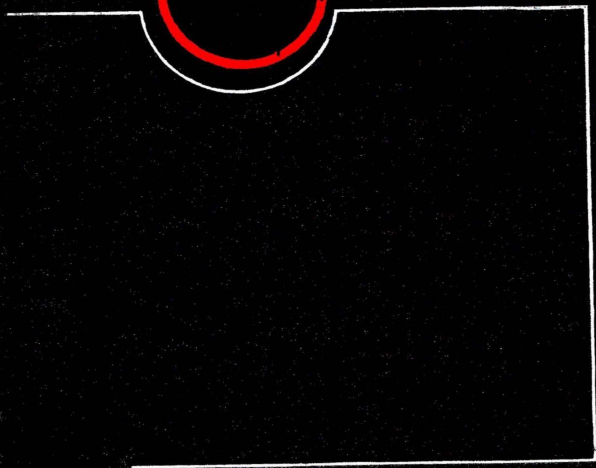


ИНФОРМАЦИОННАЯ
ТЕХНОЛОГИЯ
НА

Т
РАНСПОРТЕ



ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ НА РАНСПОРТЕ

ИЗДАНИЕ В ТРЕХ КНИГАХ

Под общей редакцией
академика АН УССР
В. С. Михалевича

И.В.В. 2017 Киев

В. И. Гриценко
В. А. Богемский
А. А. Панченко

ПРОМЫШЛЕННЫЙ ТРАНСПОРТ

**АКАДЕМИЯ НАУК УКРАИНСКОЙ ССР
ИНСТИТУТ КИБЕРНЕТИКИ им. В. М. ГЛУШКОВА**

Информационная технология на транспорте. 1. Промышленный транспорт
Гриценко В. И., Богемский В. А., Панченко А. А.; Отв. ред. Бакаев А. А.;
АН УССР. Ин-т кибернетики им. В. М. Глушкова.— Киев : Наук. думка, 1990.—
200 с.— ISBN 5-12-001287-6.

В монографии рассматриваются принципиальные основы и сущность информатики, ее роль и место в повышении качества управления промышленным транспортом. Предлагаются модели баз данных для промышленного транспорта, эффективно отображающие динамику и другую специфику транспортных систем и процессов. Рассматриваются принципы и приводятся примеры использования методов и средств новой информационной технологии (локальных сетей персональных ЭВМ, знаний, диалоговых технологий решения задач и др.) в планировании и управлении работой транспортно-технологических комплексов. Приводится методика построения и использования имитационных моделей для исследования и оптимизации транспортных объектов и процессов.

Для специалистов по применению средств информатизации и вычислительной техники на промышленном транспорте, студентов и аспирантов соответствующих специальностей.

Ил. 33. Табл. 16. Библиогр.: С. 191—195 (117 назв.).

Ответственный редактор *А. А. Бакаев*

Утверждено к печати ученым советом
Института кибернетики им. В. М. Глушкова АН УССР

Редакция физики и кибернетики

Редактор *И. Г. Васинюк*

И $\frac{140400000-478}{M221(04)-90}$ 181-90

ISBN 5-12-001287-6

© В. И. Гриценко, В. А. Богемский, А. А. Панченко, 1990

Третья книга, «Водные виды транспорта», посвящена информационному моделированию транспортных процессов как фундаментальной основы для проектирования и разработки автоматизированных систем для управления речным и морским транспортом. Рассматриваемый класс моделей призван обеспечить функциональную полноту и информационную совместимость систем, их реальную интеграцию на организационно-технологическом и информационно-программном уровнях в рамках интегрированных и комплексных АСУ.

Конечно, настоящее издание не претендует на исчерпывающее рассмотрение всего круга проблем, связанных с развитием и применением новых информационных технологий на транспорте. Тем не менее авторы выражают надежду, что все три книги окажутся полезными для разработчиков АСУ разных видов транспорта, а также помогут специалистам по управлению транспортом полнее и отчетливее представить себе возможности информатики в повышении эффективности его работы.

СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ

- АРМ — автоматизированное рабочее место
АСНИ — автоматизированная система научных исследований
АСТМ — автоматизированная система транспортировки материалов
АСУ — автоматизированная система управления
АСХД — автоматизированная система обработки данных
ВЦ — вычислительный центр
ВЦКП — вычислительный центр коллективного пользования
ГАП — гибкое автоматизированное производство
ЕТП — единый технологический процесс
ИИС — информационно-вычислительная система
КАСУ — комплексная автоматизированная система управления
ЛПР — лицо, принимающее решение
ЛС — локальная сеть
МППЖТ — межотраслевое предприятие промышленного железнодорожного транспорта
МПС — Министерство путей сообщения
НИИ — новая информационная технология
ППП — пакет прикладных программ
ПЕЭВМ — персональная ЭВМ
САПР — системы автоматизации проектирования
СКТБ — специальное конструкторско-технологическое бюро
СУБД — система управления базой данных
УАПАМ — универсальная автоматизированная имитационная модель
ЦСУ — центральное статистическое управление
ЭВМ — электронно-вычислительная машина
ЯПГ — язык графического программирования
ЯМГД — язык манипулирования графическими данными
ЯМД — язык манипулирования данными
ЯОД — язык определения данных

ПРЕДИСЛОВИЕ

Нурное развитие компьютеризации, проникновение ее во все сферы человеческой деятельности приводит к тому что все большая часть информации (научно-технической, научно-экономической, производственной и др.) перемещается в память ЭИМ и ожидается, что именно там будет сконцентрирована в недалеком будущем основная ее масса.

Овладевание методами машинной обработки информации — залог эффективного использования интеллектуального информационного потенциала, накопленного человечеством. Машинная обработка данных лежит в основе любой автоматизированной системы разнообразного назначения и разных отраслях народного хозяйства. Так, в транспорте методы и средства информатики, при умелом их использовании, станут мощным рычагом повышения эффективности его работы.

Предлагаемое издание «Информационная технология на транспорте» посвящено вопросам применения автоматизированных информационных технологий в комплексе с экономико-математическими и имитационными моделями и методами в системах различного назначения (проектирования, планирования, управления, прогнозирования и т. д.), ориентированных на транспорт.

В первой книге, «Промышленный транспорт», рассматриваются принципиальные основы и сущность информатики, ее роль и место в повышении качества управления промышленным транспортом. Анализирую в целом современное состояние, основные тенденции развития методов и средств новой информационной технологии (баз данных и знаний, локальных сетей персональных ЭВМ, диалоговых технологий решения задач и др.), и имитационного моделирования, авторы стремились продемонстрировать возможности и эффективность их практического применения на примере ряда задач проектирования, планирования и управления работой транспортно-технологических комплексов. При этом учитывалось использование персональных компьютеров периферийными специалистами — инженерами, конструкторами, административно-управленческим персоналом — на своих рабочих местах.

Вторая книга, «Железнодорожный транспорт», посвящена вопросам разработки информационно-программного обеспечения нового поколения автоматизированных информационных систем и систем управления на магистральном железнодорожном транспорте. Они содержат в своей архитектуре общесистемные блоки, позволяющие гибко реагировать на изменения как структуры самого объекта, так и внешней относительно него среды или предметной области. В основу такого подхода положено использование методов новой информационной технологии для унификации описания технологических схем обработки данных о перевозочном процессе на железной дороге. Предлагаемые информационно-программные средства ориентированы на специалистов по управлению транспортом, не имеющих специальной подготовки по электронной обработке данных,

ВВЕДЕНИЕ

Информатизация общества — проблема, требующая безотлагательного решения. Смысл ее — переход общества в новое качественное состояние, выход на новые исторические рубежи, называемые информационным обществом [1].

Информационное общество — объективно возникающая в ходе исторического процесса стадия общественного развития. Она предполагает качественно новый, более высокий уровень производительных сил (в сравнении с индустриальным обществом). При переходе от индустриального к информационному обществу устраняются ограничения в области накопления и использования информационных ресурсов во всех сферах социальной практики, что ведет к росту динамизма социальных процессов и ускорению общественного прогресса. Основу социальной динамики в информационном обществе составляют не традиционные материальные ресурсы (которые к тому же во многом исчерпаны), а информационные (интеллектуальные) ресурсы — знания, наука, организационные факторы, интеллектуальные способности людей, их инициатива, творчество.

Информационное общество характеризуется качественно новым производственным аппаратом, основанным не на традиционной механизации и автоматизации, а на принципиально новых, компьютеризованных орудиях труда и информационных технологиях.

В ходе современной научно-технической революции общество вступило в полосу всеохватывающей компьютерной автоматизации с тенденцией к замене машинами многих производительных функций человека — физических и интеллектуальных. Создается машинизированная нервная система общественного производства, как в свое время была создана его костно-мышечная система. Технической основой этого являются ЭВМ, компьютеризованные системы, телекоммуникационная техника и т. п.

Таким образом, альтернативы информатизации нет. Это объективный этап социального прогресса во всех областях, прежде всего в экономике, управлении, науке и технологии.

Что такое информатизация в более конкретном плане? Социальную информатизацию нельзя смешивать с компьютеризацией. Информатизация предполагает более широкий подход к компьютеризации, включающий преобразование всего комплекса средств и условий развертывания информационных процессов: создание соответствующей

технологической базы и модернизацию организационно-экономических, юридических и человеческих факторов.

В современных условиях с информатизацией связывается создание необходимой и качественно новой информационной среды для решения всего комплекса важнейших задач экономического, социального, научно-технического развития и повышения обороноспособности страны. Информационную среду составляют в совокупности ЭВМ, системы компьютерных коммуникаций, банки данных и базы знаний, комплексы программ и т. д., обеспеченные соответствующим контингентом подготовленных кадров и объединенные в целостные информационные технологии. Эти технологии, в свою очередь, становятся органичной частью научно-технических коммуникаций и документопотоков, процедур и регламентов управления.

Таким образом, в отличие от компьютеризации информатизация предполагает преобразование всего комплекса средств и социальных условий развертывания информационных процессов во всех сферах общественной деятельности на новой технической, технологической, организационной и юридической основе при условии их эффективного действия между собой и человеком.

Создание машинно-информационных сред в разных сферах общественной практики (производство, управление, наука, образование, медицина, сфера услуг, оборонная сфера и т. д.) вместо исторически сложившихся бумажных сред (формирование и рациональное использование машинной информатики вместо бумажной информатики) и есть информатизация.

Без информатизации общественных подсистем процессы компьютеризации и развертывания различного рода информационных систем значительную опасность и ущерб: дают затухающую отдачу и в конечном счете приносят огромные социальные потери.

Информатизация немаловажна без формирования индустрии информатики, без создания информационной инфраструктуры жизнедеятельности общества в новых условиях. Но это технологический аспект. Важно, чтобы данная инфраструктура эффективно работала, создавая новые модели деятельности и жизни людей во всех социальных сферах, включая быт.

На Западе и в Японии проблемы информатизации общества разрабатываются уже два десятилетия, а в последние годы там развернуты соответствующие общенациональные программы, поддерживаемые огромными государственными субсидиями. Информатизация считается прорывом в будущее. На это пошли все развитые страны, придав информатизации высшие приоритеты, подчинив этой цели основные ресурсы и усилия. Из объекта теоретического анализа информатизация превратилась в критерий оценки могущества, стала важнейшим ориентиром для выработки внутренней и внешней стратегии государств.

История распорядилась так, что к концу XX в. цивилизация (это касается не только нашей страны, но и всех стран) столкнулась с такими ограничителями дальнейшего развития общества, которые нельзя преодолеть традиционными путями. Это ресурсная напряженность (нехватка и резкое удорожание энергии, сырья, трудовых ресурсов),

экологический кризис (загрязнение окружающей среды), информационно-организационная сложность и неупорядоченность (катастрофический рост энтропии социальных систем). Выход из такого положения мыслим только на базе использования нетрадиционных (неизвестных или недостаточно задействованных в прошлом) ресурсов — ресурсов социальной информатизации, основанной на прогрессе компьютерных, информационных и телекоммуникационных систем. В промышленной сфере — это появление новых типов машин и механизмов со встроенными микропроцессорами, придающими оборудованию новые потребительские свойства: эффективное протекание технологических процессов, интеллектуальные средства диагностики и защиты, взаимодействие оператора с машиной или системой машин. Это интегрированные системы управления производством, сочетающие преимущества массового и единичного производства, гибко реагирующие на изменения спроса, безлюдные и малолюдные производства, насыщенные роботами с профессиональной ориентацией, заводы-автоматы. Это уже действительность нашего времени.

Сегодня мы видим примеры использования комплексов машин, информационных технологий, позволяющих на порядок повысить производительность труда. Впечатляют достижения в медицине, технологии обучения. Информатизация проникает и в быт. Используются средства безличного расчета (электронные деньги), электронная почта и т. д.

В политическом смысле информатизация выступает как информационно-технологическое обеспечение перестройки советского общества.

1. Обеспечение доступа широких масс к информации (знаниям), а также возможность оперативно излагать собственное мнение по самым разным вопросам жизни страны с уверенностью, что это мнение вовремя дойдет до уровня анализа и принятия решений (непременное условие демократизации).

2. Возможность установления действенных информационных связей между предприятиями в процессе составления и реализации планов.

3. Возможность видения на верхних уровнях управления реальной обстановки на местах, т. е. информационная реализация вертикальных связей и тем самым обеспечение оптимального сочетания централизованного управления с системой самостоятельности на уровне предприятий и объединений (непременное условие осуществления радикальной экономической реформы).

4. Новые возможности межотраслевого взаимодействия, более гибкого сочетания отраслевого и регионального управления.

Таким образом, информатизация общества открывает широкие возможности повышения эффективности общественного производства, совершенствования управления экономикой, наукой и научно-техническим прогрессом, реализацией важнейших задач в социальной сфере.

Важнейшим направлением комплекса работ по информатизации общества является формирование индустрии информатики, которая на-

ряду с производством средств информатизации должна обеспечить потребности в сборе, передаче, обработке и представлении информации во всех сферах жизни общества.

Для формирования информационной инфраструктуры общества необходимо поэтапное создание

а) баз данных (знаний) в области науки и техники, экономики, производства, социальной сферы, быта, в военной и политической сферах, в области международных отношений и т. д.;

б) мощной информационной и телекоммуникационной сети, обеспечивающей оперативный доступ к базам знаний и их актуализацию, взаимосвязь и взаимодействие;

в) автономных автоматизированных систем различного уровня и назначения с возможностью их последовательной интеграции;

г) интеллектуализированных автоматизированных рабочих мест и рабочих станций, и т. д.

Вся эта система должна быть связана с низовыми информационными структурами общественного производства, обслуживать роботизированные предприятия, цехи и участки. Отсюда вытекают и требования к государственному заказу на средства вычислительной информационной техники (виды, пропорции, количества, этапы).

Последовательно осуществляемая информатизация делает реальными и берет в свой арсенал экспертные системы, системы подготовки и поддержки решений, средства искусственного интеллекта. Особая роль принадлежит методам математического моделирования и вычислительного эксперимента. Важно создать комплексы взаимосвязанных математических моделей по всем предметным областям жизни общества. Именно они определяют возможность целесообразного использования информации для анализа ситуаций, подготовки и принятия решений. Методологически информатизация и системный анализ взаимосвязаны. Все это обеспечивает возможность системной оптимизации для реального осуществления политики ресурсосбережения.

В предлагаемой вниманию читателя монографии анализируется понятие информатики, ее особенности и основные задачи, раскрывается суть новой информационной технологии и ее роль в информатизации общества. Далее рассматриваются некоторые теоретические и практические аспекты реализации методов прикладной информатики и новой информационной технологии в такой сфере применения, как промышленный транспорт. Хотя с технологической точки зрения ему присуща специфика, значительно отличающая его от других предметных областей, предлагаемые модели, методы и информационные технологии носят достаточно унифицированный характер и могут быть использованы при автоматизации процессов обработки данных и управления, а также и в других производственных и непроизводственных сферах.

Глава 1

ИНФОРМАТИКА И ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

1.1. ИНФОРМАТИКА КАК НАУКА

Становление и развитие понятия информатики, ее особенности и задачи как комплексной научной дисциплины достаточно глубоко и всесторонне исследованы в [2]. В ней отмечается, что создание общей теории информационных процессов — одна из важнейших задач информатики, находящейся сейчас в самом начале своего становления. Концепции, функции, строение этой теории находятся в стадии выявления и кристаллизации. Формулировка основных понятий информатики, например, таких, как информация, информационная потребность, информационно-поисковая система, информационная деятельность и др., определение предмета и метода информатики, анализ взаимосвязи последней с естественными, техническими и общественными науками, с различными видами практической деятельности ведут к появлению новых методологических проблем, выходящих уже далеко за рамки информатики.

За последние тридцать лет в развитии информатики можно выделить три периода, различающихся изменением приоритетов сфер общественной практики в организации информационных процессов:

на протяжении 50—60-х годов информатика исходила из общественной потребности упорядочить обмен информацией главным образом внутри самой науки. Внешними факторами, обусловившими такую потребность, явились экспоненциальный рост научной литературы, трудности тематического отбора в связи с рассеянием и ограниченностью наличных поисковых средств, процессы дифференциации и интеграции науки;

в 70-х годах произошло дальнейшее расширение сферы приложения информатики. Научная информация во все большей степени стала играть существенную роль в управлении, политике, глобальном моделировании развития общества и т. д.,

в 80-х годах информатика претерпевает быстрые изменения, характеризуется значительно возросшим интересом к ее теоретическим и методологическим проблемам.

Возросший интерес к информатике в настоящее время связан прежде всего с тем, что резкий рост разносторонних знаний, сведений, данных, обусловленный интенсивным развитием народного хозяйства, науки, сферы услуг и т. д., привел к тому, что своевременное получение пользователем нужной ему информации достигло предела возмож-

ностей традиционных ручных методов ее сбора, преобразования, хранения и передачи.

Объективные социальные, экономические, научно-технические и другие потребности, обусловившие возросший интерес к эффективной организации практической информационной деятельности, определили и теоретическое обоснование информационной технологии. Можно говорить о наличии противоречия в развитии практики и теории информационной технологии, которое выражается в некотором отставании ее теории от практики. Информатике все еще свойственны расплывчатые границы, ее важнейшие теоретические понятия пока четко не обозначены, не оформились полностью собственные методы исследования, не установлен окончательно ее фундаментальный закон, т. е. общая теория, которая давала бы целостное представление о развитии информационной технологии, еще не создана.

Первоначальная информатика зародилась в виде совокупности дисциплин, с родственным информационно-техническим профилем. Одной из таких дисциплин являлась информатика в ее узком значении — исследование научных коммуникаций, теория и практика информационной деятельности [3]. Ее появление в 50-х годах было вызвано резким ростом потока научных публикаций, что значительно затруднило поиск нужной информации. Острая потребность в «усилителях» интеллекта проявилась не только в науке, но и в сфере управления. К этому времени развился ряд исследовательских направлений и областей техники, позволивших создать соответствующие технические средства. Теория и технические средства научно-информационной деятельности появились именно тогда, когда поиск нужной информации вынуждал значительный объем (до трети) в рабочее время исследователя.

Другим важным направлением будущей информатики является компьютерная техника и техника. Главный стимул ее развития — необходимость быстрого проведения громоздких вычислений, научных и инженерных расчетов.

Одновременно с компьютерной наукой и техникой и в тесном взаимодействии с ней рождалась кибернетика. На определенном этапе именно кибернетика сыграла роль главного интегратора всех направлений информатики [4].

За последнее время понятие информатики существенно расширилось по сравнению, например, с приводимым в работе [5], где информатика рассматривается как дисциплина, изучающая структуру и общие свойства научной информации, а также закономерность ее создания, преобразования, передачи и использования в различных сферах человеческой деятельности.

Развитие информатики сопровождается непрекращающимися дискуссиями о предмете и объеме ее исследования, понятие информатики по-разному трактуется различными авторами так же, как различно оценивается ее роль и социальная значимость. Информатика, по мнению видных советских ученых, является общенаучной дисциплиной в том смысле, что ее объект — научная информация, научные коммуникации, научно-информационная деятельность — характеризует все без исключения науки. Общенаучный статус информатики и рассмот-

рение ее предмета обусловлены повышением роли информационных процессов в современной науке и обществе. Предлагаемые информатикой методы и средства информационного обеспечения используются в естественных, технических и общественных науках, во взаимосвязи науки с производственной и другими видами практической деятельности. В связи с разнообразием областей применения средств информатики и ее общенаучным характером известные определения понятия информатики также разнообразно отражают те или иные аспекты теории и практики информационной технологии, не противоречат друг другу, а наоборот, дополняют и развивают определение одно другим.

В книге В. М. Глушкова [6] под информатикой понимается наука о совокупности процессов, методов и средств накопления, обработки и распространения информации в человеческом обществе.

В предисловии к своей последней книге академик В. М. Глушков писал, что человек, не умеющий пользоваться ЭВМ, подобен будет в начале XXI в. человеку начала XX в., не умевшему ни читать, ни писать. Все прогнозы ученых сводятся к тому, что к началу третьего тысячелетия основная масса накопленной человечеством информации будет храниться в памяти ЭВМ, т. е. в безбумажном виде, поэтому каждому образованному человеку уже в ближайшем будущем нужно быть знакомым с основами безбумажной информатики.

Основу безбумажной информатики составляет целый ряд разделов науки и техники. В. М. Глушков предвидел будущее развитие информационной технологии и много работал над тем, чтобы мы в настоящее время были заранее подготовлены к решению чрезвычайно сложных и актуальных проблем информатики.

Многие ученые считают, что путь к становлению информатики лежит через познание природы и сущности информации, закономерностей ее преобразования, применения для ее обработки новых технических средств и научных методов.

А. А. Самарский [7] подчеркивает, что сфера действия информатики и ее содержание постоянно расширяются. Если в самом начале ученые рассматривали информатику как науку об информации, то теперь информатика включает и способы получения с помощью ЭВМ новой информации, новых знаний. Такую возможность дает ученым математическое моделирование различных явлений, проведение вычислительных экспериментов. Эту же мысль подчеркивает в своих работах и Э. А. Якубайтис. Он говорит об информатике как о новой науке, которая занимается проблемами обработки различных видов информации, созданием новых высокоэффективных ЭВМ, позволяет предоставить человеку широкий спектр различных информационных ресурсов.

Эти положения развиваются в работах [8—10], где отмечается, что информатика призвана «снять» предел возможностей традиционных ручных методов обработки информации за счет использования специальной машинной технологии сбора, обработки, передачи информации. Этой же точки зрения придерживается и А. А. Дорониничин, подчеркивая, что информатика есть наука, по своему существу базирующаяся на вычислительной технике.

В работе [11] информатика понимается как комплексная научная и инженерная дисциплина, изучающая все аспекты разработки, проектирования, создания, оценки, функционирования механизированных (основанных на ЭВМ) систем обработки информации, их применения и воздействия на различные области социальной практики. Появление этой дисциплины обусловлено возникновением и распространением новой — индустриальной — технологии сбора, обработки, передачи информации, связанной с фиксацией данных на машинных носителях.

Предметом информатики как научной дисциплины и выступает технология сбора, обработки, передачи информации — технология, которая переводит практику управления, регулирования материального производства, научных исследований, образования и других областей человеческой деятельности на принципиально новый — индустриальный уровень.

Как и всякая технология в современном понимании, индустриальная информационная технология включает определенные наборы материальных средств (в данном случае информационных носителей и средств изменения их состояний), способы их соединения, совокупность определенных знаний и приемов труда, организацию, а также комплекс воздействий на ту среду, в которой она функционирует. Все это относится к информатике.

Из определения информатики как научной дисциплины следует и ее прикладное значение. Рассматривая состав информатики как «три неразрывно и существенно связанные части — технические средства, программные средства и алгоритмические средства» [12], необходимо учитывать, что она предопределяет проведение всех необходимых организационно-технических мероприятий по комплексной разработке, созданию и использованию технических, программно-информационных и алгоритмических средств, обеспечивающих эффективное удовлетворение потребностей пользователей в информационно-вычислительных услугах. При достижении достаточно высокого информационно-вычислительного потенциала, интенсивно и широко используемого потребителями, информатика неизбежно становится инфраструктурой, охватывающей различные отрасли народного хозяйства [8].

А. П. Ершов дает такое определение информатике как науке: *информатика — это наука, изучающая закономерности и методы накопления, передачи, обработки и представления информации на основе применения ЭВМ* (информационный подход центральный в трактовке информатики). Данное понятие почти идентично понятию, приведенному в [11], однако в нем больше акцентируется на понятии применения ЭВМ, что позволяет конкретизировать понятие информатики, избавить его от многих формулировок. При этом важно учитывать и дилемму: «на основе применения ЭВМ» и «понятия ЭВМ», т. е. различия, подчеркивает А. П. Ершов, когда мы говорим об ЭВМ как о вычислительном устройстве, а когда — как об абстрактной схеме автоматической обработки информации, производимой самыми различными средствами.

Поэтому необходимо и целесообразно разграничить и равноправное употребление терминов информатика и информационная техно-

логия, первое — как методология, второе — как конкретная деятельность.

Иными словами, при рассмотрении понятия информационной технологии как конкретной деятельности последняя понимается именно как способ информационного производства, как совокупность методологических и методических положений, организационных установок, инструментально-технологических средств и т. д. — всего того, что регламентирует и поддерживает деятельность людей, вовлеченных в информационное производство на основе применения ЭВМ.

Широта и чрезвычайная разнохарактерность применения средств вычислительной техники и связи в организационном управлении значительно стимулировали развитие компьютерной науки и техники, а также ясно показали, что ЭВМ — это не просто мощный арифмометр, а еще и новое средство осуществления межличностных коммуникаций и усиления интеллекта (т. е. чисто вычислительные функции ЭВМ постепенно отодвигаются на второй план). Это, в свою очередь, открыло еще более широкие пути и перспективы совершенствования вычислительной техники, систем коммуникаций, средств общения человека с ЭВМ. В результате появились персональные ЭВМ и локальные сети, средства речевого ввода информации и графопостроители, такие мощные средства усиления интеллекта, как АСНИ, САПР, экспертные информационные системы, машинные банки данных и банки знаний. В этом случае *информационная технология понимается как методология поддержки коллективного информационно-коммуникативного процесса, как обобщенный способ информационного производства.*

Представление о структуре информатики, конечно, не исчерпывается только рассмотренными подходами. Оно будет гораздо полнее, если попытаться выделить основные области применения информатики, т. е. исследовать ее функциональную структуру.

В зарубежной литературе, посвященной проблеме развития информатики, обычно рассматриваются основные области функционирования информатики: интегральные сети связи и передачи данных, «электронная почта», «электронная контора» [13].

Для отечественной литературы характерен более широкий подход. Так, в [14] выделяются следующие крупные области применения ЭВМ: информационные системы и средства коммуникации (средства поиска, хранения, передачи информации, ее обработки в реальном масштабе времени, которые используются в основном в быту, образовании, культуре); средства автоматизации и управления (АСУ, САПР, АСНИ, ГАП, средства робототехники); средства математического моделирования и вычислительного эксперимента.

Естественно, что при практической реализации эти сферы применения ЭВМ оказываются неразделимы и их дальнейший синтез представляется неизбежным. Особенно велико значение такого синтеза для современного экономического развития, связанного с комплексной автоматизацией и роботизацией производства, внедрением гибких производственных систем. Подтверждением этому является создание системы компьютеризованного производства, которая включает компьютеризованное конструирование, групповую технологию, планирова-

ние и управление производства, автоматизацию вспомогательных процессов, компьютеризацию производственных операций, робототехнику. Это позволяет использовать ЭВМ на всех стадиях производства и прогнозировании развития производства и управления им, в учете, контроле и т. д.

Современная информатика рождается в результате объединения ряда дисциплин: компьютерной науки и техники, различных областей кибернетики, информатики (в узком смысле слова), теории и практики автоматизированных систем управления, научных исследований в области сетей связи, искусственного интеллекта, исследований и практических разработок по созданию электронной почты, электронной конторы и т. д. Основой их объединения является объективное единство и фундаментальная общность закономерностей информационных процессов, а непосредственным практическим выражением — внедрение компьютерной техники, переход на новую информационную технологию в самых различных областях человеческой деятельности [4].

Следовательно, общность научных и инженерных подходов и решений, на основе которых создается и применяется информационная техника различного назначения, т. е. техника приема, обработки, хранения, передачи, использования информации, универсальность современной информационной технологии, синтез вокруг компьютерной науки ряда исследовательских направлений и областей техники, научно-технических дисциплин информационного профиля не является случайной. Постепенно, по мере накопления общего содержания и расширения области применения компьютерной техники, разворачивается процесс все более тесной интеграции и одновременно дифференциации, специализации отдельных направлений единой научно-технической области, которая связывается с понятием современной информатики.

В самой изначальной форме процесс синтеза ранее обособленных направлений информационной науки и техники, а значит, и генезиса информатики, происходил в форме смещения в сторону информатики основных понятий, принятых ранее в соответствующих направлениях. В первую очередь, осуществлялось расширенное толкование кибернетики и теории научной информации. Кибернетика, согласно определению, является наукой об управлении, которая предполагает информационное взаимодействие субъекта и объекта управления, в этом смысле и специфику управления в его кибернетическом понимании. В то же время существует множество информационных процессов, которые нельзя отнести к управлению, хотя они и осуществляются с применением ЭВМ (например, решение научных задач, инженерные расчеты, работа с базами данных и др.). В особенности это касается функционирования информационных средств общения людей, т. е. работы сетей связи, электронной почты, организации взаимодействия распределенных баз данных, а в недалекой перспективе и баз знаний и т. п. Иными словами, ЭВМ нельзя отнести только к управляющей (кибернетической) технике. В большей степени ЭВМ являются информационной техникой.

Представление о понятии информатики и ее структуре как комплексной научно-технической дисциплине будет более полным, если

рассмотреть информатику отдельно как науку и как практическую деятельность. Задача теоретической информатики включает создание и синтез знания о содержании и закономерностях информационного процесса в обществе, формирование гипотез и научных законов, разработку понятий этой научной дисциплины.

Процесс становления и развития информатики в научном и техническом аспектах можно определить как информатизацию. В техническом аспекте этот процесс проявляется прежде всего в компьютеризации. Считается, что понятие информатики является одним из наиболее адекватных и емких отражений основного содержания современного научно-технического прогресса.

Информатику как науку можно определить так: отрасль научно-технической деятельности как комплексная научно-техническая дисциплина, занимающаяся исследованием информационных процессов любой природы, разработкой на этой основе информационной техники и технологии, решением научных и инженерных проблем создания, внедрения и обеспечения эффективного использования компьютерной техники и технологии во всех сферах общественной практики.

Огромно значение информатики и в развитии самой науки и техники. Она является подлинным катализатором научно-технического прогресса. Сближение, объединение различных областей естествознания, техники, гуманитарных наук происходит сегодня в основном за счет все более широкого применения в них математического моделирования на ЭВМ и вычислительного эксперимента. Эти общенаучные методы исследований и разработок представляют собой, по сути, новую технологию науки и техники, отличающуюся гибкостью, оперативностью и универсальностью.

Важной проблемой и перспективой ближайшего будущего является внедрение методов и средств информатики в практику [12]. Связано это с развитием работ по искусственному интеллекту, с успехами в машинном представлении знаний (создание наряду с базами данных без знаний), в обеспечении диалога с ЭВМ на естественном языке (разработка языковых процессоров, систем речевого диалога), создании систем автоматического логического вывода и доказательств.

Современное развитие информатики, ее становление как новой науки обусловлено и объективно оправдано быстрым развитием принципиально новых средств социальных коммуникаций — информационно-вычислительных систем. Принципиальной новизной их является то, что здесь в отличие от других средств массовых коммуникаций (телевидения, радио) велико возмущающее воздействие социальной среды на объект проектирования — информационную технологию. Задачей информатики является, следовательно, выработка методологии, которая определяла бы условия для планомерного, научно обоснованного развития средств и методов информационной технологии. Иными словами, выработка наиболее общих принципов, методов и приемов проектирования и «встраивания» информационно-вычислительных средств в социальную среду в соответствии с закономерностями и целями развития социалистического общества, а также разработка и

применение методов оценки социальных последствий от массового внедрения новой информационной техники и технологии.

Основной задачей, стоящей перед информатикой, является не только обработка документов, но и создание, обработка и передача новых форм представлений знаний, ориентированных на решение проблем. В основе информационного обслуживания лежит не деятельность информационных органов, а взаимодействие тех, кому нужен совет (информация), и тех, кто этот совет (информацию) дает. В таких условиях каждый может создавать базы данных, доступные другим, и получать информацию без помощи информационных работников. Это является одной из главных целей создания пятого поколения вычислительных систем и новой базовой технологии для будущих вычислительных сетей, которые станут интеллектуальными сетями, предназначенными для обработки и передачи знаний. Интеллектуальные сети получают возможность автономного самоуправления. Для этого необходима развитая система сбора данных от всех элементов сети и дополнительные каналы для передачи этих данных и команд управления.

В конце 80-х — начале 90-х годов, по оценкам специалистов, в структуре производства ЭВМ, аппаратуры передачи данных, создании программных средств произойдут серьезные сдвиги. Аналоговая техника будет постепенно вытесняться цифровой, преимущества которой состоят в обеспечении высокой скорости передачи данных на расстоянии, возможности снижения влияния помех при более низких издержках производства. Тенденция к распространению цифровой техники усиливается развитием систем волоконно-оптической связи. Оптическая связь вследствие большой передающей мощности позволяет создать единую широкополосную систему цифровой связи. Эта система объединяет все компоненты информационной инфраструктуры и является общей для всех известных в настоящее время форм телекоммуникаций: телефона, телевидения, передачи данных, видеотекста и т. д.

Таким образом, перспективы интегральных цифровых сетей связи, электронной почты, персональных компьютеров и различных средств фиксации и обработки данных характеризуют становление качественно новой информационной технологии. Постепенно, по мере накопления общего содержания и расширения области применения компьютерной техники, развертывается процесс все более тесной интеграции и одновременно дифференциации отдельных направлений единой научно-технической области — информатики. Вместе с тем, развитие общенаучного понятия информации, общенаучного подхода сопровождается оформлением обобщающей науки об информации. На основе этих двух фундаментальных наук постепенно вырисовывается и прикладная наука о современной информационной технологии.

В аспекте практическом, индустриальном информатика представляет собой крупную и быстрорастущую отрасль народного хозяйства, занимающуюся его информационным обеспечением. Она включает также органы разработки, производства и обслуживания компьютерной техники. Эта индустрия обеспечивает сбор, хранение, передачу, обработку и отображение информации в интересах других отраслей хозяйства, т. е. имеет инфраструктурный характер [8].

Информатика как отрасль инженерной деятельности, особенно инженерного проектирования, также тесно связана с наукой. Большинство крупных проектов в информатике не являются типовыми, а значит, почти в каждом случае создания информационной системы требуется проводить глубокие исследования с помощью самых современных научных методов, например вычислительного эксперимента. Поэтому важно отметить, что понимание информатики только как «фундаментальной естественной науки, изучающей процессы передачи и обработки информации» [15], явно недостаточно. Практическая направленность информатики на решение различных классов задач (управленческих, научных, инженерных, производственно-технических и т. д.), а также рецептурный, методический характер многих полученных здесь решений свидетельствуют о прочной связи в информатике фундаментальных и прикладных исследований, науки и инженерии.

В то же время научный и инженерный аспекты информатики существенно отличаются от аспектов традиционной науки и инженерии. Известно, что программное обеспечение составляет до 80 % стоимости современных информационных комплексов. Иными словами, значительная часть специалистов по информатике занимается разработкой алгоритмов, программ, языков программирования. Эта специфическая продукция информатики имеет большую общественную ценность и должна получить особый юридический статус, подобный открытию в обычной науке или изобретению в традиционной инженерной деятельности.

Не менее важна в информатике проблема «стыковки» ЭВМ и строящихся на их основе информационно-технических систем с социальной средой, а также проблема органического включения (встраивания) информационной технологии в ту или иную область социальной практики — производство, управление, науку, медицину и т. д.

1.2. ПРИКЛАДНАЯ ИНФОРМАТИКА И НОВАЯ ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

Задача прикладной информатики заключается в исследовании некоторых предположений о современном этапе развития прикладной информатики, о перестройке приемов и методов практической информационной технологии в ответ на запросы практики.

Под прикладной информатикой здесь понимается практическое применение закономерностей проектирования, создания, эксплуатации и развития технологических систем обработки данных. В задачу прикладной информатики входит создание комплекса методов разработки и внедрения технологии конкретных информационных систем, т. е. конкретных информационных технологий.

Предмета прикладной информатики еще не существует (как не сформировался еще в полной мере и блок дисциплин, составляющих в целом информатику). В то же время специалисты занимаются созданием конкретных информационных технологий (в АСУ, САПР, АСНИ и т. д.), в результате деятельности которых постепенно формируется идеология прикладной информатики, ее своеобразная методология.

Внедряя методы и элементы информационной технологии в различные управляющие и информационные системы, эти специалисты вынуждены соответственно перестраивать приемы, методологические принципы, способы рассуждений и умозаключений. Особенно интенсивным процессом методологической перестройки в области информационной технологии стал в 80-х годах.

Особенности методологии и идеологии прикладной информатики (безбумажной информатики) описаны в книгах В. М. Глушкова [6, 16]. В [6] наиболее полно проанализированы важнейшие черты НИТ.

НИТ находится в самой начальной фазе своего развития, но ее значение уже сейчас далеко выходит за рамки понимания ее только как нового стиля использования ЭВМ конечным пользователем или как технологии прохождения задач в вычислительном центре. Принципиальное значение НИТ состоит в замене машинно-бумажного процесса обработки данных на безбумажный, в котором не только не используются промежуточные носители данных, но и резко снижается объем фиксации данных на обычных документах. В НИТ впервые наблюдается феноменальное явление — процессы обработки информации отделены от процессов переноса массы (только при обмене данными между человеком и машиной могут использоваться, но не обязательно используются, механические перемещения устройств).

Как область научно-технического прогресса, охватывающая в основном многочисленные применения компьютерной и телекоммуникационной технологии практически во всех сферах организационного управления, НИТ быстро развивается. Ее революционное значение заключается в кардинальной перестройке и ускорении процессов создания новой техники и реализации новых технологий. Ярким примером этому являются использование САПР и АСНИ, применение которых позволяет ускорить процесс трансформации новых научных знаний в конкретную технологию.

Существенные части НИТ и наиболее основные области ее действия представлены на рис. 1.2.1. НИТ объединяет новые технологии коммуникаций на основе локальных и распределенных сетей ЭВМ, обработки управленческой информации на основе ПЭВМ и АРМ, а также выработки управленческих решений на основе средств искусственного интеллекта (базы знаний, экспертные системы, различные системы моделирования, предусматривающие разнообразные — графические, звуковые, текстовые — формы отображения моделируемых ситуаций) (рис. 1.2.1).

НИТ — это пограничная область, охватывающая как вычислительную технологию, так и конкретную социальную информационно-коммуникативную практику [11], рационализирующую ее за счет широкого применения дешевой и малогабаритной информационно-вычислительной техники.

Наиболее ярко эффективность применения НИТ проявляется в двух нижних областях управления: автоматизация проектирования, оперативного планирования и управления промышленным производством — системы САПР, АСУ, АСНИ, ГАП и т. д.; автоматизация организационного управления (учрежденческой деятельности в

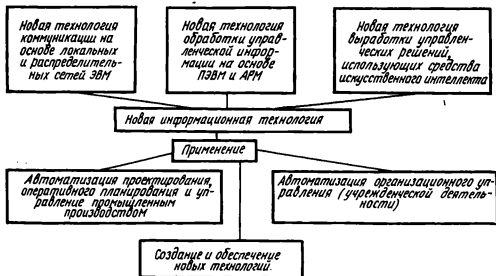


Рис. 1.2.1

самых различных ее аспектах) — текстовые системы, электронная и речевая почта, системы ведения баз данных и т. д. При этом под автоматизированным учреждением понимается учреждение, в котором средства организационной техники, обработки данных и оборудование связи используются для интенсификации информационной деятельности и улучшения условий труда работников учреждения.

Практическое использование НИТ первоначально нашло отражение во внедрении робототехники, а затем в создании на ее основе ГАП. На НИТ основаны системы комплексной автоматизации предприятий — КАСУ. Эти системы имеют интегральный характер и включают в качестве основных элементов управляющие ЭВМ, робототехнику, организованную по принципам ГАП, системы автоматизированного проектирования и планирования.

По оценкам ученых и специалистов, НИТ является в настоящее время одним из основных средств поддержки, создания и обеспечения принципиально новых технологий — электронно-лучевой, плазменной, импульсной, биологической, радиационной, мембранной, химической и др.

Таким образом, чрезвычайно важной и актуальной становится задача разработки стратегии развития НИТ и ее взаимодействия с производством, управлением, наукой, другими сферами общественной практики.

В настоящее время в нашей стране проводится активный анализ этих взаимодействий, осуществляется разработка теории и практики НИТ. С позиций оценки влияния НИТ на совершенствование управления важно отметить, что отечественный и зарубежный опыт развития НИТ как новой области научно-технического прогресса показал, что эту сферу характеризует ряд особенностей, существенно отличающих ее от других направлений науки и техники: динамичность (технология использования и поколения многих технических и программных средств изменяются два раза за 5 лет); постоянно возрастающий уро-

нень технической сложности составляющих НИТ компонентов, что вызывает необходимость постоянного повышения квалификации разработчиков и пользователей информационных систем; глубокое и долговременное влияние на развитие производительных сил и производственных отношений; высокая степень потенциальной эффективности при выполнении следующих условий: стандартизации, масштабности охвата инфраструктуры народного хозяйства, своевременного организационного обеспечения внедрения новых средств и методов НИТ.

Эти особенности НИТ во многом предопределяют специфические черты ее взаимодействия с производством, наукой, социальной сферой и требуют постоянной оценки правильности стратегии ее развития, выработки научно обоснованных подходов к созданию и внедрению новых средств и методов НИТ. НИТ — это совокупность очень многих форм, методов, навыков применения всего многообразия вычислительной техники и средств связи в области сбора, обработки, хранения и передачи информации. Вопросы о содержании и объеме понятий НИТ пока еще во многом дискуссионны и нуждаются в обстоятельном обсуждении. Поэтому дать точное определение НИТ пока невозможно.

Однако для практической работы необходимо определить понятие НИТ, которое бы ориентировало ее на практические результаты, выражало суть и было понятно широкому кругу специалистов. Учитывая эти требования, на основе общепринятого понятия информационной технологии как процесса сбора, передачи, хранения и обработки информации во всех ее возможных формах (текстовой, графической, визуальной, речевой) авторы в данном случае используют в качестве рабочего следующее понятие НИТ. *Новая информационная технология — совокупность внедряемых (встраиваемых) в системы организационного управления принципиально новых средств и методов обработки данных, представляющих собой целостные технологические системы и обеспечивающих целенаправленное создание, передачу, хранение и отображение информационного продукта (данных, идей, знаний) с минимальными затратами и в соответствии с закономерностями той социальной среды, где развивается НИТ.*

Можно предположить, что НИТ в теоретическом плане является прикладной наукой, а в практическом — инженерной деятельностью по проектированию и созданию конкретных технологических систем обработки данных. В НИТ используются достижения системотехники, теории вычислительных систем, технологии программирования, эргономики, дизайна и других прикладных наук информационно-технического профиля. Суть НИТ во многом определяется возможностями новых технологических моделей обработки данных (терминальных комплексов, локальных и распределенных сетей ЭВМ), создаваемых на основе новых элементов информационной технологии. К последним следует отнести ПЭВМ, интеллектуальные терминалы, локальные сети, системы речевого диалога, языки конечного пользователя, системы машинной графики, читающие автоматы, а также другие многочисленные новые средства организационной техники и коммуникации.

Каково бы, что на основе этих базовых элементов можно достаточно просто осуществлять проектирование и создание эффективных

конкретных информационных технологий для организационных АСУ, максимально отвечающих практическим потребностям. Однако все значительно сложнее. Дело в том, что если для нижних звеньев управления и обработки данных в учреждении сравнительно легко определить и классифицировать простые информационные технологии (текстовые системы, средства контроля сроков получения и обработки документов, средства ведения баз данных и учетной информации и т. д.), то типы и структуры информационно-управленческих процедур для высшего и среднего звена (где осуществляется выработка и принятие управленческих решений и где автоматизация приносит наиболее желаемый эффект) исследованы в значительно меньшей степени.

