от цели моделирования. На данном этапе выполняется декомпозиция системы, определяются более значительные взаимодействия между частями, составляющими систему (выполняется структурный анализ моделируемой системы), выявляются главные аспекты функциониро-

вания моделируемой системы (составляется многофункциональная модель), приводится изображение внешней среды. Выделение подсистем либо декомпозиция системы (объекта моделирования) – это процедура анализа. Составляющие таковой модели должны быть реально существующим фрагментом в системе, а сложная система разбивается на части, и охраняет при этом связи, обеспечивающие взаимодействие. Разрешено составить многофункциональную схему, проясняющая специфику динамических процессов, которые проистекают в рассматриваемой системе, и принципиально найти, какие будут вынесены во внешнюю обстановку, какие элементы будут введены в модель и какие взаимосвязи будут установлены между ними.

Итак, вначале находится «элементарность» – составляется наиболее простое дерево целей, строится упрощенная конструкция модели. Далее проводится постепенная детализация модели. Используется метод «от простого к сложному». Сначала строится простая модель, затем она усложняется. Здесь применяется принцип итеративного построения модели (иерархический метод), когда по мере исследования системы по модели, в ходе ее разработки, она меняется в результате добавления новых либо исключения некоторых её элементов и/или взаимосвязей между ними.

Как перейти от реальной системы к её упрощенному (приближенному) описанию? Упрощение, приближение – основной прием любого моделирования. Избранный уровень детализации должен позволять абстрагироваться от неточно определенных параметров из-за аспектов функционирования реальной системы вследствие недочета информации.

Под упрощением (приближением) понимается пренебрежение несущественными деталями либо принятие догадок о наиболее простых соотношениях (к примеру, предположение о линейной зависимости между переменными). При моделировании выдвигаются гипотезы, догадки, относящиеся к взаимосвязи между компонентами и переменными системы.

Иным аспектом разбора реальной системы является абстракция, она содержит в себе существенные качества поведения объекта, однако не обязательно в такой же форме и столь подробно, как в реальной системе.

После анализа элементов и подсистем приступаем к их объединению в единое целое. В концептуальной модели должно быть корректно отражено их взаимодействие. Композиция есть операция синтеза, агрегирование (при системном моделировании это не всегда сводится к элементарной сборке компонентов). В ходе операции выполняется введение отношений между элементами (к примеру, уточняется конструкция, приводится описание отношений, упорядочение и др.).

Таким образом, системное исследование построено на сочетании операций анализа и синтеза: проведение анализа взаимосвязей подсистем, установление связей между компонентами, подсистемами, элементами (построение модели структуры связей); принятие гипотез и допущений, физическая схематизация, иначе говоря, построение общей структуры системы с учетом всех подсистем и связей.

Для разработки концептуальной модели часто используются информационные системы проектирования BPwin, Erwin, Rational Rose, CASE Аналитик, ARIS Toolset и др., в которых можно провести контекстную и функциональную декомпозицию системы, потоков данных, управляющих потоков, определить структуру данных, построить диаграммы «сущность – связь».

Формализация построенной концептуальной модели осуществляется с помощью языка или аппарата математических методов, в том числе и имитационных технологий. В зависимости от сложности объекта и цели моделирования выбирается один из подходов аналитического или имитационного моделирования (статистический подход, динамическая система, дискретно-событийное, мультиагентное моделирование, системная динамика, когнитивное, SIE-моделирование). В рамках выбранного подхода проводится разработка математического описания объекта моделирования. Результатом этого этапа является разработка технического проекта компьютерной установки для моделирования.

Процесс формализации сложной системы включает следующие виды работ:

- выбор способа формализации;
- составление формального описания системы.

В процессе построения модели можно выделить три уровня ее представления:

- неформализованный (этап 2) – концептуальная модель;
- формализованный (этап 3) – формальная модель;
- программный (этап 4) – имитационная модель.

Каждый уровень отличается от предыдущего степенью детализации моделируемой системы и способами описания ее структуры и процесса функционирования. При этом уровень абстрагирования возрастает.

Концептуальная модель – это систематизированное содержательное описание моделируемой системы (или проблемной ситуации) на неформальном языке. Неформализованное описание разрабатываемой имитационной модели включает определение основных элементов моделируемой системы, их характеристики и взаимодействие между элементами на собственном языке. При этом могут использоваться таблицы, графики, диаграммы и т. д. Неформализованное описание модели необходимо как самим разработчикам (при модификации, проверке адекватности модели и т. д.), так и для взаимопонимания со специалистами других профилей. Концептуальная модель содержит исходную информацию для системного аналитика, выполняющего формализацию системы и использующего для этого определенную методологию и технологию. На основе формализованного описания осуществляется разработка более строгой и подробной системы [2].

Формализация объекта исследования осуществляется на базе той методологии имитационного моделирования, которая подходит к данной системе. Наблюдается множество схем (концепций) формализации и структуризации, которые нашли применение в имитационном моделировании. Такие таблицы формализации исходят из различных понятий об изучаемых процессах и ориентируются на разные математические теории. Отсюда множество схем формализации и трудности отбора подходящей для описаний данного предмета моделирования. В настоящее время существуют разработанные подходы и методологии формализации имитационного моделирования: методы статистического моделирования, моделирования динамических систем, системной динамики, дискретно-событийного моделирования и др.

Имитационная модель – преобразование формализованного описания в программу-имитатор, построенную в соответствии с некоторой методикой в средах программирования или моделирования. Аналогичная схема имеет место и при выполнении имитационных экспериментов: содержательная постановка отображается на формальную модель, после чего вносятся необходимые изменения и дополнения в методику направленного вычислительного эксперимента. Основная задача этапа формализации – дать формальное описание сложной системы, свободное от второстепенной информации, имеющейся в содержательном описании, а также алгоритмическое представление объекта моделирования.

Цель формализации при аналитическом моделировании – построить аналитическую модель в виде каких-либо уравнений (линейных, дифференциальных, интегро-дифференциальных и др.).

Цель формализации при имитационном моделировании – получить формальное представление логико-математической модели, т. е. алгоритмов поведения компонентов сложной системы, и отразить на уровне моделирующего алгоритма взаимодействие между компонентами.

Таким образом, концептуальное или формальное описание модели сложной системы на уровне формализации – это построение имитационной компьютерной модели «программы-имитатора» в соответствии с некоторой методикой программирования с применением языков и систем автоматизации моделирования. Выбор инструментального средства для построения компьютерной модели является основным моментом в имитационном исследовании сложной системы.

**Подготовка данных**

Включает идентификацию, спецификацию и сбор данных. Идентификация – статистический анализ модели, статистическое оценивание неизвестных параметров.

Спецификация – определение конечных целей моделирования; характеристика набора экзогенных и эндогенных переменных; состава системы уравнений, их структур; формулировка исходных предпосылок, ограничений. Спецификация опирается на имеющиеся экономические теории, специальные знания, интуицию исследователя.

В подготовку данных входит сбор и анализ исходных данных для моделирования. Если трассировку и программирование имитационной модели можно выполнять на гипотетических данных, то предстоящее экспериментальное исследование нужно выполнять на реальном потоке данных, так как от этого зависит адекватность модели реальной системе и точность получаемых результатов моделирования. Здесь перед разработчиком имитационной модели встают два вопроса: где и каким образом получить и собрать исходную информацию и как обработать собранные данные о жизненной системе [2].

Основные способы получения исходных данных:

- из экспериментальных данных (физический эксперимент);
- из документации на систему (финансовая и техническая документация для промышленных систем, данные отчетов, статистические сборники, к примеру для социально-экономических систем, и др.);
- из литературных источников по рассматриваемой системе.

Бывает так, что для задания начальной информации нужно вести предварительный, априорный синтез данных, натурные опыты на моделируемой системе либо на ее прототипах, а время от времени исходные данные имеют все шансы не быть, а моделируемая система исключает вероятность физического эксперимента, и тогда дают различные приемы предварительного объединения данных. При моделировании информационных систем длительность выполнения информационного требования оценивается на базе трудоемкости реализуемых на ЭВМ алгоритмов. К таким способам относят различные процедуры, построенные на общем обзоре проблематики, анкетировании, интервьюировании, большом применении методов экспертного оценивания [2].

Второй вопрос связан с проблемами идентификации входных данных для стохастических систем. Имитационное моделирование является весьма эффективным аппаратом исследования стохастических систем, т. е. таких систем, динамика которых зависит от случайных факторов: входные (либо выходные) переменные стохастической модели, случайные величины, векторы, функции, случайные процессы. Именно поэтому появляются лишние трудности, связанные с синтезом уравнений относительно неизвестных законов рас-

пределения и определением вероятностных характеристик (математических ожиданий, дисперсий, корреляционных функций и т. п.) для анализируемых процессов и их параметров. Необходимость статистического анализа при характеризации и сборе входных данных связана с целью определения вида функциональных зависимостей, описывающей входные данные, оценкой конкретизированных значений параметров этих зависимостей, а также проверкой значимости параметров. Для подбора теоретических распределений случайных величин используют известные приемы математической статистики, основанные на определении параметров эмпирических распределений и проверке статистических гипотез, с применением критериев согласия, сообразуются ли экспериментальные данные с известными законами распределения [2].

**Трансляция модели**

Трансляция модели – это перевод модели со специальных имитационных языков или языка математики на язык программирования, на котором будет реализована прикладная программа, соответствующая компьютерной модели. Алгоритмизация и программная реализация – это построение программного комплекса моделирования объекта исследования и проведение отладки компьютерной модели [2].