Анализ наблюдаемых в результате автоматизации эффектов показывает, что если на верхних уровнях структуры управления вследствие автоматизации происходит кардинальная «ломка» установившейся технологии управления, то, как правило, средства автоматизации либо используются нерационально, либо отвергаются. Главная причина этих отрицательных явлений заключается в отсутствии научно обоснованных рекомендаций по автоматизации рабочих мест управления высшего и среднего звена, а также методик применения вычислительной техники в конкретных управленческих технологиях. В настоящее время информационная технология проектируется и создается только интуитивно. До сих пор при создании сложных вычислительных систем почти никогда нельзя достаточно утвердительно ответить на такие вопросы: правильно ли мы проектируем технологический процесс обработки данных; достигнут ли требуемый уровень соответствия информационного аппарата реальной технологии управления; эффективно ли используются вычислительные системы; правильно ли определены режимы их использования; соответствует ли практическим потребностям степень централизации или децентрализации обработки данных и многие другие.

Далее, очень важно учитывать, что объектом исследования в информационной технологии является не традиционный инженерный (технические и программные средства), а качественно новый, «деятельный», основа которого — взаимодействие в системе человек — ЭВМ — социальная среда. Речь идет о создании и преобразовании моделей человеко-машинных систем, в которых деятельность по использованию, созданию и совершенствованию как бы сливается воедино и неразрывно взаимосвязана. Поэтому и объект проектирования информационной технологии должен как бы постепенно вписываться в социальную среду (реальную технологию управления). Следовательно, нужно говорить уже не о проектировании, а о развитии (совершенствовании) объекта и постепенном подведении его к закладываемому в проекте состоянию.

Используемые в настоящее время методы проектирования технологии обработки данных для удовлетворения регламентированных разовых информационных запросов уже не соответствуют требованиям ограниченного встраивания средств и методов информационной технологии в реальную технологию организационного управления. В условиях широкого внедрения ПЭВМ, локальных сетей, новых систем связи,

развитых средств организационной техники возникает объективная необходимость в разработке новых подходов к проектированию, созданию и эксплуатации технологических систем обработки данных.

В настоящее время особое внимание уделяется созданию организационных АСУ (по другой терминологии — административных АСУ). Они наиболее многочисленны среди АСУ иных типов — производственных, технологических, территориальных, отраслевых. Повышенное внимание к ним обусловлено не только тем, что с появлением персональных ЭВМ и локальных сетей создались новые возможности улучшения экономических, технологических, психологических и других характеристик систем автоматизации документооборота в учреждении, но и потому, что современные средства информатики заставляют специалистов резко обновлять багаж знаний и концепции проектирования, создания и эксплуатации информационной службы учреждения, от внедрения которых ожидаются не просто улучшения и качественные сдвиги, а радикальные изменения в технологии обработки данных и способах взаимодействия конечного пользователя (управленца, организатора и т. д.) с АСОД.

Традиционно считалось, что в сфере управления четыре объекта: трудовые, финансовые, материальные ресурсы и оборудование. Сейчас к ним прибавился пятый — информационный ресурс. В обычной информационной технологии этот ресурс не централизован, а разбросан между управленческими работниками. Основной способ его получения — производственные совещания, беседы, телефонные разговоры и т. д. (т. е. сбор и учет этого ресурса трудоемкий, требующий больших затрат времени управленца). НИТ позволяет кардинально улучшить процесс контроля за использованием информационного ресурса, сделать этот процесс более надежным и оперативным, менее трудоемким.

В настоящее время информационная технология обрела три наиболее характерные функции: *персонализация вычислений на основе ПЭВМ и систем интеллектуального интерфейса конечного пользователя с ПЭВМ; базы данных и базы знаний; локальные сети передачи данных.* Эти функции реализуются посредством создания специализированных и универсальных информационных комплексов. В информационной технологии предмет теоретического анализа во многом связан с поиском стратегий выбора путей синтеза структур новых информационных комплексов. Опыт показывает, что эволюция информационной технологии (и традиционной, и НИТ) представляет собой процесс непрерывного поиска новых форм улучшения и усиления ее коммуникативных свойств. При этом постоянно действует один из фундаментальных законов развития технологии ИВС: каждая элементная база (каждая структура ИВС) не только обусловлена общественной потребностью, но и ставит свои жесткие пределы сложности создаваемых на ее базе автоматизированных систем. Например, невозможно обеспечить на терминальном комплексе (эффективном в справочных системах) того многообразия информационных услуг, которые требуются в организационном управлении. Для создания больших систем необходимы и новые системные принципы проектирования информационной технологии.

Для того чтобы методы НИТ стали полноценным инструментом, они должны быть организованы в целостные технологические системы, охватывающие полностью информационное производство со всеми основными и вспомогательными процессами. Нужно не одностороннее применение новых методов обработки информации, а взаимообусловленное и сбалансированное по отдельным участкам технологического процесса обработки данных (т. е. сбалансированное по мощности ЭВМ, пропускной способности каналов связи, производительности устройств подготовки входных данных и вывода результатов и других параметров). Поэтому и прикладная информатика, проникая в новые области применения, постоянно перестраивается. Происходит выработка более гибкой тактики, изменяется стратегия проектирования, создания и внедрения информационных систем. Естественно, что этот процесс перестройки должен сопровождаться своевременными теоретическими обобщениями, методологическими выводами, установлением новых терминов и понятий.

Возникает вопрос: какие черты характерны для современной рабочей, прикладной информатики? Прежде всего это изменившаяся методология, новый набор средств, новая схема подхода к проектированию, созданию, внедрению и эксплуатации информационных систем.

Традиционная, классическая схема создания автоматизированных информационных систем была примерно такой: вначале (на стадиях предпроектного обследования и технического проектирования) разрабатывалась четкая постановка задачи, специфицировались входные и выходные документы, форма и порядок запросов пользователя, алгоритмы решения задач. Затем все это длительное время (на стадии рабочего проектирования) реализовывалось в виде программ, отлаживалось, «подгснялось» под требования практики. В итоге при подобном подходе обнаруживалось (на стадии опытной эксплуатации и внедрения), что от начальных постановок задач почти ничего не остается и начинается этап перепроектирования АСУ. Это явилось причиной скептического отношения к автоматизации управленческого труда.

Подобная схема автоматизации управленческого труда, разделяющая заказчика и исполнителя, быстро устаревает. Для современной прикладной информатики типично другое: совмещение в одном лице ставящего задачу, решающего ее и использующего результаты ее решения. Идея эта проста, но исполнение ее — процесс сложный. Нужен очень высокий класс проектирования систем, развитый инструментарий, повышение технологического уровня информационных систем. Средством и инструментарием для реализации такого подхода являются персональные ЭВМ, средства интеллектуального интерфейса пользователя с системой, средства автоматизации составления прикладных программ.

Известно, что исходная концепция АСУ создавалась в середине 60 — начале 70-х годов. В этот период основную задачу разработчики АСУ видели в создании комплекса программ, реализующих некоторый набор функций управления, в проектировании необходимой для этих программ части информационного обеспечения и комплекса технических средств. Технология использования получаемой с ЭВМ ин-

формации, методы принятия решений на новой научно-технической основе, возможная реконструкция аппарата управления и т. д. остались вне рамок разработки. Иными словами, проектировалась не система управления, а ее отдельные элементы.

В 80-е годы начался период системной интеграции АСУ. Его отличительная черта — появление и широкое использование в АСУ элементов и методов НИТ.

Таким образом, широкое применение современных средств информатики в сфере организационного управления сопровождается изменением многих первоначальных концепций и представлений о формах, методах, организации проектирования и внедрения автоматизированных систем управления и взаимодействия с ними управленческих работников. Служившие много лет классические термины — вычислительный центр, информационно-вычислительный центр, служба приема и выдачи задач, служба сопровождения архива магнитных носителей и т. д. — в настоящее время все более уступают новым понятиям — персональная ЭВМ, локальная сеть, центр обработки данных, коммуникационный центр и др. Это отражает не только смену технической базы автоматизации, но и смену концепции автоматизации, в основу которой теперь положены следующие принципы: персонализация вычислений, встраиваемость элементов информационной технологии в реальную технологию управления, «дружественный» интерфейс конечного пользователя с вычислительной системой.

Массовое производство и внедрение ПЭВМ, а также развитие средств передачи данных, создание локальных и распределенных сетей ЭВМ обеспечивают возможность подведения информационного потенциала непосредственно к рабочему месту организатора производства, управления, администратора — это и является НИТ. Одна из ее отличительных черт — возможность практически неограниченного расширения числа задач, решаемых с помощью автоматизированных информационных систем. Причем расширение спектра решаемых задач осуществляется непосредственно конечным пользователем, т. е. без посредников (системных аналитиков, программистов, операторов). Достигается это за счет развитых средств диалогового общения конечного пользователя с ЭВМ, высокой информационной культуры современной технологии обработки данных, постоянного самообучения конечных пользователей.

Таким образом, происходит переход от концепции ВЦ (с централизованной обработкой управленческой информации) к концепции децентрализованной сети АРМ. При этом предполагается, что совместное развитие и сращивание вычислительной и коммуникационной техники обеспечат работникам учреждения непосредственный доступ к нужной информации и вычислительным мощностям.

Принципы создания новых типов АСУ (в частности, административных) не являются совершенно новыми. Скорее можно говорить о новой жизни принципов создания АСУ, сформулированных В. М. Глушковым в работе [16] и затем развитых им в [6]. Прежде всего о принципах, интегрирующих основные требования к системам организационного управления и информационной технологии: созда-

для локальных (вторичных) баз данных, формируемых и актуализируемых из исходной (первичной) интегральной базы данных; создания управленческих АРМов, оснащенных более гибкими, чем раньше, общесистемными средствами, позволяющими быстро настраивать локальные базы данных на индивидуальные требования пользователей; динамичной целостности информационной модели объекта — такой принцип предусматривает постоянное автоматическое развитие базы данных о состоянии объекта (от настоящего к будущему) посредством автоматической инициации процедур внесения изменений во все взаимосвязанные локальные базы данных (если изменения вносятся хотя бы в одну из них).

Правильность этих принципов подтверждает анализ развития сетей ЭВМ, показывающий, что одной из кардинальных тенденций в развитии технологии обработки данных в системе организационного управления является, очевидно, персонализация вычислений. В этом случае массовому пользователю предоставляется возможность проводить обработку данных таким образом, что он осуществляет прямой и личный контроль над всеми стадиями обработки (т. е. контроль над управляемым процессом, не достижимый без применения ЭВМ), но в то же время обладает независимостью, которой он лишен, взаимодействуя с ЭВМ непосредственно без персональной технологической системы. В связи с этим особенно актуально создание интерфейсных средств, суть которых заключается в максимальном избавлении массового пользователя от необходимости «пробиться» к вычислительным ресурсам через «частокол» языков управления заданиями в конкретных операционных системах. Как следствие, изменяются и взгляды на оценку эффективности функционирования вычислительных систем. В настоящее время эта оценка определяется не только исходя из начальной стоимости программно-технических средств, но и их технических возможностей: простоты эксплуатации, экономичности и надежности технического обслуживания, быстрой адаптации к конкретным условиям применения.

1.3. МЕТОДЫ НОВОЙ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ В ЗАДАЧАХ УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ ТРАНСПОРТОМ

Что же может дать прикладная информатика, в частности НИТ, промышленному транспорту? Каким образом с помощью ее методов можно повысить эффективность принимаемых решений в автоматизированных системах управления промышленным транспортом? Авторы не ставили цель проводить скрупулезный анализ специфики работы промышленного транспорта и соответствующих задач планирования и управления, тем более что это уже было сделано вполне тщательно и всесторонне в публикациях [17, 109]. Поэтому, рассматривая в этой книге промышленный транспорт как основную область применения методов НИТ для автоматизации процессов обработки данных и управления, отметим в первую очередь лишь такие две его основные принципиальные особенности, как тесная взаимосвязь с обслуживаемым основным производством, особенно в таких отраслях, как металлургия,

машиностроение, горнодобывающая и угольная промышленность, а также необходимость четкого взаимодействия с железнодорожным транспортом общего пользования. Оставляя в стороне рассуждения и анализ конкретных задач управления работой промышленного транспорта (некоторые из них будут рассмотрены в следующих главах) и методов их решения, освещенных достаточно полно в многочисленной литературе, выделим лишь главные отличительные черты этих задач, а именно — многокритериальность и необходимость принятия согласованных решений. Эти черты являются прямым следствием отмеченной специфики промышленного транспорта и характерны в большей или меньшей мере практически всем его реальным объектам и задачам. Вместе с тем наличие этих черт можно рассматривать как определяющее свидетельство в пользу применения НИТ, так как только ее методы и средства дают сегодня наиболее эффективный аппарат решения таких задач. Покажем это отдельно как по фактору многокритериальности, так и по фактору принятия согласованных решений.

Говоря о многокритериальности, следует отметить, что в идеальной постановке задача оптимизации сводится к поиску решения, обращающего в максимум (минимум) единственную скалярную величину (целевую функцию). Но идеальная схема крайне редко встречается в реальных задачах. Более того, практически все реальные задачи являются задачами нелинейными, многокритериальными, т. е. с векторной целевой функцией. Одни из критериев должны быть обращены в максимум, другие, наоборот, в минимум. Эти требования, как правило, взаимно противоречивы: не существует решения, удовлетворяющего всем сразу. Попытки объединить несколько критериев в один обобщенный и оптимизировать по нему решение обычно не дают должного эффекта. Они даже могут быть вредны, так как создают иллюзию научного обоснования там, где его, по существу, нет. В таких случаях приходится, как при согласовании разных точек зрения, искать форму разумного компромисса. Например, между требованиями пользователя, возможностями разработчика, характеристиками программно-технических средств, их наличием, стоимостью оборудования и т. д.

Существующие методы многокритериальной оптимизации предоставляют пользователю некоторое множество допустимых вариантов решения, или так называемое множество эффективных альтернатив [11], при которых значения каждого из критериев не выходят за пределы установленных для него минимальных и максимальных границ. Для нахождения единственно оптимального решения в математике пока не существует полноценной теории компромисса. Поэтому практически единственной инстанцией, способной быстро и успешно выработать компромиссное решение, является человеческий разум, так называемый здравый смысл. Математика, не будучи пока готовой его заменить, должна, тем не менее, ему помогать, т. е., в процессе решения сложной многокритериальной задачи математические расчеты должны прищипываться на здравый смысл, а здравый смысл подкрепляться математическими расчетами. Это и является сутью диалога человека с НИТ. Без участия специалиста в решении многокритериальной задачи приемлемое (по совокупности нескольких критериев) решение пока

что не может быть выбрано. Современные средства информатики, прежде всего диалоговые системы, являются мощным инструментом для решения многокритериальных оптимизационных задач. Диалоговые системы, используя математические модели, позволяют в задаче оперировать специалисту такими понятиями: приемлемо, практически равноценно и др., характерными для человеческого мышления.

Осуществляя в диалоге с ЭВМ выбор решения, человек, не вдаваясь в излишние подробности, окидывает общим взглядом ситуацию и выбирает приемлемый вариант. Что касается математических методов, то их задачи в подобных случаях заключаются не в том, чтобы выдать окончательное решение, а в том, чтобы помочь человеку это решение выбрать. Задача прикладной информатики состоит в том, чтобы дать человеку, принимающему решение, максимум нужной (релевантной) ему информации в наиболее выразительной, удобовоспринимаемой форме. Такое математическое моделирование с использованием средств информатики часто может заменить недостающий человеку опыт (когда речь идет о ситуациях новых, не изученных или о мероприятиях, опыта проведения которых нет) и быть особенно эффективным с привлечением новых средств информатики: баз знаний, систем логического вывода, графического представления информации и др.

В частности, с применением баз знаний станет возможной передача опыта от человека (или коллектива), искусного в выборе решений, непосредственно ЭВМ, которая будет способна постепенно вырабатывать формализованный алгоритм выбора решения (так называемые адаптивные, или обучаемые, алгоритмы). К созданию таких алгоритмов могут быть привлечены любые средства (например, экспертные оценки, механизмы голосования и т. д.) [18], весьма далекие от математической традиции. Широкое и продуктивное практическое использование этих алгоритмов немыслимо без средств современной прикладной информатики.

Таким образом, персонализация вычислений как один из основных элементов НИТ позволяет пользователю без посредников контролировать процесс решения задачи, имея возможность при необходимости воздействовать на него посредством диалога с ЭВМ. Возможность такого воздействия позволяет оперативно корректировать технологическую схему решения задачи в зависимости от промежуточных результатов и получать окончательный вариант решения, не требующий переработки.

Необходимость принятия согласованных решений обусловливается тем, что практически все реальные системы организационного управления чаще всего представляют собой сложные иерархические комплексы, состоящие из управляющего центра и активных элементов нижнего уровня. Для промышленного транспорта такую структуру можно интерпретировать как, например, множество транспортных цехов промышленных предприятий и транспортное управление соответствующего министерства, или как множество предприятий промышленного железнодорожного транспорта и вышестоящее региональное производственное объединение и т. п. В любом случае имеет место система, в которой центр обладает определенным приоритетом действий, интересы

центра и отдельных элементов часто не совпадают, элементы располагают определенной свободой (самостоятельностью) в достижении цели и т. д. В такой системе необходимо увязать и согласовать требования оптимальности по каждому уровню и элементу с реальными интересами всей системы в целом. Теория активных систем [19] и предложенные Г. С. Поспеловым модели [74] создают теоретические и методологические предпосылки успешного решения задач такого класса. Однако практическая их реализация сталкивается с трудностями четкого обеспечения как организационного, так и информационного взаимодействия специалистов нижних уровней и центра при решении общих задач планирования и управления. Организация такого взаимодействия весьма сложна хотя бы потому, что в этом случае вместо совокупности не связанных между собой моносистем, призванных решать отдельные задачи, должна быть создана некоторая интегрированная мульти-система. Но такая мультисистема должна быть создана, ибо опыт применения различных моделей принятия решений показывает, что большинство сложных решений являются результатом организационного диалога (т. е. многочисленных обменов мнениями, встреч, переговоров, выяснения позиций и т. п.), в ходе которого управленческое решение представляется как некоторый консенсус достаточно сложного коммуникативного процесса. При этом большая часть времени затрачивается не на непосредственное решение какой-то задачи, а именно на итерационные процессы согласования этого решения. В этих условиях единственным средством, на базе которого может быть реализована автоматизированная процедура принятия оптимального согласованного решения, является включение ЭВМ в коммуникационные отношения внутри коллектива в процессе принятия решений. Здесь НИТ проявляется практически в создании локальной сети ЭВМ, в рамках которой группа взаимодействующих или даже конкурирующих пользователей совместно вырабатывает приемлемое компромиссное решение, используя как формальные методы и процедуры, так и неформальные правила принятия решений, основанные на знаниях экспертов или директивных указаниях отдельных лиц.

И, наконец, как будет показано ниже, предусматриваемое НИТ использование знаний о технологии решения функциональных задач создает хорошие предпосылки для создания интеллектуальной надстройки над совокупностью используемых экономико-математических и информационных методов для управления решением сложных комплексов взаимосвязанных задач планирования работы промышленного транспорта.

БАЗЫ ДАННЫХ И ОСНОВНЫЕ НАПРАВЛЕНИЯ ИХ РАЗВИТИЯ

2.1. ДАННЫЕ, МОДЕЛИ ДАННЫХ, БАЗЫ ДАННЫХ

Актуальность проблематики информационного моделирования предельно ясна: в настоящее время базы данных стали основой практически каждой системы обработки данных, использующей ЭВМ. Многообразие сфер применения баз данных диктует в каждом конкретном случае свои специфические требования к моделям данных, но главной задачей их построения всегда является максимально адекватное отображение реального мира и происходящих в нем процессов.

Основным фактором, определяющим эффективность системы информационного обеспечения, является степень семантической интерпретации данных. Можно сказать, что вся история развития систем электронной обработки данных — это история развития семантики данных. В начале машинной обработки данные были обезличены, представляя собой простую совокупность битовых значений, соответствующих некоторым числам или алфавитным символам. Первым шагом на пути развития семантики данных стало присвоение имен отдельным величинам, появившееся в машинных и более высокоуровневых языках программирования. Переменная, константа, элемент структуры или элемент массива — это уже не обезличенные, а поименованные отдельные элементы данных, а структуры и массивы в целом — поименованные совокупности с указанием определенных связей как между отдельными элементами данных, так и между их подмножествами. Увеличение смысловой содержательности данных обеспечивало более глубокую их семантическую интерпретацию, однако в языках программирования она не выходила, как правило, за рамки программы, использующей данные.

Последующее развитие семантики — описание сложных структур данных и их взаимосвязей, отделение описаний данных от программ их обработки и программ от данных (независимость данных и программ), разработка специальных языков для описания данных и их обработки — привело к созданию систем баз данных, в основе которых лежат модели данных, обеспечивающие глубокую и всестороннюю интерпретацию данных. Приведем кратко основные понятия моделей данных в соответствии с [19].

Модель данных — это средство абстракции, обеспечивающее представление и интерпретацию данных. Из такого определения ясно, что модель включает сами данные и их описания. Общепринятым считается представление элементарной единицы данных в модели в виде тройки

(имя объекта, свойство объекта, значение свойства). Это представление может быть реализовано различными способами, что привело к созданию множества моделей данных. Важнейшим свойством модели данных является также наличие процедур, ориентированных на обработку используемых в модели структур данных, т. е. обычно операции соотносятся со структурами данных. Исходя из этого можно дать определение модели данных M как множества правил порождения G , в соответствии с которыми описываются данные, и множества операций O , определяющих обработку данных.

Правила порождения G , соотносимые с ЯОД, специфицируют допустимые структуры данных как множество типов объектов и связей между ними (подмножество правил G_s), а также допустимые реализации путем указания для каждого типа явных ограничений целостности на значения его атрибутов (подмножество правил G_c). Совокупность поименованных типов свойств и связей между ними называется схемой S . Теперь можно определить базу данных как совокупность данных, структура которых соответствует конкретной схеме. В соответствии с выделением в G подмножеств G_s и G_c схема S базы данных будет состоять из двух частей: спецификации структуры S_s и спецификации явных ограничений целостности S_c . Кроме явных ограничений в модели данных могут поддерживаться и неявные, или внутренние, ограничения целостности, накладываемые по определению и отображаемые непосредственно в структурной части схемы S_s . В первую очередь это ограничения на связи, определяемые, например, иерархическими и сетевыми структурами. Таким образом, с помощью правил G можно задать произвольное множество схем S базы данных, специфицируя их структуры и ограничения целостности. Каждой схеме может соответствовать множество конкретных реализаций базы данных D — множество структурно идентичных совокупностей данных, имеющих разные, но не выходящие за пределы установленных ограничений целостности значения. Каждую реализацию можно рассматривать как отдельное состояние базы данных D_i .

Множество операций O , соотносимое с ЯМД, включает операции, определяющие допустимые действия над реализацией базы данных D_i при преобразовании ее в другую реализацию D_j . В результате база данных переводится из одного состояния в другое в рамках неизменной структуры. При выполнении операций, как правило, не допускаются нарушения внутренних ограничений целостности. Вместе с тем в некоторых системах привилегированным пользователям могут предоставляться так называемые операции принудительного обновления, допускающие нарушения явных ограничений целостности при реализации специфических процессов актуализации данных.

Управление базой данных осуществляется СУБД, поддерживающей средства определения схем данных и выполнение операций над данными. Обычно СУБД ориентированы на конкретную модель данных и обеспечивают эффективное интегрированное хранение и обработку больших объемов сложноструктурированных данных.

Модели данных. Известно достаточно много различных моделей данных, наиболее полно рассмотренных в первую очередь в [19], а так-

же в [20, 21]. Однако на практике при разработке баз данных широкое распространение получили лишь три известные основные модели — иерархическая, сетевая и реляционная. Наиболее перспективной безоговорочно признается реляционная, хотя большинство коммерческих систем все еще базируется на одной из двух других моделей. Однако есть все основания полагать, что с дальнейшим развитием средств вычислительной техники, особенно созданием машин баз данных, реляционная модель станет доминирующей не только в теории, но и на практике. Именно поэтому предлагаемые в гл. 3 модели данных для систем управления транспортом являются также реляционными.

Предложенная впервые Коддом в [22] базовая реляционная модель быстро завоевала всеобщее признание благодаря простоте и ясности структур данных, ограничений целостности и процедур манипулирования данными, основанных на строгих теоретико-множественных концепциях. Сама базовая реляционная модель настолько детально и многократно описана и проанализирована практически во всех монографиях, посвященных моделированию данных, что нет никакой необходимости делать это еще раз. Поэтому при возникновении у читателя вопросов по поводу отдельных определений, формальных обозначений и терминологии вообще рекомендуем ему обратиться к монографиям [19—21]. Здесь же отметим только, что упомянутая работа Кодда [22] оказала огромное влияние как на теорию баз данных в целом, так и на проблематику реляционного моделирования в частности.

Однако в ходе исследований выявились и некоторые слабости базовой реляционной модели, и в первую очередь недостаточность средств выражения ограничений целостности. Напомним, что в базовой модели могут задаваться только явные ограничения на значения отдельных атрибутов и их совокупностей (агрегатные ограничения целостности [19]), но нет неявных ограничений, специфицирующих связи между сущностями (как в иерархии, например). Отсутствие же этих ограничений делает возможными реализацию бессмысленных операций над отношениями, в частности, таких, как JOIN, а также возникновение «ловушек соединения» [20]. В результате Коддом была предложена расширенная реляционная модель данных, получившая название *RM/T* [23]. Новая работа Кодда оказалась во многом не менее основополагающей, хотя почему-то менее замеченной (по крайней мере цитируемой). Последнее обстоятельство, а также использование ряда результатов при разработке моделей данных для систем управления транспортом дают повод более подробно рассмотреть некоторые основные положения этой работы.

Важнейшим расширением *RM/T* признается введение внешнего ключа, называемого заменителем, который (это подчеркивается особо) однозначно идентифицирует сущность, а не кортеж. Фактически это индекс, автоматически формируемый системой при вводе данных о новой реализации сущности и поддерживаемый на протяжении всего времени ее нахождения в базе данных. Для каждого типа сущности имеется свое унарное *E*-отношение, содержащее внешние ключи реализаций этого типа. Сами же свойства (атрибуты) каждого типа сущности содержатся в одном или нескольких (вплоть до бинарных) *P*-

отношениях RM/T . Наличие в каждом P -отношении внешнего ключа (атрибута E -отношения) связывает свойства реализации сущности каждого типа с утверждением о его существовании в базе данных. Очевидно, что с помощью операции соединения, использующей внешний ключ E -отношения, можно получить полный набор свойств соответствующей реализации сущности как множество сцепленных кортежей нескольких P -отношений.

Служащие	Номер и фамилия		День рождения				
	Внешний ключ	Номер	Фамилия	Внешний ключ	Год	Месяц	Число
α (E -отношение)	α (P -отношение)	777	Петров	α (P -отношение)	1950	11	25

В RM/T введено семейство графовых отношений, специфицирующих имеющие смысл связи типов, иерархии обобщения, временную упорядоченность событий и другие семантические ограничения. Тем самым графовые отношения обеспечивают достоверность результатов реляционных операций, не позволяя, как в обычной реляционной модели, связывать различные типы, базируясь исключительно на значениях атрибутов.

Первая группа графовых отношений предназначена для спецификации связей. PG -отношение характеризует связь типа сущности с его атрибутами (т. е. связь E -отношений с P -отношениями). Если атрибуты одного типа сущности разбросаны по нескольким бинарным P -отношениям, то PG -отношение перечисляет для этого типа сущности все P -отношения, содержащие атрибуты этого типа сущности. AG -отношение описывает типы сущностей, участвующих в формировании данного типа связи. Например, для модели «поставщики — детали — поставки» (S, P, SP) AG имеет вид

$$AG(n \ m)$$

$$sp \ s$$

$$sp \ p,$$

т. е. показывает, что SP связывает типы сущностей S и P . CG -отношение описывает связь так называемых характеристических типов и других типов сущностей, от существования реализаций которых зависит существование характеристических. Например, если существование SP зависит от существования S и P , то CG имеет вид

$$CG(n \ m)$$

$$sp \ s$$

$$sp \ p.$$

RM . AG - и CG -отношения — это бинарные отношения, атрибуты которых определены на домене RN — множестве всех имен отношений в базе данных. Первый атрибут имеет статус SUP (старший, главный), второй — SUB (подчиненный).

В RM/T используются две известные абстракции типов сущностей — обобщение и агрегация. Два бинарных отношения описыва-

ют иерархию обобщения по включению, т. е. определяют связь между типом сущности, обобщающим его типом и категорией обобщения. *UGI*-отношение описывает безусловное обобщение по включению, когда некоторый подтип обобщается только одним из типов сущностей, а *AGI*-отношение определяет альтернативное обобщение, когда один и тот же подтип может быть обобщен несколькими типами сущностей. Первые два атрибута в *UGI* и *AGI* определены на домене *RN* (первый в роли *SUP*, второй в роли *SUB*), а третий специфицирует категорию обобщения (безусловная или альтернативная).

Кроме обобщений, в *RM/T* используется тип агрегата-покрытия. Каждая реализация этого типа есть некоторая совокупность реализаций других типов. В качестве примера типа агрегата-покрытия рассматривается конвой кораблей, реализация которого (конкретный состав) состоит из реализаций других типов (различных классов судов). Бинарное *KG*-отношение, определенное также на *RN*, показывает, какого типа сущности могут входить в агрегацию покрываемого типа.

Важным принципиальным новшеством *RM/T* являются сущности типа события. Они содержат атрибут времени, характеризующий время реализации сущности или срок ее достоверности. Четыре бинарных графовых отношения определяют допустимый порядок следования событий, а именно указывают, какие типы событий должны или могут предшествовать событиям другого типа или следовать за ними. Отношение безусловного следования *US* ($s_1 s_2$) указывает, что событие s_2 следует только за s_1 . Отношение альтернативного следования

$$AS(n \ m)$$

$$s_1 \ s_2$$

$$s_4 \ s_2$$

показывает, что событие s_2 может следовать за s_1 или s_4 . Отношение безусловного предшествования *UP* ($s_1 s_5$) говорит о том, что событию s_5 предшествует только s_1 , а отношение альтернативного предшествования

$$AP(n \ m)$$

$$s_1 \ s_5$$

$$s_2 \ s_5$$

свидетельствует, что s_5 могут предшествовать s_1 или s_2 .

В *RM/T* предусмотрена возможность обработки неопределенных значений свойств типов сущностей. Причем из двух известных трактовок неопределенности — «значение не определено в данный момент времени» и «неприменимое свойство для данной сущности вообще» — рассматривается только первая. Приводятся правила обработки неопределенных значений (они обозначаются символом ω) при реализации обычных операций реляционной алгебры, а также вводятся принципиально новые операции — внешнего объединения ($\bar{\cup}$), внешнего естественного соединения ($\bar{\bowtie}$), внешнего соединения по условию ($\bar{\theta}$).

Операция внешнего объединения позволяет объединять отношения разной арности, несовместимые с точки зрения обычной реляционной

алгебры. При этом объединяемые отношения трансформируются к некоторому общему структурному виду путем дополнения их атрибутами со значением «не определено». Например, для отношений

$$R(A \ B \ C) \text{ и } S(B \ D)$$

p	1	2	2	u
p	2	5	3	v
q	1	2		

результатом выполнения внешнего объединения будет

$$R \overline{\cup} S(A \ B \ C \ D)$$

p	1	2	ω	
p	2	5	ω	
q	1	2	ω	
ω	2	ω	u	
ω	3	ω	v	

При выполнении обычных операций соединения (естественного или по условию) происходит потеря информации, если соединяемые отношения не имеют равных проекций или не выполняется условие соединения. Внешнее соединение позволяет сохранить эту информацию, так как в случае невыполнения условия соединения необходимым атрибутам соединяемых кортежей присваивается значение «не определено» и они включаются в результирующее отношение. Например, результатом выполнения операции естественного соединения по атрибуту B для приведенных выше отношений R и S будет отношение

$$R[B \bowtie B] S(A \ B \ C \ D)$$

p	1	2	ω
p	2	5	u
q	1	2	ω
ω	3	ω	v

Внешнее соединение по условию $[RB \oplus SB]$ даст такой результат:

$$R[RB \oplus SB] S(A \ RB \ SB \ C \ D)$$

p	1	ω	2	ω
p	2	2	5	u
q	1	ω	2	ω
ω	ω	3	ω	v

Выделение операций внешнего объединения и внешнего соединения упрощает (по крайней мере с точки зрения пользователя) процесс формирования запросов к базе данных, позволяет удобно реализовать интегрированную обработку отношений разных структур. Ниже будет

показана эффективность использования этих операций в моделях данных для систем управления транспортом.

Анализ сфер, где используются базы данных, показывает, что если до недавнего времени это были традиционные деловые применения, связанные в первую очередь с организационным управлением, то сейчас базы данных получают развитие в таких новых для них областях, как моделирование, автоматизированное проектирование, управление технологией, научные и статистические применения, экспертные системы и др. Эти новые применения отличаются как от обычных, так и друг от друга в целом ряде важных аспектов. Во-первых, каждое требует своего множества инструментов моделирования данных. Типы объектов и связи, которые должны быть описаны, например, в интегрированных системах автоматизированного проектирования и управления типа *CAD/CAM*, совершенно отличны от используемых в банковском применении. Во-вторых, каждая новая область применения имеет свое специализированное множество операций, которые должны обеспечиваться СУБД. Так, для графических приложений необходимы свои типы представлений данных и методы доступа к ним. Наконец, в ряде новых областей нужна поддержка для обработки множественных версий состояния объектов в разные моменты времени. Для эффективной реализации новых применений необходима разработка как соответствующих моделей данных, так и средств их поддержки. Модели данных должны обеспечивать интегрированное представление числовых и текстовых данных, сигналов, графических и видеоданных, правил, а СУБД — поддерживать эффективный доступ к любой разновидности информации. В отношении СУБД существует единая концепция — создание расширяемых СУБД на основе стандартизации интерфейсов и совместимости по разъему отдельных модулей. Ожидается, что СУБД станет своего рода «шиной программного обеспечения», через которую могут подключаться или отключаться модули из универсальной библиотеки программного обеспечения. Это означает, что проектируемые СУБД в отличие от большинства существующих будут не монолитными, а модульными с возможностью быстрой и удобной настройки на новую область применения. Наиболее известными из разрабатываемых проектов таких СУБД являются *EXODUS* [24] и *PROBE* [25]. Что касается развития моделей данных, то здесь в качестве основных можно выделить следующие тенденции: разработка временных или темпоральных баз данных *temporal data base (TDB)*, обеспечивающих представление и обработку состояния объектов в разные моменты времени; разработка баз данных для графических приложений, в частности в графических информационных системах, для обработки пространственных данных — графических и видеоданных; разработка объектных баз данных *object data base (ODB)* для систем моделирования и проектирования. Рассмотрим подробнее эти направления, поскольку каждое из них представляет несомненный интерес с точки зрения построения моделей данных в системах управления транспортом.

2.2. ВРЕМЕННЫЕ БАЗЫ ДАННЫХ

Обычные базы данных отображают состояние реального мира в некоторый конкретный отрезок времени. И хотя база данных изменяется в результате выполнения таких операций, как вставка, удаление или модификация, в целом содержимое базы данных является как бы моментальным снимком состояния реального мира. Причем состояния не обязательно текущего, поскольку изменения в базе данных всегда будут отставать от изменений в реальном мире. В этом процессе прошлые состояния базы данных и соответственно реального мира теряются и, как правило, не могут быть восстановлены. В [26] этот тип базы данных назван *snapshot* базой данных (*snapshot* — моментальный снимок). Конечно, что такие базы данных не могут обеспечить эффективное информационное моделирование в целом ряде предметных областей, где имеет место пространственно-временная динамика объектов. А как показывает практика, такая динамика является одной из основных особенностей, в той или иной мере присущей любой предметной области, тем более такой, как железнодорожный транспорт. Динамика проявляется как во времени, так и в пространстве, причем под временной динамикой понимается изменение состояния объектов во времени, т. е. их эволюция, а под пространственной — миграция объектов одной категории внутри или между объектами другой категории в границах искомого региона. Оба проявления динамики объектов — эволюция и миграция — являются обычно результатом происходящих с ними событий.

С точки зрения моделирования данных и эволюция и миграция проявляются в изменении во времени значений некоторых атрибутов, называемых обычно динамическими, а задача отображения динамики сводится к обеспечению возможности получения по запросу не только текущих значений этих атрибутов, но и тех, которые они принимали в некоторые моменты времени в прошлом. Иерархическая, сетевая и базовая реляционная модели фактически никак не отображают динамику реального мира. Представленные в этих моделях связи, например, типа «поставщики — детали» [21], часто характеризуют скорее ограничения целостности, чем происходящие реально процессы. Значит, их можно трактовать как то, что некоторый поставщик может поставлять некоторые детали в определенном количестве, либо же рассматривать как итоговые (вторичные) данные, характеризующие количественно взаимодействие типов объектов за некоторый период времени. Но проследить эволюцию того же поставщика — когда он появился в системе, как изменялся его статус, кому и какие детали он поставил за определенный период времени — из такой модели весьма трудно.

Поэтому в последнее время при разработке моделей данных все большее внимание стало уделяться вопросам адекватного отображения динамики объектов и возможности прослеживания их состояния, местонахождения, а также взаимодействия в течение некоторого времени. В результате появились временные, или темпоральные, базы данных, в которых атрибут времени стал одним из основных, характеризую-

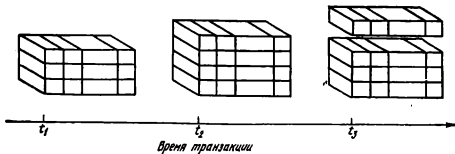


Рис. 2.2.1

щих моделируемые объекты. Как правило, *TDB* базируются на реляционной модели [26, 27], а языки соответствующих реляционных СУБД расширяются средствами для обработки временных запросов, т. е. запросов о состоянии объектов базы данных на некоторый момент времени в прошлом.

В [28] приведена достаточная библиография по истории появления и развития временных баз данных. Отметим лишь, что необходимость исчерпывающей обработки атрибута времени с целью анализа динамики изменения состояния объектов остро проявилась еще в начале 1970-х годов в медицинских информационных системах, где особенно важна история болезни пациента. Здесь уместно снова упомянуть расширенную реляционную модель Кодда, в которой категория сущностей типа события имеет атрибут времени, а графовые отношения следования и предшествования фактически задают разрешенную динамику для других категорий сущностей, участвующих в этих событиях. В дальнейшем сущности типа события не нашли почему-то широкого распространения в моделях данных для *TDB*, а развитие последних пошло по пути введения атрибута времени в отношения, характеризующие именно объекты предметной области, а не происходящие с ними события. Но принципиальная интерпретация атрибута времени осталась такой же, как и в *RM/T* — он указывает, на каком интервале времени те или иные данные достоверны.

Таким образом, в отличие от обычных (*snapshot*) баз данных в *TDB* элементарная единица данных представлена четверкой (объект — атрибут — значение — время), причем понятие времени может трактоваться по-разному. Во-первых, это может быть действительное время, т. е. время, когда произошло изменение состояния объекта реального мира. Во-вторых, это может быть время транзакции, т. е. время внесения соответствующего изменения в базу данных. Кроме того, существует еще время, указываемое в запросе к *TDB*, когда запрос касается состояния базы данных в прошлом.

Концептуально методы организации и обработки *TDB* рассмотрены наиболее полно в [26, 29]. В проводимой там классификации выделены три типа *TDB* — возвратные, исторические и собственно временные. В принципе, аналогичное разделение дается и в [27].

Возвратные (*rollback*) базы данных. Это такие базы данных, в которых значением атрибута времени является время транзакции. Состояние базы данных после каждой транзакции запоминается, так что,

по существу, возвратная база данных — это множество состояний обычной базы данных за какой-то интервал времени. Если отношение обычной базы данных можно представить двухмерной таблицей, то в возвратной базе данных это же отношение существует уже в трех измерениях, где время транзакции служит в качестве третьей оси. На рис. 2.2.1 представлено множество состояний одного отношения воз-

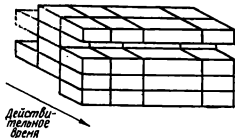


Рис. 2.2.2.

ратной базы данных за период, когда были выполнены три транзакции — добавление трех кортежей; добавление кортежа; удаление и добавление кортежа. Поскольку транзакциями управляет СУБД, то она же генерирует и два значения атрибута времени: время, когда кортеж был введен в базу данных, и время, когда он был оттуда удален. Первое значение будет всегда, а второго может не быть, если кортеж не удален из базы. Время обычно представляет собой монотонно возрастающие значения t_1, t_2, t_3, \dots и т. д. Если же в запросе пользователя указывается время t такое, что, например, $t_2 < t < t_3$, то фактически выдается состояние базы данных на время t_2 (самое последнее состояние перед t), а не состояние объектов реального мира на время t . Очевидно, что это не всегда одно и то же. Несоответствие действительного времени и времени транзакции приводит к тому, что возвратные базы данных больше отображают свою собственную динамику, чем динамику объектов реального мира. Это является очевидным их недостатком, из которого вытекает и еще один — невозможность внесения изменений по все состояния, кроме самого последнего.

Исторические (historical) базы данных. В этих базах данных атрибут времени имеет значение действительного времени, т. е. фактического времени изменения состояния объекта реального мира. В исторической базе данных каждое отношение хранится (в отличие от возвратной базы данных) в виде только одного исторического состояния. Оно содержит кортежи, данные которых являются достоверными на текущий момент времени, а также кортежи с данными, срок достоверности которых уже истек, но они не удалены из базы данных. Удаленные кортежи не сохраняются, а значит, не восстанавливаются и предыдущие состояния. Отношение исторической базы данных также можно представить в трех измерениях (рис. 2.2.2), но временная ось здесь соответствует уже действительному времени, а не времени транзакции, как в возвратных базах данных. Из рис. 2.2.2 видно, что отношение исторической базы данных состоит как бы из нескольких частей обычной базы данных, действительных на разных интервалах времени. Это показано в [26] принципом временной плотности. Атрибут времени в исторической базе данных также имеет два значения, ограничивающие интервал времени (от, до), в протяжении которого данные кортежей были достоверны. Очевидно, что если второго значения нет, то данные продолжают оставаться достоверными и на текущий момент вре-

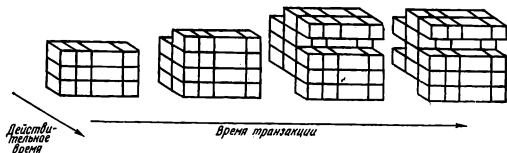


Рис. 2.2.3

Временные (*temporal*) базы данных. Как указывается в [26], сам термин «временная база данных» должен подчеркнуть необходимость представления в ней и действительного времени, и времени транзакции. Это позволяет объединить преимущества первых двух типов баз данных — возможность восстановления прошлых состояний и адекватное отображение динамики объектов реального мира в действительном времени. Пример отношения временной базы данных на рис. 2.2.3 показывает, что данные теперь существуют в двух временных осях, а само отношение представлено рядом исторических состояний, создаваемых в данном случае четырьмя такими транзакциями: добавление трех кортежей; добавление кортежа; удаление и добавление кортежа; удаление кортежа, введенного в результате первой транзакции (возможно, он был введен ошибочно или преждевременно). Очевидно, что в возвратной базе данных невозможно была бы реализация четвертой транзакции, а историческая база данных не позволила бы восстановить состояние базы, например, после первой транзакции. Из рис. 2.2.3 видно, что каждое обновление отношения во временной базе данных предполагает предварительное копирование предыдущего состояния и лишь затем внесение изменения и создание обновленного состояния.

Для обработки временных баз данных разрабатываются специальные языковые средства, в качестве которых, как правило, используются расширения известных языков запросов. Так, например, в [26] предлагается язык *TQuel*, являющийся расширением языка *Quel*, ис-

Таблица 2.2.1

Служащие

Фамилия	Должность	Действительное время пребывания в должности		Время транзакции	
		Начало	Конец	Начало	Конец
Иванов	Инженер	03.72	06.77	03.72	∞
Иванов	Старший инженер	06.77	05.81	06.77	∞
Иванов	Младший научный сотрудник	05.81		04.81	∞
Петров	Инженер	03.78	06.83	02.78	∞
Петров	Старший инженер	06.83		06.83	∞
Сидоров	Старший инженер	03.76		02.76	04.76
Сидоров	Инженер	03.76	06.81	04.76	∞
Сидоров	Старший инженер	06.81		05.81	∞

Публикации

Автор	Журнал	Дата представления статьи	Время транзакции
Иванов	Управляющие системы и машины	05.80	05.80
Петров	То же	03.79	03.79
Петров	Кибернетика	11.79	11.79
Сидоров	Автоматика	06.83	06.83

пользуемого в СУБД Ingres [30]. В *TQuel* операторы создания, ввода, удаления, модификации и поиска могут быть дополнены двумя типами специальных предложений, включающих ключевые слова и временные предикаты на переменных, представляющих атрибут времени. Эти предложения фактически накладывают дополнительные ограничения на множество обрабатываемых в запросе кортежей, выполняя две основные функции. Во-первых, обеспечивают в случае необходимости возврат базы данных к историческому состоянию на некоторый момент времени t в прошлом (*as-of*-предложение), исключая при этом из полученных отношений все дополнения и модификации, сделанные после этого времени t . Во-вторых, обеспечивают обработку этого исторического состояния с учетом условий, задаваемых на действительных значениях атрибута времени (*when*-предложение), еще более сужая тем самым подмножество обрабатываемых кортежей. На примере табл. 2.2.1 и 2.2.2 покажем в неформальном виде основные типы запросов, обрабатываемых *TQuel*.