**Оценка адекватности (верификация и валидация)**

После реализации имитационной модели на ЭВМ необходимо проводить испытания для оценки достоверности модели. В периоде испытания и исследования разработанной имитационной модели организуется ее комплексное тестирование (*testing*) – планируемый итеративный процесс, направленный на поддержку операций верификации и валидации имитационных моделей и данных.

Если в случае проведенных процедур модель окажется недостаточно подлинной, то может быть осуществлена калибровка имитационной модели (в моделирующий алгоритм встраиваются калибровочные коэффициенты) с целью снабжения адекватности модели. В более трудных случаях достижимы большие итерации на ранние периоды с целью снятия дополнительной информации о моделируемом объекте или доработки имитационной модели.

Наличие ошибок во взаимодействии компонентов модели возвращает исследователя на этап создания имитационной модели.

Причиной этого может быть с самого начала упрощенная модель процесса или явления, что приводит к неадекватности модели объекту. В случае, если выбор способа формализации оказался весьма неудачным, то должно повторить этап создания концептуальной модели с учетом свежей информации и приобретенного опыта. Наконец, если оказалось, что информации об объекте мало, то нужно вернуться к периоду составления полного описания системы и конкретизировать его с учетом итогов испытания [2].

**Стратегическое и тактическое планирование**

На этом этапе проводится стратегическое и тактическое планирование машинного эксперимента. Результатом является составленный план эксперимента и проведенный вычислительный эксперимент («прогоны» компьютерной модели с различными начальными данными). Здесь проводится определение условий машинного эксперимента с имитационной моделью, а также параметров при тестировании модели, результатов по входным данным [2].

**Постановка экспериментов**

На данном этапе предполагается прогон программы имитационной модели на ЭВМ для получения выходных данных или результатов, позволяющих оценить адекватность построенной модели. Здесь также необходимо определить условия, в которых будут осуществляться тестирование, проверка работоспособности и возможности функционирования; параметры, на которые надо обратить внимание при тестировании модели. Параметры могут быть связаны со способностью модели реагировать на какие-либо стохастические воздействия, на неверные входные данные либо полное их отсутствие, на неправильные действия персонала.

Далее проводится вычислительный эксперимент на имитационной модели. На последних стадиях имитационного моделирования необходимо вести стратегическое и тактическое планирование имитационного эксперимента. Организация направленного вычислительного эксперимента на имитационной модели определяет выбор и использование различных аналитических приемов для обработки итогов имитационного исследования. Ради этого применяются способы планирования вычислительного эксперимента, статистический, регрессионный и дисперсионный анализ, методы оптимиза-

ции. Организация и проведение опыта требует корректного использования возможных аналитических приемов. Согласно полученным результатам обследование должно позволить сделать выводы, достаточные для принятия решений по обозначенным на ранних стадиях проблемам и задачам [2].

**Анализ результатов моделирования**

Обработка, визуализация и интерпретация результатов машинного компьютерного эксперимента предполагает рассмотрение и изучение результатов имитационного эксперимента для подготовки выводов о возможности применения имитационной модели для решения некоторой проблемы [2].

**Реализация и документирование**

На основе построенной имитационной модели можно дать рекомендации о принятии того или иного управленческого решения и документально отразить процесс функционирования модели и полученные результаты; провести анализ основных проблем, возникающих при создании модели сложной системы.

Более подробно перечнем исследований на каждом этапе проектирования имитационных моделей сложных систем можно ознакомиться в работах авторов [2].

## Контрольные вопросы и задания

1. Что мы понимаем под объектом и как могут формироваться классы?
2. Что такое система, каковы ее признаки и сущность?
3. Чем отличается сложная система от большой системы?
4. Что такое связь и структура системы?
5. Что такое процесс и каковы его основные характеристики?
6. Дайте определение события и его характеристик. Что такое транзакт?
7. Какие виды имитационного эксперимента и применения компьютерных моделей Вы знаете?
8. Какие разновидности имитационного моделирования существуют на сегодняшний день?

9. Какие инструментальные средства используются для построения имитационных моделей?

10. Как соотносятся понятия «системный подход» и «системный анализ»?

11. Каковы этапы имитационного моделирования?

12. Какие среды проектирования информационных систем применяются для проектирования имитационных моделей?

13. Какие основные направления построения и использования имитационных моделей существуют на сегодняшний день?

14. Перечислите области применения имитационного моделирования.

# Глава 3

# СРЕДА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ANYLOGIC 7

## 3.1. Общие сведения о системе имитационного моделирования AnyLogic

В начале 1990-х годов в компьютерной науке наблюдался большой интерес к построению математически трактуемого описания взаимодействия параллельных процессов. Небольшая группа учёных из Санкт-Петербургского политехнического университета занималась разработкой пакета программ для анализа корректности параллельных и распределённых систем. Такой программный продукт был создан, новый инструмент был назван COVERS (параллельная верификация и моделирование). В этом пакете анализируемая система процессов задавалась графически, с помощью описания её структуры и поведения отдельных параллельных компонентов, которые могли взаимодействовать с окружением – с другими процессами и средой [2].

В 1998 году успех этого исследования вдохновил лабораторию организовать коммерческую компанию с миссией создания нового программного обеспечения для имитационного моделирования. Акцент при разработке ставился на прикладные методы: моделирование стохастических систем, оптимизацию и визуализацию модели. Новое программное обеспечение, выпущенное в 2000 году, было основано на последних преимуществах информационных технологий: объектно-ориентированный подход, элементы стандарта UML, языка программирования Java, современного GUI и т. д. [6].

Продукт получил название AnyLogic, потому что он поддерживал три известных направления имитационного моделирования (рис. 3.1), а также любую комбинацию этих подходов в пределах одной модели.

Первой версии был присвоен индекс 4 – Anylogic 4.0, так как нумерация продолжила историю версий предыдущей разработки – COVERS 3.0:

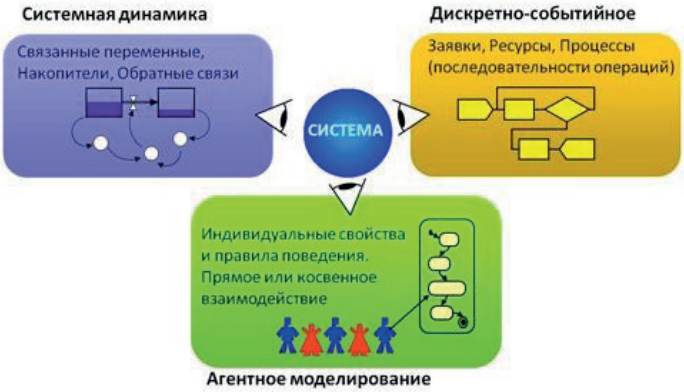



Рис. 3.1. Три подхода имитационного моделирования [6]

- системная динамика;
- дискретно-событийное (процессное) моделирование;
- агентное моделирование.

Шаг вперёд был сделан в 2003 году, когда был выпущен AnyLogic 5, ориентированный на бизнес-моделирование. С помощью AnyLogic стало возможным разрабатывать модели в следующих областях:

- производство;
- логистика и цепочки поставок;
- рынок и конкуренция;
- бизнес-процессы и сфера обслуживания;
- здравоохранение и фармацевтика;
- управление активами и проектами;
- телекоммуникации и информационные системы;
- социальные и экологические системы;
- пешеходная динамика;
- оборона.

Версия AnyLogic 6 написана на языке программирования Java в среде разработки Eclipse и является кросс-платформенным программным обеспечением, работает как под управлением операционной системы Windows, так и под Mac, OS и Linux.

AnyLogic включает в себя графический язык моделирования, а также позволяет пользователю расширять созданные модели с помощью языка Java. Интеграция компилятора Java в AnyLogic предоставляет более широкие возможности при создании моделей, а также Java апплетов, которые могут быть открыты любым браузером. Эти апплеты позволяют размещать модели AnyLogic на веб-сайтах. В дополнение к Java апплетам AnyLogicProfessional поддерживает создание Java приложений, в этом случае пользователь может запустить модель без инсталляции AnyLogic [6].

Система AnyLogic, разработанная компанией XJTechnologies (Россия), это среда компьютерного моделирования общего назначения. Это комплексный инструмент, охватывающий основные в настоящее время направления моделирования: дискретно-событийное, системной динамики, агентное. Использование AnyLogic дает возможность оценить эффект конструкторских решений в сложных системах реального мира.

Отечественный профессиональный инструмент имитационного моделирования AnyLogic нового поколения разработан на основе современных концепций в области информационных технологий и результатов исследований в теории гибридных систем и объектно-ориентированного моделирования. Построенная на их основе инструментальная система AnyLogic не ограничивает пользователя одной единственной парадигмой моделирования, что является характерным для существующих на рынке инструментов моделирования. В AnyLogic разработчик может гибко использовать различные уровни абстрагирования и различные стили и концепции и смешивать их при создании одной и той же модели [2].

Программный продукт AnyLogic основан на объектно-ориентированной концепции. Объектно-ориентированный подход к представлению сложных систем является лучшим на сегодняшний день методом управления сложностью информации, эта концепция позволяет простым и естественным образом организовать и представить структуру сложной системы. Таким образом, идеи и методы, рассчитанные на управление сложностью, выработанные в последние десятилетия в области создания программных систем, позволяют разработчикам моделей в среде AnyLogic организовать мышление,

структурировать разработку и, в конечном счете, упростить и ускорить создание моделей.