Запрос 1. Кто занимал должности старших инженеров в марте 1982 г.?

Ответ. Петров, Сидоров.

Запрос 2. Кем из старших инженеров были написаны статьи и для каких журналов?

Ответ. Ивановым — для 4 «Управляющих систем и машин», Сидоровым — для «Автоматики».

Запрос 3. Кто был младшим научным сотрудником, когда Сидоров был назначен старшим инженером?

Ответ. Иванов.

Запрос 4. Кого перевели на должность младшего научного сотрудника, в то время как по меньшей мере кто-то один оставался в должности старшего инженера?

Ответ. Никого.

Представляет интерес подход к созданию временной базы данных, используемый в проекте *TIGRE* [27]. Пользователю предоставлена возможность определять в схеме базы данных сущности в целом или отдельные их свойства (атрибуты) как динамические. Динамическими сущности объявляются явно, а также могут приниматься системой по умолчанию, если они вступают в связь с другими сущностями, объявленными как динамические. Для динамических сущностей (или их отдельных атрибутов) создаются исторические копии при внесении соот-

ветствующих изменений в данные. Причем если динамической объявлена сущность, то создается ее полная историческая копия, для динамических же атрибутов копируются только их значения. Имеется возможность модификации исторических копий. Предусмотрены три варианта организации исторических данных → запоминание по указанию пользователя (*MH*-история); периодическое запоминание через установленные интервалы времени (*PH*-история); последовательное запоминание после каждого изменения данных (*SH*-история). Такой подход обеспечивает, по мнению авторов, удобный и быстрый доступ к историческим данным, не требуя для этого сложной обработки данных о соответствующих событиях по типу отката транзакций. Обработка исторических и текущих данных осуществляется с помощью языка манипулирования данными *LAMBDA*, функционально аналогичного *TQuel* и являющегося расширением известного языка *SQL* [31]. Существенной особенностью является то, что в базе данных наряду с историей данных хранится и история схем данных, т. е. запоминаются все действия по реструктуризации базы данных. Это позволяет при обработке запросов на выдачу исторических данных о некоторой сущности обращаться к ним в соответствии именно с той схемой, которая существовала на момент времени, указанный в запросе.

В заключение следует высказать некоторые соображения об эксплуатационной эффективности временных баз данных. Актуальность этого вопроса не требует особых доказательств хотя бы потому, что очевидны значительное увеличение, во-первых, объема хранимых данных и, во-вторых, времени обработки запросов. В [32] проведен подробный количественный анализ изменения этих, а также соответствующих стоимостных показателей в зависимости от типа базы данных (обычная, возвратная, историческая, временная) и методов организации и доступа к данным (прямой, последовательный, индексно-последовательный). Анализ, сделанный по результатам специального контрольного теста, состоящего из ряда запросов с разными условиями выборки, показал, что временные базы данных требуют примерно вдвое большего объема памяти, чем возвратные или исторические. Примерно такое же соотношение и стоимостных показателей обработки запросов. Для улучшения характеристик временных баз данных предлагается организация двухуровневой памяти, т. е. выделение отдельных участков памяти для текущих и исторических данных, а также использование наиболее подходящих для динамического роста методов доступа (*B* — деревья, динамическое хеширование) в совокупности с вторичным индексированием.

2.3. БАЗЫ ДАННЫХ В ГРАФИЧЕСКИХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Системы обработки графических данных самого различного назначения — одна из новых, нетрадиционных сфер, где в последнее время наблюдается интенсивное развитие баз данных. В узком смысле графические данные понимаются как собственно графические (векторные) изображения, состоящие из множества точек и линий, заданных соот-

ответствующими координатами, в широком — включают и так называемые полутоновые (многоградационные) изображения, представляющие множеством точек растра. Из полутонового изображения специальными аналитическими методами (например, оконтуриванием) могут быть извлечены и непосредственно графические данные, связанные с этим изображением. Предложенное в [33] понятие «система баз видеоданных» (*pictorial data base system*) объединяет обе категории данных (графические и полутоновые) и определяется как «интегрированный набор видеоданных, к которому большое число пользователей имеет легкий доступ».

Трудности организации баз видеоданных по сравнению с традиционными базами данных очевидны и определяются в первую очередь такими факторами: большим объемом и сложной структурой видеоданных (даже при использовании специальных алгоритмов сжатия, объем памяти для хранения одного полутонового изображения может достигать до 10 Кбайт и более); необходимостью разработки специальных моделей для представления пространственных данных и алгоритмов их обработки (в том числе ввода-вывода); необходимостью и особыми условиями использования специализированных устройств ввода, вывода, хранения и отображения видеоданных.

Но очевидно и то, что без баз видеоданных практически невозможно обеспечить накопление, хранение и обработку все более возрастающих потоков видеoinформации. Существует достаточно много графических информационных систем, основанных на базах видеоданных. В [34, 35] приведена довольно обширная библиография по их использованию и рассмотрен ряд систем хранения и обработки изображений, классифицированных по назначению, сферам применения, структурам данных и методам их обработки.

Основными областями, где базы видеоданных уже используются или где их использование представляется наиболее эффективным, являются следующие [35].

1. Картография — хранение, обработка и отображение географических данных о состоянии земной поверхности. Эти данные могут использоваться для решения самых разнообразных задач — регионального планирования, строительства, навигации, экологического моделирования, производства дорожных карт и др. Более трети известных систем создано именно в этой области.

2. Дистанционные исследования — обработка данных, получаемых с помощью аэрофотосъемки, искусственных спутников Земли и межпланетных космических станций. С их использованием решаются задачи разведки и поиска полезных ископаемых, анализа состояния окружающей среды, космических исследований и др.

3. Медицина — хранение и обработка данных о результатах различных процедур функциональной диагностики, таких, как рентгенография, ультразвуковая томография, зондирование и др. Основные задачи — диагностика заболеваний, сравнительные и научные исследования.

4. Автоматизированное проектирование — хранение и обработка моделей и чертежей деталей и узлов разнообразных технических объ-

ектов (машин, механизмов, судов, самолетов, зданий и др.). Это область, где системы баз видеоданных начали использоваться позже других, но развивается, особенно в последнее время, очень интенсивно. Трудно перечислить все множество задач, которые могут быть решены в рамках систем автоматизированного проектирования, — эти задачи имеют место в любой сфере технической деятельности человека.

5. Другие области. Сюда можно отнести метеорологию (анализ атмосферных течений и хранение метеокарт), криминалистику (хранение фотопортретов, отпечатков пальцев, обработку фотороботов), автоматизацию научных исследований и другие области, где применение баз видеоданных находится пока на уровне единичных экземпляров или экспериментальных образцов.

Анализ проведенной в [34] классификации показывает, что в основном системы баз видеоданных используются преимущественно в таких областях, как картография, медицина, дистанционные исследования, причем явно преобладают системы обработки полутоновых изображений. Вместе с тем обращает на себя внимание невысокий процент использования баз видеоданных в системах автоматизации проектирования и моделирования, хотя именно здесь их применение сулит весьма существенное повышение эффективности. Последнее объясняется тем, что существующее программное обеспечение машинной графики как основного инструмента проектирования все заметнее не удовлетворяет возрастающие потребности пользователей. При этом самым большим и принципиальным недостатком пакетов машинной графики являются слабые средства структурирования данных. Предложения по стандартизации машинной графики [36] также оставляют этот вопрос открытым, так как преследуют в первую очередь цели независимости графических программ от устройств и реализации некоторого базового набора средств манипулирования изображениями. Следствием явно выраженной жесткой зависимости программ от данных является то, что практически каждое изображение — это самостоятельная программа, а следовательно, модификация обрабатываемого изображения или генерация нового требуют соответственно корректировки существующей или написания новой программы.

Еще один серьезный недостаток систем машинной графики состоит в том, что их интерактивные средства ориентированы на обработку изображений, а не моделей объектов. Под моделью графического объекта здесь понимается его структурированное представление в виде множества элементов типа точек и линий, описывающих пространственные характеристики этого объекта. Очевидно, что один и тот же объект (естественно, кроме точечного) может иметь множество различных изображений в зависимости от выбора точки зрения и типа проекции, а модификация модели объекта может быть совершенно неадекватна модификации его изображения и наоборот. Например, вид сверху на пирамиду не изменится при перемещении ее вершины вдоль вертикальной оси, хотя сама модель при этом изменяется. В связи с тем что в таких областях, как автоматизация проектирования первична модель, а изображение — вторично, для них важны в первую очередь средст-

ли обработки именно моделей объектов и получения (уже как результат) их модифицированного изображения.

Главное, что отличает базу видеоданных от обычных баз данных, — это наличие графических сущностей. Под графической понимается сущность, имеющая хотя бы один графический атрибут (это могут быть, например, координаты, цвет, яркость), т. е. свойство, которое при заданном значении может восприниматься визуально. Из такого определения следует, во-первых, что сущности обычных баз данных являются частным случаем графической сущности, а во-вторых, что графическая сущность, кроме графических атрибутов, характеризующих ее визуально, может иметь и неграфические атрибуты, описывающие другие ее свойства (идентифицирующие и семантические признаки) в числовой и текстовой формах. Таким образом, в базе видеоданных для каждого объекта хранится его изображение и некоторый структурированный набор свойств (крайним случаем является наличие единственного идентифицирующего признака — имени). Что касается представления структурированного набора свойств, то оно ничем не отличается от традиционного в обычных базах данных. Отметим можно, пожалуй, лишь то, что для этой цели используются, как правило, реляционные базы данных [37—40]. Методы же структурирования данных для представления самих изображений зависят от их вида (полутоновые или графические), и разделяются соответственно на два класса, называемых матричными (позиционными) и топологическими (векторными).

Матричные структуры. Простейшей и наиболее употребительной структурой представления полутоновых изображений является матрица яркостей. Плоскость всего изображения разбивается с помощью прямоугольной сетки на элементы одинакового размера, являющиеся наименьшими, неделимыми частями изображения. Очевидно, что ими будут отдельные точки раstra, каждой из которых соответствует элемент матрицы яркостей. Их число зависит обычно от разрешающей способности используемых в системе средств отображения графической информации — дисплеев или специальных растровых принтеров. Растр стандартного дисплея содержит порядка 600×300 точек. Специализированные дисплеи имеют более высокую разрешающую способность — 1200×1000 точек и более. Для кодирования каждой точки необходимо как минимум два значения (0 — нет подсветки, 1 — есть подсветка) или 1 бит информации. С учетом же реальных потребностей (индикации разных градаций яркости, мерцания, цвета и др.) для кодирования одной точки раstra может использоваться до 8 бит информации. Отсюда очевиден главный недостаток матрицы яркостей — необходимость большого объема памяти (для хранения одного изображения размером $600 \times 300 \times 8$ требуется более 180 Кбайт памяти). Особенно неэффективны матрицы яркостей при представлении ненасыщенных изображений. Со значительными трудностями и большими затратами времени связана также избирательная обработка изображений — поиск на них отдельных объектов, фрагментов, сравнение их и т. д. Зато визуализация изображения в целом максимально упрощена и требует минимальных затрат времени. Поэтому матрицы яркостей использу-

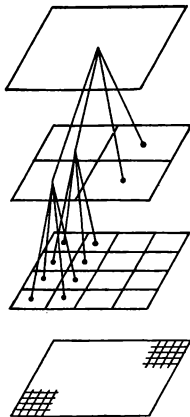


Рис. 2.3.1

ются в таких системах, где имеется достаточно большая память, время обработки строго не лимитировано, а доступ к изображению осуществляется, как правило, по его имени.

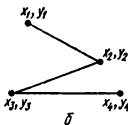
В последнее время большую популярность для организации баз видеоданных завоевывают пирамидально-рекурсивные структуры [35], в которых изображение представляется иерархическим набором матриц яркостей с различным разрешением (рис. 2.3.1). При этом элемент изображения более высокого уровня содержит некоторое обобщенное описание подчиненных элементов нижнего уровня, т. е. соответствует некоторому прямоугольному фрагменту изображения. По сравнению с обычной матрицей яркостей такой подход обладает рядом преимуществ. Во-первых, экономится память за счет возможности сжатия данных — если все подчиненные элементы имеют одинаковые (или близкие) значения, они могут быть опущены. Во-вторых, обеспечивается возможность удобной обработки отдельных фрагментов изображения. В-третьих, сокращается время обработки, так как пирами-

дально-рекурсивная структура является фактически естественным иерархическим индексом, который удобно использовать для быстрого доступа к отдельным фрагментам изображения. Вместе с тем остается недостаток, свойственный всем матричным структурам, — трудность перехода от структурного к семантическому уровню, т. е. избирательной обработки отдельных объектов изображения (особенно линейных, пересекающих несколько фрагментов).

Топологические структуры. Это принципиально иной (по сравнению с матричным) концептуальный подход, основанный на описании сложного изображения в терминах объектов, элементов и примитивов с их свойствами и отношениями между ними. Он используется в системах баз видеоданных, работающих с графическими изображениями. В простейшем случае они представляются простым набором координат точек, описывающих множество векторов или набор точечных объектов. Чаще всего используются рассматриваемые в [41] модели типа «список точек» (упорядоченный набор пар точек, рис. 2.3.2, а, б) и «список точек с управляющим параметром» (упорядоченный набор точек с указанием связей между ними). Если рассуждать в терминах реляционной модели данных, то изображение представляется отношением, каждый кортеж которого описывает объект или элемент изображения в виде отдельного графического примитива типа точка, маркер или линия. Естественно, что кортежи могут содержать и ряд неграфических атрибутов (A, B, C на рис. 2.3.2, а), характеризующих

X_i	Y_i	X_j	Y_j	A	B	C	...
x_1	y_1	x_2	y_2	a_1	b_1	c_1	...
x_2	y_2	x_3	y_3	a_2	b_2	c_2	...
x_3	y_3	x_4	y_4	a_3	b_3	c_3	...

а



б

Рис. 2.3.2

другие свойства объектов или элементов. Такая структура представления предполагает использование принципа интерпретации при визуализации изображения, т. е. специальная универсальная программно-интерпретатор просматривает отношение, анализирует значения графических атрибутов и, передавая их в качестве параметров вызываемым графическим процедурам, генерирует изображение. Очевидным преимуществом топологических структур является возможность произвольной избирательной обработки изображений, хотя при этом и возможны дополнительные затраты времени на селекцию данных и работу интерпретатора. Разумеется, эти структуры неэффективны в системах, где требуется преимущественно обработка изображений в целом.

В ряде систем баз видеоданных [38, 40] успешно используется **комбинированный** подход к представлению изображений. Например, в [38] изображение разбивается на множество иерархических фрагментов (так называемых физических карт), содержащих отдельные типы объектов и представленных матрицами яркостей. В соответствие множеству физических карт ставится множество логических карт, структурированных в виде отношений реляционной модели данных. Каждая логическая карта содержит топологическое представление объектов, изображенных на соответствующей физической карте, расширенное дополнительными семантическими признаками. Таким образом, совокупность логических карт дает исчерпывающее семантическое описание содержимого базы видеоданных и используется для поиска данных по запросу. Выдаются же в ответ на запрос соответствующие физические карты, причем они могут накладываться друг на друга, создавая в конечном счете полное изображение некоторого региона.

Архитектуру систем баз видеоданных можно считать достаточно сложившейся (рис. 2.3.3). В качестве основы программных средств поддержки этих систем чаще всего используются типовые, как правило реляционные, СУБД [39, 40]. Взаимодействие пользователей с системой осуществляется через графический интерфейс — программу, управляющую в целом процессами ввода, поиска, интерактивной обработки и выдачи графических данных.

При вводе изображений часто используются [38, 40] специальные методы их анализа (в первую очередь методы распознавания образов), позволяющие выделить отдельные объекты и их свойства. При этом сами изображения помещаются в отдельную память видеоданных (в виде матрицы яркостей), а описания объектов и их свойств структурируются и помещаются в самостоятельный раздел базы данных (обычно

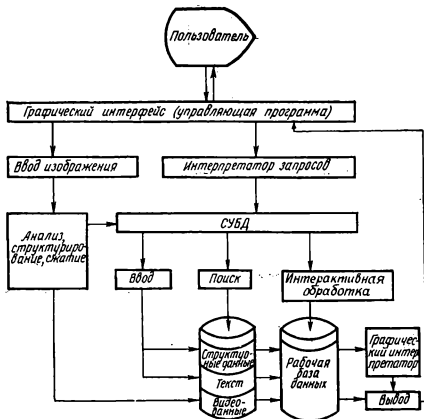


Рис. 2.3.3

это реляционная база данных). Кроме того, в отдельном разделе могут дополнительно храниться неструктурированные текстовые описания изображений. Все эти три компонента интегрированной базы видеоданных тесно связаны друг с другом. В частных случаях могут отсутствовать неструктурированные тексты, а в простейших — и структурированные данные (т. е. сохраняются только собственно изображения). В некоторых случаях может отсутствовать и раздел видеоданных, если изображения полностью структурируются и хранятся только в виде топологических структур в разделе структурированных данных.

Для поиска графических данных используются специальные графические языки [34, 38—40], являющиеся обычно расширениями известных реляционных языков манипулирования данными. Операторы графического языка с помощью интерпретатора запросов транслируются в операторы реляционного языка или специальные процедуры, с помощью которых из базы видеоданных выбирается необходимое изображение, фрагмент или отдельные объекты и помещаются в специальную рабочую область (или рабочую базу данных) пользователя. Поиск выполняется в простейшем случае по имени изображения, но обычно системы баз видеоданных обеспечивают обработку более сложных запросов, таких, как поиск по значениям графических и неграфических атрибутов; структурный поиск, определяемый типом и взаимным месторасположением объектов (например, найти объекты определенного типа, расположенные слева (справа), сверху (снизу) от

такого-то и т. п.); поиск по функции подобия, т. е. поиск изображений или объекта, подобного заданному: сложный поиск, предусматривающий интегрированную обработку структурированных и видеоданных.

Палитры рабочей базы данных у каждого пользователя обеспечивают два важных фактора в работе с базой видеоданных — удобство интерактивной обработки (что особенно важно для графических систем) и независимость пользователей друг от друга. Основными средствами интерактивной обработки являются графический редактор, графический интерпретатор и процедура корреляции. Графический редактор обеспечивает выполнение функций редактирования изображения на экране дисплея и преобразование полученного в результате изображения в формат базы данных. Графический интерпретатор осуществляет генерацию программы вывода изображения на устройстве машинной графики в соответствии с топологической структурой этого изображения в базе видеоданных. В целом он генерирует команды вывода точек по координатам соединяющих их линий (в том числе кривых), заштриховку и закраску площадей, вставку текстовых наименований и другие операции с учетом параметров геометрических преобразований — масштабирования, поворота и др. Функция корреляции обеспечивает идентификацию элементов рабочей базы данных при указании курсором их графического изображения на экране дисплея.

В заключение, рассматривая системы баз видеоданных в целом, необходимо отметить главную тенденцию их развития: интеграцию в них различных методов представления данных (матричных, пирамидальных, топологических, комбинированных) и различных методов организации поиска и доступа к данным (по имени, совокупности значений разных атрибутов, видеобразу). Эта тенденция, полностью соответствующая уже упоминавшейся концепции создания расширяемых (МБД), позволяет надеяться в итоге на создание программных систем поддержки баз видеоданных, строящихся на том же принципе «общей шины программного обеспечения» и быстро настраиваемых на разнообразные области применения.

2.4. ОБЪЕКТНЫЕ БАЗЫ ДАННЫХ

Основной специфической особенностью систем моделирования и автоматизированного проектирования является необходимость манипулирования объектами сложной, как правило, иерархической структуры, характеризующейся разнообразными типами данных: числовыми, текстовыми, графическими и др. С учетом этой специфики в последние годы был разработан ряд так называемых объектно-ориентированных языков программирования — C++, Objective-C, Common Lisp Objective System (CLOS), Smalltalk. Каждый из них имеет свои средства определения объектов в виде достаточно сложных структур, конструируемых из разных типов, являющихся фактически абстрактными типами данных. Объекты могут создаваться, разрушаться (не во всех языках), но не могут динамически изменяться путем добавления к ним или удаления из них отдельных типов. Основным средством обработки объектов являются функции в том смысле, как они понимаются в язы-

как программирования. Ни один из языков не имеет средств прямой поддержки множеств, хотя все они могут работать с ограниченными массивами, а некоторые с кортежами (*C++*, *Objective — C*) или списками (*CLOS*). Отсутствие универсальных, независимых от программ средств структурирования данных создавало определенные неудобства при разработке прикладных систем моделирования и проектирования.

Это послужило толчком к интенсификации исследований в области разработки баз данных для этих применений, в результате которых был предложен ряд объектно-ориентированных моделей данных [42—46]. Несмотря на некоторые непринципиальные различия, общая концепция в них практически одинакова. Объект определяется как сущность с внутренним содержимым и внешним поведением. Содержимое — это свойства (атрибуты) объекта. При этом значениями атрибутов могут быть элементарные значения (число, текст), кортежи или множества кортежей (отношения), представляющие, в свою очередь, другие объекты. Таким образом, объекты произвольной сложности могут быть описаны (в том числе рекурсивно) как иерархические структуры абстрактных типов данных. Поведение — это множество правил, описывающих, как и при каких условиях объект может менять свое состояние и взаимодействовать с другими объектами. Поведение описывается обычно в виде разрешенных последовательностей событий, которые могут произойти с объектом [44]. Определенный в содержательном и поведенческом аспектах объект тем самым инкапсулирует собственно данные и допустимые операции над ними. Это обстоятельство следует подчеркнуть особо, так как оно открывает новую альтернативу в информационном моделировании — переход от абстрактных типов данных к абстрактным объектам данных [44] с более разнообразной и глубокой семантикой.

При вводе в базу данных каждому объекту автоматически присваивается внешний ключ (заменитель). В моделях прямого хранения сложные объекты хранятся так, как они определены в концептуальной схеме. При этом эффективна выборка данных обо всем объекте, а выборка подобъектов — нет. В моделях нормализованного хранения сложные объекты разлагаются на множество кортежей нескольких отношений. При этом достигается лучшая производительность при доступе к подобъектам. Для получения данных о сложном объекте в целом используется соединение по значению заменителя.

Следует особо отметить, что объекты, как правило, не статичны, а изменяются во времени, причем изменения могут касаться как элементарных значений данных, так и структуры самих объектов. Значит каждый объект на протяжении своего существования в базе данных может присутствовать в ней в виде множества состояний, называемых версиями. Объект может иметь любое число версий, а из каждой версии может быть произведено, в свою очередь, также любое число версий. Кроме того, версия может иметь несколько эквивалентных представлений — временная, рабочая, окончательная. Объектные модели предполагают наличие специальных средств управления версиями [47, 48], в том числе создание, удаление, модификация, копирование и др.

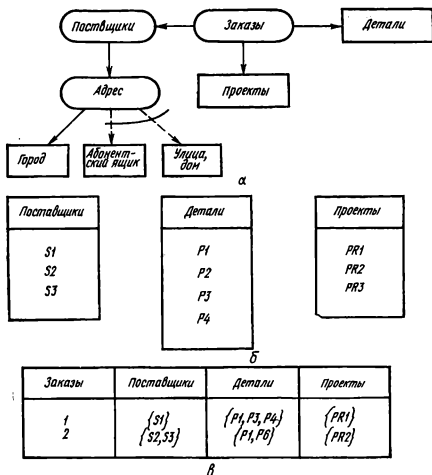


Рис. 2.4.1

Как свидетельствует анализ [49], концептуально объектная и реляционная модели достаточно близки, поскольку основаны обе на теории множеств. Видимо, поэтому большинство объектных баз данных строится именно на реляционной модели, а реляционные языки с небольшими расширениями успешно используются для поддержки этих баз. Наглядным примером такого подхода может служить рассматриваемая в [42] модель объектной базы данных.

Как утверждают сами авторы, главной целью их работы было обеспечить при описании сложных объектов однородное представление таких известных форм абстракции, как агрегация и обобщение. Основное понятие предлагаемой модели — тип молекулы. Оно описывает тип собственно объекта или тип отношения (связи) между объектами на одном уровне абстракции, т. е. объекты и связи представляются одинаковым образом, как и в обычной реляционной модели, причем связь на одном уровне абстракции может быть объектом на другом, более высоком уровне. Тип молекулы определяется исходя из других типов молекул и представляется в виде ненормализованного отношения. Если тип молекулы M определяется исходя из типов молекул M_1, M_2, \dots, M_n , тогда молекула типа M является набором молекул типов от M_1 до M_n . Каждый тип молекулы-компонента M_i представ-

ляется в M кортежем, содержащим значения ключей (или заместителей) всех реализаций, входящих в M . В определение молекулы M входят также обычные атрибуты, представляющие элементарные значения (так называемые атомарные атрибуты), характеризующие свойства молекулы в целом.

Формально тип молекулы M определяется следующим образом:

$$M = [A_1, A_2, \dots, A_m, M_1, M_2, \dots, M_n], \quad (2.4.1)$$

где A_i — атомарные атрибуты; M_i — типы молекул-компонентов; $n, m \geq 0$.

Такая молекула представляется ненормализованным отношением с кортежами

$$\{a_1, a_2, \dots, a_m, \{k_1\}, \{k_2\}, \dots, \{k_n\}\}, \quad (2.4.2)$$

где a_i — значения атомарных атрибутов; $\{k_i\}$ — множество ключей (заместителей) реализаций M_i , входящих в реализацию M (см. отношения на рис. 2.4.1, б и 2.4.1, в для молекулы «заказы» на рис. 2.4.1, а).

Каждая из молекул-компонент M_i может, в свою очередь, иметь структуру, аналогичную M . Если же какая-то из M_i не включает молекулы другого типа, т. е. M_i не содержит ни одного множества $\{k_i\}$, это означает, что молекула M_i есть не что иное, как обычное нормализованное отношение. Такие молекулы называются независимыми. Схематически молекула M изображается в виде эллипса (рис. 2.4.1, а). Исходящие стрелки указывают компоненты M . Входящие в M стрелки указывают молекулы, для которых M является компонентой. Прямоугольниками обозначены независимые молекулы.

Рассматриваемая модель позволяет представлять и обрабатывать сложные объекты с динамически изменяющимися структурными характеристиками. Это обеспечивается тем, что в реализации M представлены только ключи (заместители) входящих в нее реализаций M_i , а следовательно, изменение структуры одной из молекул-компонент M_i не требует реструктуризации M .

Рассмотрим в заключение, как достигается однородность представления агрегации и обобщения. Возвращаясь к определению молекулы, можно сказать, что она есть не что иное как агрегация, связывающая совокупность молекул из заранее определенного множества типов молекул и содержащая произвольное количество молекул любого типа. Примечательным в рассматриваемой модели является то, что обобщение в ней является частным случаем агрегации. Действительно, если M обобщает несколько типов молекул M_i , то очевидно, что в каждой реализации M только одно из множеств $\{k_i\}$ будет непустым, а остальные будут недействительными. На рис. 2.4.1, а обобщения показаны прерывистыми стрелками. Чтобы подчеркнуть, что при обобщении только одно $\{k_i\}$ будет действительным, на рисунке через соответствующие стрелки обобщения проводится дуга. Такие дуги позволяют также моделировать вариантную структуру объектов. Например, как видно из рис. 2.4.1, а, адрес поставщика может быть городом и номером абонентного ящика или городом, домом и улицей.

Глава 3

СПЕЦИФИКА ПОСТРОЕНИЯ МОДЕЛЕЙ ДАННЫХ В СИСТЕМАХ УПРАВЛЕНИЯ ТРАНСПОРТОМ

3.1. ОТОБРАЖЕНИЕ ДИНАМИКИ ОБЪЕКТОВ ПРЕДМЕТНОЙ ОБЛАСТИ

При проектировании и разработке баз данных для систем управления железнодорожным транспортом необходимо учитывать следующие специфические особенности.

1. Все множество типов объектов моделируемой предметной области можно разделить на две основные категории — стационарные, территориальное месторасположение которых не изменяется с течением времени, и подвижные, перемещающиеся в границах некоторого моделируемого региона. Примерами стационарных объектов являются станции, грузовые фронты, депо и др., подвижных — поезда, вагоны, локомотивы.

2. Стационарные и подвижные объекты имеют, как правило, иерархическую структуру, определяющую текущую подчиненность и принадлежность объектов разных уровней. Например, станция включает парки путей, грузовые фронты, локомотивные и вагонные депо; поезд состоит из локомотива и вагонов и т. д.

3. Состояние стационарных и подвижных объектов с разной степенью интенсивности изменяется во времени в результате происходящих с ними событий, т. е. проявляется явно выраженная эволюция объектов предметной области. Примером могут служить реконструкция станций, грузовые операции с вагонами и др.

4. Имеет место миграция подвижных объектов в пределах стационарных и между ними — передвижение поездов, вагонов, грузов внутри станций и между ними и т. п.

5. Происходит интенсивное функциональное взаимодействие разных типов объектов в рамках различных технологических операций — прием-отправление поездов на станциях, погрузка-выгрузка вагонов на грузовых фронтах, ремонт вагонов и локомотивов в депо и т. д.

6. Многие объекты (в первую очередь стационарные) имеют графические схематические изображения, используемые при решении ряда задач проектирования и моделирования.

Перечисленные особенности свойственны полностью или частично не только транспорту, но и другим предметным областям. Транспорт же, в частности железнодорожный, является такой сферой, где эти особенности выражены особенно отчетливо и предъявляют достаточно жесткие требования к построению моделей данных.

В 2.2 был рассмотрен ряд моделей, обеспечивающий представление данных об эволюционных изменениях состояния сущностей (объек-

тов) — возвратные, исторические и временные базы данных. Каждая из них с разной степенью точности и детализации отображает динамику объектов предметной области и обеспечивает обработку исторических запросов, т. е. запросов о состоянии объектов на некоторый момент времени в прошлом. В основе всех этих моделей лежит принцип создания так называемых исторических копий состояния базы данных, сводящийся к тому, что при изменении значения некоторого свойства (атрибута) сущности в отношении модели добавляется новый кортеж, описывающий эту же сущность, но уже с новым значением свойства, а также с действительным временем (а часто и со временем транзакции) изменения значения этого свойства. Очевидным недостатком такого подхода является дублирование (причем многократное при высокой интенсивности изменения свойств) неизменяющихся значений свойства, что приводит к большому росту памяти и увеличению времени обработки запросов. Очевидно негативное влияние этих факторов на стоимостные показатели обработки данных. Кроме того, часто требуется [27] специальное объявление как динамических тех атрибутов, для которых должны создаваться исторические копии их значений, а также необходимы специальные расширения языка манипулирования данными для обработки исторических запросов [26].

В данном разделе рассмотрены основные концепции построения реляционной модели данных типа «сущность — событие — связь», которая позволяет устранить или по крайней мере свести к минимуму влияние перечисленных недостатков, обеспечивая в то же время удобное и адекватное отображение динамики объектов моделируемой предметной области. Основным принципом построения предлагаемой модели является динамическое установление и разрушение связей между сущностями на основании обработки данных о событиях с сохранением предыстории связей и событий, происходивших с сущностями на протяжении определенного интервала времени. Такой подход позволяет восстанавливать значения любых атрибутов на любой (точнее, обозримый моделью) момент времени в прошлом без специального объявления этих атрибутов и хранения исторических копий их значений. Важнейшим свойством модели является также возможность прослеживания не только изменения состояния отдельных объектов, но и взаимодействия объектов разных типов. Все это делает модель более гибкой и универсальной, обеспечивает большую независимость данных, не требует разработки специальных языковых средств. Рассмотрим принципиальные свойства каждой базовой структуры в модели «сущность — событие — связь».

Типы сущностей описывают структуру и свойства объектов предметной области аналогично тому, как это принято в известных моделях, данных, но при этом все множество типов объектов подразделяется на два непересекающихся подмножества, описывающих соответственно стационарные и подвижные объекты.

Типы событий описывают факты изменения состояния объектов во времени в результате выполняемых с ними операций. Иначе говоря, событие — это переход стационарного или подвижного объекта из состояния S_i в некоторое новое состояние S_j . Данные о событиях мо-

гут относиться к одному типу объектов, описывая эволюционное изменение его состояния (погрузка или выгрузка вагонов), или связывать два и более типов объектов, отображая их функциональное взаимодействие — вступление в связь, или, наоборот, разрыв связи (прибытие поезда на станцию или отправление его на перегон). Так как каждое событие имеет свой временной параметр, то очевидно, что каждое состояние будет характеризоваться двумя значениями времени: t_{i1} и t_{i2} — перехода в состояние S_i и выхода из него, т. е. перехода в S_j (при этом $t_{i2} = t_{j1}$). Последнее состояние S_n , в котором объект находится в текущий момент времени, также характеризуется двумя значениями — t_{n1} и текущим временем t_i (до перехода в следующее состояние времени t_i может быть условно присвоено значение ∞ , как это делается, например, во временных базах данных [26]). Очевидно, что любое событие, помимо времени, может характеризоваться также рядом других свойств, описывающих, в частности, причины и обстоятельства изменения состояния объектов. Такого рода данные могут потребоваться и при подготовке ответов на исторические запросы пользователей к базе данных. Все это свидетельствует о целесообразности выделения типа данных «события» в качестве самостоятельного в модели данных. С точки зрения выполнения процедур над базой данных оба типа событий — об эволюционном изменении состояния объекта или о вступлении его в связь с другими объектами — эквивалентны в том смысле, что приводят к корректировке последнего элемента тройки (имя объекта, свойство объекта, значение свойства), а именно, присвоению нового значения некоторому атрибуту. Это присвоение означает не что иное, как установление новой связи между свойством и значением этого свойства, причем в качестве свойств могут выступать типы, а в качестве значений имена других объектов, с которыми устанавливается или разрывается связь рассматриваемого объекта. Поскольку установление новой связи, а значит, и разрушение старой, — результат некоторого события, переводящего объект в новое состояние, то отсюда следует, что связь реально существует на протяжении некоторого интервала времени $t_{i1}t_{i2}$, когда объект находится в состоянии S_i .

Таким образом, можно сказать, что типы связей отображают наличие связей между свойствами и их значениями или между разными типами объектов в том же смысле, что и в традиционных моделях данных, в частности в модели «сущность — связь». Однако принципиальным отличием рассматриваемой здесь модели является то, что в ней каждая связь дополнительно характеризуется двумя атрибутами времени, определяющими период ее реального существования. Это позволяет отображать в базе данных не только текущие связи, но и те, которые были установлены или разрушены за весь период моделирования, т. е. фактически отобразить всю историю изменения состояния и взаимодействия объектов, а значит, и динамику предметной области. Очевидно, что в предельном случае в качестве связи можно выделить в самостоятельное отношение любую пару «свойство — значение» из тройки (объект, свойство, значение) (аналогично тому как это делается при построении бинарных моделей данных [19]), дополнив ее

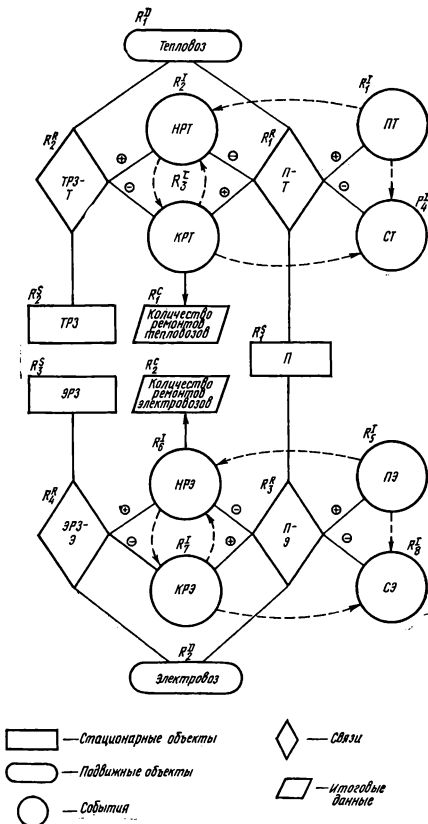


Рис. 3.1.1

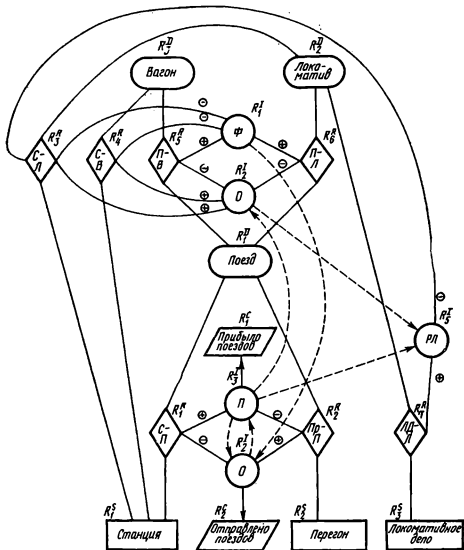


Рис. 3.1.2

атрибутами времени t_{11} и t_{12} . Однако при проектировании баз данных для конкретных приложений для обеспечения быстрой и удобной обработки данных необходимо учитывать специфику исторических запросов, с тем чтобы создавать и поддерживать отдельно связи «свойство — значение» только для тех свойств, изменение которых необходимо прослеживать во времени. Если говорить о моделях данных для систем управления железнодорожным транспортом, то это касается в первую очередь связей между разными типами объектов, таких, например, как поезд — станция, поезд — перегон, поезд — вагон, поезд — локомотив, локомотив — депо и т. п. Вместе с тем надо отметить, что при необходимости новая связь может быть введена в модель в любой момент времени.

На рис. 3.1.1 приведены основные типы сущностей, событий и связей базы данных о локомотивах. Стационарными объектами являются предприятия (Π) железнодорожного транспорта (R_1^S), тепловозо-

(ТПЗ) и электровозо-ремонтные (ЭПЗ) заводы (R_2^S и R_3^S). Подвижные объекты — тепловозы (R_1^D) и электровозы (R_2^D). В результате событий приобретение (ПТ, ПЭ) и списание (СТ, СЭ) тепловозов и электровозов (R_1^I и R_4^I , R_5^I и R_8^I); начало (НРТ, НРЭ) и окончание (КРТ, КРЭ) ремонтов тепловозов и электровозов (R_2^I и R_3^I , R_6^I и R_7^I) устанавливаются и разрываются связи предприятие — тепловоз (П — Т) (R_1^R) и тепловозо-ремонтный завод — тепловоз (ТПЗ — Т) (R_2^R) или предприятие — электровоз (П — Э) (R_3^R) и электровозоремонтный завод — электровоз (ТПЭ — Э) (R_4^R). Дуги, соединяющие типы событий и связей, помечены знаком плюс, если данный тип связи устанавливается в результате данного типа события, и знаком минус — если разрушается. Пунктирными линиями показана допустимая очередность событий.

Пример базы данных о перевозках показаны на рис. 3.1.2. Из него видно, что связи между стационарными и подвижными объектами, такие, как станция — поезд (R_3^R), перегон — поезд (R_2^R), станция — локомотив (R_3^R), станция — вагон (R_4^R), поезд — вагон (R_5^R), поезд — локомотив (R_6^R), депо — локомотив (R_7^R), устанавливаются и разрушаются в результате событий: формирование (R_1^I), отправление (R_2^I), прибытие (R_3^I) и расформирование (R_4^I) поезда, а также ремонт локомотива (R_5^I).

Рассмотрим подробнее состав и принципы обработки данных в модели «сущность — событие — связь», выделив три основные компоненты модели данных: собственно структуры данных об объектах, событиях и связях; ограничения целостности, определяющие множества допустимых состояний отдельных объектов, и базы данных в целом, а также разрешенные эволюцию, миграцию и взаимодействие объектов; операции манипулирования данными, обеспечивающие обработку запросов к базе данных.

3.1.1. СТРУКТУРЫ ДАННЫХ

1. Данные о стационарных объектах представляются в составе отношений вида $R^S (N^S, A^S)$, где N^S — уникальное имя (идентификатор) стационарного объекта; $A^S = \{A_1^S, \dots, A_n^S\}$ — множество атрибутов, характеризующих N^S . Имя N^S может использоваться в качестве первичного ключа при отсутствии в составе СУБД средств поддержки заменителей.

Пример состава данных о стационарных объектах для базы данных о локомотивах (рис. 3.1.1) приведен в табл. 3.1.1.

2. Данные о подвижных объектах представляются в составе отношений вида

$$R^D (N^D, N^I, T^I [, N_A^S, N_F^S, N_T^S, N^S], A^D).$$

Здесь N^D — уникальное имя (идентификатор) подвижного объекта, используемое при необходимости в качестве первичного ключа; N^I

и T^I — код и время последнего по времени события, в котором принимал участие N^D ; $A^D = \{A_1^D, \dots, A_m^D\}$ — множество атрибутов, характеризующих другие свойства N^D .

Для многих моделей, в том числе и для базы данных о локомотивах (рис. 3.1.1), такого состава атрибутов может оказаться достаточно для описания подвижных объектов. Пример отношения для этой базы данных показан в табл. 3.1.2.

Однако в ряде случаев, в частности для базы данных о перевозках (см. рис. 3.1.2), целесообразно ввести в состав некоторых R^D ряд дополнительных атрибутов. Так, если подвижный объект приписан к одному из стационарных, который считается его постоянным владельцем (например, локомотивы и локомотивные депо), в R^D имеет смысл ввести определяющий владельца атрибут N_A^S . Для лучшего отображения миграции некоторых объектов (например, поездов) удобно иметь в составе также атрибуты N_P^S , N_T^S и N^S , определяющие соответственно стационарные объекты отправления, назначения и текущего местонахождения N^D .

При моделировании данных о подвижных объектах следует учитывать, что эти объекты могут либо все время находиться в границах отображаемой предметной области, либо периодически уходить за ее пределы и вновь возвращаться, либо, появившись однажды, исчезать

Таблица 3.1.1

R_1^S (предприятия)

Наименование N_1^S	Город A_{11}^S	Вагонооборот A_{12}^S
П1	Киев	250
П2	Киев	1320
П3	Харьков	4100
П4	Одесса	2580
П5	Киев	3400
П6	Харьков	295

R_2^S (тепловозоремонтные заводы)

Наименование N_2^S	Город A_{21}^S
ТРЗ1	Киев
ТРЗ1	Одесса

R_1^D (тепловозы)

Номер N_1^D	Событие		Серия A_{11}^D	Год изготовления A_{12}^D
	Код N^I	Дата T^I		
2510	НРТ14	14.08	ТЭМ2	1975
3800	КРТ5	12.04	ТЭМ2	1986
1440	КРТ10	22.07	ТЭМ3	1974
7830	КРТ8	08.06	ТЭМ3	1982
9510	СТ1	05.07	ТЭМ3	1985
1850	НРТ12	17.07	ТГМ6	1976
2470	ПТ6	12.01	ТГМ6	1978
4800	КРТ11	26.08	ТГМ6	1981
7190	ПТ11	14.05	ТГМ6	1988
3010	СТ2	13.08	ТГМ4	1968

Таблица 3.1.2

через некоторое время навсегда. Например, локомотивы и вагоны могут выходить за пределы отдельной железной дороги или промышленного предприятия и вновь появляться на них через некоторое время, поезд — расформировываться, те же локомотивы и вагоны — списываться и т. п. В любом случае в базе данных должны быть представлены данные как об объектах, находящихся в границах моделируемого региона на текущий момент времени (текущие данные), так и об объектах, находившихся в нем в прошлом (исторические данные). Это обеспечивает реализацию широкого спектра исторических запросов и разнообразную статистическую обработку данных. Учитывая, что частота обращения к текущим данным, как правило, всегда гораздо выше, чем к историческим, представляется эффективной двухуровневая организация памяти для представления данных о подвижных объектах с целью отделения относительно небольшой обычно части текущих данных от большого объема исторических. Текущие данные размещаются при этом в основной памяти (отношения R_T^D), а исторические — в архивной (отношения R_A^D), причем при выходе подвижного объекта за границы региона или прекращении его существования вообще данные о нем переписываются из R_T^D в R_A^D , т. е. из основной памяти в архивную.