Другой базовой концепцией AnyLogic является представление модели как набора взаимодействующих параллельно функционирующих активностей. Такой подход к моделированию интуитивно очень понятен и естественен во многих приложениях, поскольку системы реальной жизни состоят из совокупности активностей, взаимодействующих с другими объектами. Активный объект AnyLogic – это объект со своим собственным функционированием, взаимодействующий с окружением. Он может включать в себя любое количество экземпляров других активных объектов. Активные объекты могут динамически порождаться и исчезать в соответствии с законами функционирования системы. Так могут моделироваться социальные группы, холдинги компаний, транспортные системы и т. п.

Графическая среда моделирования AnyLogic поддерживает проектирование, разработку, документирование модели, выполнение компьютерных экспериментов с моделью, включая различные виды анализа – от анализа чувствительности до оптимизации параметров модели относительно некоторого критерия.

В результате AnyLogic не ограничивает пользователя одной единственной парадигмой моделирования, что является характерным фактически для всех инструментов моделирования, существующих сегодня на рынке. В AnyLogic разработчик может использовать различные уровни абстрагирования, различные стили и концепции, строить модели в рамках той или иной парадигмы и смешивать их при создании одной и той же модели, использовать ранее разработанные модули, собранные в библиотеки, дополнять и строить свои собственные библиотеки модулей. При разработке модели на AnyLogic можно использовать концепции и средства из нескольких «классических» областей моделирования, например в агентной модели использовать методы системной динамики для представления изменений состояния среды или в непрерывной модели динамической системы учесть дискретные события. В AnyLogic легко строятся подобные модели с требуемым уровнем адекватности, позволяющие ответить на многие вопросы, интересующие исследователя. Богатые возможности анимации и визуального представления результатов в процес-

се работы модели позволяют понять суть процессов, происходящих в моделируемой системе, упростить отладку модели [6].

Многие крупные компании, как отечественные Билайн, Газпром, Сбербанк России, Русский алюминий, Северсталь и т. д., так и зарубежные General Motors, Mitsubishi, McDonalds, IBM и др. являются клиентами компании и используют AnyLogic для своих потребностей и исследований (рис. 3.2, 3.3, 3.4).

Рис. 3.2. Клиенты компании AnyLogic [6]

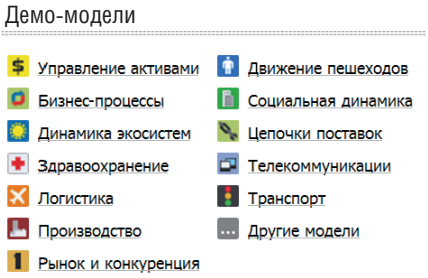
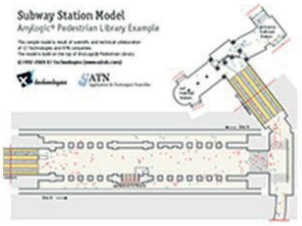



Рис. 3.3. Отрасли и сферы использования программного продукта [6]

## Движение пешеходов

Subway Station

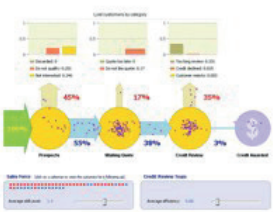



Subway Entrance Hall

## Бизнес-процессы

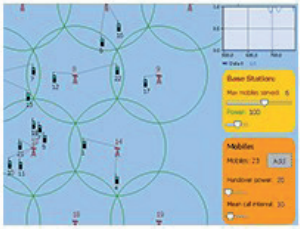
Airport Terminal

Sales Funnel



## Телекоммуникации

Cellular Phones



Leader Election

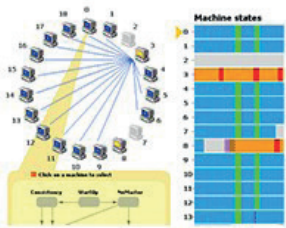



Рис. 3.4. Области применения имитационного моделирования [6]

Графическая среда AnyLogic 5 построена по тому же принципу, что и в Rockwell Arena. Моделирующие конструкции располагаются в палитрах (аналог шаблонов в Arena). Для создания модели, как и в Arena, моделирующие конструкции перетаскивают в область модели и соединяют.

На сегодняшний день актуальной версией программы является AnyLogic 7.3.6. Принциальные изменения произошли в значительном раширении линейки Библиотек [6].

## 3.2. Средства AnyLogic для имитационного моделирования систем

### 3.2.1. Основные концепции

***Две фазы моделирования***. AnyLogic используется для разработки имитационных исполняемых моделей и последующего их прогона для анализа. Разработка модели выполняется в графическом редакторе AnyLogic с использованием многочисленных средств поддержки, упрощающих работу. Построенная модель затем компилируется встроенным компилятором AnyLogic и запускается на выполнение. В процессе выполнения модели пользователь может наблюдать ее поведение, изменять параметры модели, выводить результаты моделирования в различных формах и выполнять разного рода компьютерные эксперименты с моделью [4].

Для реализации специальных вычислений и описания логики поведения объектов AnyLogic позволяет использовать мощный современный язык Java.

***Активные объекты, классы и экземпляры активных объектов***. Основными строительными блоками модели AnyLogic являются *активные объекты*, которые позволяют моделировать любые объекты реального мира.

Класс в программировании является мощным средством, позволяющим структурировать сложную систему. Класс определяет шаблон, в соответствии с которым строятся его отдельные экзем-

пляры. Эти экземпляры могут быть определены как объекты других активных объектов.

Активный объект является *экземпляром класса* активного объекта. Чтобы создать модель AnyLogic, Вы должны создать классы активных объектов (или использовать объекты библиотек AnyLogic) и задать их взаимосвязи. AnyLogic интерпретирует создаваемые Вами графически классы активных объектов в классы *Java*, поэтому Вы можете пользоваться всеми преимуществами объектно-ориентированного моделирования.

Активные объекты могут содержать вложенные объекты, причем уровень вложенности не ограничен. Это позволяет производить декомпозицию модели на любое количество уровней детализации.

Активные объекты имеют четко определенные интерфейсы взаимодействия – они взаимодействуют со своим окружением только посредством своих интерфейсных элементов [4].

Это облегчает создание систем со сложной структурой, а также делает активные объекты повторно используемыми. Создав класс активного объекта, Вы можете создать любое количество объектов – экземпляров данного класса.

Каждый активный объект имеет структуру (совокупность включенных в него активных объектов и их связи), а также поведение, определяемое совокупностью переменных, параметров, стейтчартов и т. п. Каждый экземпляр активного объекта в работающей модели имеет свое собственное поведение, он может иметь свои значения параметров, функционирует независимо от других объектов, взаимодействуя с ними и с внешней средой.

***Визуальная разработка модели***. При построении модели используются средства визуальной разработки (введения состояний и переходов стейтчарта, введения пиктограмм переменных и т. п.), задания численных значений параметров, аналитических записей соотношений переменных и аналитических записей условий наступления событий. Основной технологией программирования в AnyLogic является визуальное программирование – построение с помощью графических объектов и пиктограмм иерархий структуры и поведения активных объектов.

***Встроенный язык Java***. AnyLogic является надстройкой над языком *Java* – одним из самых мощных и в то же время самых простых современных объектно-ориентированных языков. Все объекты, определенные пользователем при разработке модели с помощью графического редактора, компилируются в конструкции языка *Java*, а затем происходит компиляция всей собранной программы на *Java*, задающей модель, в исполняемый код. Хотя программирование сведено к минимуму, разработчику модели необходимо иметь некоторое представление об этом языке (например, знать синтаксически правильные конструкции).

***Средства описания поведения объектов***. Основными средствами описания поведения объектов являются переменные, события и диаграммы состояний. *Переменные* отражают изменяющиеся характеристики объекта. *События* могут наступать с заданным интервалом времени и выполнять заданное действие. *Диаграммы состояний (или стейтчарты)* позволяют визуально представить поведение объекта во времени под воздействием событий или условий, они состоят из графического изображения состояний и переходов между ними (т. е. по сути это конечный автомат). Любая сложная логика поведения объектов модели может быть выражена с помощью комбинации стейтчартов, дифференциальных и алгебраических уравнений, переменных, таймеров и программного кода на *Java*. Алгебраические и дифференциальные уравнения записываются аналитически.

Интерпретация любого числа параллельно протекающих процессов в модели AnyLogic скрыта от пользователя. Никаких усилий разработчика модели для организации *квазипараллелизма* интерпретации не требуется; отслеживание всех событий выполняется системой автоматически.

***Модельное и реальное время***. Понятие модельного времени является базовым в системах имитационного моделирования. *Модельное время* – это условное логическое время, в единицах которого определено поведение всех объектов модели. В моделях AnyLogic модельное время может изменяться либо непрерывно, если поведение объектов описывается дифференциальными уравнениями, либо дискретно, переключаясь от момента наступления одного события

к моменту наступления следующего события, если в модели присутствуют только дискретные события. Моменты наступления всех планируемых событий в дискретной модели исполнительная система хранит в так называемом календаре событий, выбирая оттуда наиболее раннее событие для выполнения связанных с ним действий. Значение текущего времени в моделях AnyLogic может быть получено с помощью функции *time* ().

Единицу модельного времени разработчик модели может интерпретировать как любой отрезок времени: секунду, минуту, час или год. Важно только, чтобы все процессы, зависящие от времени, были выражены в одних и тех же единицах. При моделировании физических процессов все параметры и уравнения должны быть выражены в одних и тех же единицах измерения физических величин.

Интерпретация модели выполняется на компьютере. Физическое время, затрачиваемое процессором на имитацию действий, которые должны выполняться в модели в течение одной единицы модельного времени, зависит от многих факторов. Поэтому единица физического времени и единица модельного времени не совпадают.