3. Данные о событиях содержатся в составе отношений, принципиальная схема которых в общем случае выглядит следующим образом:

$$R^I(N^I, T^I, N_1^S, \dots, N_n^S, N_1^D, \dots, N_m^D, A^I).$$

Кортежи R^I для каждого типа событий содержат атрибуты, определяющие код события N^I , время T^I и множество других характеристик этого события, и устанавливают связь между участвовавшими в событии стационарными N_1^S, \dots, N_n^S и подвижными N_1^D, \dots, N_m^D объектами. Для некоторых событий могут быть введены два значения атрибута T^I , соответствующие временам начала и окончания события. Это целесообразно тогда, когда значения остальных атрибутов остаются одинаковыми как в начале, так и в конце события (например, такие события, как выгрузка и погрузка вагонов и др.). Соответствующий событию кортеж в R^I будет формироваться в этом случае на основании двух сообщений — о начале и об окончании события N^I .

Очевидно, что данные о событиях могут описывать не только взаимодействие нескольких объектов, но и обычные эволюционные изменения состояния отдельных стационарных или подвижных объектов. В любом случае с каждым из объектов в базе данных будет связана история его динамики (эволюции, миграции, взаимодействия) в виде множества данных о соответствующих событиях. В конкретных системах данные в R^I могут накапливаться и сохраняться на протяжении принятого технологического цикла или некоторого установленного периода времени. Как будет показано ниже, они позволяют восстановить ретроспективно хронологию происходящих изменений состояния стационарных и подвижных объектов, а также прогнозировать развитие ситуации.

Отношения базы данных о локомотивах, содержащие данные о некоторых событиях, приведены в табл. 3.1.3.

Таблица 3.1.3

 R_1^I (приобретение тепловоза)

Событие		Предприя- тие N_1^S	Номер тепловоза N_1^D	Тип A_{11}^I	Серия A_{12}^I
Код N_1^I	Дата T_1^I				
ПТ1	05.01	П1	2510	Тепловоз	ТЭМ2
ПТ2	07.01	П2	3800	»	ТЭМ2
ПТ3	09.01	П3	7830	»	ТЭМ3
ПТ4	11.01	П3	9510	»	ТЭМ3
ПТ5	12.01	П3	1850	»	ТГМ6
ПТ6	12.01	П3	2470	»	ТГМ6
ПТ7	14.01	П4	4800	»	ТГМ6
ПТ8	19.01	П5	3010	»	ТГМ4
ПТ9	15.03	П6	1440	»	ТЭМ3
ПТ10	14.05	П5	7190	»	ТГМ6

 R_2^I (начало ремонта тепловоза)

Событие		Номер тепло- воза N_1^D	Предприятие N_1^S	Завод N_2^S	Вид ремонта A_{21}^I
Код N_2^I	Дата T_2^I				
НРТ4	21.01	3800	П2	ТР31	КР1
НРТ7	06.04	2510	П1	ТР31	КР1
НРТ8	26.04	7830	П3	ТР32	КР2
НРТ9	03.06	3010	П5	ТР32	КР1
НРТ10	14.06	1440	П6	ТР31	КР2
НРТ11	10.07	4800	П4	ТР31	КР1
НРТ12	17.07	1850	П3	ТР32	КР1
НРТ14	14.08	2510	П1	ТР31	БПР

 R_3^I (конец ремонта тепловоза)

Событие		Номер тепло- воза N_1^D	Завод N_2^S	Предприятие N_1^S	Стоимость ре- монта A_{31}^I
Код N_3^I	Дата T_3^I				
КРТ5	12.04	3800	ТР31	П2	11.8
КРТ6	10.05	2510	ТР31	П1	22.2
КРТ8	08.06	7830	ТР32	П3	14.3
КРТ9	13.07	3010	ТР32	П5	28.1
КРТ10	22.07	1440	ТР31	П6	17.8
КРТ11	26.08	4800	ТР31	П4	21.9

 R_4^I (списание)

Событие		Предприятие N_1^S	Номер тепловоза N_1^D	Тип A_{41}^I
Код N_4^I	Дата T_4^I			
СТ1	05.07	П3	9510	Тепловоз
СТ2	13.08	П5	3010	»

Таблица 3.1.4

 R_1^R (предприятие—тепловоз)

Предприятие N_1^S	Номер тепловоза N_1^D	Событие 1		Событие 2	
		Код N_1^I	Дата T_1^I	Код N_1^F	Дата T_1^F
П1	2510	ПТ1	05.01	НРТ7	06.04
П2	3800	ПТ2	07.01	НРТ4	21.02
П3	7830	ПТ3	09.01	НРТ8	26.04
П3	9510	ПТ4	11.01	СТ1	05.07
П3	1850	ПТ5	12.01	НРТ12	17.07
П3	2470	ПТ6	12.01	—	∞
П4	4800	ПТ7	14.01	НРТ11	10.07
П5	3010	ПТ8	19.01	НРТ9	03.06
П6	1440	ПТ9	15.03	НРТ10	14.06
П2	3810	КРТ5	12.04	—	∞
П1	2510	КРТ6	10.05	НРТ14	14.08
П5	7190	ПТ10	14.05	—	∞
П3	7830	КРТ8	08.06	—	∞
П5	3010	КРТ9	13.07	СТ2	13.08
П6	1440	КРТ10	22.07	—	∞
П4	4800	КРТ11	26.08	—	∞

 R_2^R (завод—тепловоз)

Завод N_2^S	Номер тепловоза N_1^D	Событие 1		Событие 2	
		Код N_1^I	Дата T_1^I	Код N_1^F	Дата T_1^F
ТР31	3800	НРТ4	21.02	КРТ5	12.04
ТР31	2510	НРТ7	06.04	КРТ6	10.05
ТР32	7830	НРТ8	26.04	КРТ8	08.06
ТР32	3010	НРТ9	03.06	КРТ9	13.07
ТР31	1440	НРТ10	14.06	КРТ10	22.07
ТР31	4800	НРТ11	10.07	КРТ11	26.08
ТР32	1850	НРТ12	17.07	—	∞
ТР31	2510	НРТ14	14.08	—	∞

4. Данные о связях представляются в составе отношений вида

$$R^R(N^S, N^D, N_S^I, T_S^I, N_F^I, T_F^I).$$

Как и в модели сущность — связь [19], каждая реализация связи (кортеж) в R^R содержит имена N^S и N^D связываемых объектов (в принципе, возможны пары N^S, N^S и N^D, N^D). Но, кроме того, в рассматриваемой модели каждая связь характеризуется дополнительно кодами N_S^I, N_F^I и временами T_S^I, T_F^I событий, в результате которых эта связь соответственно устанавливается или разрывается. При этом отсутствие значений N_F^I и T_F^I (точнее, T_F^I в этом случае для удобства обработки присваивается значение ∞) означает, что кортеж R^R описывает существующую на данный момент времени связь. Если T_F^I имеет кон-

кретное, отличное от ∞ значение t , то это означает, что связь носит характер исторический, т. е. она существовала на интервале времени $T_S^i T_F^i$.

Для ускорения обработки запросов о текущем состоянии объектов данные об исторических связях могут быть вынесены в отдельную, архивную, область памяти и сохраняться там на протяжении требуемого времени.

Отношения базы данных о локомотивах, содержащие данные о некоторых типах связей, приведены в табл. 3.1.4.

Отношения R^R , описывающие связи разных типов объектов, позволяют удобно моделировать структуры характерных для железнодорожного транспорта объектов типа агрегата-покрытия [23], примером которого может служить такой тип объекта, как поезд, представляющий собой совокупность объектов других типов, а именно локомотива и вагонов. Как видно из рис. 3.1.2, формированию поезда соответствует установление в модели связей типа поезд — локомотив и поезд — вагон, т. е. каждая реализация объекта поезд будет состоять из некоторой совокупности реализаций этих связей. Используя операцию естественного соединения всех отношений R^R , относящихся к одному типу агрегата — покрытия, можно получить полный состав каждой реализации объекта такого типа. Наличие же в составе данных о связях атрибутов времени существования связи обеспечивает прослеживание динамики возникновения и исчезновения реализаций объектов — агрегатов (в данном случае — формирования и расформирования поездов).

Наконец, с помощью связей можно моделировать и достаточно сложные иерархические структуры стационарных объектов, также относящихся к типу агрегатов — покрытий. Например, рассматривая отделение дороги как совокупность станций (C) и перегонов (Π) (рис. 3.1.3), можно с помощью связей типа отделение — станция ($O - C$) $R_1^R (N_0^S, N_1^S)$ и отделение — перегон ($O - \Pi$) $R_2^R (N_0^S, N_2^S)$ для каждой реализации объекта отделение дороги поставить в соответствие подмножества реализаций объектов станция и перегон, входящих в состав этого отделения.

При таком подходе создаются предпосылки моделирования сложных структур, используемых в объектных базах данных. О концептуальной

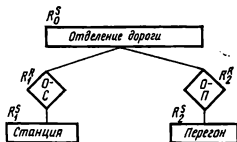


Рис. 3.1.3

Таблица 3.1.5

Π_1^C (общий объем ремонтов)

Завод N_2^S	Ремонт		
	Вид A_{21}^I	Количество ΣA_{11}^C	Общая стоимость, тыс. руб. ΣA_{31}^I
ТР31	КР1	3	55,9
ТР31	КР2	1	28,1
ТР32	КР2	1	14,3

близости рассматриваемой здесь модели к объектным, например к описанной в [42], свидетельствует простота перехода от одной к другой. Так, если сгруппировать в множестве $\{k_1\}$ и $\{k_2\}$ имена N_1^S и N_2^S или соответствующие заменители кортежей из R_1^S и R_2^S , описывающих входящие в N_0^S реализации, и ввести эти множества как молекулярные атрибуты в кортеж отношения R_0^S , описывающий N_0^S , то рассматриваемая модель трансформируется в модель, предлагаемую в [42] и представленную выражениями (2.4.1) и (2.4.2) в параграфе 2.4.

5. *Итоговые данные* накапливаются в отношениях вида

$$R^C (\{N_1^S, \dots, N_n^S, N_1^D, \dots, N_m^D\} A^C),$$

кортежи которых для каждой совокупности взаимодействующих в процессе отдельного вида событий стационарных и подвижных объектов содержат множество $A^C = \{\Sigma A_1^C, \dots, \Sigma A_k^C\}$ значений некоторых итоговых показателей, характеризующих это взаимодействие за определенный интервал времени. Например, для базы данных о локомотивах это будет общее количество и суммарная стоимость завершенных ремонтов по видам ремонтов и тепловозоремонтным заводам (табл. 3.1.5), полученные в результате суммирования соответствующих данных из R_2^I и R_3^I (см. табл. 3.1.3).

Очевидно, что данные отношений R^C достаточно просто генерируются из данных отношений первых четырех групп, т. е. R^S , R^D , R^I и R^R . Однако в ряде систем, где невозможно обеспечить своевременную обработку данных о событиях, т. е. реализовать пооперационное отслеживание динамики объектов, итоговые данные могут вводиться в базу данных после предварительной ручной подготовки.

В заключение рассмотрения структур данных отметим, что схемы отношений для разных типов объектов в рамках отдельных рассмотренных выше пяти категорий данных могут отличаться друг от друга степенью за счет разного состава атрибутов в A . Значит, в общем случае данные по каждой из категорий будут представлены в модели соответствующими множествами отношений $R^S = \{R_1^S, \dots, R_n^S\}$, $R^D = \{R_1^D, \dots, R_m^D\}$, $R^I = \{R_1^I, \dots, R_l^I\}$, $R^R = \{R_1^R, \dots, R_k^R\}$, $R^C = \{R_1^C, \dots, R_f^C\}$.

3.1.2. ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ

Рассматриваемая модель данных содержит четыре основные группы ограничений целостности.

1. *Явные ограничения целостности*, относящиеся к отдельным атрибутам, такие, как ограничения на значения и агрегатные ограничения. Они декларируются обычно с помощью спецификаций языка определения данных используемой СУБД.

2. *Ограничения, относящиеся к структурам объектов*. Они определяют допустимое членство в покрытии, т. е. показывают, какие типы объектов могут входить в реализацию объектов типа агрегата-покрытия. Ограничения описываются с помощью бинарных отношений

RID		
R^D	R_i^I	R_j^I
R_1^D	R_1^I	R_2^I
R_2^D	R_1^I	R_4^I
R_3^D	R_2^I	R_3^I
R_4^D	R_3^I	R_2^I
R_1^D	R_3^I	R_4^I

а

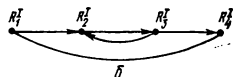


Рис. 3.1.4

RS (R_{AGR}^S, R^S) и RD (R_{AGR}^D, R^D), где R_{AGR}^S и R_{AGR}^D — имена типов агрегатов-покрытий, а R^S и R^D — имена типов объектов, входящих в них в качестве членов. Описывая разрешенные (обычно уже существующие в модели) иерархии стационарных и подвижных объектов, RS и RD , вместе с тем позволяют удобно моделировать вновь возникающие в процессе эволюции предметной области сложные структуры с помощью определения новых абстракций в терминах уже существующих.

Ниже приведен состав RS для базы данных, показанной в табл. 3.1.3, и RD для базы данных — в табл. 3.1.2.

RS		RD	
R_{AGR}^S	R^S	R_{AGR}^D	R^D
R_0^S	R_1^S	R_1^D	R_2^D
R_0^S	R_2^S	R_1^D	R_3^D

3. Ограничения технологической целостности, определяющие разрешенную динамику объектов. Для каждого типа объектов могут быть определены некоторое множество допустимых состояний и правила переходов из одного в другое. Поскольку переходы являются обычно следствием выполнения событий, то динамику объектов можно представлять как некоторую последовательность событий, выглядящую схематически в виде вектора, однозначно определяющего очередность их выполнения, либо дерева или ориентированного графа, описывающего возможность альтернативного развития событий. Ограничения на переходы описываются тернарными отношениями RIS (R^S, R_i^I, R_j^I) и RID (R^D, R_i^I, R_j^I), кортежи которого указывают для объекта R^S или R^D возможность выполнения события R_j^I , если ему предшествовало событие R_i^I . Очевидно, что из пар R_i^I, R_j^I для каждого типа объекта R^S или R^D нетрудно установить всю возможную последовательность событий (в виде вектора, дерева или сети), в которых этот тип объекта может участвовать.

Например, допустимую последовательность событий, которая может произойти с объектом тепловоз (она показана на рис. 3.1.1), можно

представить графом (рис. 3.1.4, б), соответствующим фрагменту отношения RID (рис. 3.1.4, а). Интерпретация этой последовательности очевидна — после приобретения локомотива может следовать начало его ремонта или списание, после начала ремонта только конец ремонта, после конца ремонта — начало следующего ремонта или списание.

4. *Ограничения, определяющие возможность участия объектов в событиях*, перечисляются в составе бинарных отношений RTS (R^S, R^I) и RTD (R^D, R^I) и устанавливают соответствие между типами объектов R^S и R^D и видом события R^I , в котором эти объекты могут быть задействованы. Из RTS и RTD несложно определить для каждого типа объекта (R^S или R^D) все множество типов событий, в которых он может участвовать, а также установить все типы объектов, которые могут быть задействованы в отдельном виде события. Для этого достаточно выполнить операцию селекции над RTS и/или RTD , задавая в качестве условия конкретные значения R^S, R^D и R^I . Например, как видно ниже из отношений RTS и RTD для базы данных о локомотивах (см. рис. 3.1.1), стационарный объект R_3^S может участвовать в событиях R_6^I и R_7^I , в событиях R_2^I могут участвовать объекты R_1^S, R_2^S , и R_1^D и т. п.

Таким образом, RTS и RTD фактически устанавливают разрешенное взаимодействие между разными типами объектов на уровне выполняемых с ними операций, а следовательно, и определяют возможность установления и поддержания в модели соответствующих связей R^R .

RTS	RTD
$R^S \quad R^I$	$R^D \quad R^I$
$R_1^S \quad R_1^I$	$R_1^D \quad R_1^I$
$\vdots \quad \vdots$	$R_1^D \quad R_2^I$
$R_1^S \quad R_8^I$	$R_1^D \quad R_3^I$
$R_2^S \quad R_2^I$	$R_1^D \quad R_4^I$
$R_2^S \quad R_3^I$	$R_2^D \quad R_5^I$
$R_3^S \quad R_6^I$	$R_2^D \quad R_6^I$
$R_3^S \quad R_7^I$	$R_2^D \quad R_7^I$
	$R_2^D \quad R_8^I$

Следует отдельно сказать еще об одном виде ограничений целостности, а именно, об *ограничениях, специфицирующих разрешенную миграцию подвижных объектов между стационарными и в их границах*, т. е. возможность поступления и нахождения определенного типа подвижного объекта на территории некоторого стационарного. Эти ограничения сознательно не выделены в самостоятельную группу, так как присутствуют неявно в составе RTS и RTD . Действительно, разрешенное взаимодействие стационарного и подвижного объектов в процессе некоторого события определяет само по себе и возможность нахождения последнего на территории первого. Однако, при необ-

ходимости эту группу ограничений можно представить и явно в составе, например, бинарного отношения $RSD (R^S, R^D)$.

Как видно, все атрибуты отношений, описывающих ограничения целостности, определены на одном общем домене, включающем все имена отношений из множеств R^S, R^D, R^I . Рассматриваемая система ограничений целостности, во многом аналогичная предлагаемой в расширенной реляционной модели RM/T , существенно усиливает семантику модели, позволяет не только отслеживать, но и контролировать динамику объектов и блокировать выполнение некорректных процедур обработки данных.

3.1.3. ПРИНЦИПЫ МАНИПУЛИРОВАНИЯ ДАННЫМИ

Будем рассматривать процедуры манипулирования данными на примере основных обобщенных типов запросов, для которых приводятся принципиальные схемы их обработки, определяющие в целом стратегию выборки данных. Тексты запросов приведены на языке реляционной алгебры, причем порядок операций достаточно условный и может не соответствовать требованиям оптимизации запросов. Читателю, знакомому с конкретными языками манипулирования данными, не составит труда перевести приводимые ниже запросы на тот или иной язык навигационного или спецификационного типа [19].

1. Запросы о наличии и текущем состоянии стационарных и подвижных объектов обрабатываются обычным образом с использованием операций селекции и проекции, выполняемых над R^S и R^D .

2. Запросы на получение данных о текущей дислокации подвижных объектов относительно стационарных обрабатываются по схеме

$$W \leftarrow \pi_X (\sigma_Q (R^S \bowtie R^R \bowtie R^D)), \quad (3.1.1)$$

т. е. на интегрированном отношении, полученном в результате выполнения операции естественного соединения \bowtie , с помощью селекции σ по условию Q с последующей проекцией π на множество атрибутов X можно получить результирующее отношение W , содержащее ответ на любой запрос указанного типа. Здесь и далее подразумевается, что естественное соединение выполняется на основании значений одноименных атрибутов соединяемых отношений. Примером будет такой запрос.

3.1. «Перечислить номера и серии тепловозов, работающих в настоящее время на предприятиях Харькова». Обработка этого запроса в общем то тривиальна и определяется выражением

$$W \leftarrow \pi_{N_1^S, N_1^D, A_{11}^D} (\sigma_{A_{11}^S = \text{ХАРЬКОВ} \wedge T_F = \infty} (R_1^S \bowtie R_1^R \bowtie R_1^D)).$$

В результате получим такое W :

N_1^S	N_1^D	A_{11}^D
П3	2470	ТГМ6
П3	7830	ТЭМ3
П6	1440	ТЭМ3

Следует заметить, что приведенная схема запроса пригодна лишь тогда, когда он касается одного типа стационарных и одного типа подвижных объектов. Если это ограничение снять, то схема запроса трансформируется к виду

$$W \leftarrow \pi_X (\sigma_Q (\bar{\cup} / (R_2^S \bowtie R_2^R \bowtie R_2^D))), \quad (3.1.2)$$

где $\bar{\cup}$ — операция внешнего объединения (см. параграф 2.1), позволяющая обрабатывать отношение разных структур. Косая черта / означает, что объединение выполняется для всех триад соединений отношений из подмножеств $R_2^S \subseteq R^S$, $R_2^R \subseteq R^R$ и $R_2^D \subseteq R^D$. Состав R_2^S , R_2^R и R_2^D определяется сформулированными в запросе информационными потребностями.

Рассмотрим такой запрос.

3.2. «Перечислить номера тепловозов, работающих в настоящее время на предприятиях с вагонооборотом более 3000 вагонов, а также находящихся в ремонте на заводах Киева»:

$$W \leftarrow \pi_{N_1^D, N_1^S, N_2^S, A_{21}^S} (\sigma_{A_{12}^S > 3000 \wedge T_{F-\infty}^I \wedge A_{21}^S = \text{КИЕВ}} \times \\ \times ((R_1^S \bowtie R_1^R \bowtie R_1^D) \bar{\cup} (R_2^S \bowtie R_2^R \bowtie R_2^D))).$$

В данном случае $R_2^S = \{R_1^S, R_2^S\}$, $R_2^R = \{R_1^R, R_2^R\}$, $R_2^D = \{R_1^D, R_2^D\}$. Однако из всех возможных триад соединений отношений из R_2^S , R_2^R и R_2^D в запросе допускаются только те, которые разрешены соответствующими ограничениями целостности RTS и RTD . В этом смысле, как нетрудно проверить, запрос 3. 2 корректен, и результатом его обработки будет отношение W :

N_1^D	N_1^S	N_2^S	N_{21}^S
7190	П5	—	—
2470	П3	—	—
7830	П3	—	—
2510	—	ТР31	Киев

В случае нарушения ограничений целостности RTS и RTD обработка запроса прекращается с выдачей пользователю соответствующего уведомления о некорректности запроса. Таким образом, можно избежать выполнения дорогостоящих операций естественного соединения, когда такая попытка предпринимается только на основании значений атрибутов, но без учета семантики модели. Например, обработка запроса «Какие тепловозы находятся на ремонте на электровозоремонтных заводах», т. е.

$$W \leftarrow \pi_{N_1^D} (R_3^S \bowtie R_2^R \bowtie R_1^D),$$

была бы заблокирована из-за бессмысленности этого соединения (если тепловозы не могут ремонтироваться на электровозоремонтных заводах).

Очевидно, что в случае отсутствия в составе ЯМД операции внешнего объединения запрос типа (3.1.2) распадается на ряд запросов типа (3.1.1).

Рассмотрим пример еще одного запроса о текущем состоянии объектов предметной области, но требующих обработки данных о событиях.

3.3. «Перечислить номера тепловозов, работающих в настоящее время на предприятиях Харькова после ремонтов КР2 стоимостью более 15 тыс. руб.».

$$W \leftarrow \pi_{N_1^D, A_{21}^I, A_{31}^I} (\sigma_{A_{12}^S = \text{ХАРЬКОВ} \wedge T_F^I = \infty \wedge A_{21}^I = \text{КР2} \wedge A_{31}^I > 15} \times \\ \times (R_1^D \bowtie R_2^I \bowtie R_2^R \bowtie R_3^I \bowtie R_1^S)).$$

Здесь в соединении участвуют еще R_2^I и R_3^I , содержащие атрибуты A_{21}^I — вид ремонта и A_{31}^I — стоимость ремонта. Результатом будет:

N_1^D	A_{21}^I	A_{31}^I
1440	КР2	17.8

3. Запросы об исторической дислокации подвижных объектов, т. е. об их расположении относительно стационарных на некоторый момент времени в прошлом, обрабатываются в целом по схемам, аналогичным (3.1.1) и (3.1.2). Значение времени может указываться явно, как в следующих двух запросах.

3.4. «Перечислить номера тепловозов и города, на заводах которых они находились в ремонте КР1 1 марта»:

$$W \leftarrow \pi_{N_1^D, A_{21}^S} (\sigma_{A_{21}^S = \text{КР1} \wedge T_S^I \leq 01.03 < T_F^I} (R_2^S \bowtie R_2^D \bowtie R_1^D \bowtie R_1^I)),$$

В результате получим

N_1^D	A_{21}^S
3800	Киев

3.5. «Перечислить предприятия, на которых работали, заводы, на которых ремонтировались, и виды ремонтов для тепловозов серии ТЭМЗ на 15 мая».

$$W \leftarrow \pi_{N_1^S, N_1^D, N_2^S, A_{21}^I} (\sigma_{A_{11}^D = \text{ТЭМЗ} \wedge T_S^I \leq 15.05 < T_F^I} \times \\ \times ((R_1^R \bowtie R_1^D) \bar{\cup} (R_2^S \bowtie R_2^D \bowtie R_1^D \bowtie R_1^I))).$$

Результатом будет

N_1^S	N_1^D	N_2^S	A_{21}^I
ПЗ	9510	—	—
П6	1440	—	—
—	7830	ТР32	КР2

В некоторых исторических запросах значение времени может указываться неявно, как, например, в таком запросе:

3.6. «Перечислить тепловозы серии ТЭМЗ, оставшиеся в работе на предприятии ПЗ в то время, когда хотя бы один тепловоз этого же предприятия и этой же серии был отправлен в ремонт КР2». Такой запрос можно разбить на два следующих.

3. 7. «Определить время, когда был отправлен в ремонт КР2 тепловоз серии ТЭМ3 предприятия ПЗ»:

$$W \leftarrow \pi_{T_2^I} (\sigma_{N_1^S = \text{ПЗ} \wedge A_{11}^D = \text{ТЭМ3} \wedge A_{21}^I = \text{КР2}} (R_1^R \times R_2^I \times R_1^D))_{(N_1^I = N_2^I)}$$

Полученное в результате значение времени $T_2^I = 26.04$ подставляется в составе условия Q в запрос 3. 8.

3. 8. «Перечислить номера тепловозов серии ТЭМ3, находившихся в работе на предприятии ПЗ 26 апреля»:

$$W \leftarrow \pi_{N_1^D} (\sigma_{N_1^S = \text{ПЗ} \wedge A_{11}^D = \text{ТЭМ3} \wedge T_5^I \leq 26.04 < T_1^I} (R_1^R \times R_1^D))$$

Результат:

$$N_1^D \\ 9510$$

4. Запросы на получение данных об эволюции (траектории динамики) объектов, состоящей из перечня событий, в которых участвовал этот объект на протяжении некоторого интервала времени или же всего периода своего существования, обрабатываются по следующей обобщенной схеме:

$$W \leftarrow \pi_X (\sigma_Q (\overline{U} / (R_2^S \times R_2^I)))$$

или

$$W \leftarrow \pi_X (\sigma_Q (\overline{U} / R_2^D \times R_2^I)) \quad (3.1.3)$$

Указывая в условии Q значение интервала времени $t_1 \leq T^I \leq t_2$, можно проследить пооперационно упорядоченную во времени динамику изменения состояния интересующего объекта за период с t_1 до t_2 .

Приведем в качестве примера такой запрос.

3. 9. «Перечислить даты и типы событий, в которых участвовал тепловоз № 3010, за период с 1 февраля по 1 августа»:

$$W \leftarrow \pi_{N^I, T^I} (\sigma_{N_1^D = 3010 \wedge T^I \geq 01.02 \wedge T^I < 01.08} \times \\ \times ((R_1^D \times R_1^I) \overline{U} (R_1^D \times R_2^I) \overline{U} (R_1^D \times R_3^I) \overline{U} (R_1^D \times R_4^I)))$$

В результате обработки получим

N^I	T^I
ПТ8	19.01
НРТ9	03.06
КРТ9	13.07
СТ2	13.08

5. Запросы о взаимодействии объектов за некоторый период времени обрабатываются по схеме

$$W \leftarrow \pi_X (\sigma_Q (R^S \times R^R \times R^D \times R^I)) \quad (3.1.4)$$

и позволяют получить данные о том, какие операции и с какими подвижными объектами выполнялись на некоторых стационарных,

или — на каких стационарных объектах, когда и в каких операциях участвовали определенные подвижные объекты и т. п.

Рассмотрим два примера наиболее характерных запросов такого типа.

3. 10. «Перечислить номера и серии локомотивов предприятий Киева, а также даты их поступления в ремонт КР1 на заводы Одессы за период с 1 апреля по 1 августа»:

$$W \leftarrow \pi_{N_1^D, A_{11}^D, N_1^S, A_{21}^I} (\sigma_{A_{11}^S = \text{КИЕВ} \wedge A_{21}^I = \text{КР1} \wedge T_2^I \geq 01.04 \wedge T_2^I < 01.08} \times (R_1^S \times R_2^I \times R_2^S \times R_1^D)).$$

Ответом на этот запрос будет отношение

N_1^D	A_{11}^D	N_1^S	A_{21}^I
3010	ТГМ4	П5	КР1

3. 11. «Перечислить виды и стоимость ремонтов, выполненных на заводах Киева для тепловозов постройки до 1980 г., за период с 1 апреля по 1 июня»:

$$W \leftarrow \pi_{A_{21}^I, A_{31}^I, N_1^D, A_{12}^D} (\sigma_{A_{21}^S = \text{КИЕВ} \wedge A_{12}^D < 1980 \wedge T^I \geq 01.04 \wedge T^I < 01.06} \times (R_2^S \times R_2^I \times R_2^R \times R_3^I \times R_1^D)).$$

$(N_1^S = N_2^I) (N_1^I = N_2^S)$

В результате обработки получаем

A_{21}^I	A_{31}^I	N_1^D	A_{12}^D
КР1	22.15	2510	1975

6. Запросы на получение итоговых данных просто реализуются на отношениях R^C или, если последние не материализуются в базе данных, выполняются с помощью использования встроенных арифметических и агрегатных функций типа *SUM*, *COUNT* и др. в запросах (3.1.1) — (3.1.4). Выбор конкретного варианта реализации зависит от временных ограничений на выполнение запросов.

7. Запросы на ввод и обновление данных. Принципиальной особенностью предлагаемой модели данных является то, что обычному пользователю возможность обновления базы данных доступна только через обращение к специальным программным процедурам обработки данных о событиях. Эти процедуры программируются разработчиками базы данных в конкретной предметной области и для каждого типа событий выполняют соответствующие функции контроля ограничений целостности, а также реализацию необходимых запросов на модификацию отношений R^S , R^D , R^I , R^R и R^C в соответствии с принятыми технологическими схемами.

Как показывает анализ, при информационном моделировании любой предметной области, в том числе и транспортных систем, все множество событий можно разделить на четыре основных типа — возникновение,

изменение состояния, исчезновение и перемещение объектов. Причем в первых трех типах могут участвовать как стационарные, так и подвижные объекты, а в последнем — только подвижные. В зависимости от типа события различаются состав атрибутов и интерпретация их значений в кортежах отношений R^I , кроме атрибутов N^I и T^I , трактуемых всегда однозначно. Соответственно различаются и схемы процедур обработки данных о событиях, хотя каждая из них в обязательном порядке включает как минимум две такие операции.

Перед внесением изменений в базу данных проверяется в соответствии с ограничениями целостности RIS и RID возможность наступления события типа R^I в зависимости от того, какое событие ему предшествовало. Проверяется также согласно RTS и RTD допустимость участия каждого из объектов в событии типа R^I , а следовательно, и их взаимодействие в рамках этого события. При нарушении ограничений целостности данные о событии не обрабатываются, а пользователю выдается соответствующее уведомление.

Помимо внесения изменений в отношения R^S , R^D , R^R и R^C кортеж с данными о событии включается в отношение R^I , т. е.

$$\downarrow R^I_{N^I, T^I, A^I_1, \dots, A^I_l}(n, t, a_1, \dots, a_l),$$

где N^I , T^I , A^I_1 , ..., A^I_l имена всех или некоторых атрибутов отношения R^I ; n , t , a_1 , ..., a_l — их значения в новом кортеже. Атрибутам, не перечисленным в составе A^I_1 , ..., A^I_l , присваиваются значения «не определено».

Рассмотрим несколько подробнее состав данных и особенности процедур их обработки для каждого из типов событий.

Возникновение объекта. Для стационарного объекта этот тип события характеризуется следующими атрибутами: $R^I(N^I, T^I, N^S, A^I)$. Атрибут N^S обозначает имя самого объекта. Атрибуты множества A^I определяют свойства нового объекта. Примером такого события может служить постройка локомотивного депо, открытие нового перегона и др.

Несколько иной состав атрибутов будет характеризовать возникновение подвижного объекта: $R^I(N^I, T^I, N^S, N^D_1 [N^D_2, \dots, N^D_m], A^I)$. В этом случае N^S обозначает имя стационарного объекта, на территории которого возникает подвижный объект N^D_1 . Причем если N^D_1 объект-агрегат (т. е. создается реализация типа агрегата-покрытия), то данные о событии могут включать имена его членов — N^D_2, \dots, N^D_m . Таким событием является, например, формирование поезда N^D_1 на станции N^S , в котором участвуют локомотив N^D_2 и вагоны N^D_3, \dots, N^D_m .

При обработке данных о событиях такого типа в отношении R^S или R^D записывается кортеж, компонентами которого являются атрибуты N^S или N^D_1 и A^I . Кроме того, формируется кортеж в отношении R^R , связывающем N^S с N^D_1 , а также если создается объект-агрегат,

кортежи в отношениях R^R , связывающих N_1^D с N_2^D , N_1^D с N_3^D и т. д., причем значению T'_F в R^R присваивается значение ∞ .

Изменение состояния стационарного или подвижного объекта.

Этот тип события характеризуется таким составом атрибутов: $R^I(N^I, T^I, \{N^S\}_{N^D}, A^I)$. N^S или N^D обозначает имя стационарного или подвижного объекта, а A^I — множество атрибутов, приобретающих в результате события новые значения. Примером события такого типа являются реконструкция станций, погрузка или выгрузка вагонов и др. Обработка данных о таких событиях сводится к обновлению значений атрибутов A^I в кортежах отношений R^S или R^D , описывающих соответствующий объект.

Исчезновение объекта. Для этого типа сообщений состав атрибутов такой: $R^I(N^I, T^I, \{N^S\}_{N^D}, A^I)$. Атрибуты N^S или N^D обозначают

имя исчезающего объекта, а A^I — обстоятельства исчезновения. Примерами событий этого типа являются закрытие станций, списание локомотивов, расформирование поездов и т. п. Их обработка предусматривает в первую очередь перезапись данных об исчезнувшем объекте из основной памяти (отношений R^S или R^D) в архивную (R^A_S или R^A_D). Затем в кортежах всех отношений R^R , описывающих связи исчезающего объекта, формируются значения $N'_F = N^I$ и $T'_F = T^I$, характеризующие код и время события, в результате которого эти связи разрушаются.

Перемещение объекта. Этот тип события характеризуется такими атрибутами: $R^I(N^I, T^I, N_1^S, N_2^S, N^D, A^I)$. Они обозначают имя подвижного объекта N^D , перемещающегося с одного стационарного объекта N_1^S на другой N_2^S . Таким событием является, например, отправление поезда N^D со станции N_1^S на перегон N_2^S . При обработке данных о событиях этого типа в кортеже отношения R^D , описывающего объект N^D , атрибуту N^S , определяющему текущее местонахождение N^D , присваивается значение атрибута N_2^S . Если N^D является объектом-агрегатом, то это же значение N_2^S присваивается атрибуту N^S всех кортежей отношений R^I , описывающих объекты-члены N^D . Кроме того, корректируются данные о связях таким образом, что в кортеже R^R , описывающем связь N_1^S с N^D , формируются значения $N'_F = N^I$ и $T'_F = T^I$, фиксирующие разрыв этой связи, в R^R , описывающем связь N_2^S с N^D , добавляется новый кортеж с $N'_S = N^I$, $T'_S = T^I$ и $T'_F = \infty$, фиксирующий установление новой связи. Разновидностью перемещения можно считать такое событие, как выход подвижного объекта за границы моделируемого региона (например, сдача поезда на соседнюю железную дорогу или на подъездной путь). В этом случае

в схему процедуры обработки включаются операции по перезаписи кортежа, описывающего N^D (а также кортежей отношений R_i^D , описывающих объекты-члены N^D , если N^D — объект-агрегат), из основной области памяти (R_T^D) в архивную (R_A^D).

Организация доступа обычных пользователей к обновлению базы данных только через процедуры обработки данных о событиях имеет два положительных свойства. Во-первых, обеспечивается автоматический контроль ограничений целостности, описывающих разрешенную динамику и взаимодействие объектов. Во-вторых, упрощается с точки зрения пользователей сам процесс внесения изменений в базу данных, так как ему (пользователю) для обработки данных о некотором событии достаточно инициировать выполнение соответствующей процедуры. Вместе с тем администратор базы данных, а также привилегированные пользователи имеют возможность выполнять при необходимости принудительное (без контроля ограничений целостности) обновление данных посредством реализации запросов непосредственно на языке манипулирования данными СУБД, неся, однако, за него полную ответственность.

Проведем в заключение краткий сопоставительный анализ предложенной здесь модели с рассмотренными в параграфе 2.2 моделями временных баз данных, коснувшись двух основных аспектов — отображения динамики объектов моделируемой предметной области и обеспечения при этом целостности базы данных.

Для отображения динамики в обоих случаях при изменении значений атрибута старое его значение сохраняется, а новое фиксируется вместе со временем изменения. Но если во временных базах данных для этого предусмотрено создание так называемых исторических копий состояния объекта в определенные моменты времени, то в модели «сущность — событие — связь» все данные об изменениях состояния объектов хранятся в виде записей о соответствующих событиях. Принципиальность этого различия состоит в том, что для модели сущность — событие — связь не требуется разработка специальных средств ЯОД и ЯМД для спецификации динамических атрибутов или объектов в целом и обработки исторических запросов. Для обеспечения обработки любого запроса о значении отдельного атрибута или состояния объекта в целом на некоторый момент времени t в прошлом необходимо лишь, чтобы для каждого динамического атрибута существовал соответствующий тип события, в результате которого изменяется значение этого атрибута. При этом необязательно, чтобы каждому динамическому атрибуту соответствовал отдельный тип события, т. е. в результате события одного типа могут изменяться значения не только одного, но и нескольких или даже всех динамических атрибутов. Обязательно, чтобы всякое изменение значения динамического атрибута было следствием обработки данных о событии, а не принудительного обновления. Только в этом случае можно восстановить значение любого атрибута или состояние объекта в целом на время t , обработав совместно данные об этом объекте и всех типах событий, в которых он участвовал и которые были последними, произошедшими с ним до наступления ука-

данного времени t . Все это делает модель сущность — событие — связь более гибкой и независимой в том смысле, что она не требует переопределения схемы базы данных, когда, например, некоторые атрибуты становятся динамическими. Иначе говоря, динамическим подразумевается любой атрибут, который фигурирует в схемах отношений R^t , описывающих события. При этом появление нового типа события фактически соответствует автоматическому присвоению статуса динамических всем входящим в его схему атрибутам и не требует их специального переопределения, как, например в [27].

Рядом преимуществ обладает, на наш взгляд, модель сущность — событие — связь и с точки зрения обеспечения целостности базы данных. Во-первых, динамика объектов не просто отображается, но и контролируется с помощью ограничений RIS и RID , определяющих разрешенную последовательность событий, а следовательно, и допустимость перехода базы данных из состояния D_i в состояние D_j при обработке данных об этих событиях. Во-вторых, что особенно важно, отображается не просто история эволюционного развития каждого объекта в отдельности, а и динамика взаимодействия (в том числе и миграции) разных объектов в рамках разных событий, причем взаимодействия не произвольного, а регламентированного и контролируемого с помощью специальных ограничений целостности RTS и RTD .

Таким образом, модель сущность — событие — связь можно рассматривать как многоцелевое универсальное средство организации баз данных в системах проектирования, моделирования и управления, ориентированных на железнодорожный транспорт. Вполне возможно, что рассмотренные принципы моделирования данных могут оказаться полезными и при разработке базы данных в других предметных областях, для которых характерны в большей или меньшей мере перечисленные в начале этого параграфа специфические особенности. Вместе с тем необходимо отметить, что реализация некоторых концепций может столкнуться с определенными трудностями (в первую очередь это сложность обеспечения обработки данных о событиях в реальном масштабе времени), которые необходимо учитывать при разработке моделей конкретных баз данных.

3.2. ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА СХЕМ ТРАНСПОРТНЫХ ОБЪЕКТОВ

Транспорт вообще и железнодорожный в частности является одной из сфер, наиболее перспективных с точки зрения применения систем баз видеоданных. На их основе могут быть реализованы высокопроизводительные системы проектирования транспортных объектов, эффективные системы моделирования и управления, широко использующие средства и устройства машинной графики. В этом параграфе рассмотрим модель данных для графических приложений, ориентированную на такую предметную область, как железнодорожный транспорт.

Принципиальные концепции построения баз данных в графических системах — независимость и интерпретация графических данных — были изложены еще в [37] и дальнейшее их развитие (напри-

мер, [38—40]) связано в основном с использованием более совершенных программных средств ведения баз данных. Однако рассматриваемые в этих системах модели данных сильно упрощены, что затрудняет практическое их использование для решения достаточно сложных задач проектирования на транспорте. Здесь следует еще раз сказать, что развитие баз данных в системах автоматизации проектирования даже без учета графического аспекта показало недостаточность средств традиционных моделей данных [19] в первую очередь для представления сложных структур объектов реального мира и семантики связей между ними. В результате был предложен ряд специальных моделей (например [42, 46—48, 57]) с явно выраженной объектной ориентацией. Основным свойством этих моделей с точки зрения структурирования данных является введение более мощных средств представления абстракций типа обобщения и агрегации, таких, например, как «молекулы» [42] или «молекулярные агрегаты» [50]. Во многом эти нововведения продолжают идеи расширенной реляционной модели Кодда *RM/T* [23].

Основные принципы построения рассматриваемой модели данных для представления и обработки схем транспортных объектов (железных дорог, станций, парков, депо и др.) были предложены еще в [52]. Модель строится на основе ряда идей *RM/T*, а также результатов других исследований [42, 50]. Это позволяет структурировать графические данные с необходимой степенью детализации и обеспечивает достаточно глубокую их семантическую интерпретацию. В качестве программных средств поддержки предполагается комплексное использование типовых реляционной СУБД и пакета машинной графики. Средствами СУБД выполняются структурирование и избирательная обработка графических данных, а визуализация изображений осуществляется с помощью графического пакета, эффективно реализующего при этом различные функции геометрических преобразований — масштабирование, поворот, отсечение и др. Такой подход обеспечивает также и возможность интерактивной модификации моделей и изображений объектов с последующей их записью в базу данных, что является общим требованием к универсальным графическим системам высокого уровня [41]. Относительно систем, ориентированных на проектирование, где модели объектов могут часто и в определенных границах произвольно изменяться, оно становится решающим. Поэтому предлагаемую модель можно рассматривать как проблемно-ориентированную в первую очередь для решения задач проектирования. Проблемная ориентация, позволяющая быстро настраивать и эффективно использовать систему для решения задач в некоторой предметной области, отмечается в [34] как одна из тенденций развития баз данных в графических системах.

3.2.1. СТРУКТУРЫ ДАННЫХ

В параграфе 2.3 модель графического объекта была определена как множество элементов типа точек и линий, описывающих пространственные характеристики этого объекта. Исходя из этого содержательного определения формально модель графического объекта можно представить как помеченный граф $G(V, R)$, где V — множество вер-

шин, соответствующих элементам v типа точка, а R — множество ребер, соответствующих соединяющим точки элементам $r = \{v_1, v_2 | v_1, v_2 \in V\}$ типа линия. Поскольку графические объекты часто имеют иерархическую природу, т. е. элемент объекта i -го уровня может представлять собой самостоятельный объект на $(i + 1)$ -ом уровне, граф G , моделирующий их структуру, будет рекурсивным. Таким образом, абстрактная модель предметной области графической системы описывается фактически множеством графов $G = \{G_1, \dots, G_n\}$, каждому из которых соответствуют наборы помеченных вершин V_i и ребер R_i , определяющих некоторые физические структуры (объекты и связи между ними) и объединенных в составе отношений RV_i и RR_i . Данные о вершинах и ребрах в RV_i и RR_i рассматриваются формально как кортежи $v = (V, A)$ (или $r = (R_i, A)$), где V_i (или R_i) — синтаксический символ (имя), обозначающий структуру (вершину или ребро), а $A = \{A_1, \dots, A_m\}$ — семантический вектор, обозначающий m атрибутов (графических и неграфических) структуры V_i (или R_i). Любая структура, представленная в виде вершины или ребра графа G_i i -го уровня, может интерпретироваться как самостоятельный граф на $(i + 1)$ -ом уровне (детализированное представление), а сам граф G_i — как отдельная вершина на $(i - 1)$ -ом уровне (агрегированное представление). Такое представление наиболее адекватно реальной иерархии моделируемых физических объектов, когда каждый объект i -го уровня — это некоторая абстракция, рассматриваемая как комплекс объектов разных типов $(i + 1)$ -го уровня. Эта абстракция в полной мере соответствует введенному в *RM/T* типу агрегата покрытия. Каждая реализация такого типа является совокупностью реализаций других типов, обладающих своими множествами прямых свойств, представленными значениями соответствующих атрибутов. Графическими атрибутами в RV_i и RR_i являются в первую очередь координаты объектов, а также другие атрибуты, используемые обычно в системах машинной графики — тип, цвет, яркость и др. Все они помечаются специальным идентификатором (например, начальным символом имен всех графических атрибутов является буква G), т. е. фактически реализуется принцип самоописания данных. Значения графических атрибутов могут указываться явно при вводе данных, приниматься по умолчанию или автоматически устанавливаться в зависимости от значений неграфических атрибутов при существовании соответствующих функциональных зависимостей.