В AnyLogic приняты два режима выполнения моделей: режим виртуального времени и режим реального времени. В *режиме виртуального времени* процессор работает с максимальной скоростью без привязки к физическому времени. Данный режим используется для факторного анализа модели, набора статистики, оптимизации параметров модели и т. д. Поскольку анимация и другие окна наблюдения за поведением модели обычно существенно замедляют скорость интерпретации модели на компьютере, для повышения скорости выполнения эти окна нужно закрыть.

В *режиме реального времени* пользователь задает связь модельного времени с физическим временем, т. е. устанавливает ограничение на скорость процессора при интерпретации модели. В этом режиме задается количество единиц модельного времени, которые должны интерпретироваться процессором в *одну секунду*. Обычно данный режим включается для того, чтобы визуально представить функционирование системы в реальном темпе наступления событий, проникнуть в суть процессов, происходящих в модели.

Соотношение физического и модельного времени при работе модели можно понять на таком примере. При коэффициенте ускорения 4, если процессор успевает выполнить менее чем за 1 с все операции, которые в модели определены в течение четырех единиц модельного времени, то он будет ждать до конца секунды. Если же процессор не успевает выполнить все операции, то у него не будет интервала ожидания, и коэффициент ускорения будет меньше того, который установлен пользователем.

***Анимация поведения модели***. AnyLogic имеет удобные средства представления функционирования моделируемой системы в живой форме динамической анимации, что позволяет «увидеть» поведение сложной системы. Визуализация процесса функционирования моделируемой системы дает возможность проверить адекватность модели, выявить ошибки при задании логики.

Средства анимации позволяют пользователю легко создавать виртуальный мир (совокупность графических образов, ожившую мнемосхему), управляемый динамическими параметрами модели по законам, определенным пользователем с помощью уравнений и логики моделируемых объектов. Графические элементы, добавленные на анимацию, называются динамическими, поскольку все их параметры: видимость, цвет и т. п. – можно сделать зависимыми от переменных и параметров модели, которые меняются со временем при выполнении модели.

С помощью совершенной технологии визуализации работающих моделей AnyLogic можно создавать интерактивные анимации произвольной сложности, связывая графические объекты (в том числе импортированные чертежи) во встроенном редакторе с объектами модели. Как и модель, анимация имеет иерархическую структуру, которая может динамически изменяться. Возможно создание нескольких точек зрения или нескольких уровней детальности в пределах одной анимации. Элементы управления и развитая бизнес-графика превращают анимацию модели в настоящую панель управления для оценки эффективности решений. В AnyLogic поддерживается как двумерная, так и трёхмерная анимация.

***Интерактивный анализ модели***. Многие системы моделирования позволяют менять параметры модели только до запуска моде-

ли на выполнение. AnyLogic позволяет пользователю вмешиваться в работу модели, изменяя параметры модели в процессе ее функционирования. Примером таких средств являются *слайдеры*, которые могут быть введены в окно анимации [4].

### 3.2.2. Базовые инструменты для разработки модели в среде AnyLogic

Окно редактора AnyLogic имеет следующие элементы (рис. 3.5) [2]:

- панель инструментов;
- панель «Проекты» (дерево всех объектов проекта);
- панель «Палитра»;
- панель «Ошибки»;
- окно свойств;
- окно графического редактора.

***Панели инструментов*** обеспечивают быстрый доступ к большинству функций, автоматически подстраиваются под текущий режим, а также могут быть адаптированы пользователем. В AnyLogic существуют следующие панели инструментов:

- стандартная («Создать», «Открыть модель», «Сохранить модель», «Сохранить все модели»)

;

- построение («Построить модель», «Построить все», «Отладка», «Запуск»)

;

- редактирование («Отменить», «Повторить», «Вырезать», «Скопировать», «Вставить», «Удалить»)

;

- рисование (100 %, «Отдалить», «Масштаб», «Приблизить», «Отобразить/скрыть сетку»)

.

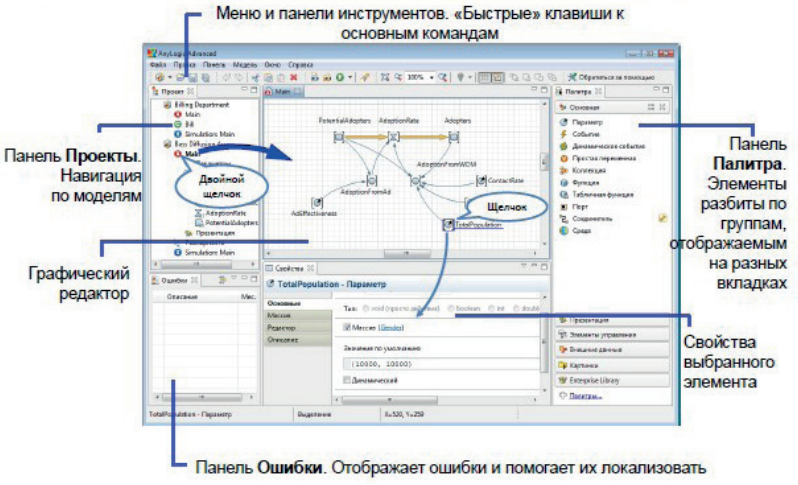



Рис. 3.5. Окно редактора AnyLogic [2]

**Панель «Проекты».** При открытии проекта (нового или существующего) AnyLogic всегда открывает среду разработки проекта – графический редактор модели. Рассмотрим основные составляющие этого редактора (рис. 3.6).

***Окно проекта*** обеспечивает легкую навигацию по элементам проекта, таким как пакеты, классы и т. д. Поскольку проект организован иерархически, то он отображается в виде дерева: сам проект образует верхний уровень дерева рабочего проекта, пакеты – следующий уровень, классы активных объектов и сообщений – следующий и т. д. Можно копировать, перемещать и удалять любые элементы дерева объектов, легко управляя рабочим проектом (рис. 3.7).

Одна из ветвей в дереве проекта имеет название *Simulation: Main* (эксперимент). Эксперимент хранит набор настроек, с помощью которых конфигурируют работу модели. Один эксперимент создается автоматически при создании проекта. Это *простой эксперимент* с именем *Simulation*, позволяющий визуализировать модель с помощью анимации и поддерживающий инструменты для отладки модели. Простой эксперимент используется в большинстве случаев. Поддерживается ещё несколько типов экспериментов для различных задач моделирования.

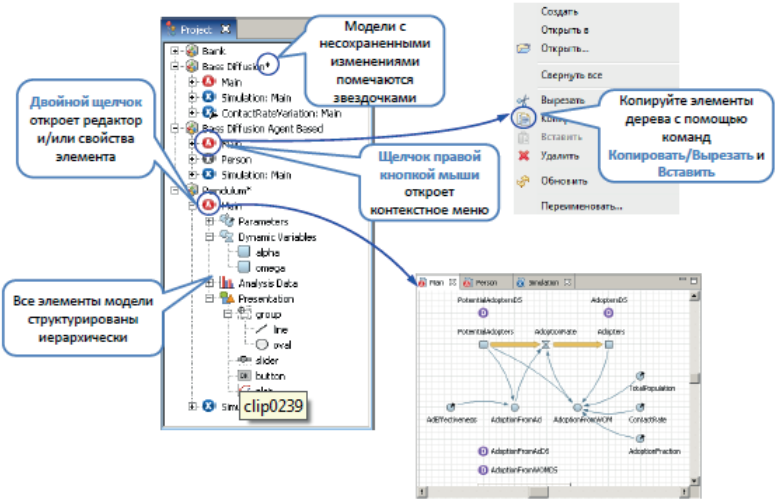



Рис. 3.6. Панель «Проекты» [2]

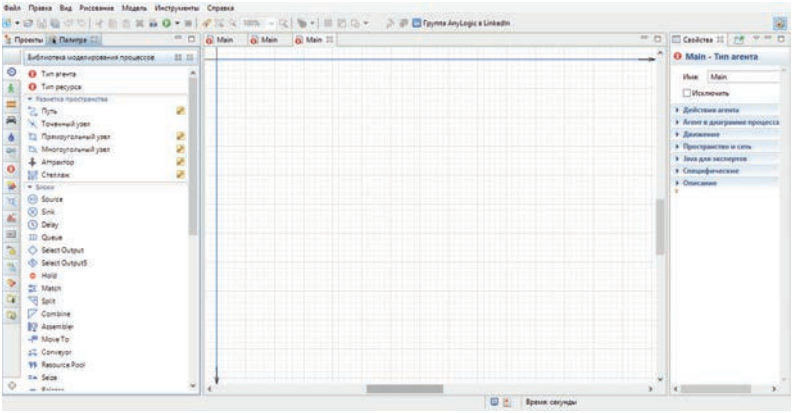


Рис. 3.7. Окно проекта [2]

**Панель «Палитра».** Окно палитры содержит элементы (графические объекты), которые могут быть добавлены на структурную диаграмму. Элементы разбиты по группам, отображаемым на разных вкладках. Чтобы добавить объект палитры на диаграмму, щелкните сначала по элементу в палитре, а затем по диаграмме на рис. 3.8.

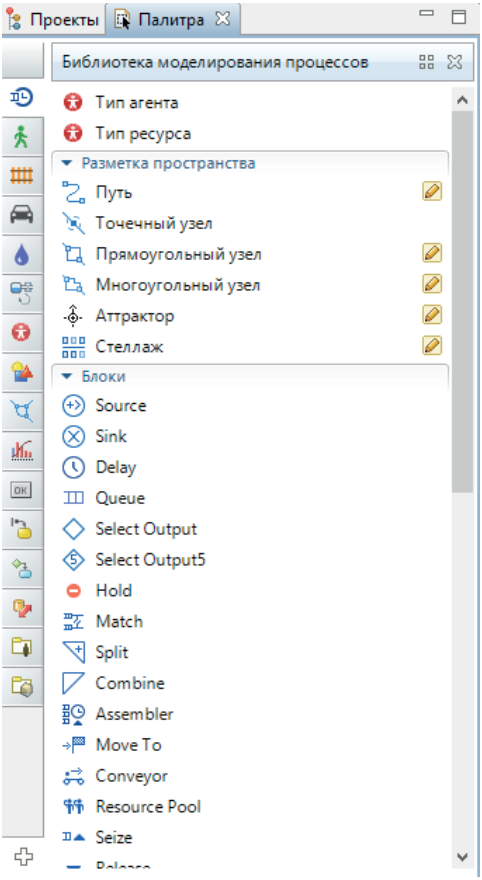



Рис. 3.8. Панель «Палитра»

Состоит из нескольких вкладок (палитр), каждая из которых содержит элементы, относящиеся к определенной задаче:

- основная содержит базовые элементы, с помощью которых Вы можете задать динамику модели, ее структуру и данные;
- системная динамика содержит элементы диаграммы потоков и накопителей, а также параметр, соединитель и табличную функцию;
- диаграмма состояний содержит блоки диаграмм, позволяющих графически задавать поведение объекта;

• диаграмма действий содержит блоки структурированных блок-схем, позволяющих задавать алгоритмы визуально;

• статистика содержит элементы, используемые для сбора, анализа и отображения результатов моделирования;

• презентация содержит элементы для рисования: примитивные фигуры, а также элементы управления для придания интерактивности;

• внешние данные содержат инструменты для работы с базами данных и текстовыми файлами;

• картинки содержат набор изображений наиболее часто моделируемых объектов: человек, грузовик, фура, погрузчик, склад, завод и т. д.

**Панель «Ошибки»** находится в нижнем левом углу. Она позволяет быстро с помощью двойного щелчка отобразить ошибку (курсор расположится там, где будет находиться ошибка) и локализовать ее.

***Окно свойств.*** В редакторе AnyLogic для каждого выделенного элемента модели существует свое окно свойств, в котором указываются свойства (параметры) этого элемента. При выделении какого-либо элемента в окне редактора снизу появляется окно свойств, показывающее параметры данного выделенного элемента. Окно свойств содержит несколько вкладок. Каждая вкладка включает элементы управления, такие как поля ввода, флажки, переключатели, кнопки и т. д., с помощью которых можно просматривать и изменять свойства элементов модели. Число вкладок и их внешний вид зависит от типа выбранного элемента.

В окне свойств редактора AnyLogic для каждого выделенного элемента модели указываются его свойства (параметры). При выделении какого-либо элемента в любом из окон редактора (в окне структуры, окне поведения, окне анимации или в окне классов) справа появляется окно свойств, показывающее параметры именно этого выделенного элемента. Каждый класс может иметь произвольное количество параметров различных типов (int, real, Boolean и др.). Каждый экземпляр класса может иметь собственные значения параметров, которые могут быть изменены динамически в процессе выполнения модели (рис. 3.9).

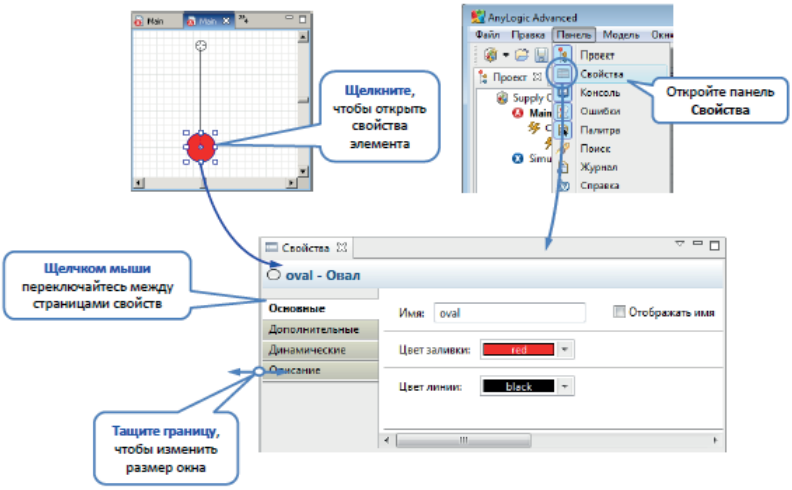



Рис. 3.9. Окно свойств [2]

В ***Окне графического редактора*** активного объекта для модели строится двухмерное или трехмерное анимационное представление, которое помогает понять, что происходит с моделью с течением времени. Именно в этом окне визуально представляется имитация поведения моделируемой системы. Элементы анимационной картинки имеют свои параметры, которые могут быть связаны с переменными и параметрами модели. Изменение переменных модели во времени ведет к изменению графического образа, что позволяет пользователю наглядно представить динамику моделируемой системы с помощью динамически меняющейся графики (рис. 3.10).

**Исполняемая модель.** При исполнении модели запустится компилятор, который построит исполняемый код модели в языке Java, оттранслирует его и затем запустит модель на исполнение (рис. 3.11).

При этом откроется окно презентации (рис. 3.12).

AnyLogic имеет развитый базовый язык дискретного и смешанного дискретно-непрерывного моделирования, на основе которого разработаны стандартные библиотеки:

- Enterprise Library – конструкции для построения дискретно-событийных моделей;

- Dynamic Systems Library (системная динамика);
- Material Flow Library (потоки материалов).

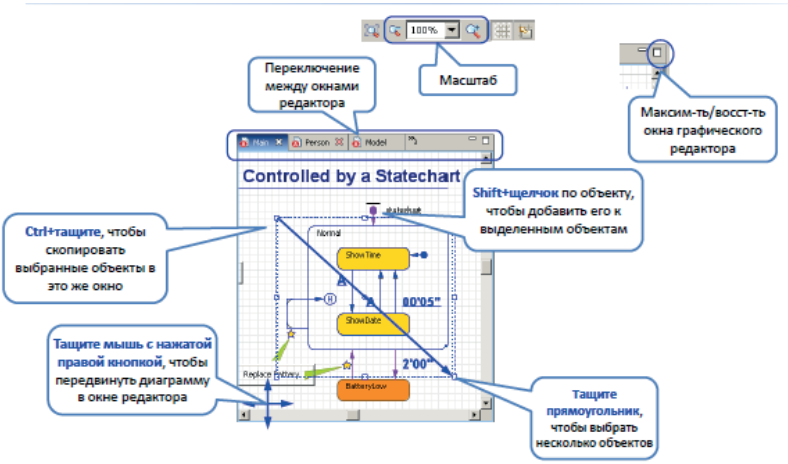



Рис. 3.10. Окно графического редактора [2]

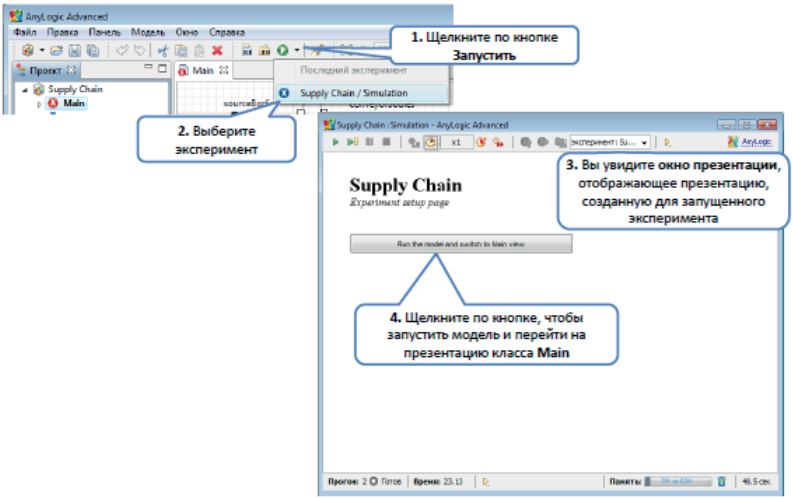



Рис. 3.11. Запуск модели

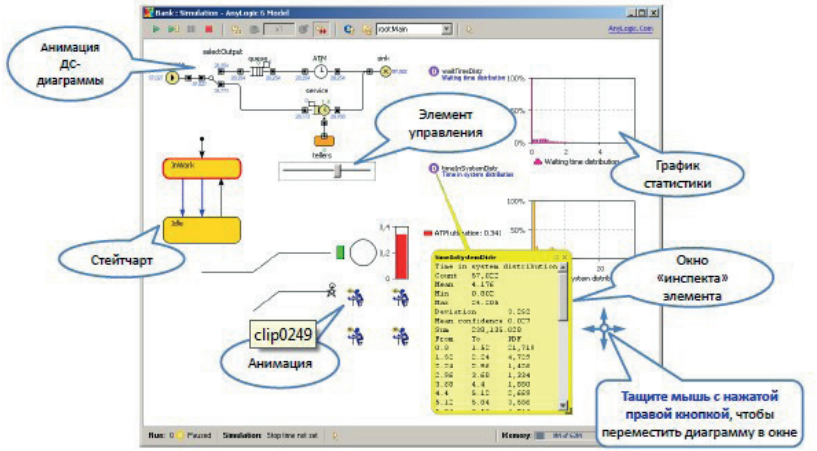



Рис. 3.12. Окно презентации

Модель и анимация быстро строятся в стиле drag-and-drop и гибко параметризуются. Реализация стандартных объектов открыта для пользователя, их функциональность может быть как угодно расширена, вплоть до создания собственных библиотек. Используя иерархию и регулярные структуры объектов, можно создавать масштабируемые модели.