Рассмотренная абстрактная модель ориентирована на представление графических данных в виде топологических структур. Очевидно, что для железнодорожного транспорта, а точнее, для представления и обработки схем транспортных объектов, она наиболее естественна. Например, железнодорожную сеть в масштабах некоторого региона можно рассматривать как множество станций нескольких типов (узловые, сортировочные, грузовые и др.), соединенных перегонами. Сами станции можно представить как комплекс парков, грузовых дворов, локомотивных и вагонных депо и других объектов, каждый из которых, в свою очередь, состоит из таких элементов, как пути, стрелочные переводы, светофары, тупики и др. В общем случае объект i -го уровня,

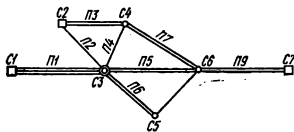


Рис. 3.2.1

будучи агрегацией покрытия по отношению к объектам $(i + 1)$ -го уровня, в то же время сам является членом агрегации $(i - 1)$ -го уровня. Определим подробнее состав данных в отношении концептуальной модели.

Представление вершин.

Принципиальный состав RV_i

следующий: $RV_i(V_i, V_{i-1}, X_i, Y_i, T_i, \phi_i, G_i^V, A_i^V)$. Здесь V_i — это уникальное имя (или другой идентификатор) вершины. Оно может быть определено как первичный ключ, если используемая СУБД не имеет средств автоматического формирования и поддержки заменителей; V_{i-1} — имя (идентификатор) объекта $(i - 1)$ -го уровня (вершины или ребра), являющегося типом агрегата покрытия, включающим в качестве члена вершину V_i . Очевидно, что V_{i-1} может отсутствовать в кортеже RV_i (иметь неопределенное значение), если V_i не является членом агрегата покрытия; X_i и Y_i — координаты вершины V_i , определяющие ее размещение на некоторой площади. Они могут быть как абсолютными объектными координатами в некоторой региональной системе (тогда они могут выступать в качестве альтернативного первичного ключа RV_i), так и относительными для каждого отдельного уровня иерархии. В последнем случае при построении изображения координаты V_{i-1} служат началом координатных осей для привязки соответствующих вершин i -го уровня. Атрибут T_i характеризует тип вершины, от значения которого любой объект при визуализации изображения может интерпретироваться как примитив типа маркер (точка или какой-то специальный знак) или как некоторый стандартный элемент (круг, многоугольник и др.). $G_i^V = \{G_{i1}^V, \dots, G_{in}^V\}$ — множество других графических атрибутов V_i (цвет, яркость и др.). $A_i^V = \{A_{i1}^V, \dots, A_{in}^V\}$ — множество атрибутов, характеризующих неграфические свойства объекта (технические параметры и др.). На рис. 3.2.1 представлен фрагмент схемы железной дороги (Д1), состоящей из станции (С) и перегонов (П). Пример отношения RV для схемы приведен в табл. 3.2.1.

Таблица 3.2.1

RV_1 (станции)

v_i	v_o	x_i	y_i	t_i	G_i^V	A_i^V
C1	Д1	100	50	Стыковая
C2	Д1	120	100	»
C3	Д1	200	50	Сортировочная
C4	Д1	250	100	Участковая
C5	Д1	280	10	Грузовая
C6	Д1	300	50	Участковая
C7	Д1	360	50	Стыковая

Таблица 3.2.2

 RR_i (перегоны)

R_i	V_{i1}	V_{i2}	G_i^R	Число путей N_i^R	Скорость, км/ч, V_i^R	Вес, т, P_i^R	A_i^R
П1	С1	С3		3	90	8000	...
П2	С2	С3		1	60	4000	...
П3	С2	С4		2	80	5000	...
П4	С3	С4		1	70	5000	...
П5	С3	С6		2	90	8000	...
П6	С3	С5		2	70	4000	...
П7	С4	С6		2	80	6000	...
П8	С5	С6		1	70	5000	...
П9	С6	С7		2	90	8000	...

Представление ребер. Ребра графа G_i отображают наличие связей между объектами одного и того же уровня. При этом сама связь представляет собой самостоятельный физический объект — путь, перегон, участок железной дороги. Каждому отношению RV_i в соответствие ставится отношение RR_i , содержащее данные о ребрах графа G_i . Принципиальный состав RR_i следующий: $RR_i(R_i, V_{i1}, V_{i2}, G_i^R, A_i^R)$, где R_i — имя ребра или цепи графа G_i ; V_{i1}, V_{i2} — имена вершин, соединяемых R_i ; G_i^R и A_i^R — множества графических и других атрибутов R_i .

Выбор первичного ключа для RR_i зависит от следующих возможных ситуаций:

а) R_i и пара (V_{i1}, V_{i2}) уникальны для каждого кортежа, а значит, могут рассматриваться как альтернативные ключи;

б) G_i является мультиграфом, т. е. имеет кратные ребра. Тогда функциональная зависимость вида $R_i \rightarrow (V_{i1}, V_{i2}), G_i^R, A_i^R$ оставляет R_i единственным претендентом на роль первичного ключа;

в) R_i не уникально, так как определяет некоторую цепь, а не ребро графа. Этому соответствует, например, случай, когда две станции соединены перегоном с именем R_i , состоящим из нескольких линейных участков, описываемых в модели отдельными кортежами. Первичным ключом в этом случае будет пара (V_{i1}, V_{i2}) , так как $(V_{i1}, V_{i2}) \rightarrow R_i, G_i^R, A_i^R$. Возможное также при этом наличие транзитивных зависимостей вида $R_i \rightarrow G_i^R, A_i^R$ порождает два типа отношений концептуальной модели: $RR_i'(V_{i1}, V_{i2}, R_i)$ и $RR_i''(R_i, G_i^R, A_i^R)$;

г) комбинация ситуаций «б» и «в» приводит к возникновению многозначной зависимости вида $R_i \twoheadrightarrow (V_{i1}, V_{i2})$ и функциональной зависимости $(R_i, V_{i1}, V_{i2}) \rightarrow G_i^R, A_i^R$, а следовательно, первичным ключом RR_i будет совокупность атрибутов (R_i, V_{i1}, V_{i2}) .

В табл. 3.2.2 приведен фрагмент модели данных о ребрах для схемы, изображенной на рис. 3.2.1.

Очевидно, что отношения RV_i и RR_i разных уровней могут иметь разную степень (арность), которая определяется в первую

очередь составом атрибутов A_i^V и A_i^R . Более того, вершинам и ребрам одного и того же уровня могут соответствовать разные типы объектов, характеризующихся разными наборами свойств. В результате объекты каждого уровня могут описываться не одним, а множествами отношений, т. е. $RV_i = \{RV_{i1}, \dots, RV_{in}\}$ и $RR_i = \{RR_{i1}, \dots, RR_{im}\}$.

Модель «вершины — ребра» (VR) имеет ряд преимуществ по сравнению, например, с рассмотренными в [38—40], которые являются практически частными случаями предложенной модели. Основные ее достоинства:

возможность совместного и раздельного манипулирования данными о вершинах и ребрах, что позволяет удобно представлять и обрабатывать модели объектов, изображаемых графами разной степени связности — от полных до вполне несвязных;

обеспечение многоуровневого интерфейса, т. е. возможности манипулирования данными и изображениями на разных уровнях детализации — от наиболее укрупненного представления предметной области как совокупности объектов сложной структуры до максимально детализированного их рассмотрения на уровне составляющих элементов;

отсутствие избыточности и связанной с ней проблемы управления целостностью, присущих моделям [38—40].

Помимо вершин и ребер как основных типов данных в модели могут быть представлены и такие типы, как текстовые наименования, входящие в состав изображений, а также площади.

Представление текстовых наименований. Вообще-то текстовые наименования можно было бы ввести в состав RV_i и RR_i в качестве значений соответствующих атрибутов из G_i^V и G_i^R . Однако вряд ли это целесообразно делать, учитывая, что в реальных условиях далеко не каждая вершина или ребро имеют собственное текстовое наименование. Это привело бы к тому, что для многих вершин и ребер значения текстовых атрибутов были бы неопределены. Поэтому удобнее выделить текстовые наименования в отдельные отношения RT_i^V и RT_i^R следующего состава: $RT_i^V (V_i, G_i^T, TV_i)$, $RT_i^R (R_i, G_i^T, TR_i)$, где G_i^T — множество графических атрибутов текста (координаты начала, угол наклона, размер текстового ящика и т. д.), а TV_i и TR_i — сами текстовые наименования, относящиеся к вершине V_i или ребру R_i .

Представление площадей. Весьма вероятно, что при решении ряда задач проектирования появится необходимость представления и обработки некоторых площадей (например, производственных зданий) в целом. В известных моделях [37—39] площади представляются обычно упорядоченным последовательным набором кортежей, описывающих ограничивающие площадь ребра. Это не всегда удобно, так как требование упорядочения достаточно строгое, придающее реляционной модели излишнюю жесткость. Удобнее, на наш взгляд, представлять площади на каждом уровне в составе двух отношений: $RS_i (S_i, G_i^S, A_i^S)$ и $SR_i (S_i, R_i)$.

Атрибуты множеств G_i^S и A_i^S описывают графические (цвет, яркость, тип штриховки и др.) и другие характеристики площади S_i в целом (S_i — идентификатор площади), а бинарное SR_i для каждой площади S_i перечисляет ограничивающие ее ребра R_i .

3.2.2. ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ

Важнейшим ограничением целостности, помимо обычных ограничений на значения отдельных атрибутов и их совокупностей, в рассматриваемой модели является ограничение на состав агрегатов покрытия. Оно представлено графовым GR -отношением и специфицирует сложные иерархические структуры объектов реального мира, описывая вхождение объектов ($i - 1$)-го уровня в объекты i -го уровня. Введя в базу данных графической системы домен наименований всех RV - и RR -отношений (RN -домен), можно определить GR -отношение как бинарное, оба атрибута которого определены на домене RN , первый в роли SUP — старший, второй в роли SUB — подчиненный. Кортеж ($SUP : n, SUB : m$) принадлежит GR в том случае, если тип объекта с именем m является (или может быть) членом агрегата покрытия с именем n , т. е. GR имеет следующий вид: $GR (RV_n, RV_m)$.

С помощью GR -отношения (как и RS -отношения модели, описанной в 3.1) не только описываются ограничения целостности, но и моделируются структуры сложных объектов, аналогичные тем, что используются в объектных моделях [42—46].

3.2.3. ПРИНЦИПЫ МАНИПУЛИРОВАНИЯ ДАННЫМИ

Под манипулированием графическими данными понимаем выполнение следующих процедур их обработки: ввод, модификацию, удаление графических данных; получение символьной (алфавитно-цифровой) информации о графических объектах; получение изображений объектов на устройствах машинной графики (графических дисплеях и графопостроителях); интерактивную обработку моделей и изображений объектов.

Для реализации перечисленных процедур используется специальный ЯМГД. Он включает в себя средства ЯМД СУБД, обеспечивающие традиционные функции обработки баз данных, а также средства ЯГП для формирования и вывода изображений объектов на устройства машинной графики.

Функции интерпретации операторов ЯМГД выполняет специальный программный графический процессор, который осуществляет (рис. 3.2.2):

прием от прикладных программ или терминального пользователя запросов, определяющих действия над базой графических данных; выполнение этих запросов с использованием средств ЯМД СУБД — реализацию операций запоминания либо выборки с формированием при выборке рабочей базы данных, включающей результирующие рабочие отношения W^V и W^R , описывающие модель запрашиваемого объекта;

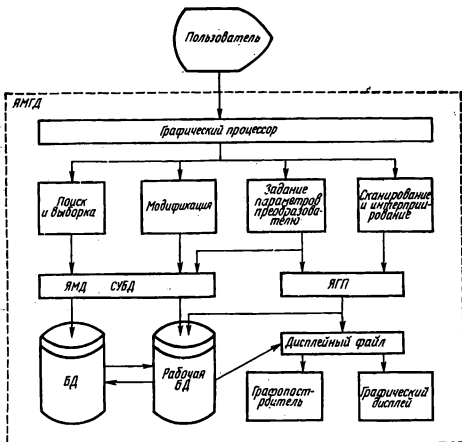


Рис. 3.2.2.

анализ значений графических атрибутов W^V и W^R и параметров преобразований и генерацию операторов ЯГП, обеспечивающих формирование и вылачу изображения на устройства машинной графики; интерактивную обработку W^V и W^R с одновременной модификацией изображения на экране.

Рассмотрим особенности выполнения процедур манипулирования графическими данными.

Ввод, модификация, удаление, выборка данных. Поскольку информация о графических объектах представлена в базе данных в символьном виде, очевидно, что все эти процедуры могут быть успешно выполнены средствами реляционного ЯМД используемой СУБД. Более того, следует отметить, что практически все разновидности реляционных языков — (alpha, Quel, SQL и др. [20]) обеспечивают с разной степенью процедурности реализацию функционально полного набора операций над иерархическими графическими структурами, определенного, например, в [54]. Причем это касается как операций над отдельными элементами структур (добавление и удаление вершин и ребер), так и операций над структурами в целом (расщепление, подстановка, деконпозиция, объединение, разность). Выполнение таких специальных процедур обработки графических структур, как автоматическое удаление ребер, инцидентных удаляемой вершине (например, *DELETER*),

определение изоморфизма и равенства графических структур (*ISOMOR* и *EQUEL*), может быть осуществлено с помощью соответствующих операторов, расширяющих используемый язык. При этом *DELETER* и *EQUEL* просто интерпретируются базовыми средствами языка, а *ISOMOR* реализуется как специальная встроенная функция.

Получение изображений на устройствах машинной графики. В зависимости от состава запрашиваемых данных можно выделить следующие основные типы запросов на получение изображений, состоящих 1) только из изолированных вершин; 2) из вершин (в том числе возможно, изолированных) и соединяющих их ребер; 3) из ребер и только инцидентных им вершин; 4) только из ребер.

В изображение, выдаваемое по запросу, относящемуся к любой из перечисленных групп, могут быть включены текстовые наименования вершин и ребер, если они имеются.

Обобщенная принципиальная схема запроса независимо от принадлежности к одной из этих групп выглядит в общем случае так:

$$GET\ W \leftarrow \pi_X(\sigma_Q(RV_i \bowtie RR_i \overline{\bowtie} RT_i^V \overline{\bowtie} RT_i^R)).$$

Виртуальное результирующее отношение W , описывающее модель запрашиваемого объекта, содержит с точки зрения пользователя все атрибуты отношений RV_i , RR_i , RT_i^V , RT_i^R или некоторое их подмножество X , задаваемое проекцией π . В состав W включаются кортежи, соответствующие условию Q селекции σ , определенному на значениях атрибутов RV_i и RR_i . Селекции предшествует операция внешнего естественного соединения отношений RV_i , RR_i , RT_i^V и RT_i^R , которая превращает неизбежную при использовании обычного соединения потерю информации в W о вершинах и ребрах, не имеющих текстовых наименований.

Для удобства спецификации запроса в смысле принадлежности его одной из перечисленных выше групп оператор *GET* целесообразно дополнить специальными идентификаторами V и R . В этом случае каждой группе запросов будет соответствовать свой оператор *GETV* [T], *GETVR* [T], *GETRV* [T] и *GETR* [T], определяющий состав данных в W . Наличие в запросе идентификатора T указывает на необходимость включения в изображение текстовых наименований.

Схема обработки запроса на получение изображения включает в себя три этапа: предобработку запроса; поиск данных и формирование результирующего W ; сканирование W и генерацию изображения.

На этапе предобработки выполняется декомпозиция вводимого пользователем запроса на два самостоятельных, выполняющих выборку данных отдельно из RV_i и RR_i . Для каждого из них из общего условия Q селекции формируются собственные условия, определенные соответственно на атрибутах RV_i и RR_i . Полученные запросы формулируются на ЯМД, используемой СУБД.

На втором этапе осуществляется обработка этих запросов и выборка соответствующего подмножества данных, используемых в дальнейшем при генерации изображения. При этом за счет высокой селективной мощности реляционных языков пользователь получает широкие

и гибкие возможности конструирования новых объектов и их изображений: включать в них по своему желанию в каждом конкретном случае интересующие его отдельные объекты или их совокупности, фрагменты или элементы объектов. Результатом выполнения этого этапа обработки запроса является рабочее отношение W . Точнее говоря, каждый из двух полученных на первом этапе запросов формирует свое результирующее отношение — W^V и W^R , хотя с точки зрения пользователя он видит одно общее W . Внутрисистемное разделение W на W^V и W^R удобно по ряду причин. Во-первых, оно позволяет избежать сложностей реализации соединения RV_i и RR_i , которые связаны с неоднозначностями из-за наличия в RR_i двух атрибутов имен вершин; во-вторых, при этом не происходит дублирование данных о вершинах, когда одной вершине инцидентны несколько ребер, и значительно упрощается поддержка интерактивной модификации модели и изображения. При необходимости включения в состав изображения текстовых наименований выполняется соединение W^V и W^R с RT_i^V и RT_i^R . Если модель объекта включает вершины и ребра разных типов, представленных не отдельными отношениями, а множествами отношений разной степени, то перед поиском данных выполняются операции внешнего объединения, т. е. $RV_{i1} \cup RV_{i2} \cup \dots \cup RV_{in}$ и $RR_{i1} \cup RR_{i2} \cup \dots \cup RR_{im}$.

На третьем этапе на основании содержащихся в W^V и W^R данных генерируется дисплейный файл или программа дисплейного процессора, обеспечивающая вывод запрашиваемого изображения на экран дисплея или стол графопостроителя. Генерация программы осуществляется специальным интерпретатором, включаемым в состав ЯМГД. Интерпретатор сканирует рабочие отношения и на основании анализа значений графических атрибутов каждого кортежа генерирует соответствующие команды перевода пишущего элемента (луча или пера) либо вычерчивания графических примитивов (точек, маркеров, линий) или стандартных элементов с определенными визуальными свойствами — типом, цветом, яркостью и др. Таким образом, результатом работы интерпретатора является последовательность *CALL*-операторов вызова графических процедур, которым передаются в качестве фактических графических параметров соответствующие значения графических атрибутов из W^V и W^R . Интерпретатор может генерировать команды формирования изображений на языке конкретных устройств машинной графики либо на некотором ЯГП более высокого уровня, входящем в состав пакетов машинной графики. Целесообразность и эффективность использования средств графических пакетов на этом этапе очевидна, так как помимо терминальной независимости они обеспечивают автоматическое выполнение ряда процедур специальных преобразований изображений — отсечения, масштабирования, поворота и др. Генерация команд формирования изображений выполняется интерпретатором с учетом значений атрибутов, определяющих указанные преобразования.

Рассмотрим примеры основных типов запросов для модели, представленной в табл. 3.2.1 и 3.2.2.

3. 1. «Показать все сортировочные и участковые станции, примыкающие к станции С6».

$GETV W \leftarrow \pi_X (\sigma_{V_1=C6 \wedge (V_{11}=C6 \vee V_{12}=C6)} \wedge (V_1=\text{сорт} \vee V_1=\text{участ})) (RV_1 \times RR_1)$.

Хотя запрашивается изображение, состоящее только из вершин, условие селекции включает и атрибуты из RR_1 , так как RR_1 не содержит информацию о смежности вершин, т. е. при поиске должны анализироваться данные как из RV_1 , так и из RR_1 . А общая схема обработки будет такой:

1.1. Формируется W^V как результат селекции над RV_1 по условию $V_1 = C6$ (кортеж 6).

1.2. Формируется W^R из RR_1 по условию $V_{11} = C6 \vee V_{12} = C6$, т. е. выбираются ребра, в которых одной из вершин является С6 (кортежи 5, 7—9).

1.3. По именам смежных с С6 вершин (V_{11} или V_{12}) находятся данные в RV_1 , и если они удовлетворяют условию $V_1 = \text{сорт} \vee V_1 = \text{участ}$, то заносятся в W^V (кортежи 3—5).

1.4. Сканируется W^V и генерируются команды визуализации изображения, приведенного на рис. 3.2.3, а.

3. 2. «Показать все станции дороги и соединяющие их двухпутные перегоны с весом поезда не менее 5000 т».

$GETVR W \leftarrow \pi_X (\sigma_{N_1^R=2 \wedge P_1^R \geq 5000}) (RV_1 \times RR_1)$.

Общая схема обработки.

2.1. Формируется W^V из всех кортежей RV_1 .

2.2. Формируется W^R из кортежей RR_1 по условию $N_1^R = 2 \wedge P_1^R \geq 5000$ (кортежи 3, 6, 7, 9).

2.3. Сканируются W^V и W^R и генерируются команды визуализации изображения, представленного на рис. 3.2.3, б.

3. 3. «Показать перегоны с числом путей не менее 2 и скоростями движения не менее 80 км/ч».

$GETRV W \leftarrow \pi_X (\sigma_{N_1^R \geq 2 \wedge V_1^R \geq 80}) (RV_1 \times RR_1)$.

Общая схема обработки.

3.1. Формируется W^R из RR_1 по условию $N_1^R \geq 2 \wedge V_1^R \geq 80$ (кортежи 1, 3, 5, 7, 9).

3.2. Формируется W^V из RV_1 по условию $V_1 = V_{11} \vee V_1 = V_{12}$ (кортежи 1—4, 6, 7).

3.3. Сканируются W^V и W^R и генерируются команды визуализации изображения, приведенного на рис. 3.2.3, в.

В отличие от запросов типа $GETVR$ (запрос 3. 2) изображения, полученные в результате обработки запросов типа $GETRV$, не имеют изолированных вершин.

3. 4. «Показать перегоны, соединяющие сортировочные станции с участковыми».

$GFTR W \leftarrow \pi_X (\sigma_{(T_1=\text{сорт} \vee T_1=\text{участ}) \wedge ((V_{11}=\text{сорт} \wedge V_{12}=\text{участ}) \vee (V_{11}=\text{участ} \wedge V_{12}=\text{сорт}))}) \times (RV_1 \times RR_1)$.

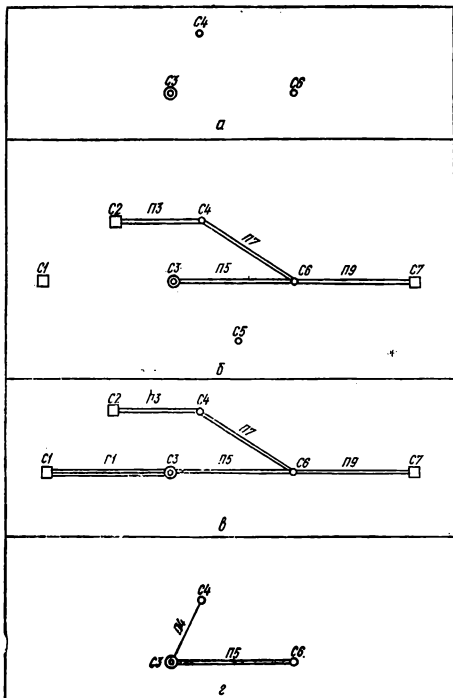


Рис. 3.2.3

Общая схема обработки.

4.1. Формируется W^V из RV_1 по условию $T_1 = \text{сорт} \vee T_1 = \text{участ}$ (кортежи 3, 4, 6).

4.2. Формируется W^R из RR_1 по условию $(V_{11} = \text{сорт} \wedge V_{12} = \text{участ}) \vee (V_{11} = \text{участ} \wedge V_{12} = \text{сорт})$ (кортежи 4, 5).

4.3. Сканируется W^R и генерируются команды визуализации изображения, приведенного на рис. 3.2.3, г.

Так как изображения, состоящие только из ребер, часто теряют наглядность, запросы типа *GETR* по умолчанию могут обрабатываться как *GETRV*, т. е. в изображение включаются вершины, инцидентные ребрам из W^R .

Возвращаясь к обобщенной принципиальной схеме запроса на вывод изображений, следует сказать, что условие селекции Q в общем виде можно записать как тройку предикатов:

$$Q = [P_1(RV_i)] [P(RR_i)] [P_2(RV_i)],$$

где $P_1(RV_i)$ определяет условия поиска для указываемых непосредственно в запросе вершин; $P(RR_i)$ — для ребер, а $P_2(RV_i)$ — для вершин, смежных задаваемым в запросе. Присутствие каждого предиката в запросе обязательно, но наличие $P_2(RV_i)$ возможно только если $P(RR_i)$ содержит условие вида $V_{11} = V_1 \vee V_{12} = V_1$, т. е. когда требуется выборка вершин, смежных вершинам, отобранным по условию $P_1(RV_i)$. Как видно из приведенных примеров, схемы обработки запросов в общем достаточно тривиальны, за исключением случаев, когда условие селекции содержит предикат $P_2(RV_i)$, как, например, в запросе 3. 1. Отличительной особенностью обработки таких запросов является двухэтапное формирование W^V . На первом этапе в него включаются кортежи из RV_i , удовлетворяющие условию $P_1(RV_i)$, затем в RR_i ищутся имена вершин, смежных вершинам из W^V , и по именам смежных вершин выбираются из RV_i и дописываются в W^V кортежи, удовлетворяющие условию $P_2(RV_i)$.

Несколько слов об обработке площадей. Хотя площади могут быть частями общего изображения, выдаваемого по одному из приведенных выше типов запросов, рассмотренная модель позволяет выполнять и их самостоятельную обработку. Например, запрос вида

$$GETS W \leftarrow \sigma_Q(RS_i \bowtie SR_i \bowtie RR_i \bowtie RV_i)$$

обеспечивает для площади S_i , выбранной из RS_i по условию Q , нахождение в SR_i имен всех ограничивающих ее ребер, получение о них необходимой информации из RR_i и RV_i с формированием W^R и W^V и последующим выводом изображения площади на экран.

3.2.4. ИНТЕРАКТИВНАЯ ОБРАБОТКА МОДЕЛЕЙ И ИЗОБРАЖЕНИЙ

Основными требованиями интерактивной обработки являются, как известно [41], немедленное воспроизведение модифицированного изображения на экране, а также возможность записи модели вновь созданного объекта в базу данных. Пакеты программ машинной графики обеспечивают выполнение только первого из этих требований, так как обычно манипулирование изображениями на экране с помощью курсора, светового пера и других аналогичных средств не выходит за пределы соответствующей модификации дисплейного файла. А поскольку дисплейный файл описывает лишь изображение, а не модель объекта, его содержимое не достаточно для быстрого и удобного восста-

новления модели изображаемого объекта с целью помещения ее в базу данных.

При работе с предлагаемой моделью базы данных интерактивный режим может быть успешно реализован следующим принципиальным образом. Запрос пользователя на вывод первоначального изображения обрабатывается аналогично тому, как это описано ранее для неинтерактивного режима. Отличием является лишь то, что полученные при этом W^V и W^R сохраняются в основной памяти ЭВМ на все время взаимодействия пользователя с базой данных. Таким образом, в основной памяти постоянно хранится фрагмент базы данных, описывающий модель объекта, изображение которого выводится на экран. Интерактивная модификация модели объекта осуществляется посредством операторов ЯМД используемой СУБД над отношениями W^V и W^R . Средства ЯМД, как уже указывалось, позволяют пользователю гибко менять состав рабочих отношений W^V и W^R , удаляя и добавляя отдельные кортежи или обновляя значения графических атрибутов — координат, визуальных свойств и др. — произвольным образом или в зависимости от значений неграфических атрибутов. Например, если в RR_1 (см. табл. 3.2.2) графический атрибут G_{i1}^R обозначает цвет ребра, то в результате обработки запроса 3.3 изображение (см. рис. 3.2.3, в) можно выделить красным цветом двухпутные участки сети с помощью запроса такого вида: $UPDATE (\sigma_{N_1^R=2} (W^R)); G_{i1}^R = \langle RED \rangle$ или $\rightarrow W_{N_1^R; G_{i1}^R}^R (2; \langle RED \rangle)$.

После обработки запросов на обновление W^V и W^R интерпретатор сканирует модифицированные отношения и генерирует команды формирования соответствующего модифицированного изображения объекта. При этом сканирование может осуществляться как автоматически после выполнения отдельного оператора, модифицирующего W^V или W^R , так и по специальному указанию пользователя после выполнения группы таких операторов. Таким образом, результат действий пользователя может быть немедленно отображен на экране и вместе с тем модифицированная модель объекта, описываемая текущим состоянием W^V и W^R , в любое время может быть помещена в базу данных.

Отдельно следует подчеркнуть такую важную возможность, как динамическая сегментация изображений. Сегмент в машинной графике — это поименованная совокупность выходных графических примитивов (маркеров, линий, текстовых строк), определяющая некоторую часть полного изображения. Применительно к рассматриваемой модели сегменту будет соответствовать определенное подмножество кортежей из W^V и/или W^R . Кроме имени, сегмент имеет свои графические атрибуты, такие, как видимость, обнаруживаемость, световое и цветовое выделение. Атрибутами сегмента могут быть также параметры геометрических преобразований — перемещения, поворота, масштабирования. Если все эти атрибуты сегмента ввести в состав отношений W^V и W^R , то с помощью ЯМД можно в интерактивном режиме,

во-первых, присваивать атрибутам сегментов необходимые значения, а во-вторых, динамически формировать сам состав сегментов, группируя в них графические примитивы, описываемые кортежами W^V и W^R , у которых значения графических и других атрибутов соответствуют задаваемым условиям. Например, если W^R содержит кортежи, описывающие всю схему (см. рис. 3.2.1), то для выделения в отдельный сегмент с именем $N^S = S1$ части схемы, содержащей только двухпутные перегоны, необходимо выполнить обработку такого запроса: $UPDATE (\sigma_{N^R=2} (W^R)); N^S = S1 \text{ или } \rightarrow W_{N^R, N^S}^R (2; S1)$.

3.3. ОПТИМИЗАЦИЯ ВЫВОДА ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ

Одной из важных проблем обработки графической информации вообще и баз графических данных в частности является задача эффективной организации формирования изображений объектов при их хранении в памяти и выводе на устройства машинной графики [55, 58, 59].

В данном параграфе эта задача рассматривается как задача минимизации времени вычерчивания изображения на экране векторного дисплея или на столе графопостроителя. Суть ее состоит в следующем.

Постановка задачи. При вычерчивании практически любого изображения имеют место холостые переводы l_{xi} пишущего элемента (пера графопостроителя или луча дисплея). Количество холостых переводов K_x и, что особенно важно, их суммарная длина L_x существенно зависят от принятой очередности вычерчивания элементов и фрагментов изображения. Отсюда следует, что задача минимизации времени построения изображения состоит в отыскании такого варианта вычерчивания изображения, при котором количество холостых переводов K_x принимало бы минимальное значение K_x^* и, кроме того, общая длина холостых переводов пишущего элемента была бы минимальной:

$$\min \left\{ L_x = \sum_{i=1}^{K_x^*} l_{xi} \right\}.$$

Практическим эффектом решения этой задачи являются улучшение качества изображения на экране дисплея за счет увеличения частоты его регенерации или возможность увеличения «емкости» (насыщенности) изображения при той же частоте регенерации; повышение производительности графопостроителя как естественное и очевидное следствие сокращения времени вычерчивания изображения; уменьшение объемов памяти, необходимой для хранения изображения, за счет исключения избыточной информации о холостых переводах пишущего элемента, сводимых до минимума в результате оптимального формирования графических образов; повышение пропускной способности ВЦКП, оснащенных техническими средствами, ориентированными на обработку высокоинтенсивных потоков задач графического вывода.

Будем рассматривать изображение, выводимое на экран дисплея или стол графопостроителя в виде графа $G(V, R)$, где R — множество ребер (отрезков графического изображения), а V — множество вершин

графа (точек пересечения отрезков, из которых состоит изображение). Холостые переводы пишущего элемента оказываются неизбежными вследствие того, что реальные графы $G(V, R)$ практически никогда не являются эйлеровыми, т. е. такими, что

$$\forall (v_i, v_j) \in R, \exists S(v_i, v_j), \quad (3.3.1)$$

где $i \neq j$; S — цепь в графе,

$$\forall v_i \in V, d(v_i) = 2q, \quad (3.3.2)$$

где $q = 1, 2, 3, \dots, d$ — степень вершины графа.

Превратить граф $G(V, R)$ в эйлеров можно, если добавить в него некоторое множество ребер P таким образом, чтобы условия (3.3.1) и (3.3.2) оказались справедливы.

Для графов $G(V, R)$, удовлетворяющих условию (3.3.1) и не удовлетворяющих условию (3.3.2), минимальное множество дополняющих ребер P , соответствующее минимальному количеству холостых переводов K_x^* , (т. е. неподсвечиваемых или невычерчиваемых отрезков графического изображения), зависит от множества N вершин нечетной степени графа таким образом, что $K_x^* = |P| = |N|/2 = n/2$, где n — количество вершин нечетной степени в $G(V, R)$.

Нетрудно подсчитать, что количество возможных вариантов построения множества P для таких графов равняется $(n-1)!!$. Понятно, что решить задачу нахождения дополнительного множества ребер, суммарная длина L_x которых была бы минимальной при достаточно мощном множестве N , методом полного перебора не представляется возможным.

На практике обычно пользуются простым интуитивным правилом построения изображения, которое состоит в том, что в качестве следующего элемента вычерчивается ближайший элемент изображения. Однако очевидно, что для достаточно сложных изображений использование этого правила дает далеко не оптимальную последовательность вычерчивания и не позволяет существенно минимизировать количество холостых переходов K_x и их общую длину L_x .

Методы решения. Из множества вершин V исходного графа $G(V, R)$ выделяется подмножество N вершин нечетной степени. На подмножестве N задается полный взвешенный граф $F(N, T)$ с функцией веса c_{ij} на множестве его ребер T . В соответствии с [58] вершины $v_i \in V$, а следовательно, и $v_i \in N$, представляют собой точки плоскости L_∞ и вес c_{ij} ребер графа F , понимаемый как расстояние между v_i и v_j , вычисляется в соответствии с чебышевской метрикой, т. е.

$$c_{ij} = c_{ji} = \rho(v_i, v_j) = \max(|x_i - x_j|, |y_i - y_j|),$$

где x_i, y_i и x_j, y_j — координаты вершин графа F на плоскости L_∞ . При этом координаты вершин могут представлять собой как объектные, так и картинные координаты, получаемые из объектных с помощью масштабных преобразований, которые учитывают тип устройства вывода и линейные размеры воспроизводимого изображения.

На графе $F(N, T)$ решается задача о совершенном паросочетании P минимального веса [62], т. е. в F определяется такое множество ребер P , что никакие два из них не являются смежными (не имеют об-

шей вершины) и каждой вершине N инцидентно ровно одно ребро из P , а сумма их длин минимальна.

Оптимальное множество P вносится в исходный граф $G(V, R)$, что приводит к образованию эйлерова графа $E(V, R + P)$. На графе F с помощью алгоритма обнаружения и сращивания циклов за время $O(r)$ решается задача почтальона [61], выявляющая непосредственный эйлеров цикл (здесь r — количество ребер графа $E(V, R + P)$). В результате оказывается сформированной оптимальная последовательность элементов графического изображения, выводимая на внешнюю графическую периферию с максимальной скоростью и требующая минимума памяти для хранения.

Остановимся подробнее на задаче о совершенном паросочетании P минимального веса. Точное решение этой задачи можно получить с помощью алгоритма Эдмондса [60—62]. Однако вычислительная сложность такого алгоритма, оцениваемая как $O(n^3 m \log n)$, где m — количество ребер графа, при большой размерности задачи ограничивает возможность практического применения этого алгоритма, а также некоторых его модификаций, несколько улучшающих оценки сложности: $O(n^3 + nm \log n)$ в [63] и $O(nm \log n)$ в [64]. Другими словами, при большом количестве вершин графа E время нахождения точного решения может оказаться сопоставимо или больше времени, которое эконоится при вычерчивании изображения за счет такой оптимизации последовательности вывода.

В то же время с практической точки зрения необязательно иметь точное решение задачи о совершенном минимальном взвешенном паросочетании. В [65] был предложен класс алгоритмов сложности $O(n)$ для приближенного решения данной задачи на плоскости. Идея состоит в следующем. Пусть на плоскости произвольным образом расположены n точек (нечетных вершин графа G). Накладываем на эти точки квадратную сетку из $n' \times n'$ ячеек таким образом, чтобы покрыть все точки. Определяем точки, принадлежащие одной ячейке, и устанавливаем какую-то последовательность просмотра ячеек. Точки, находящиеся в одной ячейке, соединяем произвольным, не увеличивающим вычислительной сложности образом, а точки из разных ячеек соединяем в выбранном порядке ячеек. Пройдя таким образом все ячейки и соединив последнюю точку с первой, получим маршрут коммивояжера, содержащий два совершенных паросочетания. В качестве решения задачи выбирается паросочетание меньшего веса.

При условии, что n' выбирается порядка \sqrt{n} , сложность такого алгоритма будет равна $O(n)$. В то же время при зафиксированном значении n' алгоритм может быть улучшен при условии, что нечетные вершины графа G не распределены равномерно в квадрате и имеется представление об их действительном расположении на плоскости. Например, разбивая (если это возможно) исходное множество нечетных вершин на несколько подмножеств, покрывая множество или подмножества (если это возможно) не квадратами, а прямоугольниками, а также подбирая специальные типы обходов ячеек, можно существенно сократить время работы алгоритма и получить решение, достаточно близкое к оптимальному.

Один из наиболее практичных и эффективных подходов к решению задачи оптимизации формирования и вывода состоит в разработке эвристических алгоритмов. На основании учета семантики графических изображений они позволяют в лучшем случае найти приемлемое приближенное решение задачи, в худшем — существенно снизить ее размерность в результате сокращения количества вершин нечетной степени в исходном графе за счет соединения их ребрами с минимальными весовыми коэффициентами, которые наверняка вошли бы в точное решение. В последнем случае задача оптимизации решается в два этапа и предполагает комплексное использование эвристических правил предобработки графа, описывающего изображение на первом шаге, а также применение точного или приближенного алгоритма для отыскания окончательного решения на втором шаге.

Экспериментальные исследования. Использование методов оптимизации формирования изображения предполагается для автоматического построения схем путевого развития транспортных объектов — железнодорожных станций, узлов, локомотивных и вагонных депо — на основе описаний этих схем, хранящихся в реляционной базе графических данных.

Данные о схемах объектов представлены в составе двух отношений (RV и RR), содержащих сведения соответственно о вершинах и ребрах графа $G_q(V_q, R_q)$, описывающего q -й объект. Принципиальный состав RV и RR следующий:

$$RV(V, X, Y, A_1, \dots, A_k),$$

$$RR(R, V_1, V_2, B_1, \dots, B_l),$$

где V и R — имена вершин и ребер; X, Y — координаты вершин; V_1, V_2 — пары имен вершин из $V(v_i, v_j | i \neq j)$; A_1, \dots, B_l — атрибуты, определяющие некоторые дополнительные характеристики вершин и ребер ($i = \overline{1, k}; j = \overline{1, l}$).

Для оптимизации формирования и вывода изображений в среде ОС ЕС был реализован на языках PL/1 и ФОРТРАН комплекс программ, решающий следующие задачи: представление схемы путевого развития станции, хранящейся в базе графических данных, в виде графа; эвристический алгоритм преобразования исходного графа, позволяющий на основе анализа схемы путевого развития в среднем на порядок уменьшить количество нечетных вершин; выделение из графа множества вершин нечетной степени и построение на этом множестве взвешенного графа; определение совершенного паросочетания минимального веса для взвешенного графа; приближенное решение задачи о совершенном минимальном паросочетании на плоскости; внесение найденного паросочетания в исходный граф и выявление в нем непосредственного эйлера цикла; упорядочение в базе данных схемы путевого развития для хранения в форме кратчайшего обхода; отрисовка схемы путевого развития на графопостроителе или дисплее.

Рассмотрим подробнее применение методов оптимизации и их эффективность для формирования изображения схемы путевого развития станции, показанной на рис. 3.3.1 (здесь $ГС$ — границы станции; T —

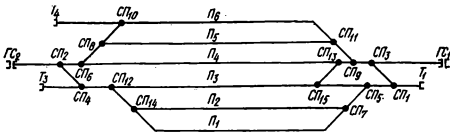
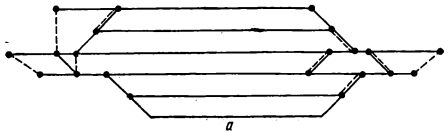
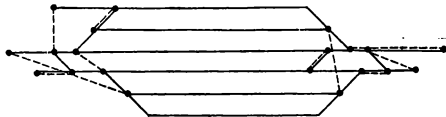


Рис. 3.3.1.



а



б

Рис. 3.3.2

тупики; СП — стрелочные переводы; П — пути). На основании хранящегося в базе данных описания схемы формируется граф $G(V, R)$, который является связным неэйлеровым графом с большинством вершин нечетной степени. Такой граф представлен на рис. 3.3.2, где ребра графа изображены сплошной линией, ребра, дополняющие граф до эйлерова, — пунктирной, нечетные вершины графа — точками.

Для рассматриваемого примера проанализируем использование точного и приближенного методов оптимизации без предобработки схемы с помощью эвристического алгоритма. В данном случае количество нечетных вершин $n = 20$, наименьшее количество дополняющих ребер $K_x^* = 10$ и количество возможных вариантов дополнения графа до эйлерова $(n - 1)!! = 19 \cdot 17 \cdot 15 \dots \cdot 3 \cdot 1$. Минимальная суммарная длина дополнения L_x^0 , найденная в результате работы точного алгоритма, равняется 29 усл. ед. (см. рис. 3.3.2, а). При полезной длине линий L_n , из которых состоит изображение, равной 171 усл. ед., общая длина линий пути, необходимая для отрисовки изображения, составляет 200 ед.

Минимальная длина дополнения L_x^n при работе приближенного алгоритма равняется 59 усл. ед. (см. рис. 3.3.2, б), и общая длина пути

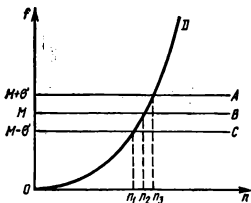


Рис. 3.3.3

варианта вывода (точного или приближенного) равны.

Принимая, что t — это время, необходимое для полезной длины изображения, можем составить уравнение

$$t + \frac{L_x^0}{L_n} t + t_p^0 = t + \frac{L_x^n}{L_n} t + t_p^n, \quad (3.3.3)$$

решая которое определяем, что условие удовлетворяется при значении $t = 57$ с. Время t зависит от формата воспроизводимого изображения и может быть оценено предварительно. В данном случае оно было больше вычисленного значения, а это говорит о том, что предпочтительнее было бы воспользоваться точным алгоритмом.

Соотношение (3.3.3) может быть преобразовано к виду

$$\frac{L_x^n - L_x^0}{L_n} = \frac{t_p^0 - t_p^n}{t}. \quad (3.3.4)$$

Вычислительные эксперименты, проведенные с помощью программ, реализующих точный и приближенный алгоритмы оптимизации, показали, что левая часть уравнения (3.3.4) практически не зависит от сложности схем путевого развития. Как показано на рис. 3.3.3, функции A , B и C являются константами и представляют собой соответственно M , $M + \sigma$ и $M - \sigma$, где M — математическое ожидание и σ — среднеквадратичное отклонение случайной величины

$$f = \frac{L_x^n - L_x^0}{L_n}.$$

Правая часть соотношения (3.3.4) зависит от сложности схемы (количества вершин нечетной степени) и от размеров воспроизводимого изображения. При зафиксированном значении параметра t график функции

$$f = \frac{t_p^0 - t_p^n}{t}$$

показан на рис. 3.3.3 (кривая D).

Из рис. 3.3.3 видно, что в случае, когда количество вершин нечетной степени меньше, чем n_1 , предпочтительнее воспользоваться точным алгоритмом для нахождения совершенного паросочетания мини-

пера графопостроителя или луча дисплея будет составлять 230 ед.