С помощью библиотеки Enterprise Library пакета AnyLogic можно быстро создавать сложные дискретно-событийные модели, такие как:

- модели производственных процессов;
- модели систем обслуживания (банки, аэропорты и т. д.);
- модели бизнес-процессов с оценкой затрат операций;
- модели логистики и цепочек доставки.

Библиотека объектов Enterprise Library позволяет создавать гибкие модели с наглядной визуализацией моделируемого процесса и возможностью сбора необходимой статистики, разработана для поддержки дискретно-событийного моделирования в таких областях, как производство, цепи поставок, логистика и здравоохранение. Используя Enterprise Library, можно смоделировать системы реального мира с точки зрения заявок (сделок, клиентов, продук-

тов, транспортных средств и т. д.), процессов (последовательности операций, очередей, задержек) и ресурсов. Процессы определены в форме блочной диаграммы (рис. 3.13).

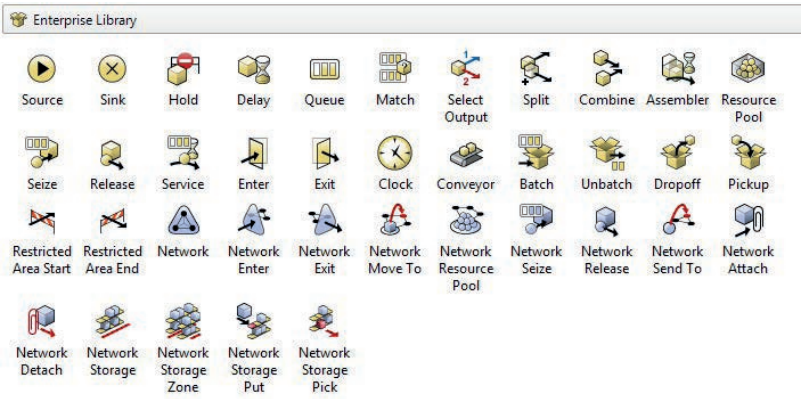



Рис. 3.13. Библиотека Enterprise Library [2]

Pedestrian Library создана для моделирования пешеходных потоков в «физической» окружающей среде. Это позволяет создавать модели с большим количеством пешеходного трафика (как станции метро, проверки безопасности, улицы и т. д.). Модели поддерживают учёт статистики плотности движения в различных областях. Это гарантирует приемлемую работу пунктов облуживания с ограничениями по загруженности, оценивает длину простаивания в определённых областях и обнаруживает потенциальные проблемы с внутренней геометрией – такие как эффект добавления слишком большого числа препятствий – и другими явлениями. В моделях, созданных с помощью Pedestrian Library, пешеходы двигаются непрерывно, реагируя на различные виды препятствий (стены, различные виды областей) так же, как и обычные пешеходы. Пешеходы моделируются как взаимодействующие агенты со сложным поведением. Для быстрого описания потоков пешеходов Pedestrian Library обеспечивает высокоуровневый интерфейс в виде блочной диаграммы (рис. 3.14).

Rail Yard Library поддерживает моделирование, имитацию и визуализацию операций сортировочной станции любой сложно-

сти и масштаба. Модели сортировочной станции могут использовать комбинированные методы моделирования (дискретно-событийное и агентное моделирование), связанные с действиями при транспортировке: погрузками и разгрузками, распределением ресурсов, обслуживанием, различными бизнес-процессами.

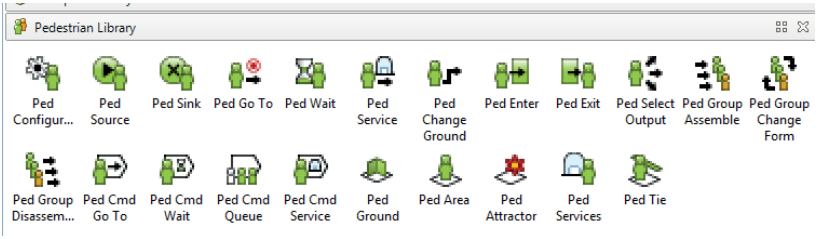



Рис. 3.14. Библиотека Pedestrian Library

Детализировать моделирующие конструкции можно, выделив их и изменив параметры, используя панель свойств.

AnyLogic поддерживает подход агентного моделирования, в качестве агентов могут быть: люди – потребители, жители, работники, пациенты, доктора, клиенты, солдаты и др.; транспорт, оборудование – автомобили, краны, самолёты, вагоны, станки и др.; нематериальные вещи – проекты, продукты, инновации, идеи, инвестиции и др.; организации – компании, политические партии, страны и др. Эти объекты в информационной системе AnyLogic могут создаваться и уничтожаться динамически, перемещаться, общаться друг с другом, иметь поведение, знания, цели, стратегию – т. е. обладают всеми свойствами агентов.

Агентный подход используют для моделирования социальных систем, в частности, рынков (агент – потенциальный покупатель), конкуренции и цепочек поставок (агент – компания), населения (агент – семья, житель города или избиратель) и многого другого. Только агентный подход позволяет получить представление об общем поведении системы исходя из предположений о поведении её элементов при отсутствии знания о глобальных законах – т. е. в наиболее общем случае.

AnyLogic основан на Java и базируется на платформе Eclipse – современном стандарте для бизнес-приложений. Благодаря Eclipse

AnyLogic работает на всех распространенных операционных системах (Windows, Mac, Linux и т. д.).

В редакторе AnyLogic возможно разработать анимацию и интерактивный графический интерфейс модели. Анимация может быть иерархической и поддерживать несколько перспектив. Например, есть возможность определить глобальный взгляд на процесс производства, а также детальные анимации конкретных операций и переключаться между ними. Имеется перспектива использования различных видов технологических решений для реализации анимации (рис. 3.15).

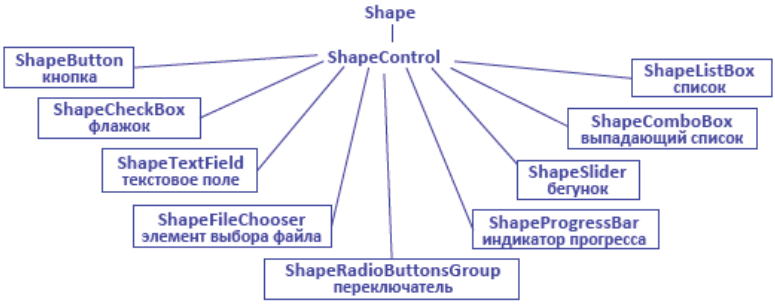



Рис. 3.15. Методы технологических решений

В AnyLogic пользователю доступно 35 стандартных теоретических распределений, также можно определить собственные распределения. AnyLogic позволяет строить как стохастические, так и детерминированные модели и проводить анализ результатов моделирования. С моделью могут быть проведены различные эксперименты:

- моделирование (simulation);
- оптимизация (optimization);
- эксперименты Монте-Карло;
- анализ чувствительности;
- эксперименты по сценарию пользователя.

В AnyLogic встроен оптимизатор OptQuest. Комбинируя эвристики, нейронные сети и математическую оптимизацию, OptQuest позволяет находить значения дискретных и непрерывных параметров модели, соответствующие максимуму или минимуму целевой

функции, в условиях неопределённости и при наличии ограничений. Для создания отчетов в AnyLogic отведена специальная палитра «Статистика», в которой содержатся конструкции для сбора данных по ходу работы модели. В этой палитре также находятся различные диаграммы, графики и гистограммы.

После запуска AnyLogic открывается рабочее окно, в котором для продолжения работы надо создать новый проект или открыть уже существующий.

Начиная с версии 6.4 AnyLogic предоставляет пользователям возможность использовать шаблоны существующих моделей при создании новых. Чтобы создать новый проект, щелкните по соответствующей кнопке на панели инструментов или выберите пункт меню **Файл | Создать проект** и затем из ниспадающего меню – **Модель**. Откроется диалоговое окно **Новая модель**, где задается имя и местоположение нового проекта. Далее следуйте указаниям *Мастера создания модели*. Можно создавать новую модель «с нуля» или использовать шаблон.

***Структурная диаграмма***. При построении модели нужно задать ее структуру (т. е. описать, из каких частей состоит модель системы) и поведение отдельных объектов системы. В AnyLogic структурными элементами модели являются так называемые *активные объекты*. Активный объект имеет структуру и поведение. Элементы структуры – это другие активные объекты, включенные как составные элементы данного активного объекта, и связи, которые существуют между включенными активными объектами. Активные объекты могут содержать события, стейтчарты, переменные, функции, уравнения, параметры. Структура активного объекта задается графически на структурной диаграмме. Поведение задается с помощью стейтчарта и определяет реакции активного объекта на внешние события – логику его действий во времени.

***Диаграмма состояний*** (или стейтчарт – *statechart*) – это модифицированные графы переходов конечного автомата. Стейтчарт позволяет графически задать пространство состояний алгоритма поведения объекта, а также события, которые являются причинами срабатывания переходов из одних состояний в другие, и действия, происходящие при смене состояний. Стейтчарты соответствуют

стандарту UML. Они сохраняют графический вид, атрибуты и семантику выполнения, определенную в UML (*Unified Modeling Language*). Стейтчарты в AnyLogic поддерживают следующие типы событий: *сигнал* – объект может послать сигнал другому объекту, чтобы уведомить его о чем-то; *таймаут* – в течение заданного промежутка времени в стейтчарте ничего не происходит; *событие* – событие, при котором значением булева выражения становится «истина».