Таким образом, при работе точного алгоритма в данном случае достигается 15 % сокращения времени вывода изображения. Однако точный алгоритм требует большего времени счета и для данного примера эти времена соответственно $t_p^0 = 11,4$ с и $t_p^n = 1,4$ с. Возникает вопрос, при каких условиях общие временные затраты на вывод изображения и на время нахождения

мального веса. При $n > n_3$ целесообразно использовать приближенный алгоритм, а при $n_1 \leq n \leq n_3$ может быть использован любой из этих алгоритмов. Таким образом, n_1 , n_2 и n_3 представляют собой граничные значения сложности графического объекта, устанавливающие пределы эффективного использования рассматриваемых методов оптимизации.

Анализ соотношения (3.3.4) и рис. 3.3.3 позволяет предположить, что при более тонком учете в приближенном алгоритме особенностей рассматриваемого класса графических объектов можно добиться уменьшения значения величины M , тем самым понижая значения величин n_1 , n_2 и n_3 . Иными словами, учет особенностей графических объектов позволяет расширить границы эффективного использования приближенного алгоритма за счет соответствующего сужения границ использования точного метода. С другой стороны, использование точных алгоритмов с улучшенной оценкой вычислительной сложности [63, 64] позволило бы применить точный метод расчета для формирования и вывода более сложных изображений. Кроме того, необходимо отметить, что увеличение формата воспроизводимого изображения также позволяет в известных пределах без потери эффективности использовать точный метод оптимизации.

Очевидно, что рассмотренный подход к формированию изображений графических объектов позволяет достичь существенного сокращения времени их вывода. Учитывая вычислительную сложность алгоритмов, можно рекомендовать следующую специфику их использования.

Точный алгоритм целесообразно применять для оптимизации моделей объектов в базе графических данных. Это относится в первую очередь к объектам, модели которых достаточно постоянны, а изображения воспроизводятся многократно, и к наиболее часто воспроизводимым стандартным элементам и фрагментам изображений, т. е. принципом использования точного алгоритма должен быть «один расчет, многократный вывод». В результате оптимизации моделей объектов в базе данных упорядочиваются кортежи в RR в соответствии с очередностью отрисовки линий изображения, а также пары имен вершин V_1, V_2 в RR в соответствии с оптимальным направлением отрисовки каждого R .

При оптимизации формирования и вывода очень сложных изображений, для которых применение точного алгоритма может оказаться неэффективным, целесообразно использовать эвристические правила преобработки с целью уменьшения сложности задачи.

Глава 4

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В РЕШЕНИИ ЗАДАЧ УПРАВЛЕНИЯ ТРАНСПОРТОМ

4.1. ЛОКАЛЬНЫЕ СЕТИ — ТЕХНИЧЕСКАЯ ОСНОВА ПОСТРОЕНИЯ СИСТЕМ УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ ТРАНСПОРТОМ

4.1.1. ОПРЕДЕЛЕНИЯ, ПРИНЦИПЫ ПОСТРОЕНИЯ, ТРЕБОВАНИЯ

Актуальность использования локальных сетей (ЛС) в качестве технической основы построения систем управления промышленным транспортом определяется двумя главными факторами, непосредственно связанными с основными принципами новой информационной технологии — персонализацией вычислений и принятием согласованных решений. Во-первых, даже в условиях персонализации вычислений большая часть данных остается предназначенной для коллективного использования разными лицами при решении разных задач. Поэтому наряду с предоставлением каждому пользователю своих вычислительных ресурсов (персональных ЭВМ) совершенно необходимо создание локальной сети для обеспечения доступа к общим данным. Во-вторых, без ЛС как средства коммуникации невозможно практически автоматизировать процесс принятия согласованных оптимальных решений группой пользователей.

В [66] ЛС определяется как «коммуникационная система, поддерживающая в пределах здания или некоторой другой ограниченной территории один или несколько высокоскоростных каналов передачи цифровой информации, предоставляемых подключенным устройствам для кратковременного монопольного использования». Основным принципиальным отличием ЛС от глобальных сетей является именно наличие единого высокоскоростного канала, а не размеры охватываемых ими территорий, которые могут существенно различаться. Типовыми для ЛС считаются такие характеристики протяженности: расстояние между соседними абонентами (от десятков до сотен метров), максимальное удаление абонентов (до нескольких километров), т. е. ЛС могут обслуживать такие территории, как города, заводы, учреждения, суда, самолеты и др.

Скорость передачи данных в ЛС может колебаться в значительных пределах, что делает термин «высокоскоростной канал» весьма относительным. Например, для ЛС, поддерживающих только терминалы, достаточна скорость 50 Кбит/с, а для сети мини-ЭВМ необходима скорость 5 Мбит/с.

Единые магистрали передачи данных — общие каналы или общие шины — впервые были использованы в мини-ЭВМ. В отличие от применяемого в больших ЭВМ прямого соединения различных компонент в мини-ЭВМ каждый модуль подключается к общей шине и может посылать сообщения, адресованные другим модулям. Аналогичный прин-

цип используется и в ЛС. Важным его преимуществом является то, что шина более всего подходит для работы в режиме высокого соотношения пиковой и средней нагрузок, который наиболее характерен для систем реального времени. Кроме того, шинная архитектура имеет и ряд других достоинств — высокую надежность, универсальное подключение различных устройств, возможность широковещательного обращения, намного меньшую длину линий связи по сравнению с эквивалентным числом прямых соединений. В связи с неравномерным характером потока данных они обычно передаются через шину в составе пакетов, включающих адрес получателя и отправителя, собственно данные и контрольную сумму. В ЛС передача данных может осуществляться как в рамках устанавливаемых сеансов взаимодействия ЭВМ, так и в режиме дейтаграмм, при котором пакеты обрабатываются отдельно, а коммуникационная система не устанавливает продолжительных соединений между устройствами.

Требования к ЛС изучались специальным комитетом по стандартам и были опубликованы в [66]. Вот основные.

1. В общем случае ЛС должна выполнять разнообразные функции по передаче данных, включая пересылку файлов, поддержку терминалов (в том числе графических), электронную почту, обработку сообщений, доступ к файлам и базам данных, передачу речевых сообщений, видеообслуживание. Для этого она должна допускать подключение широкого набора стандартных и специальных устройств, таких, как ЭВМ, терминалы, устройства внешней памяти, принтеры, графопостроители, фотокопировальные устройства, факсимильные устройства, контрольно- и управляющее оборудование, мосты к другим ЛС и шлюзы к другим сетям, телефонов, телекамер.

2. Коммуникации между подключенными устройствами должны осуществляться непосредственно через шину без использования промежуточных устройств. Информационный поток может проходить через промежуточные устройства только в том случае, если они выполняют преобразование протоколов или функции регистрации потока.

3. Одна ЛС должна обеспечивать поддержку не менее 200 устройств и охватывать территорию не менее чем 2 км. Это требование не является критичным при использовании связанных ЛС.

4. Скорость передачи данных должна составлять 1—20 Мбит/с. Для связи больших ЭВМ могут потребоваться более высокие скорости, в то же время для сети микро-ЭВМ и особенно для терминальной сети достаточно скоростей менее 1 Мбит/с.

5. ЛС должны допускать простое подключение новых устройств и отключение старых без нарушения работы сети.

6. Для всех устройств должен быть обеспечен равноправный доступ к совместно используемым ресурсам сети, и особенно к общей шине.

7. Каждое устройство ЛС должно быть способно адресовать пакеты другому устройству, группе устройств, всем устройствам сети.

8. Пропускная способность ЛС не должна существенно снижаться при достижении полной нагрузки и даже при превышении реальной мощности сети. Максимальная задержка передачи пакета через ЛС должна поддаваться предварительному расчету.

9. ЛС должна обладать высокой надежностью и развитыми средствами обнаружения и обработки сбоев.

10. Стоимость ЛС должна быть низкой. При этом стоимость подключения устройств к сети должна быть малой по сравнению со стоимостью самих устройств.

Очевидно, что удовлетворить все перечисленные требования практически невозможно, так как это привело бы к чрезмерному увеличению сложности и стоимости сети, что в свою очередь снизило бы ее надежность. Поэтому при проектировании ЛС необходимо с учетом конкретных условий объекта обеспечить выполнение основных функциональных требований и по возможности уменьшить последствия невыполнения других.

4.1.2. КЛАССИФИКАЦИЯ ЛОКАЛЬНЫХ СЕТЕЙ

Локальные сети классифицируются обычно по топологии, физической среде и методам передачи данных, категории (назначению) сети.

Основными топологическими конфигурациями ЛС являются «шина», «дерево», «звезда» (рис. 4.1.1), «кольцо», «цепочка», «снежинка» (рис. 4.1.2). Имеются и другие, но они являются, как правило, разновидностями перечисленных. Первые три относятся к так называемым ширококвещательным конфигурациям, в которых каждая станция сети передает сигналы, воспринимаемые всеми остальными станциями. Три последние составляют группу последовательных конфигураций, в которых каждая станция может передавать информацию только одной другой.

Основным типом конфигурации ЛС является шина. Несколько шин могут быть соединены в конфигурацию типа дерево с помощью специальных цифровых повторителей, обеспечивающих регенерацию цифровых сигналов, передаваемых от одной шины к другой. Конфигурация типа звезда характерна для крупных сетей больших ЭВМ. Но если в них в центре звезды обычно размещается коммутирующее устройство, то в ЛС вместо него может находиться простой соединитель или цифровой повторитель. ЛС типа шина и дерево считаются более надежными и потребляют намного меньше кабеля, чем звездообразные ЛС. Пакеты в ширококвещательных конфигурациях передаются обычно непосредственно от отправителя к получателю.

В сети типа кольцо каждая станция участвует в приеме, регенерации и передаче пакета информации, следующего по кольцу от отправителя до получателя. И в других последовательных конфигурациях (типа цепочка и снежинка) в передаче пакета участвуют все промежуточные станции, расположенные между отправителем и получателем. Именно этот фактор делает ЛС типа кольцо, цепочка и снежинка особенно уязвимыми в отношении отказов, так как выход из строя любой станции может привести к прекращению обслуживания всех пользователей сети.

В качестве физической среды передачи данных в ЛС целесообразнее всего использовать коаксиальный кабель. Он прост, надежен, недорог и обеспечивает передачу данных с высокими скоростями (до

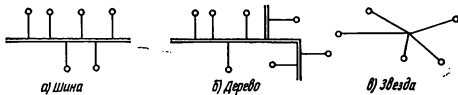


Рис. 4.1.1

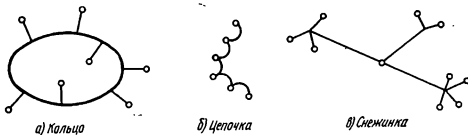


Рис. 4.1.2

30—50 Мбит/с) и на большие расстояния (до 100 км). Другие среды имеют или высокую стоимость (многожильные кабели) или жесткие ограничения на дальность и скорость передачи (телефонный кабель), а также плохо защищены от помех. Перспективными, особенно для сетей, где требуется высокая скорость передачи данных (сотни и тысячи Мбит/с), являются оптоволоконные кабели (световоды). Они имеют небольшую массу, невосприимчивы к электрическим помехам, пожаробезопасны. Вместе с тем существует ряд нерешенных проблем, затрудняющих применение световодов: трудности сращивания, невозможность передачи электроэнергии для работы повторителей и других устройств, сложность врезки и сильное ослабление сигнала при этом, невозможность двухсторонней передачи сигнала и др. Но, несмотря на это, считается, что с развитием технологии изготовления, и особенно стыковки оптоволоконных кабелей, они станут одним из наиболее удобных средств передачи данных.

По методам организации передачи данных сети разделяются на следующие группы.

Сети с опросом. В таких сетях центральное управляющее устройство последовательно опрашивает каждое устройство в порядке очереди, определяемой списком опроса. Такая технология наиболее характерна для глобальных сетей. Частота и очередность опроса могут динамически изменяться в зависимости от активности опрашиваемых устройств. Общим недостатком сетей с опросом является простояние устройства, содержащего готовые для передачи данные, пока идет опрос других устройств. Опрос обычно осуществляется центральной ЭВМ, через которую идет передача данных (только от центрального устройства к опрашиваемому и наоборот) и от надежности которой зависит надежность сети в целом. Производительность сетей с опросом в целом считается приемлемой для небольших ЛС, но принцип централизации не соответствует характеру распределенной обработки данных в ЛС. Конфигурация сетей с опросом может быть самой разнообразной.

Сети с передачей маркера. Это сети с распределенным доступом, в которых опрос (передача маркера) идет последовательно от устройства

к устройству, а управление сетью распределено между всеми станциями, т. е. рассматриваемая ЛС представляет собой логическое кольцо (физически это может быть шина, дерево, звезда и др.), формируемое маршрутом маркера. Станция может начать передачу данных (пакета) в сеть только после получения маркера и после завершения передачи отсылает маркер дальше. Сети с передачей маркера обладают высокой производительностью и привлекают повышенный интерес после разработки фирмой IBM экспериментальной кольцевой ЛС [67], ставшей основой предложений по стандартизации для сетей такого типа.

Сети с доступом в режиме соперничества. Принципиальной особенностью таких систем является то, что их станции могут предпринять попытку передачи в любой момент, «захватив» передающую среду. Специальные алгоритмы реализуют необходимую дисциплину обслуживания при наложении сигналов в результате попыток одновременного захвата шины несколькими соперничающими станциями. Однако всегда существует вероятность весьма продолжительной задержки передачи при возникновении таких конфликтов, что снижает производительность сети. Опыт эксплуатации систем с соперничеством за рубежом свидетельствует об их популярности благодаря простому и надежному доступу к передающей среде и разработке стандартов.

Сети с резервированием времени. В ЛС с резервированием времени все время функционирования шины разделяется на интервалы. Каждой станции для передачи данных предоставляется один или несколько интервалов, специально зарезервированных для нее. Резервирование может производиться при генерации системы, в начале соединения и в произвольные моменты времени с помощью специальной регулировочной процедуры. Это позволяет хорошо учитывать характер трафика и эффективно использовать пропускную способность каналов. Главным достоинством этих сетей считается возможность передачи интегрированного трафика, объединяющего информационные потоки разных видов — речь, изображения, данные.

Перечисленные разновидности ЛС могут иметь различную, в том числе и кольцевую, конфигурацию. Ниже рассматриваются два метода передачи данных, разработанные специально для кольцевых систем.

Кольцевые сети с сегментированной передачей. В этих сетях набор пустых сегментов циркулирует по сети с большой скоростью от станции к станции. Станция-отправитель может поместить свои данные в пустой сегмент и направить их другой станции — получателю. Каждая станция может принимать пакеты от любой станции-отправителя, только от одной станции-отправителя, не принимать пакеты. То же относится и к передаче пакетов.

Кольцевые сети со вставкой регистров. Каждая станция этой сети имеет свой регистр или буфер, подключаемый параллельно к кольцу. В нормальном положении регистр отключен от кольца. Станция-отправитель помещает в пустой регистр передаваемый пакет, в определенный момент времени (в интервале между передачей двух последовательных пакетов по кольцу) регистр подключается к кольцу и передает данные в линию. Каждая станция считывает пакеты из кольца, опознает адресованные именно ей и удаляет их из кольца или помечает как пустые

для последующего удаления центральной мониторинной станцией. Дублирование кольца передачи (двойное кольцо) и специальные методы передачи обеспечивают высокую надежность и скорость передачи данных.

Широкополосные сети. Передача сигнала в коаксиальном кабеле может осуществляться двумя способами. Первый — работа в основной полосе частот, когда цифровой сигнал передается непосредственно в кабель. Дальность передачи в сети может быть просто увеличена последовательным включением цифровых повторителей, подавляющих шумы (помехи) и воспроизводящих чистый сигнал. Такие системы передачи дешевы и надежны, однако пропускная способность кабеля в них используется неэффективно, так как скорость работы электронных устройств составляет весьма незначительную часть по сравнению со скоростью передачи данных по коаксиальному кабелю. Для ее повышения используется второй способ передачи, основанный на принципе радиочастотной модуляции, когда вся пропускная полоса (5—300 МГц) делится между несколькими независимыми каналами с предоставлением каждому из них некоторого ограниченного частотного спектра (ширина телевизионного канала 6 МГц, канала терминальной сети 120—300 КГц и т. д.). Такой способ позволяет с помощью одной физической шины обслуживать несколько независимых ЛС, работающих в своей полосе частот, или комбинировать ЛС с другими видами цифровых коммуникаций. Такие системы называются широкополосными сетями передачи данных и имеют обычно конфигурацию «дерево с корнем». Цифровые данные в них передаются после модуляции несущей потоком битов. Дальность передачи в широкополосных ЛС ограничена (до десятков км) тем, что используемые в них аналоговые усилители одинаково усиливают и полезный сигнал, и шум. К тому же стоимость их значительно выше из-за необходимости оборудования каждой станции ЛС фильтром и высокоскоростным модемом, которые стоят примерно в 10 раз дороже, чем аппаратура основной полосы частот, работающая на той же скорости. Преимуществом широкополосных систем считается возможность совместной передачи звуковой, видео и цифровой информации. Основной сферой применения является кабельное телевидение.

Классификация ЛС по категориям (назначению) определяется типом поддерживаемых ими устройств.

Сети поддержки терминалов обеспечивают подключение к ЭВМ множества терминалов и поддерживают работу в основной полосе частот или в широкополосном режиме (последняя намного дороже).

Сети микро- и мини-ЭВМ обеспечивают передачу сообщений, доступ к общим ресурсам памяти, базам данных и библиотекам. Сети этой категории включают также такие устройства, как рабочие станции (автоматизированные рабочие места), которые строятся на базе микро-ЭВМ, обеспечивают пользователей собственными вычислительными ресурсами и поддерживают различные виды обслуживания — обработку данных, текстов, изображений, речи и др.

Отличительной особенностью *интегрированных учрежденийских сетей* является то, что они рассчитаны на взаимодействие разнообразных

Назначение ЛС	Название ЛС	Методы передачи данных
Сети поддержки терминалов	<i>Clearway</i>	Вставка регистров Соперничество
	<i>Local Net System 20</i>	
Сети микро-ЭВМ	<i>Net/One</i>	»
	<i>Econet</i>	»
	<i>Z-net</i>	»
	TOKEN RTNG Netvork	»
Сети мини-ЭВМ	STARLAN	»
	<i>Ethernet</i>	
	<i>ARCNet</i>	Передача маркера
	<i>Cambridg Ring</i>	Пустые сегменты
Интегрированные учрежденческие сети	<i>SILK</i> <i>Ubits</i>	Вставка регистров Соперничество

устройств и машин, включая системы обработки текстов, копировальную и множительную технику, телефоны и др.

В крупных вычислительных центрах организуют *сети больших ЭВМ*. Они имеют небольшую протяженность (до сотен метров) и предоставляют пользователям мощные вычислительные ресурсы и запасающие устройства, но по более высокой цене, чем в сетях мини-ЭВМ. В табл. 4.1.1 приведены основные обобщенные характеристики некоторых зарубежных ЛС.

4.1.3. МЕЖСЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ

Соединение однородных сетей, обслуживающих большую территорию, осуществляется с помощью специальных устройств, называемых в [66] фильтрами. Основная часть фильтра — буфер, в который временно помещается пакет, передаваемый из сети *A* в сеть *B*, если последняя в момент передачи занята. Возможна потеря пакетов при переполнении буфера из-за его малой емкости или длительного времени занятия канала сети, в которую должны быть переданы пакеты из буфера. Учитывая это, фильтр может снабжаться средствами уведомления отправителей о потере пакетов. При межсетевой передаче сообщений используется или изначальная маршрутизация, когда отправитель сам указывает последовательность фильтров, которые должны пройти пакеты до выхода на сеть, содержащую станцию-получателя, или же фильтры сами управляют потоком сообщений на основании содержащихся в них таблиц маршрутизации.

Взаимодействие удаленных однородных ЛС требуется в случае, когда необходима связь между территориально разнесенными объектами, каждый из которых имеет свою локальную сеть. Считается целесообразным устанавливать прямую связь между удаленными ЛС с

Таблица 4.1.1

Физическая среда	Топология	Скорость передачи, Бит/с	Примечание
Коаксиальный кабель	Кольцо	50К	—
» »	Дерево	128К	Широкополосная
» »	Шина	10М	—
Четырехпроводный кабель	»	210К	—
Коаксиальный кабель	»	800К	—
» »	Кольцо	4М	—
Витая пара	Звезда	1М	—
Коаксиальный кабель	Шина	10М	—
» »	соединенные звезды	2,5М	—
Четырехпроводный, коаксиальный или оптоволоконный кабель	Кольцо	10М	—
Коаксиальный кабель	»	17М	—
Многожильный кабель	Шина	140М	—

помощью одной или нескольких арендованных линий либо по постоянно действующему виртуальному каналу в рамках сети коммутации пакетов общего пользования, но не путем установления соединений для каждого сеанса передачи с помощью телефонной сети общего пользования или сети передачи данных. Управляют потоком данных в межсетевых каналах (включая маршрутизацию и согласование скоростей передачи в ЛС и телефонном канале) устройства, называемые мостами.

Подключение ЛС к глобальным сетям имеет проблемы, связанные с эксплуатационными различиями локальных и глобальных сетей. Высокие скорости и небольшие помехи передачи в ЛС позволяют использовать более простые протоколы, чем в низкоскоростных телефонных каналах глобальных сетей. Вместе с тем необходимо учитывать такие специфические особенности ЛС, как обязательная адресация отправителя и возможность широковещательной передачи, не предусмотренная протоколами глобальной сети. Разрешить проблемы взаимодействия ЛС и глобальной сети помогают скоростные телефонные сети пакетной коммутации общего пользования со скоростью передачи 48 Кбит/с, базирующиеся на интерфейсе X.25 (стандарт протокола сетевого уровня). За рубежом ожидается выпуск систем со скоростью 2,048 Мбит/с, но наиболее перспективным решением считается создание сетей передачи общего пользования на базе широкополосного кабеля.

Межсетевое взаимодействие однородных ЛС (в том числе удаленных) между собой и с глобальными сетями в принципе может быть выполнено достаточно просто. Для этого фильтры должны поддерживать протоколы только физического и канального уровня, а мосты (при использовании телефонной связи) еще и сетевой протокол. Поэтому при оборудовании ЛС подразделений промышленных предприятий и объединений весьма важно соблюдать по возможности однотипность этих сетей для обеспечения в перспективе их простого взаимодействия.

При необходимости установления взаимодействия между разнотипными сетями (в частности, подключения сетей микро-ЭВМ к другим ЛС) оно осуществляется с помощью специальных устройств, называемых шлюзами. Шлюзы обеспечивают преобразование сетевых протоколов взаимодействующих систем, а также поддерживают протокол транспортного уровня (т. е. устанавливают виртуальный канал между процессами разных сетей и обеспечивают передачу данных по нему). В перспективе ожидается разработка «прозрачных» фильтров, мостов и шлюзов, которые позволяют обращаться к файлам в других системах и сетях как к локальным, не заботясь об адресации.

4.1.4. МОДЕЛЬ ЛОКАЛЬНОЙ СЕТИ ДЛЯ УПРАВЛЕНИЯ ТРАНСПОРТОМ НА УРОВНЕ ПРОИЗВОДСТВЕННОГО ОБЪЕДИНЕНИЯ И ОТРАСЛИ

Анализ специфики задач управления транспортом в производственных объединениях и отраслях позволяет выделить среди них две группы.

К первой относятся задачи контроля, анализа и технико-экономического планирования работы транспорта, основными из которых являются анализ технико-экономических показателей работы и прогнозирование потребности в технических средствах транспорта (локомотивах, вагонах, путевых машинах и др.); планирование перевозок грузов железнодорожным транспортом; планирование, учет и анализ ремонтов подвижного состава и железнодорожных путей; проектирование организации транспортного обслуживания промышленных предприятий; обработка форм отчетности ЦСУ; другие задачи.

Задачи этой группы характеризуются большим объемом вводимых, хранимых и выводимых данных достаточно сложной структуры, необходимостью их интегрированной обработки, относительно невысокой периодичностью решения (месяц, квартал, год). При их решении используются в основном процедуры массовой избирательной обработки данных, типичные для систем баз данных. Перечисленные задачи решаются обычно в рамках отделов транспорта (или транспортных управлений), а в ряде случаев еще и в СКТБ объединений и отраслей.

Ко второй группе относятся задачи оперативного управления транспортом, в первую очередь управление перевозками, решение которых возлагается на персонал специальных диспетчерских групп. Для этих задач при относительно небольшом объеме обрабатываемых данных и получении их в реальном времени (с трехчасовыми интервалами и менее) преимущественно по каналам связи требуются адекватное отображение в памяти ЭВМ текущего поездного положения, моделирование продвижения поездов, индикация поездного положения на сети на графическом дисплее или специальном табло, вывод исполненного графика движения на графопостроитель. Здесь наряду с обычной (традиционной) обработкой данных должны быть реализованы функции моделирования, отображения динамики объектов предметной области, представления и обработки графических данных.

Кроме того, в задачах обеих групп могут потребоваться процедуры логического вывода, используемые при принятии решений на основании опыта и знаний, накопленных специалистами по управлению транспортом.

Решение всех перечисленных задач может быть организовано на базе микро- или персональных ЭВМ и рабочих станций, построенных на основе этих же микро-ЭВМ и расширенных дополнительными техническими средствами, в частности устройствами ввода-вывода графической информации. В принципе представляется возможной реализация всех задач управления промышленным транспортом на уровне производственного объединения или отрасли на базе нескольких (не более 5—6) взаимодействующих в рамках ЛС однотипных персональных ЭВМ, связанных с большими ЭВМ информационно-вычислительных центров производственных объединений и отрасли, если таковые имеются.

С точки зрения специфики задач и характеристик технических средств ЛС наиболее подходящими для реализации такой системы управления транспортом являются две категории локальных сетей — сеть поддержки терминалов и сеть микро-ЭВМ.

Сети поддержки терминалов обладают следующими достоинствами.

1. Возможность взаимодействия разнотипных устройств (терминалов, микро-ЭВМ, текстовых процессоров и др.) сети как с главной ЭВМ, так и непосредственно между собой на основе развитых средств преобразования протоколов либо путем отображения всех типов терминалов на некоторый виртуальный сетевой терминал.

2. Простое и удобное расширение сети за счет увеличения числа терминалов, установки дополнительной большой ЭВМ, подключения принципиально новых устройств — коммутаторов, блоков сопряжения с телекомом и др.

Вместе с тем обеспечение взаимодействия разнотипных устройств достигается за счет необходимости наличия специальных программных средств преобразования протоколов, что приводит к дополнительным затратам и делает целесообразным использование таких сетей только при достаточно большом (не менее 20) количестве терминалов. Разнотипность устройств ограничивает также скорость передачи данных (от десятков до сотен Кбит/с), а надежность сети сильно зависит от надежности главной ЭВМ. Большим недостатком считается также отсутствие развитых средств организации и совместного использования сетевых ресурсов (кроме ресурсов главной ЭВМ), в первую очередь общей памяти на магнитных дисках и общих принтеров.

Очевидно, что в случае однотипности и небольшого числа (менее 10) устройств достоинства сети поддержки терминалов не перекрывают ее недостатков. Это практически исключает возможность использования сети в качестве технической основы системы управления транспортом объединения или отрасли.

Локальная сеть микро-ЭВМ (ЛЭВМ) — наиболее целесообразная форма организации комплекса технических средств для построения системы управления транспортом регионального и отраслевого уровня. Объединение в ЛС совместимых (однотипных) микро-ЭВМ представляет

собой наиболее дешевый способ организации их взаимодействия и образует, вместе с тем, полноценную скоростную магистраль передачи данных, обеспечивающую пользователей гибкими средствами организации вычислительного процесса.

Одним из основных достоинств ЛС микро-ЭВМ, которое становится решающим с учетом специфики задач управления транспортом, является возможность совместного использования ресурсов, в первую очередь запоминающих устройств на жестких магнитных дисках. Это связано с тем, что при решении задач разными пользователями часто применяются одни и те же данные (документы паспортов железнодорожных хозяйств предприятий, нормативно-справочная информация и др.), которые целесообразно организовать в виде общей базы данных, доступной всем пользователям. Не менее важным положительным фактором коллективного использования ресурсов памяти является возможность организации общих библиотек программ. Благодаря высокому быстродействию ЛС доступ к общим ресурсам памяти происходит так, как если бы они были подключены локально, а эффективность их использования повышается за счет снижения стоимости и улучшения условий организации хранения и обработки данных.

Общесистемным ресурсом, совместно используемым несколькими микро-ЭВМ, могут быть также скоростные принтеры, обеспечивающие высококачественную печать. Это дорогие устройства, обращение к которым отдельных пользователей осуществляется эпизодически при подготовке документов больших объемов. Печать этих документов на обычных устройствах занимает много времени, а использование скоростных принтеров позволяет, помимо сокращения времени печати, продолжить работу на машине.

Эти же аргументы можно привести в отношении совместного использования межсетевых шлюзов, когда каждый пользователь может обратиться к общим программно-техническим ресурсам, обеспечивающим межсетевое взаимодействие.

- Совместное использование ресурсов при работе в сетевом режиме должно поддерживаться специализированными обслуживающими процедурами, так называемыми серверами, к которым имеется коллективный доступ. В первую очередь это файловые серверы, обеспечивающие работу с общими устройствами памяти, серверы печати, управляющие принтерами, коммуникационные серверы для межсетевого обмена данными, а также серверы для ведения библиотек программ, редактирования текстов, электронной почты и др. Наличие серверов освобождает от необходимости иметь для каждой ЭВМ свое программное обеспечение для реализации общих функций по управлению совместными ресурсами. Это снижает стоимость ЭВМ и повышает их эффективность при решении других задач.

Совместное использование ресурсов, высокая скорость передачи данных (до нескольких Мбит/с) с помощью общей шины и возможности практически неограниченного наращивания делают оправданной установку ЛС даже для небольшого числа микро-ЭВМ (менее 10).

Принципиальная схема ЛС микро-ЭВМ для управления транспортом на уровне объединения или отрасли приведена на рис. 4.1.3.

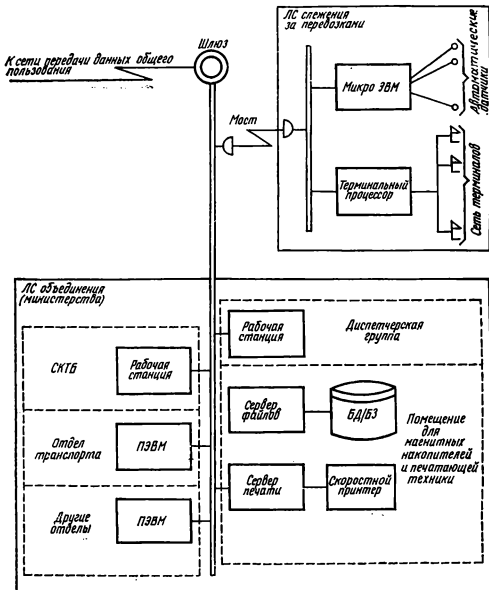


Рис. 4.1.3

ЛС объединения может иметь выход через шлюзы на сети больших ЭВМ информационно-вычислительных центров (отраслей, железных дорог и т. д.), другие сети ЭВМ и передачи данных.

Многие производственные объединения имеют предприятия, расположенные иногда на большом удалении друг от друга (в границах области, республики или другого крупного региона). Каждое предприятие может быть оборудовано самостоятельной ЛС терминалов или микро-ЭВМ. Как правило, большинство необходимых при управлении связей замыкаются внутри этих ЛС, однако в ряде случаев для решения некоторых глобальных задач требуется связь между территориально удаленными предприятиями и центральным управлением. Взаимодействие ЛС для таких объединений не сложно организовать с использованием мостов, обеспечивающих связь отдельных ЛС (рис. 4.1.4).

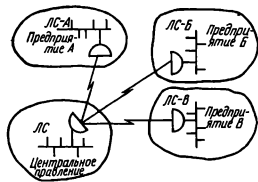


Рис. 4.1.4

При этом следует подчеркнуть, что именно соединение ЛС является наиболее эффективным с точки зрения удовлетворения коммуникационных потребностей при объединении территориально разобщенных предприятий. Это объясняется тем, что ЛС располагают всеми необходимыми средствами для обслуживания информационных систем будущего [2], в частности электронной почты, обеспечивающей факсимильную

передачу изображений (текстовых и графических) по сети связи.

Специфика передачи данных в ЛС (коммутация пакетов с малой задержкой времени на передачу), а также необходимость установления связи на достаточно длительное время (до нескольких часов в сутки) делают невыгодным взаимодействие локальных сетей через телефонные каналы общего пользования. Более целесообразно устанавливать прямую связь между ЛС с помощью арендованных линий связи или же по постоянно действующему виртуальному каналу сети коммутации пакетов общего пользования, если таковая имеется. Вообще выбор способа взаимодействия ЛС, а кроме перечисленных это могут быть микроволновая, спутниковая связь и др., должен определяться в первую очередь объемом передаваемой информации, чтобы обеспечить необходимые требования по времени передачи данных при минимальных затратах.

В заключение отметим, что мосты и шлюзы включают в себя всю необходимую аппаратуру для соединения удаленных однородных и разнотипных сетей, в том числе адаптеры и модемы, обеспечивающие согласование протоколов (способов кодирования данных в разных ЭВМ и сетях) и функции преобразования данных при передаче их по линиям связи определенного типа (телефонные, телеграфные и др.). При этом в качестве мостов и шлюзов целесообразно использовать специально предназначенные для этого микро-ЭВМ, обеспечивающие комплексную реализацию функций межсетевое взаимодействия, включая буферизацию, адресацию, квитирование, преобразование протоколов, перекодирование данных и др.

4.2. БАНКИ ДАННЫХ — ОСНОВНАЯ ФОРМА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ СИСТЕМ УПРАВЛЕНИЯ

Банк данных — совокупность программных, организационных и технических средств, предназначенных для централизованного накопления и коллективного использования данных. Основой банка данных является база данных, которая поддерживает модель предметной области и обеспечивает решение широкого круга функциональных задач на единой информационной основе и многоаспектный избирательный поиск данных при обработке оперативных запросов пользователей. Для ведения базы данных используется обычно одна из типовых систем

управления базой данных. Технологический цикл обработки данных является достаточно устоявшимся и включает в себя ряд периодически повторяющихся процессов обработки данных, присущих в общем случае практически всем системам массовой обработки данных (ввод, контроль, корректировка данных, решение функциональных задач, выдача выходных форм, обработка оперативных запросов пользователей, копирование базы данных, перезапись данных в архивный раздел и переход к вводу и обработке данных за очередной отчетный период).

Рассмотрим на примере [68] некоторые основные общесистемные (не зависящие от состава функциональных задач) аспекты реализации банка данных для систем управления промышленным транспортом.

База данных строится на основе концепций, изложенных в параграфах 3.1, 3.2 и включает в себя динамическую модель предметной области, а также ряд специальных служебных данных.

Динамическая модель содержит данные о текущем состоянии и наличии подвижных и стационарных объектов — локомотивном и вагонном парках, путевых машинах и механизмах, локомотивных и вагонных депо, пути и путевом хозяйстве и т. д.; данные о событиях (операциях с объектами) — сведения о выполнении перевозок, ремонтов, производстве элементов верхнего строения пути, приобретении и списании локомотивов и др.; данные о текущих и исторических связях стационарных и подвижных объектов; итоговые данные (статистические сводки), характеризующие плановые и аналитические показатели работы транспорта — объемы перевозок, ремонтов, грузовых работ и др.; нормативно-справочную информацию. Данные об объектах и событиях вводятся в базу данных в составе входных документов, итоговые данные по предприятиям отрасли или объединения частично генерируются из первых двух типов, а частично вводятся в составе самостоятельных входных документов.

В состав служебных данных входят открытый (расширяемый) каталог, журнал и справочники, регламентирующие обработку данных, доступ и полномочия пользователей. Каталог содержит сведения о каждом отношении в базе данных — имя отношения, тип данных (об объектах или событиях, итоговые, нормативные или служебные), наличие определенных на отношении внешних моделей (взглядов), наличие средств первичной и вторичной индексации, имена специальных процедур контроля целостности и др. По назначению каталог аналогичен каталогу *RM/T*, хотя и отличен от него по структуре и составу. В журнале базы данных специальными векторами описывается обязательная очередность выполнения рассмотренных выше процессов обработки данных, дифференцированная по отношениям динамической модели и предприятиям отрасли. В журнале фиксируется также фактическое выполнение отдельных процессов с указанием кодов их завершения (без ошибок или с ошибками). Таким образом, журнал отображает ход выполнения технологического цикла обработки данных и позволяет получать информацию, какие типы входных документов и по каким предприятиям были введены и загружены в базу данных, какие требуют корректировки по результатам контроля с последующей дозагрузкой, какие не были введены вообще, какие функцио-

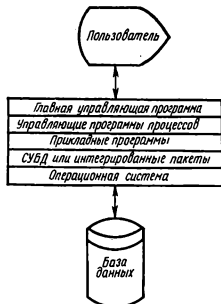


Рис. 4.2.1

нальные задачи были решены, какие выходные формы выданы и т. п.

Проектирование базы данных выполняется в целом в соответствии с рассмотренной в [70, 71] техблогией, однако при этом целесообразно выделить в качестве самостоятельных этапов проектирование концептуального и внешнего уровней [72] модели базы данных.

Концептуальная модель проектируется с использованием формальных методов теории проектирования баз данных [19—21] — анализа функциональных и многозначных зависимостей и приведения отношений к третьей и четвертой нормальной формам. На полученных в результате базовых отношениях, удовлетворяющих требованиям избыточности, целостности и независимости, определяются средства первичной и вторич-

ной индексации, обеспечивающие прямой доступ к данным.

Внешние модели (взгляды) проектируются с учетом информационных потребностей пользователей и неформальной семантики данных. Ключевым здесь должно быть рассмотрение вопроса, поддерживать взгляды на уровне определений и генерировать данные из базовых отношений при обработке соответствующих запросов или же материализовать их (взгляды) и актуализировать по мере внесения изменений в базовые отношения, на которых эти взгляды определены? Для принятия решения рекомендуется использовать простые правила:

если время генерации данных из базовых отношений превышает допустимое время ответа на запрос, взгляды целесообразно материализовать и поддерживать динамически;

в противном случае определяющим должно быть сопоставление суммарных затрат, с одной стороны, на генерацию, а с другой — на поддержание материализованного взгляда. Причем при высокой интенсивности внесения изменений в базовые отношения и относительно нечастых запросах пользователей взгляды лучше поддерживать на уровне определений, а когда имеет место обратная ситуация — соответствующие взгляды лучше материализовать и поддерживать динамически. Основным показателем при расчете затрат является количество обменов с внешней памятью, определяемое как функция от числа запросов в программе пользователя, объема данных базовых отношений и размера физического блока обмена.

Такой подход к проектированию базы данных позволяет достаточно полно учесть не всегда согласующиеся, а часто и противоречивые требования избыточности, целостности и независимости данных, с одной стороны, и минимизации времени ответа на запрос — с другой. Изменение состава прикладных задач (по крайней мере не связанное с расширением состава базовых отношений) требует при этом реструк-

туризации схемы базы данных только на внешнем уровне, не затрагивая концептуальный.

Основными принципами построения программного комплекса управления банком данных является многослойная архитектура (рис. 4.2.1) и диалоговое управление процессами обработки данных [73]. Каждый программный слой имеет самостоятельную функциональную специализацию. Главная управляющая программа поддерживает взаимодействие пользователей с банком данных и управляет работой всего программного комплекса. Каждый из рассмотренных ранее процессов обработки данных реализуется самостоятельным программным режимом, в состав которого входит управляющая программа этого режима и комплекс общесистемных и прикладных программ, взаимодействующих через среду СУБД, и операционной системы с базой данных и обеспечивающих содержательную обработку данных. Рассмотрим особенности работы каждого слоя программного комплекса.

Главная управляющая программа запрашивает последовательно код доступа пользователя к банку данных и код иницируемого им режима. Если работа с запрашиваемым режимом разрешена, вызывается его управляющая программа. Основными ее функциями является планирование вычислительного процесса и контроль технологии обработки данных.

Планирование вычислительного процесса подразумевает автоматизацию формирования многошагового пакета заданий в процессе диалога с пользователем и определения его потребностей. Пакет формируется в виде специального набора данных, фактически представляющего собой командный файл, и в качестве отдельных шагов включает вызов прикладных программ содержательной обработки данных (ввода и контроля данных, решения функциональных задач, выдачи выходных форм и др.), соответствующих введенным в процессе диалога указаниям пользователей. В диалоге также запрашиваются и уточняются коды задач, подразделений, для которых необходимо их решение, даты отчетных периодов и другие сведения. Кроме того, в пакет при необходимости автоматически включаются некоторые дополнительные общесистемные процедуры управления наборами данных — создания, удаления, копирования, сортировки и др. Необходимость и эффективность диалога обуславливается тем, что в каждом отдельном случае перечень и очерочность процедур обработки данных плохо предсказуемы заранее, так как они зависят, во-первых, от конкретных потребностей пользователя и, во-вторых, от текущего состояния базы данных — полноты и корректности исходных данных, выполненного ранее решения функциональных задач и т. д. Сформированный пакет автоматически стартует по специальному указанию пользователя после завершения диалога. При необходимости изменения структуры пакета старт может быть отменен и пакет сформирован заново. Возможен также отсроченный старт. В этом случае после завершения диалога пакет сохраняется во внешней памяти и стартуется в удобное время с помощью вводимой в ЭВМ команды.

Одной из важнейших задач ведения банка данных является поддержание технологической целостности базы данных, под которой

понимается строгое соблюдение определенной в журнале базы данных очередности выполнения отдельных процессов обработки данных. Нарушения очередности могут привести к получению некорректных результатов, например, при преждевременном решении функциональных задач до ввода и корректировки всех исходных данных, либо к непроизводительным затратам машинного времени при повторном выполнении уже выполненных ранее процедур. Контроль соблюдения технологии обработки данных является серьезной проблемой, стоящей перед администрацией и пользователями банка данных. Это обусловливается рядом факторов, осложняющих выполнение функций контроля при ведении больших банков данных — наличием большого числа пользователей, обилием входных данных, различной интенсивностью решения пользовательских задач, разнообразием технологических схем обработки данных при их решении, взаимозависимостью отдельных процессов и др. Необходимым условием гарантированного обеспечения пользователей достоверной информацией в этой ситуации является автоматизация процедур контроля технологии обработки данных.

В общем случае при ведении банка данных, содержащего множество отношений $R = \{R_1, \dots, R_n\}$, выполняется множество процессов обработки данных $P = \{p_1, \dots, p_m\}$. Анализ особенностей выполнения позволяет выделить среди них два типа — безусловно-выполняемые (БВ) и условно-выполняемые (УВ) процессы.

БВ-процессом называется такой процесс p_j , который требует своего обязательного выполнения после каждого выполнения непосредственно предшествующего ему процесса p_{j-1} . Например, семантический контроль данных является БВ-процессом, так как должен выполняться обязательно после каждого ввода данных.

УВ-процессом называется такой процесс p_j , который не требует своего выполнения при условии успешного завершения предшествующего ему процесса p_{j-1} . Например, корректировка данных является УВ-процессом, так как не требует своего выполнения при условии, что в результате семантического контроля данных ошибки не были обнаружены.

В общем случае существует разбиение $C(P)$ множества P на такое, что в результате будут получены два непересекающихся подмножества $P^{БВ}, P^{УВ} \subseteq P$, т. е. $P^{БВ} \cap P^{УВ} = 0$ и $P^{БВ} \cup P^{УВ} = P$.

В реальных условиях функционирования банка данных над каждым отношением R_i может выполняться как все множество процессов обработки P , так и некоторое его подмножество $P_i^* \subseteq P$, причем для каждого R_i соответствующее P_i^* может быть различным. При этом в P_i^* предполагается строгое упорядочение элементов, соответствующее последовательности выполнения отдельных процессов, т. е. p_{ij} всегда выполняется раньше $p_{i'j}$, если $j' < j$. Таким образом, каждое P_i^* представляет собой вектор, описывающий технологию обработки R_i . Подмножество процессов P_i^* и взаимосвязь между ними для каждого R_i можно представить в виде ориентированного циклического графа $G_i(P_i^*, X_i)$, где $X_i = \{p_{ij}, p_{i'j} | j' \neq j\}$. Пример графа G_i приведен на рис. 4.2.2.

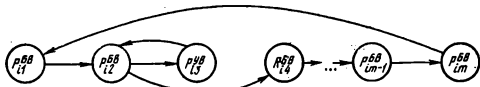


Рис. 4.2.2.

В целом, технология обработки данных описывается набором $T = \{P_1^*, \dots, P_n^*\}$ непустых множеств P_i^* , равной с P мощностью. Элементами P_i^* являются пары кодов $\{pv_{ij}, pl_{ij} \mid j = 1, \dots, m\}$, первый из которых указывает, выполняется ли процесс p_j при обработке R_i (1 — да, 0 — нет), а второй определяет тип процесса (1 — БВ-процесс, 0 — УВ-процесс).

Ход выполнения технологического цикла обработки данных отображается в составе множества $V = \{V_1, \dots, V_n\}$ векторов текущего состояния обработки. Элементами V_i являются пары кодов $\{kv_{ij}, kz_{ij} \mid j = 1, \dots, m\}$, первый из которых фиксирует факт выполнения процесса p_j над отношением R_i (1 — процесс выполнялся, 0 — нет), а второй является кодом завершения этого процесса (1 — успешное завершение, 0 — завершение с ошибками).

Наличие векторов P_i^* и V_i позволяет формализовать процедуры контроля соблюдения технологии обработки и автоматизировать их выполнение. При инициировании процесса p_j над отношением R_i предварительно анализируются состояния соответствующих элементов P_i^* и V_i и в зависимости от их значений разрешается либо блокируется выполнение p_j . С учетом специфики процессов, изложенной в определении их типов, можно сформулировать следующие условия разрешения выполнения процесса p_j .

Для БВ-процессов:

- 1) $kz_{ij-1} = 1$ — предшествующий процесс успешно завершен;
- 2) $kz_{ij} = 0$ — иницируемый процесс ранее не был выполнен успешно.

Для УВ-процессов:

- 1) $kv_{ij'} = 1 \wedge kz_{ij'} = 0$ ($j' < j$) — предшествующий БВ процесс был выполнен, но завершен с ошибками и, следовательно, требуется выполнение p_j ; ($j' < j$);

- 2) $kz_{ij} = 0$ — иницируемый процесс ранее не был выполнен успешно.

Очевидно, что несоблюдение первого условия для любого p_j свидетельствует о нарушении технологии, а второго — о попытке повторного инициирования уже успешно выполненного ранее процесса.

Программная реализация функций контроля технологии выполняется с помощью специальных процедур, включаемых в управляющую программу и проверяющих соответствие реальной очередности процессов обработки данных с допустимой, описанной в составе векторов журнала базы данных. При этом выполнение процессов разрешается либо блокируется с соответствующим уведомлением системы и пользователя. Уведомление системы выражается в занесении в журнал кодов завершения процесса, а уведомление пользователя осуществля-

ется посредством выдачи соответствующих сообщений на экран терминала и твердую печать.

Процесс взаимодействия пользователя с банком данных протоколируется, а протокол распечатывается после окончания сеанса работы.

Таким образом, архитектура программного комплекса обеспечивает удобное диалоговое многотерминальное взаимодействие удаленных и локальных пользователей с банком данных, автоматизацию управления вычислительным процессом с контролем технологической целостности базы данных, независимость программного обеспечения (по крайней мере на уровне управляющих программ) от состава функциональных задач, безбумажную технологию обработки данных.

Опыт эксплуатации банка данных [68] подтвердил, в целом, правильность принятых при его разработке основных концепций. Вместе с тем при расширении состава функциональных задач выявились серьезные трудности и неудобства использования традиционных (чисто машинных) методов для решения некоторых из них. В первую очередь это касается ряда задач планирования, не имеющих строго формализованных схем решения и требующих согласования на разных этапах решения между разными лицами. В связи с этим дальнейшее развитие подсистем управления транспортом предполагает в первую очередь использование новых информационных технологий [2], обеспечивающих интеллектуальную поддержку процесса решения сложных задач. Некоторые аспекты реализации такой технологии рассмотрены в следующем параграфе.

4.3. ИСПОЛЬЗОВАНИЕ ЗНАНИЙ В ПРИНЯТИИ РЕШЕНИЯ

Одним из основных прагматических направлений научных исследований в области искусственного интеллекта является в настоящее время так называемая интеллектуализация ЭВМ [74], подразумевающая в числе основных задач создание комплексных диалоговых интерфейсов с ЭВМ (в первую очередь персональными).

Интеллектуализация интерфейса позволяет резко повысить эффективность автоматизированных систем самого разного назначения за счет интенсификации работы конечного пользователя. Это достигается в результате непосредственного общения пользователя с ЭВМ в процессе решения задач; диалогового задания условий (исходных данных) для решаемой задачи, автоматической генерации программы ее решения и диалогового контроля и управления всеми стадиями вычислительного процесса; накопления в памяти ЭВМ знаний экспертов о предметной области и их использования в процессе принятия тех или иных решений.

Очевидно, что реализация такого интерфейса, при котором ЭВМ берет на себя все посреднические функции, возможна только при условии соответствующего повышения интеллекта ЭВМ путем оснащения ее процедурными знаниями обо всех технологических особенностях решения задач.

Таковыми знаниями располагают интеллектуальные пакеты прикладных программ [74], которые обеспечивают решение задач по описаниям

и исходным данным, вводимым пользователем, с автоматическим синтезом программ решения из отдельных программных модулей. Знания представляются функциональной семантической сетью, являющейся в общем случае двудольным неориентированным графом с двумя типами вершин. К первому относятся вершины, описывающие параметры решаемых задач (в том числе исходные данные), а ко второму — вершины, отображающие математические отношения между этими параметрами (соотношения значений, функции, математические модели). Каждой вершине второго типа соответствует обычно некоторый программный модуль, вычисляющий это математическое отношение, причем результаты вычислений могут выступать в качестве параметров других отношений. Дуги, определяющие смежность вершин разных типов, связывают все множество параметров со всем множеством существующих между ними математических отношений.

При решении конкретной задачи задается обычно некоторое подмножество известных параметров, для которого необходимо выбрать подмножество отношений, последовательное вычисление которых обеспечит решение поставленной задачи. Программа-планировщик вычленяет из неориентированного графа ориентированный граф решения задачи, определяя входы и выходы для необходимых отношений в соответствии с заданным подмножеством параметров, т. е. устанавливает фактически цепочку модулей рабочей программы. На таком принципе построена работа известных пакетов ПРИЗ и МАВР [74], предназначенных для автоматизации проектирования сложных технических систем модульного типа.

Самостоятельным этапом в развитии искусственного интеллекта является использование экспертных систем [74—77]. Экспертная система определяется в [75] как «вычислительная система, в которую включены знания специалистов о некоторой конкретной проблемной области и которая в пределах этой области способна принимать экспертные решения». Более широкое практическое применение экспертных систем обусловлено в [74] тремя основными факторами: 1) экспертные системы ориентированы на решение широкого круга задач в ранее не формализуемых областях, которые считались малодоступными для ЭВМ; 2) экспертные системы предназначены для решения задач в диалоговом режиме со специалистами, от которых не требуется знания программирования, что резко расширяет сферу использования вычислительной техники и усиливает способности специалиста к логическому выводу; 3) за счет аккумуляции в системе знаний экспертов высшей квалификации использующий ее специалист может при решении своих задач достигать, а иногда и превосходить по результатам возможности экспертов в данной области знаний.

Рассмотрим некоторые принципиальные возможности использования основанных на знаниях интеллектуальных технологий принятия решений в задачах планирования и управления промышленным транспортом.

4.3.1. УПРАВЛЕНИЕ РЕШЕНИЕМ СЛОЖНЫХ КОМПЛЕКСОВ ЗАДАЧ

Задачи планирования и управления, решаемые в автоматизированных системах управления крупными промышленными объектами, обычно весьма сложны, многосвязны и многовариантны, требуют учета разных критериев и согласования с различными лицами и подразделениями в процессе разработки планов и принятия управленческих решений. Кроме того, динамически изменяющиеся условия функционирования объектов управления вносят такие непредсказуемые возмущения, которые требуют соответствующей оперативной корректировки планов или пересчета их вообще. Наряду со сложностью технологических схем решения задач необходимо отметить их детерминированный характер. Это обусловлено тем [74], что функционально-иерархические структуры автоматизированных систем управления достаточно устойчивы и в них заранее известен (или может быть установлен) порядок взаимодействия между отдельными звеньями в процессе принятия решений и согласования планов. В целом, решение комплекса задач представляет многошаговый процесс, в котором выбор очередного шага зависит от исходного состояния, а также результатов, полученных на предыдущих шагах, т. е. в зависимости от этих факторов процесс может развиваться альтернативно. Число альтернатив может быть достаточно велико, а выбор наилучшей из них не всегда очевиден и связан с квалификационной оценкой сложившейся ситуации. Рассмотрим некоторые вопросы управления решением сложных комплексов взаимосвязанных задач на основе знаний об особенностях принятия решений в той или иной ситуации.

На рис. 4.3.1 приведен фрагмент технологической схемы решения комплекса задач планирования ремонтов локомотивов [78], включающей следующие этапы. По типовым нормативам межремонтных периодов рассчитывается план ремонтов локомотивов. В диалоге пользователем дается указание согласовать этот план с планом перевозок или рассчитать потребность в запчастях либо распечатать план ремонтов. Если необходимо согласование, то, кроме плана ремонтов, необходимо рассчитать еще план перевозок и в зависимости от анализа результатов согласования реализовать программно (без вмешательства пользователя) переход к расчету потребностей в запчастях и последующей печати плана или же откорректировать (в диалоге) требуемым образом нормативы и сделать пересчет плана ремонтов сначала. В реальных условиях эта схема значительно сложнее, однако в принципиальном отношении приведенный фрагмент отражает основные типы связей между отдельными блоками, а именно — наличие несомкнутых, альтернативных или обязательных, входов в блок и нескольких альтернативных выходов из блока. Каждому блоку соответствует программный модуль, реализующий некоторый вычислительный алгоритм, информационный запрос к базе данных или диалоговую процедуру. В зависимости от результата работы этого модуля выбирается очередной шаг расчета.

Приведенная на рис. 4.3.1 блок-схема может быть представлена

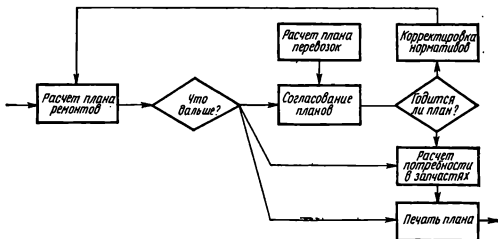


Рис. 4.3.1

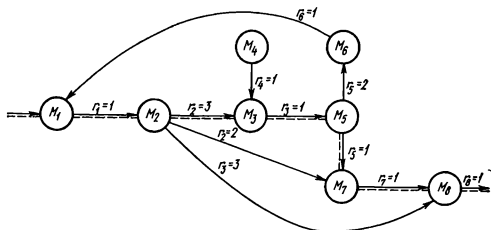


Рис. 4.3.2

в виде семантической сети (рис. 4.3.2), в которой вершинам сопоставляются программные модули из множества $M = \{M_1, \dots, M_n\}$, а дугам — отношения следования. Семантическая сеть является замкнутой и детерминированной, так как все ситуации на ней перечислены, а отношения следования однозначно определены*. Однозначность отношений следования превращает сеть в ориентированный граф, а процедура поиска альтернативного пути решения на этом графе сводится к выделению на нем ориентированного подграфа, описывающего процесс решения поставленной задачи.

Каждый программный модуль M_i ($i = \overline{1, n}$) формирует свой код завершения $r_i \in R_i$, значение которого определяется на основании интерпретации результата работы процедуры расчета или поиска или вводимого в диалоге указания пользователя. В простейшем случае $R_i = \{0, 1, 2\}$, что означает соответственно, что модуль не выполнялся

* Строго говоря, семантическая сеть остается открытой в том смысле, что при появлении новой технологии решения задач она может быть дополнена соответствующими вершинами и дугами,

($r_i = 0$), при выполнении получен удовлетворительный ($r_i = 1$) или неудовлетворительный ($r_i = 2$) результат. В действительности спектр оценки результатов работы каждого модуля может быть значительно шире, причем для модулей, реализующих диалоговые процедуры, каждому ответу пользователя может ставиться в соответствие свое значение r_i , т. е. в общем случае $R_i = \{1, 2, \dots, n_i\}$. Совокупность значений r_i представляет собой вектор $R = \{r_1, r_2, \dots, r_n | r_i \in R_i\}$, описывающий состояние сети в целом.

Семантическая сеть, дуги которой помечены значениями r_i , определяющими альтернативные отношения следования, является, по сути, моделью знаний о технологии поиска решения сложных комплексов взаимосвязанных задач в разных условиях. Используя эти знания, можно создать своего рода интеллектуальную программную надстройку над совокупностью функциональных программных модулей, которая обеспечивает автоматический поиск альтернативного пути решения поставленной задачи в зависимости от исходных условий и получаемых на каждом шаге результатов, подключая при необходимости к этому процессу пользователя. Рассмотрим два варианта организации такой надстройки.

Использование продукционной системы для представления и обработки знаний. База знаний включает в себя три компоненты:

набор $P = \{P_1, P_2, \dots, P_m\}$ продукционных правил, определяющих выбор очередного шага решения задачи (комплекса задач);

вектор R текущего состояния семантической сети;

отношение $A(M, R, T)$, перечисляющее для каждого модуля M_i множество R_i значений его кодов завершения и соответствующих пояснительных текстов $T_i = \{t_1, t_2, \dots, t_{n_i}\}$.

Программа-интерпретатор в соответствии с изложенными в [74] принципами последовательно выбирает и выполняет правила из набора P . Каждое P_j предполагает анализ значений некоторого подмножества элементов вектора R и выбор для выполнения очередного программного модуля M_i , т. е. $P_j : r_{1j} \wedge r_{2j} \wedge \dots \wedge r_{nj} \rightarrow M_i$.

После выполнения M_i , переводящего сеть в новое состояние, преобразуется и вектор R за счет помещения в него нового значения r_i . Таким образом, в процессе решения задачи будет происходить и эволюция семантической сети, проявляющаяся в последовательной активизации ее вершин. Новое состояние сети на каждом этапе определяется $R_{k+1} = P_k R_k$, где R_k — состояние на этапе k , а P_k — правило, примененное на этом этапе.

Применение любого правила в процессе поиска решения зависит от состояния вектора R и не зависит от того, какие правила были применены ранее. При возврате по семантической сети в случае необходимости повторной активизации некоторых M_i (пересчет планов) соответствующим r_i в векторе R присваиваются нулевые значения.

Функция объяснения просто реализуется путем выбора из отношения A по текущим значениям кодов r_i вектора R соответствующих текстов t_i , мотивирующих принятие решения на каждом шаге.

Для представленных на рис. 4.3.1 и 4.3.2 блок-схемы и семантической сети набор продукционных правил будет таким:

- P_0 : ЕСЛИ начало работы, ТО все $r_i = 0$.
 P_1 : ЕСЛИ $r_1 = 0$, ТО выполнить M_1 .
 P_2 : ЕСЛИ $r_1 = 1$, ТО выполнить M_2 .
 P_3 : ЕСЛИ $r_2 = 1$, ТО перейти к правилу P_{11} .
 P_4 : ЕСЛИ $r_2 = 2$, ТО перейти к правилу P_{10} .
 P_5 : ЕСЛИ $r_2 = 3$ и $r_4 = 0$, ТО выполнить M_4 .
 P_6 : ЕСЛИ $r_2 = 3$ и $r_4 = 1$, ТО выполнить M_3 .
 P_7 : ЕСЛИ $r_3 = 1$, ТО выполнить M_5 .
 P_8 : ЕСЛИ $r_5 = 2$, ТО выполнить M_6 .
 P_9 : ЕСЛИ $r_6 = 1$, ТО перейти к правилу P_0 .
 P_{10} : ЕСЛИ $r_5 = 1$ или $r_2 = 2$, ТО выполнить M_7 .
 P_{11} : ЕСЛИ $r_7 = 1$ или $r_2 = 1$, ТО выполнить M_8 .
 P_{12} : ЕСЛИ $r_8 = 1$, ТО закончить работу.

Отношение A будет иметь вид

M	R	T
M_1	1	Выполнен расчет плана ремонтов
M_2	1	Получено указание распечатать план ремонтов
M_2	2	Получено указание о расчете потребности в запчастях
M_2	3	Получено указание о согласовании планов ремонтов и перевозок
M_3	1	Проведено согласование планов ремонтов и перевозок
M_4	1	Выполнен расчет плана перевозок
M_5	1	План ремонтов согласуется с планом перевозок
M_5	2	План ремонтов не согласуется с планом перевозок
M_6	1	Выполнена корректировка нормативов межремонтных периодов
M_7	1	Выполнен расчет потребностей в запчастях
M_8	1	Напечатан план ремонтов

Решение задачи по выделенному на рис. 4.3.2 пунктирной линией пути соответствует последовательности выполнения правил $P_0, P_1, P_2, P_5, P_6, P_7, P_{10}, P_{11}, P_{12}$, переводящих вектор R из начального состояния $R^0 = \{0, 0, 0, 0, 0, 0, 0, 0, 0\}$ в состояние $R^K = \{1, 3, 1, 1, 1, 0, 1, 1\}$. Объяснение выбора именно этого пути решения будет включать в себя тексты, выделенные в отношении A курсивом.

Использование сетей Петри для моделирования процесса принятия решений, представленного семантической сетью. Привлекательность сетей Петри [79] можно объяснить двумя основными факторами. Во-первых, логика их функционирования практически полностью адекватна логике развития процесса решения сложных комплексов взаимосвязанных задач. Во-вторых, наличие четкого математического аппарата, описывающего функционирование сетей Петри, позволяет построить на его базе универсальный программный комплекс для управления процессом решения сложных задач.

На рис. 4.3.3 приведена сеть Петри, представляющая семантическую сеть (рис. 4.3.2). Узлам семантической сети ставятся в соответствие события (переходы) сети Петри t_i , т. е. событие t_i соответствует выполнению программного модуля M_i ($i = \overline{1, n}$), а дугам — условия их реализации (позиции) p_j ($j = \overline{1, m}$) [79].

Рассмотрим особенности моделирования динамики поведения системы поиска решения, основанного на матричном представлении сетей Петри, при котором сеть задается четверкой показателей [80] $C = (P,$

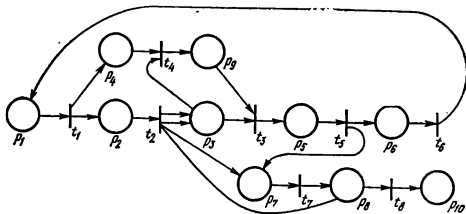


Рис. 4.3.3

T, D^-, D^+). Здесь помимо множества позиций $p_i \in P$ и переходов $t_i \in T$ используются матрицы входов в переходы D^- и выходов из переходов D^+ , имеющие n строк (по одной на переход) и m столбцов (по одному на позицию).

Для сети Петри, представленной на рис. 4.3.3, матрицы имеют следующий вид:

$$D^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad D^+ = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Приведенные матрицы задают структуру сети Петри. При этом наличие в векторе $d^+ [i, j]$ ($j = \overline{1, m}$) (i -й строке матрицы D^+) одного ненулевого элемента соответствует существованию одной выходной позиции для перехода t_i . Если $d^+ [i, j]$ содержит несколько ненулевых элементов, то это означает (для рассматриваемого приложения сети Петри) либо наличие нескольких выходных позиций, которые должны быть помечены после срабатывания перехода t_i , либо наличие нескольких альтернативных выходов, из которых после срабатывания t_i должен быть выбран один. Правильная интерпретация строк с несколькими ненулевыми элементами обеспечивается с помощью табл. 4.3.1, в которой для переходов с альтернативными выходными позициями указываются имена программных

Т а б л и ц а 4.3.1

Номер перехода	Имя модуля	Код завершения	Номер позиции
t_2	M_2	1	p_3, p_4
t_2	M_2	2	p_7
t_3	M_3	3	p_8
t_5	M_5	1	p_7
t_5	M_5	2	p_8

Таблица 4.3.2

модулей, коды их завершения и соответствующие этим кодам помечаемые позиции сети.

Принципиальная схема функционирования сети при поиске решения выглядит следующим образом. Начиная со стартовой маркировки сети последовательно определяются и срабатывают переходы t_i до тех пор, пока не будет достигнута конечная выходная позиция. Для срабатывания

Состояние	Соответствующий переход	Примечание
$W_0 = (1,0,0,0,0,0,0,0,0,0)$	$t_1 (M_1)$	Код завершения = 3; помечается p_3 и p_4
$W_1 = (0,1,0,0,0,0,0,0,0,0)$	$t_2 (M_2)$	
$W_2 = (0,0,1,1,0,0,0,0,0,0)$	$t_4 (M_4)$	Код завершения = 1; помечается p_7
$W_3 = (0,0,1,0,0,0,0,0,1,0)$	$t_3 (M_3)$	
$W_4 = (0,0,0,0,1,0,0,0,0,0)$	$t_5 (M_5)$	Код завершения = 1; помечается p_7
$W_5 = (0,0,0,0,0,0,1,0,0,0)$	$t_7 (M_7)$	
$W_6 = (0,0,0,0,0,0,0,1,0,0)$	$t_6 (M_6)$	Конец работы
$W_7 = (0,0,0,0,0,0,0,0,0,1)$	—	

перехода t_i необходимо выполнение условия [79] $W \geq e [i] D^-$, где $W = (w_1, w_2, \dots, w_m)$ — текущая маркировка сети (w_j — метка позиции p_j); $e [i]$ — единичный вектор-строка.

Результат запуска последовательности переходов $\sigma = \{t_1, t_2, \dots, t_k\}$ определяется $\delta (W, \sigma) = W + f (\sigma) D$, где $D = D^+ - D^-$ — составная матрица изменений, а $f (\sigma) = (e [i_1] + e [i_2] + \dots + e [i_k])$ — вектор запусков переходов, содержащий в качестве координат число запусков соответствующих переходов.

После срабатывания очередного перехода t_i , а фактически — завершения работы модуля M_i , формируется новая маркировка сети. При этом если вектор $d^+ [i, j]$ выхода из перехода t_i содержит несколько ненулевых элементов, а номер этого перехода имеется в табл. 4.3.1 (например, t_2), то это означает необходимость выбора альтернативного выхода в зависимости от значения кода завершения работы модуля M_i и пометки соответствующей позиции (для t_2 это будет p_3 или p_7 , или p_8). В противном случае (если t_i отсутствует в табл. 4.3.1) помечаются все выходные позиции перехода t_i , соответствующие ненулевым элементам в $d^+ [i, j]$ (например, позиции p_2 и p_4 после срабатывания перехода t_1).

Таким образом, осуществляется продвижение по сети (с возможным возвращением назад, как после срабатывания t_6) до достижения конечной выходной позиции, т. е. до завершения решения комплекса задач. В табл. 4.3.2 в качестве примера показана динамика изменения текущей маркировки сети для выделенного пунктиром на рис. 4.3.2 пути решения.

Очевидно, что поиск решения может выполняться полностью автоматически, если коды завершения работы модулей M_i , реализующих переходы t_i , генерируются программно (как в переходе t_5) или при участии пользователя, вводящего в диалоге указания, в зависимости от которых формируются коды завершения диалоговых процедур (переход t_2) и определяются направления дальнейшего развития процесса поиска решения.

Формальный аппарат сетей Петри, обеспечивая подсчет числа срабатываний переходов, не позволяет восстановить их траекторию, т. е.

последовательность запусков переходов, что затрудняет реализацию функции объяснения выбранного пути решения. Поэтому целесообразно вести протокол работы сети, который фиксирует последовательность имен и кодов завершения отработавших программных модулей. Ставя каждой такой паре в соответствие текстовый комментарий, несложно организовать функцию объяснения, аналогично тому как она описана для производственной системы. Сети Петри, в частности подкласс иерархических сетей [79], позволяют эффективно моделировать процессы принятия согласованных решений в сложных многоуровневых системах управления.

В заключение отметим, что в отличие от известных интеллектуальных пакетов [74], обеспечивающих в первую очередь автоматический синтез математических моделей и программ функционирования сложных технических систем, предлагаемая методика более проста и ориентирована на класс задач планирования, где определяющим фактором является не проектирование внутренней вычислительной структуры модуля, а интерпретация результатов его работы. Описанная выше надстройка, которая работает по принципу интерпретации знаний о технологии принятия решений, полностью независима от используемых функциональных модулей и невосприимчива к изменениям технологии (меняется только база знаний). Это выгодно отличает ее от традиционных управляющих программ, которые содержат знания о порядке вычислений или обработке данных в самих себе.

4.3.2. ПРИНЦИПЫ ПОСТРОЕНИЯ ЭКСПЕРТНОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ УПРАВЛЕНИЯ ПЕРЕВОЗКАМИ

Характер многих задач управления работой железнодорожного транспорта таков, что довольно часто для их решения малоэффективно или вовсе неприемлемо использование точных математических методов и алгоритмов. Типичным примером такой задачи является оперативное управление перевозочным процессом, где решения должны приниматься специалистами по управлению на основании анализа сложившейся ситуации и с учетом опыта их профессиональной деятельности. Следовательно, имеют место те характерные условия, которые предопределяют целесообразность использования системы принятия решений, основанной на знаниях, или экспертной системы. Здесь имеются свои сложности, обусловленные в первую очередь высокой пространственно-временной динамикой объектов такой предметной области, как железнодорожный транспорт. Она проявляется в непрерывном изменении местонахождения и состояния транспортных объектов, в первую очередь поездов, вагонов, локомотивов. Для принятия эффективных решений по управлению необходимо, чтобы в экспертной системе (точнее, в ее базе данных/знаний) оперативно и адекватно отображался временной характер технологических процессов на транспорте. Информация о состоянии объекта должна быть достоверной и не устаревшей, так как правила принятия решений включают (или по крайней мере могут включать) в себя переменные, характеризующие как время, так и место происходящих событий.

Принципиальная архитектура экспертной системы реального времени для управления перевозочным процессом на железнодорожной сети в границах некоторого региона представлена на рис. 4.3.4.

Основой системы является база данных/знаний (БД/БЗ). Она содержит, как и в других экспертных системах, декларативные знания о текущем состоянии управляемого объекта — в данном случае это динамическая модель перевозочного процесса или непосредственно база данных и процедурные знания, т. е. правила, используемые при принятии решений, а также правила выбора и использования этих правил.

Динамическая модель перевозочного процесса поддерживается в реальном времени на основании поступающих из ЛС данных о движении, погрузке и выгрузке поездов. Данные могут поступать в виде сигналов автоматических датчиков или сообщений, передаваемых через сеть терминалов. Связной процессор принимает эти данные, преобразует в стандартный формат и заносит в БД/БЗ. При этом вместе с новыми значениями атрибутов туда заносятся и времена их изменений.

Раздел знаний БД/БЗ представляет собой семантическую сеть, вершинами которой являются фреймы знаний. Каждый фрейм содержит одно или несколько правил логического вывода (принятия решений), анализируемых системой. Правила группируются так, что каждый фрейм содержит совокупность правил, определенных на множестве атрибутов (слотов) именно этого фрейма. Это могут быть обычные или динамические атрибуты, значения которых контролируются системой в реальном времени и для которых указываются дополнительно к значению атрибута время его последнего изменения и срок годности. Кроме того, фрейм может содержать запросы к БД/БЗ на языке манипулирования данными используемой СУБД для получения необходимой в данном случае дополнительной информации о процессе, а также операторы вызова программных процедур, реализующих некоторые формальные методы расчета, в том числе оптимизационные. Тогда в правила вывода подставляются в качестве анализируемых параметров результаты выполнения соответствующих запросов и процедур.

В общем случае сеть фреймов БД/БЗ может состоять из ряда относительно независимых фрагментов, описывающих технологию контроля и принятия решений для отдельных задач управления перевоз-

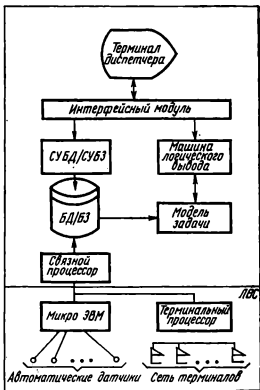


Рис. 4.3.4

ками или отдельных этапов перевозочного процесса — погрузка, выгрузка, перевозка, распределение порожних вагонов и др.

Принципиальная схема функционирования системы выглядит следующим образом. Как только на основании поступившего из ЛС сообщения в модели объекта изменяется значение контролируемого системой динамического атрибута, из БД/БЗ выбирается соответствующий фрагмент сети фреймов, т. е. формируется модель задачи. Затем машина логического вывода анализирует правила начиная с фрейма, содержащего атрибут, для которого зафиксировано новое значение. В первую очередь анализируются правила, использующие в качестве условий значения этого атрибута. Если отклонение нового значения от номинального находится в допустимых пределах (например, исполненный график движения поездов соответствует заданному), то система регистрирует новое значение атрибута и время его формирования. Если новое значение выходит за границы допустимых отклонений, то срабатывает соответствующее правило и начинает работать цепочка прямого вывода. Если очередное выбранное правило содержит другие динамические атрибуты, то правило анализируется лишь в том случае, если срок их годности не истек. Иначе система требует установления новых значений этих атрибутов, причем они могут генерироваться автоматически в результате обработки соответствующего запроса к БД/БЗ либо запрашиваться в диалоговом режиме у пользователя. Результатом срабатывания правила является переход к следующему фрейму и т. д. до прихода в некоторую конечную вершину сети. Так как каждый фрейм может содержать, кроме анализируемых правил перехода, еще и соответствующие правила принятия решений, то в результате такого прямого прохода по сети система выдает перечень директивных указаний для приведения объекта в нормальное состояние.

В системе должна быть также возможность построения и обратной цепочки рассуждений. Она необходима, когда требуется проверить целесообразность и эффективность некоторого планируемого управляющего воздействия (правила высокого уровня). Такая проверка предусматривает обратный проход по сети фреймов с целью установления, соответствуют ли текущие значения параметров объекта необходимости принятия планируемого управляющего воздействия.

Рассматриваемая архитектура предполагает в отличие от обычных экспертных систем генерацию в реальном времени из БД/БЗ основной массы необходимой для принятия решений информации, а не получение ее в результате часто утомительного диалогового опроса пользователей. Кроме того, в рассмотренном проекте обеспечивается более целенаправленный поиск решения за счет автоматического выбора и анализа только тех правил, которые содержат в составе условий атрибуты с изменившимися значениями.

4.4. ДИАЛОГОВАЯ ТЕХНОЛОГИЯ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

Промышленный транспорт представляет собой крупномасштабный транспортно-технологический комплекс средств доставки и переработки грузов. Сложный характер взаимодействия объектов и элемен-

тов этого комплекса, непрерывное его развитие по пути интеграции в макросистемы придадут особую актуальность проблеме создания эффективных инструментальных средств для его исследования и оптимизации. Эффективным подходом к ее решению является использование концепции программирования жизненных циклов объектов и систем новой техники и технологии. Жизненный цикл можно рассматривать как основную структуру, упорядочивающую во времени всю совокупность взаимосвязанных этапов проектирования, постройки и целевого использования создаваемых объектов и систем. Такая структура дает ясную концептуальную основу для интеграции в единую целевую программу всех задач перечисленных выше этапов. Как показывает опыт, наиболее продуктивной формой реализации концепции программирования жизненных циклов являются интегрированные системы автоматизированного проектирования и управления (САПР/АСУ или CAD/CAM системы). Они обеспечивают сквозное управление жизненным циклом объекта на всех этапах, начиная с проектирования и постройки до программируемой эксплуатации. Основным аппаратом этих систем являются методы моделирования, использующие аналитические или имитационные модели, наиболее адекватно отображающие суть физических процессов функционирования объектов и комплексов.

Принципы построения САПР транспортных объектов и технологий [81] определяются двумя основными группами факторов. Первая характеризует общие требования, предъявляемые сегодня к системам моделирования: концептуальность описания моделируемых объектов, наличие развитых средств диалога пользователя с системой, сочетание конструктивной универсальности с возможностью проблемной ориентации и др. Вторая отражает существующие особенности сферы предполагаемого использования САПР — специфику задач моделирования разных этапов жизненного цикла, разнообразие структур объектов, использование методов и средств машинной графики, наличие широкого круга пользователей с разным уровнем профессиональной подготовки и т. д.

Анализ совокупности перечисленных факторов позволяет сформулировать три основных принципа построения САПР для оптимизации жизненных циклов крупномасштабных транспортно-технологических комплексов: концептуальное единство, конструктивная универсальность и проблемная ориентация.

Концептуальное единство заключается в системной интеграции всех видов обеспечений задач моделирования в единую информационно-вычислительную среду, обеспечивающую взаимоувязанное решение задач программирования жизненных циклов транспортно-технологических комплексов.

Конструктивная универсальность означает типовость структурного описания и внутреннего представления моделируемых объектов в базе данных, использование принципа интерпретации в моделировании, т. е. отделение описания модели (ее структуры и параметров) от программных средств моделирования, а также накопление знаний экспертов и их использование при анализе результатов моделирования и принятии решений по оптимизации проектируемых объектов и технологий.

В соответствии с изложенными принципами структура САПР должна включать в себя следующие основные компоненты:

базу данных, содержащую описание схем базовых конструкций транспортно-технологического комплекса, модели конкретных комплексов и составляющих их объектов и элементов. При этом в перечень атрибутов включаются специальные параметры, используемые программными средствами моделирования, а также данные, обеспечивающие обработку графических изображений объектов;

банк программных средств моделирования, включающий как универсальные средства (SIMULA-67, GPSS, GASP и др.), так и узкоспециализированные, но высокопроизводительные для конкретных приложений средства;

базу знаний, содержащую правила анализа параметров моделируемых объектов и технологий, которые обеспечивают их квалифицированную экспертную оценку и выработку рекомендаций по оптимизации проектных решений;

комплекс инструментальных средств поддержки взаимодействия пользователей с САПР, генерации моделей объектов и программ моделирования, управления процессом моделирования.

Принципиальная схема функционирования САПР при решении конкретной задачи моделирования выглядит следующим образом. По запросу пользователя из базы данных выбирается существующая либо генерируется новая модель исследуемого объекта или системы. Затем выполняется настройка модели, т. е. с учетом специфики используемого аппарата (метода) моделирования иницируются конкретные значения параметров, характеристик процессов, указываются процедуры генерации случайных чисел, формирования очередей, обработки статистики и т. п. Сформированная модель передается программе моделирования, которая выполняется под контролем пользователя, причем процесс моделирования может быть в любой момент приостановлен, модель объекта или отдельные ее параметры модифицированы и процесс продолжен. Принятые пользователем по результатам моделирования решения, касающиеся изменения структуры или отдельных характеристик исследуемого объекта, могут быть отображены соответствующим образом в базе данных.

Рассмотрим в качестве примера диалоговую систему проектирования оптимальной организации транспортного обслуживания промышленных предприятий [82], разработанную в соответствии с изложенными выше концепциями.

Железнодорожный транспорт является основным видом транспорта, с помощью которого реализуются перевозки практически на всех крупных промышленных предприятиях различных отраслей народного хозяйства. При этом практически все внешние перевозки (ввоз сырья и вывоз готовой продукции) осуществляются вагонным парком общего пользования, т. е. вагонами МПС. Основным технико-экономическим показателем использования вагонов МПС на предприятии является оборот вагона — время его нахождения на подъездном пути предприятия от приема с сети МПС до сдачи на сеть. Норма оборота в числе других показателей определяется при расчете ЕТП работы

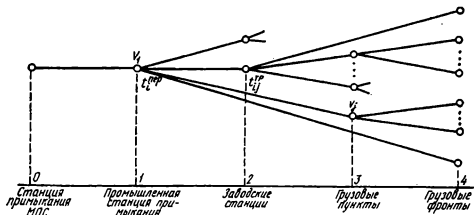


Рис. 4.4.1

станции прицепки МПС и подъездного пути. Используемый для расчета ЕТП графоаналитический метод не учитывает в полной мере неравномерность прибытия поездов на подъездной путь и вероятностный характер времен выполнения отдельных операций в зависимости от количества и назначений вагонов в поезде. В результате при задании нового графика прибытия и отправления поездов вновь рассчитанные показатели ЕТП могут значительно измениться. Кроме того, громоздкость и трудоемкость графоаналитического метода не позволяют оптимизировать расчет оборота вагона с точки зрения минимизации суммарных затрат железной дороги и промышленного предприятия, а также оперативно перерасчитывать ЕТП при изменении условий перевозок и грузовой работы — приобретении или списании локомотивов, реконструкции путевого развития, техническом перевооружении грузовых фронтов и др.

В Институте кибернетики им. В. М. Глушкова АН УССР разработана система [82], обеспечивающая автоматизированные расчеты и оптимизацию основных показателей ЕТП в соответствии с основными положениями изложенной в [83] методики. Система реализована на базе персональной ЭВМ и ориентирована на диалоговое взаимодействие с работниками железнодорожного транспорта, не имеющими подготовки в области программирования. Основная форма диалога — меню, инициируемое системой.

Модель подъездного пути и расчет основных показателей ЕТП. Схема путевого развития в общем случае представляется графом $G(V, X)$ древовидной структуры, приведенном на рис. 4.4.1.

Вершинам графа соответствуют сгруппированные по уровням иерархии технические объекты — станции, группы фронтов (грузовые пункты), фронты погрузки-выгрузки, на которых выполняются операции по переработке вагонов, прием-сдача и расформирование-формирование поездов, погрузка-выгрузка вагонов и др. Как видно из рис. 4.4.1, в частных случаях объекты второго и/или третьего уровня могут отсутствовать, т. е. вагоны могут подаваться, например, с промышленной станции прицепки прямо на грузовой пункт или непосредственно на грузовой фронт. Ребрам графа соответствуют участки железнодорожных путей, связывающие между собой перечислен-

ные выше технические объекты. Для каждой вершины и ребра указывается объем переработки и транспортировки вагонов, который рассчитывается на основании данных о количестве прибывающих с сети МПС вагонов, размеров погрузки-выгрузки по каждому фронту, а также с учетом внутренних перестановок порожних вагонов при выполнении сдвоенных операций.

Время переработки в вершине v_i (здесь не учитываются грузовые операции) зависит от общего количества вагонов n_i^n и числа групп m_i^n вагонов разного назначения в перерабатываемом поезде, т. е. $t_i^{nep} = f(n_i^n, m_i^n)$ и рассчитывается в соответствии с установленными нормативами. При этом состав поезда (передачи) вычисляется $n_i^n = W_i/\lambda_n$ или $n_i^n = W_i/\lambda_y$, где W_i — суточный вагонопоток, следующий через v_i , а λ_n и λ_y — суточное число подачи и уборок поездов с сети МПС на подъездной путь. Количество назначений m_i^n определяется по предложенной в [83] формуле

$$m_i^n = \frac{\min(N_i, n_i^n)}{1 + 0,55 \sqrt{N_i \sum_{j=1}^{N_i} p_{ij}^2 - 1}},$$

где N_i — число непосредственно примыкающих к v_i объектов более низкого уровня иерархии (для рис. 4.4.1 $N_i = 4$); p_{ij} — вероятность назначения вагона с v_i на v_j , которая определяется как отношение вагонопотока W_{ij} , следующего с v_i на v_j , ко всему вагонопотоку W_i , отправляемому с v_i на все N_i , т. е. $p_{ij} = W_{ij}/W_i$, или, что то же самое, $p_{ij} = W_{ij}/\sum_j W_{ij}$.

Время транспортировки (в том числе подачи-уборки на грузовые фронты) вагонов с v_i на v_j ($v_i, v_j \in X$) зависит от количества вагонов n_{ij}^n в передаче, способа транспортировки (учитывается с помощью нормативных коэффициентов a_{ij} и t_{ij} на выполнение маневровых операций), скорости V_{ij} и расстояния L_{ij} транспортировки, т. е. в общем виде

$$t_{ij}^{tp} = a_{ij} + b_{ij}n_{ij}^n + L_{ij}/V_{ij}.$$

При наличии на подъездном пути приоритетных технологических перевозок возможны задержки выполнения операций по переработке вагонов МПС. Они учитываются при расчете t_i^{nep} и t_{ij}^{tp} с помощью специального коэффициента, значение которого вычисляется в зависимости от времени технологических перевозок, определяемого по контактному графику.

Следует подчеркнуть очевидную зависимость t_i^{nep} и t_{ij}^{tp} от организации передачи поездов между сетью МПС и подъездным путем, а точнее, от λ_n и λ_y , определяющих значения n_i^n и m_i^n .

Таким образом, общее время переработки вагонопотока МПС на подъездном пути без учета грузовых операций равно

$$T^{n/n} = \sum_i t_i^{nep} + \sum_{ij} t_{ij}^{tp},$$

причем это время будет разным для разных сочетаний λ_n и λ_y .

Отсюда определим величину рабочего парка локомотивов, необходимого для реализации перевозок

$$n_p^n = T^{n/n} / (24 - t_{np}^n),$$

где t_{np}^n — время непроизводительных простоев локомотива (экипировка, смена бригад и др.).

При наличии в схеме подъездного пути внутривзаводских станций, что характерно для крупных промышленных предприятий, расчет n_p^n дифференцируется по промышленной станции приыкания и внутривзаводским станциям, что позволяет определить для каждой из них необходимое число локомотивов из общего парка по подъездному пути.

В основу расчета времени оборота вагона положено рассмотрение подъездного пути (см. рис. 4.4.1) как многофазной многоканальной системы массового обслуживания, где отдельной фазе соответствует множество вершин одного уровня иерархии графа или множество ребер, для которых правые смежные вершины принадлежат одному иерархическому уровню. Обслуживающими приборами являются локомотивы и грузовые механизмы, а в роли заявок на обслуживание выступают группы вагонов, причем в начальной фазе это будет поезд в целом, передаваемый с сети МПС на промышленную станцию приыкания, а далее — группа перерабатываемых (включая уже грузовые операции) или транспортируемых вагонов, вплоть до отдельного вагона в предельном случае. Полный цикл обслуживания включает в себя все операции обработки вагонов в прямом и обратном направлениях — прием, расформирование, транспортировка (подача), выгрузка, перестановка, погрузка, транспортировка (уборка), формирование, сдача.

Оборот вагона складывается из времен обслуживания заявки в каждой фазе и времен ожидания обслуживания в случае занятости обслуживающего прибора (локомотива), т. е.

$$T^{об} = \sum t_i + \sum t_i^{ож}.$$

Время обслуживания рассчитывается как средневзвешенное для каждой фазы $t_i = \sum_i t_i^{сп} p_i$ или $t_i = \sum_{ij} t_{ij}^{сп} p_{ij}$, где $p_i = W_i / \sum_i W_i$, а $p_{ij} = W_{ij} / \sum_{ij} W_{ij} (\sum_i W_i)$ — вагонопоток, перерабатываемый во всех вершинах одного уровня).

Время обслуживания на грузовых фронтах (погрузка-выгрузка и дополнительные операции по очистке, взвешиванию и т. д.) определяется по нормативам.

Время ожидания обслуживания $t_i^{ож}$ вычисляется согласно [83], исходя из предположения, что поток заявок является пуассоновским, а время обслуживания заявки распределено по показательному закону.

Системная реализация. Все исходные данные, характеризующие технические объекты моделируемых промышленных предприятий, — промышленную станцию приыкания, внутривзаводские станции, грузовые пункты и грузовые фронты — вводятся в состав входных документов стандартной формы и хранятся в специально организованной

Таблица 4.4.1

Промышленная станция примыкания — Заводская
Предприятие — Киев-Московский

№ ш/п	Показатели	Значение		Вариант		Отклоне- ние (+, -)
		мини- мальное	макси- мальное	опти- мальный	рассмат- риваемый	
1.	Количество подач	4	7	6	6	0
2.	Количество уборок	4	7	4	4	0
3.	Рабочий парк тепловозов, ед.	5	7	6	6	0
4.	Инвентарный парк тепло- возов, ед.	6	10	8	8	0
5.	Оборот вагона, ч	5.36	5.90	5.75	5.75	0.00
6.	Затраты предприятия, тыс. руб.	1526.1	1649.3	1531.7	1531.7	0.0
7.	Затраты железной дороги, тыс. руб.	193.2	212.9	207.2	207.2	0.0
8.	Приведенные затраты, тыс. руб.	1739.0	1845.0	1739.0	1739.0	0.0

базе данных системы. Для проведения конкретных расчетов генерируется модель подъездного пути, в которую из базы данных включаются объекты, указываемые пользователем в процессе диалога с системой. Такая избирательная генерация позволяет автоматически создавать модели отдельного транспортного района крупного промышленного предприятия, всего предприятия в целом, нескольких или всех предприятий, обслуживаемых одной станцией примыкания (т. е. всего подъездного пути).