Кроме того, в окне редактора для модели можно построить двумерное или трехмерное анимационное представление, которое помогает понять, что происходит с моделью во времени. Именно в этом окне визуально представляется имитация поведения моделируемой системы. Элементы анимационной картинки имеют свои параметры, которые могут быть связаны с переменными и параметрами модели. Изменение переменных модели во времени ведет к изменению графического образа, что позволяет пользователю наглядно представить динамику моделируемой системы с помощью меняющейся графики.

***Параметры.*** Активный объект может иметь параметры. Параметры обычно используются для задания характеристик объекта. Вы можете задать различные значения параметров для разных объектов одного и того же класса, что требуется в тех случаях, когда объекты имеют одинаковое поведение, но их характеристики не тождественны. Возможно описание параметров любых *Java*-классов.

Чтобы создать параметр класса активного объекта (рис. 3.16), в окне **Проект** щелкните мышью по классу активного объекта. В окне **Свойства** щелкните по кнопке **Новый параметр**. Задайте свойства параметра в открывшемся диалоговом окне **Параметр**.

***Переменные.*** Активный объект может содержать переменные. Переменные могут быть либо внутренними, либо интерфейсными. Активный объект может иметь переменные, моделирующие, меняющиеся во времени величины. Переменные могут быть вынесены в интерфейс активного объекта и связаны с переменными других активных объектов. Тогда при изменении значения одной переменной будет немедленно меняться и значение связанной с ней зависимой переменной другого объекта. Этот механизм обеспечивает непрерывное и/или дискретное взаимодействие объектов.

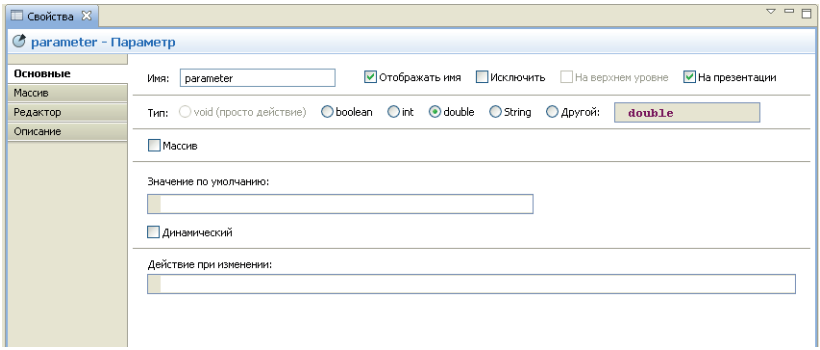


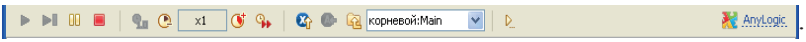

Рис. 3.16. Окно параметра класса активного объекта

***Передача сообщений.*** AnyLogic позволяет передавать информацию от одного объекта другому путем передачи специальных пакетов данных – сообщений. С помощью передачи сообщений можно реализовать механизм оповещения – сообщения будут представлять команды или сигналы, посылаемые системой управления. Можно также смоделировать поток заявок, в этом случае сообщения будут представлять собой заявки – объекты, которые производятся, обрабатываются, обслуживаются или еще каким-нибудь образом подвергаются воздействию моделируемого процесса (документы в модели бизнес-процесса, клиенты в модели системы массового обслуживания, детали в производственных моделях).

Сообщения принимаются и посылаются через специальные элементы активных объектов – *порты*. Обмен сообщениями возможен только между портами, связанными соединителями – элементами, играющими роль путей движения сообщений.

Чтобы соединить порты вложенных объектов, щелкните мышью по кнопке панели инструментов **Соединитель**, а затем щелкните мышью поочередно по обоим портам. Чтобы добавить точку изгиба, щелкните мышью по кнопке панели инструментов **Редактировать точки**.

***Запуск и просмотр модели***. Запускать и отлаживать модель можно с помощью меню **Модель** и панели инструментов:



При исполнении модели запустится компилятор, который построит исполняемый код модели в языке *Java*, скомпилирует его и затем запустит модель на исполнение.

Для запуска модели щелкните по кнопке **Выполнить**, затем выберите эксперимент из выпадающего списка. После этого откроется окно презентации, отображающее созданную презентацию для запущенного эксперимента. Щелкните по кнопке, чтобы запустить модель и перейти на презентацию.

В окне презентации можно увидеть анимированную диаграмму модели, окна инспекта элементов модели, ожившую анимацию, диаграммы состояний, графики статистики.

**Стохастическое моделирование**

Существуют различные способы описания стохастического поведения: использование различных законов распределения и/или статистики для описания времени между событиями; реализация недетерминированных алгоритмов, например утеря с определенной вероятностью заявки. В случае, если несколько событий должны произойти в одно и то же время, AnyLogic позволяет выбрать событие, которое должно произойти случайно, используя равномерное распределение.

AnyLogic поддерживает *Stat:Fit* – специализированное программное обеспечение для обработки статистики, которое позволяет подбирать распределения по имеющейся выборке.

Рассмотрим средства для разработки стохастических моделей в AnyLogic. В AnyLogic включено 37 генераторов случайных величин с наиболее часто встречающимися вероятностными распределениями: равномерным, экспоненциальным, Бернулли, биномиальным и т. п. Их описание можно найти в *руководстве пользователя* AnyLogic.

Все классы вероятностных распределений унаследованы от класса *Distr*. Они называются *Distr Exponential*, *Distr Chi*, *Distr Normal* и т. д. Класс *Distr* имеет только один абстрактный метод *get* (), возвращающий случайное значение, сгенерированное по этому закону распределения. Пользователь может определить свое вероятностное распределение, для чего нужно создать свой класс распределения и унаследовать его от базового класса *Distr*. Мето-

ды классов распределений подробно описаны в *справочнике классов* AnyLogic.

Вызвать метод очень просто, например, *exponential* (0,6) или *uniform* (–1,1), который вернет соответствующее случайное значение.

В общем случае все параметры производительности систем, функционирующих в условиях неопределенности, являются стохастическими. AnyLogic включает средства, позволяющие выполнять анализ случайных величин и визуализировать их распределения. Случайная величина (СВ) здесь задается набором данных, в котором хранятся не только все ее конкретные значения, но и автоматически подсчитываются статистические характеристики: количество реализаций, среднее, минимальное и максимальное значения, дисперсия, среднеквадратичное отклонение (СКО) и доверительный интервал для среднего значения. На выходе стохастической модели СВ легко представить графически в виде гистограммы.

**Проведение экспериментов**

С помощью экспериментов задаются конфигурационные настройки модели. AnyLogic поддерживает несколько типов экспериментов: простой эксперимент, эксперимент для варьирования параметров, оптимизационный и др. На рис. 3.17 показано окно выбора эксперимента.

***Простой эксперимент***. Задачи вида «что – если» (так называемая *прямая задача имитационного моделирования ИМ*) в AnyLogic решаются с помощью *простого эксперимента*. Простой эксперимент (с именем *Simulation*) создается автоматически при создании проекта. Он позволяет визуализировать модель с помощью анимации, графиков (диаграмм) и т. п. Широкие возможности для отображения данных предоставляет библиотека бизнес-графики (Business Graphics Library).

Для построения, например, графика зависимости переменных от времени в поле анимации сначала нужно построить прямоугольник, в пределах которого будет размещаться график, после чего в любое место поля редактора перенести экземпляр объекта *Chart Time* из Business Graphics Library. Затем в окне свойств данного объекта следует настроить параметры, определяющие цвет и толщину линий, имена отображаемых переменных, названия переменных, которые будут отображаться, цвет текста и т. д.

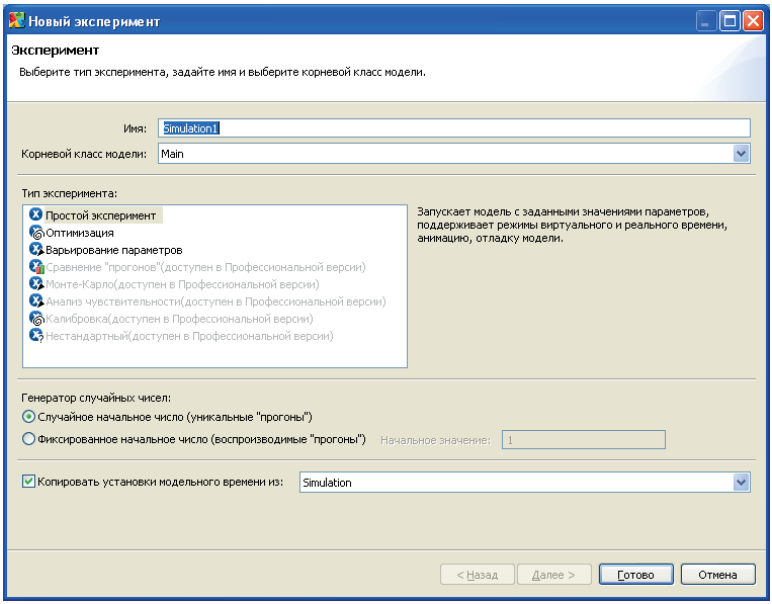



Рис. 3.17. Окно выбора эксперимента

Простой эксперимент используется в большинстве случаев при разработке и анализе моделей, созданных в AnyLogic. В частности, он поддерживает средства для отладки модели. Можно организовать несколько простых экспериментов с различными значениями исходных факторов и, сделав один из этих экспериментов *текущим*, запустить модель на выполнение.

***Эксперимент для варьирования параметров. Анализ чувствительности модели***. Анализ чувствительности модели – процедура оценки влияния исходных гипотез и значений ключевых факторов на выходные показатели модели. Обычно эксперимент с варьированием параметров и анализом реакции модели помогает оценить, насколько чувствительным является выдаваемый моделью прогноз к изменению гипотез, лежащих в основе модели. При анализе чувствительности обычно рекомендуется выполнять изменение значений факторов по отдельности, что позволяет ранжировать их влияние на результирующие показатели.

В AnyLogic доступен механизм автоматического запуска модели заданное количество раз с изменением значений выбранных параметров – это *эксперимент для варьирования параметров*. При запуске данного эксперимента пользователь может изучить и сравнить поведение модели при разных значениях параметров с помощью графиков.

Чтобы запустить такой эксперимент, нужно выполнить следующее:

- создать эксперимент для варьирования параметров;
- сконфигурировать эксперимент, выбрав параметры модели, которые вы хотите варьировать, и задав значения, которые эти параметры должны будут принять за определенное вами количество прогонов модели, в окне свойств данного эксперимента;
- запустить модель, выбрав данный эксперимент в качестве текущего.

Такой вид эксперимента не поддерживает визуализацию работы модели с помощью анимации.

***Оптимизационный эксперимент***. Используется для решения задач *количественного анализа* (расчет показателей эффективности системы). Поиск тех значений факторов, которые определяют наиболее предпочтительный вариант решения, называется *обратной задачей ИМ*. Обратные задачи моделирования отвечают на вопрос о том, какое решение из области допустимых решений обращает в максимум показатель эффективности системы. Для решения обратной задачи многократно решается прямая задача. В случае, когда число возможных вариантов решения невелико, решение обратной задачи сводится к простому перебору всех возможных решений. Сравнивая их между собой, можно найти оптимальное решение.

Если перебрать все варианты решений невозможно, то используются *методы направленного перебора с применением эвристик*. При этом оптимальное или близкое к оптимальному решение находится после многократного выполнения последовательных шагов (решений прямой задачи и нахождения для каждого набора входных параметров модели вектора результирующих показателей). Правильно подобранная эвристика приближает эксперимент к оптимальному решению на каждом шаге. В качестве блока регистрации значе-

ний выходных показателей и выбора очередного приближения при оптимизации (рис. 3.18) пользователь может использовать любой внешний оптимизатор или же оптимизатор *OptQuest*, встроенный в AnyLogic. Оптимизатор *OptQuest* разработан недавно на основе метаэвристик рассеянного поиска (scattersearch) и поиска «табу» (tabusearch). Этот оптимизатор является лучшим из предлагаемых на рынке профессиональных пакетов оптимизации для решения сложных проблем оптимизации.

Рис. 3.18. Условие прекращения эксперимента выполнено

Оптимизатор *OptQuest* запускается прямо из среды разработки модели. Чтобы настроить оптимизацию в AnyLogic, необходимо выполнить следующее:

- создать в разработанной модели оптимизационный эксперимент;
- задать оптимизационные параметры и области их изменения;
- задать условие остановки модели после каждого прогона. Это может быть либо остановка по времени выполнения прогона, либо остановка по условиям, накладываемым на переменные модели;
- задать целевую функцию, т. е. исследуемую реакцию системы;
- задать ограничения, которые в конце каждого прогона определяют, допустимо ли значение вектора исходных входных факторов. Ограничения можно не задавать (т. е. это опционально);
- задать условия прекращения эксперимента.

После запуска модели оптимизационный эксперимент найдет наилучшие значения входных параметров, при которых заданная целевая функция обратится в минимум или максимум.

## Контрольные вопросы и задания

1. Назовите три подхода имитационного моделирования.
2. Почему в первой версии был присвоен индекс 4 – Anylogic 4.0?
3. Какой язык в себя включает Anylogic? Что он позволяет расширять? С помощью какого языка?
4. Из каких двух фаз моделирования состоит Anylogic?
5. Могут ли активные объекты содержать вложенные объекты?
6. Что является основными средствами описания поведения объектов?
7. Что такое модельное время?
8. Поддерживается ли в AnyLogic трёхмерная анимация?
9. Какие элементы имеют в окне редактора AnyLogic?
10. Что обеспечивают окна проектов?
11. Какие библиотеки имеет AnyLogic?
12. Какие эксперименты могут быть проведены с моделью?
13. Что такое элемент структуры?
14. Что такое диаграмма состояний?
15. Что позволяет передача сообщений? Сообщения принимаются и посылаются через специальные элементы активных объектов, что это значит?

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Куприяшкин, А. Г. Основы моделирования систем : учеб. пособие / А. Г. Куприяшкин; Норильский индустр. ин-т. – Норильск : НИИ, 2015. – 135 с.

2. Маликов, Р. Ф. Практикум по имитационному моделированию сложных систем в среде AnyLogic 6 : учеб. пособие / Р. Ф. Маликов. – Уфа : Изд-во БГПУ, 2013. – 296 с.

3. Замятина, О. М. Моделирование систем : учеб. пособие / О. М. Замятина. – Томск : Изд-во ТПУ, 2009. – 204 с.

4. Киселева, М. В. Имитационное моделирование систем в среде AnyLogic : учеб.-метод. пособие / М. В. Киселева. – Екатеринбург : УГТУ – УПИ, 2009. – 88 с.

5. Боев, В. Д. Компьютерное моделирование : пособие для практических занятий, курсового и дипломного проектирования в AnyLogic 7 / В. Д. Боев. – СПб. : ВАС, 2014. – 432 с.

6. Григорьев, И. Anylogic за три дня : учеб.-метод. пособие / И. Григорьев. – СПб. : СППУ, 2016. – 202 с.

7. Моделирование сложных вероятностных систем : учеб. пособие / В. Г. Лисиенко, О. Г. Трофимова, С. П. Трофимов, Н. Г. Дружинина, П. А. Дюгай. – Екатеринбург : УРФУ, 2011. – 200 с.

8. Боев, В. Д. Концептуальное проектирование систем в Any Logic 7 и GPSS World / В. Д. Боев. – М. : ИНТУИТ.ru, 2013. – 513 с.

9. Боев, В. Д. Об адекватности систем имитационного моделирования GPSS World и AnyLogic. Ч. 1 / В. Д. Боев // Прикладная информатика. – 2010. – № 6 (30). – С. 69–82.

10. Боев, В. Д. Об адекватности систем имитационного моделирования GPSS World и AnyLogic. Ч. 2 / В. Д. Боев // Прикладная информатика. – 2011. – № 4 (34). – С. 50–62.

11. Девятков, В. В. Мир имитационного моделирования: взгляд из России / В. В. Девятков // Прикладная информатика. – 2011. – № 4 (34). – С. 9–29.

12. Карпов, Ю. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю. Карпов. – СПб. : БХВ-Петербург, 2005. – 400 с.

13. Маликов, Р. Ф. Основы математического моделирования : учеб. пособие / Р. Ф. Маликов. – М. : Горячая линия – Телеком, 2010. – 368 с.

14. Сьерра, К. Изучаем Java / К. Сьерра, Б. Бэйтс [пер. с англ.]. – М. : Эксмо, 2012. – 720 с.

15. Официальный сайт компании AnyLogic [Электронный ресурс]. – Режим доступа: www.anylogic.ru.

16. Кельтон, Д. Имитационное моделирование. Классика CS / Д. Кельтон, А. Лоу [пер. с англ.]. – СПб. : Питер, 2004. – 847 с.

# ОГЛАВЛЕНИЕ

**Введение** ..... **3**
**Глава 1. Методологические основы имитационного моделирования** ..... **6**
1.1. Моделирование как научный метод ..... 6
1.2. Исходные понятия и определения ..... 11
1.3. Классификация моделей ..... 17
1.3.1. Классификация моделей по степени абстрагирования от оригинала ..... 18
1.3.2. Классификация моделей по степени устойчивости ..... 21
1.3.3. Классификация моделей по отношению к внешним факторам ..... 21
1.3.4. Классификация моделей по отношению ко времени ..... 22
1.4. Разновидности моделирования ..... 22
1.4.1. Математическое моделирование ..... 24
1.4.2. Аналитическое моделирование ..... 28
1.4.3. Информационные системы моделирования и проектирования ..... 58
Контрольные вопросы и задания ..... 67
**Глава 2. Системная динамика** ..... **69**
2.1. Методология системной динамики ..... 69
2.1.1. Системный анализ и этапы имитационного моделирования сложных систем ..... 75
2.1.2. Проектирование и разработка имитационных моделей сложных объектов ..... 79
Контрольные вопросы и задания ..... 91
**Глава 3. Среда имитационного моделирования AnyLogic 7** ..... **93**
3.1. Общие сведения о системе имитационного моделирования AnyLogic ..... 93
3.2. Средства AnyLogic для имитационного моделирования систем ..... 99
3.2.1. Основные концепции ..... 99
3.2.2. Базовые инструменты для разработки модели в среде AnyLogic ..... 104
Контрольные вопросы и задания ..... 123
**Библиографический список** ..... **124**

Учебное издание

**Эльберг** Мария Сергеевна
**Цыганков** Никита Сергеевич

# ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

Учебное пособие

Редактор *М. В. Саблина*
Компьютерная верстка *О. А. Кравченко*

Подписано в печать 22.09.2017. Печать плоская. Формат 60×84/16
Бумага офсетная. Усл. печ. л. 8,0. Тираж 100 экз. Заказ № 1002

Библиотечно-издательский комплекс
Сибирского федерального университета
660041, Красноярск, пр. Свободный, 82а
Тел. (391) 206-26-67; http://bik.sfu-kras.ru
E-mail: publishing_house@sfu-kras.ru