Для сгенерированной модели рассчитываются основные технико-экономические показатели работы. Расчет выполняется для каждой пары значений числа подач λ_n и уборок λ_y , т. е. для каждого λ_n перебираются все λ_y . Минимальные и максимальные значения λ_n и λ_y вычисляются в зависимости от входного вагонотока с сети МПС, но могут быть указаны и в диалоге с пользователем. При расчете, кроме числа локомотивов и оборота вагонов, определяются согласно [83] затраты предприятия на реализацию перевозок (сюда входят капиталовложения и затраты на содержание пути и подвижного состава) и производство грузовых операций, затраты железной дороги, связанные с простоем вагонов, а также общие народнохозяйственные затраты как сумма затрат предприятия и железной дороги. По окончании расчета на экран, а по указанию пользователя и на печать, выдается итоговая таблица (табл. 4.4.1), содержащая показатели оптимального с точки зрения минимума суммарных затрат варианта, а также допустимые границы изменения значений каждого из этих показателей. Изменения значений основных показателей с индикацией оптимального варианта могут быть также показаны на экране в графической форме.

В реальных условиях может оказаться, что рассчитанный оптимальный вариант не может быть принят в качестве окончатель-

Таблица 4.4.2

Промышленная станция примыкания — Заводская
Предприятие — Киев-Московский

Форма транспортного обслуживания	Время оборота вагона, ч	Тепловозный парк, ед.		Затраты, тыс. руб.	
		рабочий	инвентарный	приведенные	фактические
Транспортный цех	4.18	4	28	3171.1	3191.1
МПС (пути предприятия)				2309.1	1955.6
МПС (пути железной дороги)	5.75	6	8	2269.4	1955.6
Автотранспорт				4132.7	4132.7
МППЖТ				1739.0	1865.5

ного, если критерий суммарных затрат по тем или иным причинам не имеет определяющего значения. В этом случае задача приобретает характер многокритериальной с функцией цели $f = \{f_i(\alpha)\}$ (в данном случае $i = \overline{1,8}$), где α представляет собой множество эффективных альтернатив [84], определяемое границами изменения допустимых значений каждого критерия, т. е. от $f_{i(\min)}$ до $f_{i(\max)}$ (см. табл. 4.4.1). Выбор компромиссного решения из этих альтернатив (т. е. окончательного варианта) осуществляется пользователем с помощью специальной диалоговой процедуры, работающей в двух режимах. Первый режим реализует известную идею многокритериальной оптимизации по схеме, в целом соответствующей изложенной в [84] методике. Целенаправленный поиск решения осуществляется пошаговым заданием значений критериев (с экрана указывается номер и желаемое значение показателя табл. 4.4.1) с пересчетом границ $f_{i(\min)}$ и $f_{i(\max)}$, в рамках которых уточняются значения соответствующих критериев. Второй режим предусматривает простую оценку альтернатив — изменение значений показателей табл. 4.4.1 в зависимости от последовательной корректировки отдельных из них. Это позволяет определить, например, какой необходим локомотивный парк при заданных λ_r , λ_y и обороте вагона, какие при этом окажутся затраты и т. п. Для выбранного в качестве окончательного варианта по указанию пользователя на экран или твердую печать выдается таблица расчлененного по отдельным элементам времени оборота вагона.

Одной из основных функций системы является анализ экономических показателей для разных форм транспортного обслуживания и выбор оптимальной из них. Вначале для отдельных предприятий определяются затраты по таким формам, как транспортный цех, обслуживание средствами железной дороги с передачей железнодорожных путей на ее баланс и с содержанием путей силами предприятия, обслуживание автотранспортом с раскрытием подъездного пути. Затем генерируется модель, включающая предприятия, которые предполагается объединить в МППЖТ. Для них вычисляются суммарные затраты по каждой из вышеперечисленных форм обслуживания, а также рассчитываются

затраты при объединении их в МППЖТ. Эффективность создания МППЖТ определяется в первую очередь сокращением размеров локомотивного парка и снижением удельных расходов на содержание пути и подвижного состава. Краткие сравнительные оценки для разных форм обслуживания (табл. 4.4.2) выдаются на экран и твердую печать, а более детализированные — только на печать.

Для улучшения работы подъездного пути могут использоваться разные технические и технологические мероприятия — приобретение локомотивов, укладка вторых путей, замена типов рельсов с повышением скоростей движения, замена погрузочно-разгрузочных механизмов и т. д. Каждое из этих мероприятий, направленное на сокращение простоя вагонов и снижение соответствующих затрат, требует, вместе с тем, и определенных капиталовложений. Система позволяет моделировать реализацию таких мероприятий и оценить их эффективность. Для этого выполняются расчеты для существующих условий перевозок и грузовой работы, затем в исходные данные в диалоговом режиме вносятся соответствующие планируемому мероприятию коррективы и вновь выполняются расчеты по модели. Сопоставление затрат по существующему и проектируемому вариантам позволяет оценить эффективность последнего.

В заключение отметим, что разработанная система позволяет оперативно (вплоть до ежесуточного режима) пересчитывать и анализировать различные варианты и основные показатели работы подъездного пути при изменении условий перевозок и грузовой работы, что практически невозможно выполнять вручную из-за высокой трудоемкости и многовариантности расчетов.

Описанная система может быть дополнена программным блоком, реализующим функции экспертного анализа рассматриваемого варианта работы подъездного пути и выработки рекомендаций по его оптимизации. Основанием для такого анализа может служить расчлененный по всем элементам модели (вершинам и ребрам графа G) оборот вагона, позволяющий выявить узкие места и рекомендовать технические и технологические мероприятия по их устранению.

Глава 5

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОИЗВОДСТВЕННО-ТРАНСПОРТНЫХ СИСТЕМ

5.1. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ В НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ ТРАНСПОРТОМ

В условиях широкого внедрения новых информационных технологий в управлении промышленным транспортом особую актуальность приобретает внедрение современных экономико-математических методов в практику планирования и управления производственно-транспортными процессами.

Техническая реализация НИТ характеризуется развитой сетью ПЭВМ, имеющих интерфейс с большой ЭВМ центра обработки данных системы управления производственно-транспортным процессом. ПЭВМ является интеллектуальным терминалом, составляя техническую оснащенность автоматизированного рабочего места ЛПР. Это позволяет ЛПР получить, с одной стороны, прямой доступ к информации баз данных центральной системы и — с другой — иметь в качестве математического обеспечения ПЭВМ мощный инструмент для выработки оптимальных управленческих решений как по формализованным критериям, так и по неформальным (интуитивным) соображениям.

Среди актуальных экономико-математических методов, используемых в таких системах, особую роль приобретает имитационное моделирование, позволяющее учитывать случайные воздействия на управляемую транспортную систему и с большой степенью вероятности предсказывать ее поведение при текущем и долгосрочном планировании. При этом представляется весьма перспективным создание в рамках НИТ имитационных систем, в которых наряду со всеми необходимыми атрибутами АСУ реализуется диалог ЛПР — ЭВМ, а имитационные процедуры используются в сочетании с оптимизационными экономико-математическими методами [87—95].

Диалог ЛПР — ЭВМ позволяет решать практически все задачи управления транспортно-технологическими объектами в интерактивном режиме с внутримашинным генерированием случайных процессов и имитацией нештатных ситуаций в производственно-транспортном процессе. Наиболее эффективные решения из выработанных путем имитации ЛПР сможет принимать с помощью оптимизационных методов, используя развитую систему экспертных оценок, которая тоже может служить частью НИТ в управлении промышленным транспортом.

Известно, что имитационное моделирование получает все большее распространение как инструмент повышения качества управленчес-

ких решений в управлении производственно-экономическими системами. Оно позволяет проводить широкий многовариантный анализ данных, имеющих ключевое значение для принятия как текущих в реальном масштабе времени, так и стратегических решений относительно новых технологических и транспортных процессов, совершенствования организации и управления производственно-экономическими объектами [88]. С помощью имитационного моделирования становится возможным выбор наиболее эффективных вариантов создания и размещения новых производственных мощностей и транспортных коммуникаций, выполнять анализ резервов производства и оптимизацию режимов транспортно-технологических операций [89, 90].

На базе ограниченного числа исходных допущений и развитой системы экспертных оценок с помощью методов имитации можно получить надежное решение проблемы управления запасами, сопряженных проблем поставок и транспортировки складированных материалов. При этом решаются задачи определения минимального страхового запаса для предупреждения риска неудовлетворенного спроса [91] и оптимизации графиков погрузочно-разгрузочных и транспортных операций.

В данном обзоре остановимся кратко на ряде публикаций по проблеме использования имитационного моделирования в динамических системах управления внутризаводским транспортом. В статье [92] анализируются некоторые направления использования методов имитации для совершенствования разработки и эксплуатации АСТМ. Здесь методы моделирования предлагается применять для аргументированной оценки целесообразности внедрения различных проектов АСУ. С помощью имитации представляется возможным разработать обоснованные правила управления АСТМ, исследовать их функционирование при критических нагрузках, совершенствовать транспортные системы в связи с изменением условий функционирования предприятий. В работе делается вывод о возможности значительного повышения эффективности АСТМ на основании использования новых методов автоматизации управления и имитационного моделирования.

Ряд публикаций посвящен разработке специального программного обеспечения для имитации деятельности промышленного транспорта.

В работе [93] приводится описание пакета прикладных программ имитации поведения заводской транспортной системы, а в [94] дана сравнительная характеристика наиболее распространенных программных средств для микро-ЭВМ, реализующих задачи маршрутизации и оптимального планирования работы транспорта.

Новые информационные технологии требуют и новых подходов к созданию имитационных систем для управления производственно-транспортными объектами. В этом отношении представляют интерес разработки имитационного моделирования на основе искусственного интеллекта и экспертных систем. В [95] отмечается, что искусственный интеллект должен играть возрастающую роль в качестве инструмента интеллектуальной поддержки имитации и расширения потенциальных возможностей моделирования. Как перспективным рас-

считается использование искусственного интеллекта для разработки языка перевода декларативных (факты) и процедурных (как сделать) знаний в область конкретного имитационного анализа. Здесь могут быть полезны методы логического программирования [96], которые в сочетании с элементами искусственного интеллекта позволяют облегчить процесс выбора и принятия решений путем имитации множества альтернативных вариантов на основе закономерностей человеческой логики и мышления. В целом, подводя итоги данного краткого обзора, следует подчеркнуть, что развитие имитационного моделирования идет в направлении его гибкости, модульности, сочетания с оптимизационными методами, искусственным интеллектом и экспертными системами, что особенно важно для «встраивания» его в новые информационные технологии управления транспортными процессами.

В последующих параграфах подробнее остановимся на классических схемах имитационного моделирования и методике его применения к управлению производственно-транспортными системами.

5.2. ОСНОВНЫЕ ЭТАПЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Широкий диапазон применения имитационного моделирования производственно-транспортных систем, разнообразие решаемых задач и исследовательских проблем затрудняют выработку каких-либо универсальных рекомендаций по проведению имитационных экспериментов. Не существует твердых правил и обоснованных методов, как формализовать задачу исследований, очертить необходимый круг переменных и параметров модели и связать их функциональными соотношениями, провести эксперименты на ЭВМ и оценить их результаты.

В каждом конкретном случае построение имитационной модели основано на изучении действительного хода процессов в реальной системе и представлении их с помощью избранной системы показателей. При этом исследователю нет необходимости подгонять математическую модель под известную схему, заранее предполагающую принадлежность модели к какому-либо классу задач математического программирования. Это позволяет описывать производственно-экономический процесс без существенных упрощений как можно ближе и точнее к реальности исследуемой системы показателей. Математическая модель здесь, невзирая на ее сложность и громоздкость, должна адекватно отражать сущность производственно-экономической системы.

Таким образом, имитационное исследование на всех его этапах требует творческого подхода и неординарности решений экспериментатора, индивидуального подхода при решении каждой конкретной задачи.

Любая задача, решаемая методами имитационного моделирования, может иметь свои особенности при реализации отдельных фаз эксперимента. Опыт позволяет сформировать методику имитационного моделирования производственно-транспортных систем, а также выделить некоторые закономерности в построении математических моделей алгоритмизации, программировании и проведении имитации на ЭВМ.

Имитационное моделирование широкого класса производственно-экономических и транспортных систем в большинстве своем требует проведения следующих этапов [97].

1. Изучение моделируемого объекта или процесса.
2. Формулировка проблемы и целей моделирования.
3. Формализованное описание и разработка математической модели.
4. Разработка моделирующего алгоритма.
5. Составление программы для ЭВМ.
6. Оценка пригодности модели.
7. Планирование и обработка результатов эксперимента.

Следует подчеркнуть, что выполнение перечисленных этапов обязательно для большинства имитационных исследований, но для решения некоторых реальных практических задач может оказаться недостаточным. Так, при решении некоторых задач текущего и долгосрочного планирования работы отрасли и транспортных процессов в отраслевом масштабе потребовалось выполнение процедуры отсеивания построенных с помощью имитации, бесперспективных вариантов планов [98]. Здесь потребовалось реализовать такой дополнительный этап, как решение оптимизационной задачи по выбору лучших вариантов из построенных допустимых.

Вообще проблема стыковки имитационных и оптимизационных моделей является весьма актуальной и ей уделяется много внимания авторами [99, 100]. Отметим, что в конкретных разработках такая стыковка может быть сформулирована отдельным этапом методики проведения имитационных экспериментов.

В процессе имитационного моделирования реальных производственно-экономических систем каждый из перечисленных этапов данной методики может оказаться далеко неравнозначным как по объему, так и по содержанию. Зачастую некоторые из этапов необходимо объединять и рассматривать как одну из фаз проводимого исследования.

Прежде всего это относится к этапам изучения моделируемой системы и формулировки проблем и целей моделирования. В один этап они объединяются при проведении имитационных экспериментов для хорошо изученных процессов и систем либо в том случае, если проблема или конкретная цель формулируется до начала всех проводимых исследований.

Поскольку изучаемые нами производственно-транспортные и экономические системы относятся к классу сложных динамических систем с большим числом вероятностных характеристик, то следует особое внимание уделять построению математических моделей и моделирующих алгоритмов. При выполнении указанных этапов возникает ряд принципиальных вопросов, требующих специального рассмотрения.

Вероятностные характеристики могут быть связаны со следующими показателями: 1) эффективностью системы, т. е. вероятностью работы с необходимым качеством и издержками выходных показателей; 2) надежностью системы, т. е. вероятностью безаварийной работы; 3) простотой отдельных элементов, узлов, блоков и системы в целом,

не связанных с аварийными ситуациями; 4) вероятностью различных событий и воздействий на систему со стороны внешней среды.

Таким образом, в математических моделях наряду с аналитическими функциональными зависимостями должны быть описаны и стохастические зависимости, отражающие перечисленные вероятностные характеристики, т. е. входные и выходные переменные модели могут быть связаны между собой как детерминированными, так и стохастическими соотношениями.

Что касается моделирующих алгоритмов, то они должны представлять собой комбинации методов вероятностного моделирования реализующие случайные процессы в системах, с детерминированными расчетами.

Особое внимание в математических моделях должно уделяться формализованному описанию динамических характеристик, т. е. описанию зависимостей переменных функционирования системы от времени. Соответственно в моделирующих алгоритмах первостепенное значение приобретает вопрос реализации системного времени.

Если алгоритмизация модели не встречает технических трудностей, а также при наличии специального алгоритмического языка моделирования, легко описывающего исследуемый процесс, целесообразно в один этап объединить разработку моделирующего алгоритма и составление программы для ЭВМ. При построении вероятностных и детерминированных автоматов в один этап практически объединяются разработка математической модели и разработка моделирующего алгоритма, которые задаются в виде таблицы условных функционалов переходов (ТУФП).

Предлагаемая методика является развитием идей моделирования и имитационных экспериментов, сформулированных в работах советских ученых Н. П. Бусленко, А. А. Бакаева, И. Н. Коваленко, Н. Н. Моисеева, Н. М. Яровицкого и др. [101—105], а также зарубежных авторов [98, 106—108]. Методика послужила отправной точкой для создания проблемно-ориентированных моделей различных производственно-транспортных и экономических систем [109—111] и проведения авторами имитационных экспериментов с ними.

5.3. ИЗУЧЕНИЕ МОДЕЛИРУЕМОЙ СИСТЕМЫ И ФОРМУЛИРОВКА ПРОБЛЕМЫ

Вопрос предварительного исследования моделируемой системы носит принципиальный характер, поскольку многие авторы считают его второстепенным. В работах [106—108] предлагается начинать имитационное моделирование с формулировки проблемы, предполагая распределения многих случайных величин заданными априори. При таком подходе исследуются внешние проявления изучаемого процесса, а внутреннее его механизмы остаются практическими без внимания.

Известно, что один из самых сложных вопросов анализа реальной системы — это изучение внутренних механизмов ее функционирования, а также формулировка гипотез о законах распределения случайных величин и зависимостях между ними. Машинная имитация без

предварительного изучения системы может создать только иллюзию, что, симитировав все явления и процессы, мы непременно их исследовали [103].

Изучение моделируемой системы или процесса, предшествующее имитационным экспериментам, необходимо для выявления закономерностей функционирования системы и оценки влияния на нее внешней среды. В результате изучения системы определяется ее функциональная и организационная структура, очерчивается круг задач, решаемых в процессе управления работой системы и требующих для своего решения методов имитационного моделирования.

В каждом конкретном случае построение имитационной модели основано на изучении действительного хода процессов в системе и описания их с помощью некоторой избранной в результате анализа системы показателей. Применение имитационного моделирования может оказаться плодотворным только в том случае, когда предварительно проведен обстоятельный анализ исследуемой системы, выявлены главные ее компоненты и дана удовлетворительная оценка погрешности при отбрасывании или игнорировании второстепенных составляющих. Все это позволяет перейти к адекватному формализованному описанию системы, т. е. к созданию действительно пригодной математической имитационной модели.

Для производственно-транспортных систем наиболее приемлемым и универсальным методом их изучения, позволяющим ответить на основные вопросы предварительного обследования, является экономико-статистический анализ.

Экономико-статистический анализ позволяет.

1. Выявить переменные и характеристики процессов в системе, подверженные в значительной степени случайным воздействиям и имеющие вероятностный характер поведения (например, интенсивность транспортных потоков, скорость движения составов и т. п.).

2. Построить законы распределения случайных величин, оценить и рассчитать параметры законов и выявить парную, а также множественную корреляцию между случайными величинами.

3. Определить количественные и качественные зависимости между детерминированными и случайными переменными системы в виде разного рода аналитических и регрессионных уравнений.

4. Построить экономико-статистические модели процессов, связывающие детерминированными и стохастическими функциональными соотношениями входные и выходные переменные с производственно-экономическими показателями и параметрами процессов.

Следует отметить, что последняя фаза экономико-статистического анализа может входить составной частью в разработку математической имитационной модели — третьего этапа описанной выше методики имитационного моделирования.

В процессе имитационных экспериментов, т. е. при эксплуатационных испытаниях имитационной модели, конкретные значения случайных переменных генерируются на ЭВМ с помощью специальных датчиков случайных чисел. Очевидно, что генерируемые случайные числа должны подчиняться тем законам вероятностных распределе-

ний, которым подчиняются реальные переменные исследуемой системы. Это является непременным условием адекватности имитационной модели реальным процессам и воздействиям внешней среды на моделируемые явления. Отсюда вытекает важность и целесообразность выполнения основных этапов экономико-статистического анализа.

Рассмотрим некоторые примеры предварительного обследования производственно-экономических и транспортных систем. Так, выполненный авторами экономико-статистический анализ функционирования транспортно-технологического комплекса некоторых горно-обогатительных комбинатов позволил определить функциональную и организационную структуру карьерного транспорта для автоматизированного управления, оценить тесноту взаимосвязи транспортных работ и технологических процессов добычи и переработки руды, а также вероятностный характер некоторых экономических и производственных показателей работы транспорта.

На ряде карьеров Криворожского бассейна проводился экономико-статистический анализ простоев транспорта и технологического погрузочно-разгрузочного оборудования как с помощью хронометражных наблюдений, так и по выборкам регистрируемых простоев в отчетных графиках исполненного движения. В результате обработки статистических данных построены гистограммы, сформулированы гипотезы вероятностных распределений и с помощью χ -критерия Пирсона проверены достоверности гипотез. Для получения численных значений параметров распределений использовались известные численные методы математической статистики, а также методы графоаналитических построений. Были сформулированы гипотезы о логарифмическом нормальном и гамма-распределении простоев экскаваторов в ожидании транспортного обслуживания и о нормальном распределении простоев транспорта в ожидании освобождения пунктов погрузки и разгрузки. Вероятности того, что принятые гипотезы не противоречат опытным данным после аналитических проверок, оказались в пределах 0,05—0,15.

Выявлено, что простои экскаваторов на карьерах в ожидании транспортного оборудования составляют 55—70 % всех простоев основного технологического оборудования и при этом занимают 15—20 % их рабочего времени. Таким образом, можно сделать вывод, что имеется большой резерв повышения производительности добычного и транспортного оборудования горно-обогатительных комбинатов.

Для установления причин столь высоких простоев транспортно-технологического оборудования и потерь горнорудного производства авторами проведены статистические исследования интенсивности потоков информации с участков транспортной сети в управляющий центр горнотранспортного комплекса карьера. Установлено, что интенсивность потока сообщений, поступающих к транспортному диспетчеру, весьма значительна и подчинена нормальному закону распределения вероятностей. Исследовался также вероятностный характер длительностей интервалов между соседними сообщениями, для которых выдвинута гипотеза об экспоненциальном законе. Проверка с помощью критерия Пирсона показала, что принятые гипотезы не противоречат статистическим наблюдениям.

Дальнейший анализ позволил установить, что высокая информационная нагрузка на диспетчерский персонал горно-транспортного комплекса карьера оказывает существенное влияние на простои основного технологического оборудования и транспорта. Простои по указанной причине для карьеров Южного и Новокриворожского горно-обогатительных комбинатов составляют 25—30 % всех простоев технологических агрегатов и локомотивосоставов, а суммарные потери по этой причине составляют более 1 млн. руб. в год для каждого комбината.

Полученные результаты привели к выводу о необходимости автоматизировать основные функции и задачи диспетчерского управления карьерным транспортом, а также построить имитационную модель, позволяющую наметить круг наиболее эффективных задач, подлежащих автоматизации, и выбрать для них критерии оптимальности. Полученные в результате статистических исследований законы распределения случайных величин, в том числе длительности погрузки, разгрузки и пробега вагонов, интервалы поступления транспортных средств на погрузку и разгрузку и др., были непосредственно использованы в имитационной модели [110].

Во втором примере при разработке авторами диалоговой системы планирования поставок сырья для агломерационного производства в черной металлургии УССР был выполнен экономико-статистический анализ влияния технико-экономических показателей работы аглофабрик и транспорта на объем производства агломерата в отрасли [112]. На основании статистических данных для каждой аглофабрики были построены линейные регрессионные зависимости, которые в общем виде записываются следующим образом:

$$\hat{y}_i = \sum_{j=1}^m a_{ij} x_j, \quad i = \overline{1, n},$$

где \hat{y}_i — объем производства агломерата i -й аглофабрики; x_j — транспортно-технологические и экологические факторы, влияющие на объем производства агломерата; a_{ij} — параметры регрессии, характеризующие степень влияния факторов.

Был выполнен ряд стандартных процедур, позволивших установить значения параметров регрессии, а также определить тесноту связи между объемами производства и выбранными факторами. При этом были вычислены коэффициенты детерминации, на основе которых проверены гипотезы о значимости регрессионных зависимостей с помощью F -статистики.

На данном этапе анализа возникла проблема выделения факторов, значительно влияющих на функцию регрессии. В последующем это позволило сократить размерность имитационной модели путем исключения из нее второстепенных факторов. Для построения «наилучшего» подмножества факторов был использован алгоритм формально-логического поиска, который подробно изложен в работе [109].

На основании выполненных исследований построены регрессионные зависимости между текущими простоями технологических агрегатов, химическим составом исходного сырья, температурными режима-

ми процесса агломерации, транспортными факторами, с одной стороны, и объемом производства агломерата — с другой. Показано, что существует довольно тесная корреляционная связь между перечисленными показателями, что предопределило необходимость включения их в имитационную модель планирования производства агломерата.

Результаты статистических исследований показали целесообразность стыковки детерминированной балансовой модели распределения и поставок сырья на производство агломерата и экономико-статистической модели планирования производства, что значительно повысило адекватность имитационных моделей реальному процессу. Имитационные эксперименты с построенным комплексом моделей позволили значительно повысить точность и эффективность плановых решений ЛПР. Следует отметить, что в данной разработке успешно решена проблема стыковки имитационной экономико-статистической модели и балансовой оптимизационной модели.

По результатам статистических исследований объекта можно сделать ряд принципиальных выводов о путях его совершенствования и необходимости автоматизации управления, а также сформулировать задачу имитационного моделирования.

Формулировка задачи (проблемы) является первым этапом построения имитационной модели. Данный этап тесно переплетается с этапом изучения моделируемой системы. Результаты экономико-статистического анализа становятся основой для формулировки замысла имитационных экспериментов и определения предположительной или так называемой концептуальной [103] модели. Для исследователя это этап «обдумывания и планирования», который вкратце можно охарактеризовать как процедуру ясного и точного изложения целей имитационного моделирования.

Цели формулируются обычно либо в виде вопросов, на которые надо ответить, либо гипотез, которые надо проверить, либо действий, которые надо оценить.

Если имитационный эксперимент преследует цель получить ответы на конкретные вопросы, то очевидно, что в самом начале имитации после обследования соответствующей системы необходимо детально и точно сформулировать указанные вопросы. Машинную имитацию целесообразно проводить, отвечая, например, на следующие вопросы экономического или управленческого характера: «Как изменится объем производства продукции в отрасли, если часть капитальных вложений направить не на строительство новых производственных мощностей, а на реконструкцию старых?»; «Как влияет правильность принимаемых решений в конкретной процедуре оперативного управления транспортными перевозками на конечные экономические показатели работы производственного участка, цеха, предприятия?».

Целью имитационного моделирования может быть также проверка одной или нескольких гипотез относительно поведения сложной производственно-транспортной системы. Например, может потребоваться проверка, существенно или не существенно влияет изменение уровня тех или иных факторов на снижение простоев оборудования, транспорта, или уменьшение издержек производства. Может возник-

нужно необходимость проверки того, что выбранная стратегия управления способствует увеличению прибыли предприятия или отрасли и т. п.

Имитационное моделирование может быть использовано как метод решения конкретных задач планово-управленческого характера. В этом случае имитационные эксперименты позволяют оценить количественно и качественно управленческие воздействия на ход производственно-экономического процесса. Например, с помощью имитации решается задача эффективного распределения прибыли предприятия на нужды развития производства и мероприятия бытового и социально-культурного строительства. В этом случае машинные эксперименты позволяют оценить воздействия выбранной системы самофинансирования на конечные экономические показатели планового периода.

В ходе обсуждения концептуальной модели и проблемных вопросов может оказаться, что информации, полученной на стадии исследования системы, недостаточно для четкой формулировки проблемы. В таком случае исследователю целесообразно обращаться за консультациями к ЛПР и специалистам-экспертам по вопросам функционирования данной системы.

С помощью специалистов выдвигаются гипотезы для заполнения пробелов в понимании задачи и принимаются необходимые предположения и допущения. Предположения принимаются, если некоторые данные неизвестны и их практически невозможно получить. С другой стороны, предположения нужны для упрощения модели в случае ее неприемлемой громоздкости или сокращения размерности задачи относительно известных данных. Предположения также дают возможность преобразовать сложные, не поддающиеся учету характеристики в величины, с которыми удобно оперировать в процессе имитационных экспериментов [108]. Особенно часто приходится выдвигать предположения и принимать допущения при наличии в модели качественных переменных, не поддающихся количественному описанию. Так, в системах диспетчерского управления карьерным транспортом принималось допущение о снижении качества управленческих решений при увеличении информационной нагрузки на диспетчерский персонал. Как показано выше, это допущение подтверждается анализом косвенных к качеству решений данных (простой агрегатов и транспортных средств, экономические потери карьера и комбината).

5.4. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Методы имитационного моделирования, как и методы исследования операций, основываются на математическом описании тех или иных фактов, процессов, явлений, происходящих в реальной системе. Совокупность абстрактных (формализованных) описаний отдельных сторон функционирования производственно-транспортной системы является ее математической моделью. Построение математических моделей является основой всего системного анализа. По словам академика Н. Н. Моисеева, это центральный этап исследования или проектирования любой системы [103]. Разработка математических моделей — все-

гда процедура неформальная и, конечно, она очень зависит от исследователя, его опыта и таланта. Как отмечает академик А. А. Дородницын, построение математических моделей экономических систем — пока еще искусство [113].

Исследование математических моделей можно проводить различными методами, а для моделей производственно-экономических систем наибольшее распространение получили методы исследования операций. Однако эти методы входят в противоречие с необходимостью более адекватного описания реальных систем, поскольку такое описание делает математические модели весьма сложными и громоздкими. При этом практически невозможно использовать классические методы математического программирования, особенно если возникает необходимость учитывать стохастический характер поведения систем и воздействий внешней среды. Таким образом, использование имитационного моделирования практически не ставит ограничений на характер и структуру математического описания, что позволяет достичь максимальной адекватности математической имитационной модели реальной системе.

Своеобразие и оригинальность математических моделей конкретных производственно-экономических и транспортных систем, однако, не отрицает, что им присущи некоторые общие черты. Общность моделей прежде всего относится к структурным элементам, из которых они строятся. Структура модели в математическом выражении может быть записана в общем виде так

$$Z = F(Y, X, A), \quad (5.4.1)$$

где Z — выходные переменные системы, т. е. конечные результаты ее действия: производственные, экономические, социальные; Y — вектор зависимых или управляемых переменных; X — вектор независимых или управляющих переменных; A — вектор параметров системы; F — функциональные зависимости, связывающие переменные и параметры системы.

Из (5.4.1) можно выделить структурные элементы, присущие практически всем математическим моделям (переменные, параметры и функциональные зависимости).

При проведении имитационных экспериментов переменные математической модели суть величины, принимающие либо случайный характер, либо детерминированный и связаны между собой определенным видом функциональной зависимости. Переменные, как указывалось ранее, могут быть независимыми или управляющими, а также зависимыми или управляемыми.

В отличие от переменных параметры остаются постоянными в процессе эксперимента и могут лишь меняться экспериментатором при переходе от одной серии экспериментов к другой.

В математических моделях производственно-транспортных систем переменными являются объемы поставок сырья и материалов, текущие простои транспортных средств, запасы, себестоимости транспортно-технологических работ и т. д. В качестве параметров чаще всего используются различные нормы и нормативы, протяженность

транспортных коммуникаций, плановые задания по основным показателям транспортировки и производства и т. п.

В литературе по имитационному моделированию часто различают эндо- и экзогенные переменные. К первым относятся выходные переменные математической модели — Z , а входные переменные X и Y являются, таким образом, экзогенными, или, как иногда их называют, переменными состояния системы.

Учитывая принятую терминологию, цель имитационных экспериментов можно сформулировать как исследование поведения эндогенных переменных в зависимости от принятых значений параметров и изменения экзогенных переменных.

Построение математической модели производственно-экономической системы начинается с выбора необходимого круга независимых переменных. При выборе независимых переменных, участвующих в эксперименте и воздействующих на управляемые переменные, возникает ряд трудностей. Если таких переменных слишком мало, то модель может оказаться неадекватной реальности, если их слишком много, то из-за недостаточного объема памяти ЭВМ или сложности вычислительных процедур имитация может оказаться трудноразрешимой.

Выбор выходных (эндогенных) переменных модели, как правило, не составляет никаких затруднений, так как они определяются уже в процессе формулировки целей исследования. Например, если цель исследований формулируется так: «Существенно ли влияет изменение тех или иных факторов на снижение простоев оборудования и издержек производства?». То выходными переменными математической модели будут простои оборудования и издержки производства. Количество входных переменных сказывается на объеме вычислений в процессе экспериментов. Одна дополнительная входная переменная увеличивает объем вычислений практически вдвое.

Функциональные зависимости математической модели выражают математически соотношения между параметрами и переменными исследуемой системы. Эти зависимости по своей природе являются либо детерминированными, либо стохастическими. Детерминированными зависимостями описываются процессы, в которых конечные показатели однозначно определяются параметрами и входными переменными системы. Такие зависимости выражаются в виде линейных, нелинейных, дифференциальных и интегральных уравнений.

Стохастическими зависимостями описываются процессы, в которых входные переменные принимают вероятностный характер, что весьма характерно для транспортных процессов крупных промышленных предприятий. Выходные показатели транспортных систем обычно имеют неопределенный результат. Указанные функциональные зависимости чаще всего представляются в виде регрессионных уравнений, построенных на основе вероятностных гипотез методами статистического анализа.

В математических моделях производственно-экономических систем, в том числе и транспортных, функциональные зависимости разделяют на целевые функции и ограничения. Структурно и математи-

чески эти зависимости могут не отличаться друг от друга, но в процессе исследований они несут в себе разное производственно-экономическое содержание.

Целевые функции или критерии — это точные математические описания целей функционирования систем и заранее сформулированных проблем имитационных экспериментов. В процессе экспериментов мы обычно стремимся к достижению экстремальных численных значений целевых функций, например, к достижению максимума прибыли, минимума производственных и транспортных затрат и т. д. Нетрудно заметить, что в целевых функциях математически описываются зависимости между, с одной стороны, выходными (эндогенными) переменными и — с другой — параметрами и переменными состояния системы. В общем виде целевая функция может быть записана в следующем виде:

$$Z = f(c_k, x_i, y_j) \rightarrow \min (\max). \quad (5.4.2)$$

В качестве ограничений модели используются функциональные зависимости, связывающие управляемые переменные с параметрами и управляющими переменными, например:

$$y_j \leq f_j(a_i, x_i); \quad j \in M. \quad (5.4.3)$$

Здесь Z , y_j — управляемые переменные; x_i — управляющие переменные; c_k , a_i — параметры модели; M — множество всех функциональных соотношений.

Функциональные соотношения (5.4.3) устанавливают пределы изменения управляемых переменных, т. е. являются условиями, ограничивающими распределение тех или иных средств и ресурсов, заданных нагрузок, плановых заданий и т. п. Если зависимости (5.4.2) — (5.4.3) детерминированы и достаточно просты, то пространственная математическая модель может исследоваться известными методами математического программирования.

Примером такой модели может служить рассмотренная далее производственно-транспортная модель планирования поставок сырья на производство чугуна. Поскольку указанная модель недостаточно адекватно описывает систему управления потоками железорудного сырья на металлургическом предприятии, в модели были реализованы дополнительно отдельные компоненты стохастической системы управления запасами, и дальнейшее ее исследование проводилось методами имитационного моделирования.

Здесь следует ввести понятие компонентов, или подсистем, часто встречающееся в определении элементов структуры математической модели. Понятие компонентов (подсистем) взято из теории сложных систем, где под ними понимаются составные части, которые при соответствующем объединении образуют систему. Система определяется как совокупность объектов или процессов, функционально и информационно связанных между собой и объединенных некоторой формой регулярного взаимодействия для выполнения заданной функции.

Построение математической модели сложной производственно-транспортной системы, охватывающей математическими соотношениями

все аспекты ее функционирования, часто оказывается практически невозможным. В этих случаях по некоторым принципам производится декомпозиция исследуемого объекта на ряд подсистем при сохранении функциональных и информационных связей между ними.

Не вдаваясь подробно в основные принципы декомпозиции и последующего агрегирования систем, отметим, что построение математических моделей для выделенных подсистем значительно упрощается. Указанные модели являются компонентами математической модели сложной системы, и вся проблема заключается лишь в том, как построить схему взаимодействия между компонентами.

Задача разделения математической модели на компоненты и организации их взаимодействия является далеко не тривиальной. Так, в моделях, относящихся к классу производственно-транспортных, можно выделить два относительно независимых блока ограничений.

В первом блоке описываются производственные возможности исследуемых систем. При этом систему разбивают на отдельные «элементарные» производственные единицы. После этого с помощью производственных функций различных типов описываются, во-первых, производственные возможности каждой из единиц, во-вторых, возможности обмена ресурсами и продукцией между ними. Очевидно, что такие описания являются компонентами математической модели. При описании взаимодействия между компонентами модели главную роль играют балансовые соотношения, которые связывают между собой, с одной стороны, производственные задания (госзаказы), плановые экономические показатели — прибыль и нормы ее распределения, а с другой — производственные мощности, потребности в сырье, материалах, энергетических и других ресурсах. Очевидно, что технологические и производственные ограничения сами по себе не определяют развитие экономического процесса, и существует огромное количество вариантов выполнения плановых заданий, укладываемых в данные ограничения.

Второй блок является непосредственно транспортным и отражает связи поставщиков с потребителями в самом широком их понимании. Здесь описываются балансы поставок и потребления грузов с выделенными для этого транспортными средствами, пропускными способностями транспортных сетей и транспортными издержками.

В математических моделях выделяют специальные выходные переменные, по значениям которых определяют оптимальный или близкий к оптимальному вариант функционирования производственно-транспортного процесса. Эти переменные называют обычно управляющими воздействиями или управлениями. В имитационной модели на уровне производственно-экономических процессов работает механизм выбора лучших из множества допустимых управляющих воздействий. Следует подчеркнуть, что на этом уровне возникают самые серьезные трудности в решении задачи координации функционирования отдельных компонентов математической модели. Вопросы определения допустимых управляющих воздействий и соответственно выбора наилучших из них тесно переплетаются с проблемами выбора целевых функций и их оптимизацией в процессе имитационных экспериментов.

Отметим, что проблема сочетания имитационных моделей и известных методов оптимального программирования является одной из наиболее актуальных и наименее исследованных в теории имитационного моделирования. Имеются частные примеры применения линейного программирования при имитации распределения капитальных вложений [100], попытки использовать оптимизационные методы в имитации транспортных процессов на отраслевом уровне [99]. Известны различные подходы к решению задач оптимального проектирования транспортных коммуникаций, определения потребностей в подвижном составе и т. п. Некоторые примеры сочетания оптимизационных и имитационных методов, разработанные авторами для решения производственно-транспортных задач, будут приведены далее.

5.5. РАЗРАБОТКА МОДЕЛИРУЮЩИХ АЛГОРИТМОВ

При разработке моделирующих алгоритмов возникает несколько специфических проблем, связанных с особенностями имитационных экспериментов производственно-экономических систем. В моделирующих алгоритмах практически для всех имитационных моделей можно выделить четыре необходимые процедуры: 1) организующая ввод исходных данных; 2) организующая вычислительный процесс; 3) генерирующая недостающие случайные переменные; 4) регламентирующая последовательность выполнения событий и процессов в системе.

Процедуры, реализующие ввод исходных данных и организацию вычислительного процесса, имеют место практически во всех задачах экономического характера, решаемых с помощью ЭВМ. Особенности их алгоритмов практически полностью определяются построенной ранее математической моделью, и их разработка в большинстве случаев принципиальных трудностей не встречает. Техническая сторона алгоритмизации указанных процедур, освещена в [98, 104]. Разработка алгоритмов процедур ввода исходных данных и организации вычислительного процесса тесно переплетается с разработкой машинных программ, в частности с вопросами выбора языка программирования. Эти вопросы мы подробно рассмотрим в следующем параграфе.

Процедура генерации недостающих случайных переменных — это особенность, характерная для алгоритмов имитационного моделирования. Рассмотрим кратко некоторые методы генерирования последовательности случайных чисел. В математическом обеспечении современных ЭВМ имеются ППП, реализующие получение псевдослучайных чисел для наиболее распространенных распределений. В качестве исходной совокупности случайных чисел, необходимых для генерации любого закона распределения, наиболее часто используется совокупность равномерно распределенных в интервале $[0,1]$ случайных чисел. Для данной совокупности функция плотности распределения случайной величины имеет вид

$$F(x) = \begin{cases} 0, & \text{если } x \leq 0; \\ x, & \text{если } 0 < x < 1; \\ 1, & \text{если } x \geq 1, \end{cases}$$

а математическое ожидание и среднее квадратическое отклонение соответственно равны: $M(\beta) = 1/2$; $\sigma(\beta) = 1/2\sqrt{3}$.

Преимуществом совокупности равномерно распределенных в интервале $[0,1]$ чисел является то, что для ее построения не требуется больших затрат машинного времени и ее можно достаточно просто и удобно преобразовать в последовательности случайных чисел различных законов распределения.

Равномерно распределенные числа в современных ЭВМ наиболее часто генерируются с помощью датчиков псевдослучайных чисел, построенных на основе рекуррентной формулы $\beta_{i+1} = f(\beta_i)$ при заданном β_0 . Формулы эти достаточно просты и определяются разрядной сеткой ЭВМ, для которой разрабатывается датчик. Основным недостатком таких датчиков является ограниченность количества реализаций случайных чисел, поскольку рано или поздно будет получено число, встречающееся в последовательности.

Совокупность равномерно распределенных случайных чисел позволяет, как отмечалось выше, построить последовательность случайных чисел, распределенных по различным другим законам. Например, с помощью простой формулы, построенной на основе центральной предельной теоремы, реализуется последовательность чисел, распределенных по закону, близкому к нормальному:

$$x_j = \left(\sum_{i=12(j-1)+1}^{12j-1+12} \beta_i - 6,0 \right) \sigma_x + \bar{x},$$

где \bar{x} и σ_x — математическое ожидание и среднеквадратическое отклонение нормального распределения.

Второй метод генерации случайных чисел как равномерно распределенных, так и по другим законам распределения заключается в предварительном построении таблиц случайных чисел. Такие таблицы, обычно большого объема, необходимо заранее составить, записать в память ЭВМ, а затем использовать в процессе имитации. Этот метод приобретает распространение с ростом мощности современных ЭВМ, и в частности с ростом объема оперативной и долговременной памяти, а также с увеличением скорости обращения к внешним магнитным накопителям, на которых записана таблица.

Наиболее важной и ответственной процедурой, реализуемой в алгоритме любой имитационной модели, является процедура регламентации последовательности возникновения событий и процессов и выполнения соответствующих им вычислений. Для каждой конкретной модели разработка указанной процедуры обычно требует индивидуального подхода.

Поскольку большинство имитационных моделей производственно-экономических систем призваны исследовать их поведение на некотором отрезке времени, то регламентация последовательности событий и процессов определяется тем, как организовано системное время в модели.

Остановимся подробнее на некоторых принципах построения механизма системного времени.

Более просто регламентацию событий можно организовать в моделях вариантного планирования, в которых под событием понимается начало расчета очередного варианта, а конец экспериментов определяется расчетом заранее заданного количества вариантов. Аналогично регламентируются события в простых системах массового обслуживания, где предусматривается обработка заявок (событий) в той последовательности, в которой они поступили на вход системы. В обоих случаях требуется минимум сведений о предыдущем событии и нет необходимости в какой-то особой координации во времени между отдельными блоками системы. Это вызвано тем, что в указанных моделях практически исключено одновременное возникновение событий, а модели вариантного планирования вообще не связаны с реальным временем.

Принцип организации системного времени для данного класса моделей по аналогии с [114] назовем «принципом последовательной проводки заявок (вариантов)».

Для моделей сложных производственно-экономических систем, связанных с реальным временем, регламентация возникновения событий и процессов включает два аспекта: корректировку временной координаты в процессе имитации работы системы и обеспечение согласованности во времени действий различных блоков и подсистем.

В процессе функционирования реальных систем различные внешние воздействия, а также события внутри подсистем возникают спорадически, и имеется вероятность одновременного их появления. Кроме того, процессы и действия различных подсистем взаимосвязаны, т. е. зависят от состояния других блоков системы, внешних воздействий и предыстории процесса, поэтому они должны быть скоординированы во времени и синхронизированы. Задача синхронизации значительно упрощается тем, что отдельные блоки и компоненты имитационной модели функционируют последовательно, поскольку принцип действия ЭВМ (выполнения отдельных команд) последовательный.

Предположим, что мы по какому-либо принципу скорректировали временную координату, т. е. перешли от реального и соответственно системного времени t_i к моменту времени t_{i+1} . Учитывая состояние отдельных компонентов в предыдущий момент времени t_i , а также события и возмущения (случайные и детерминированные), возникшие на отрезке времени между t_i и t_{i+1} , мы последовательно рассчитываем новое состояние каждого блока и элемента модели в очередной точке временной координаты t_{i+1} . Имитация заканчивается в момент достижения очередной точкой временной координаты величины заранее намеченного горизонта времени T .

Остановимся подробнее на принципах корректировки временной координаты. Обычно различают два основных принципа задания системного времени: с помощью фиксированных интервалов времени и переменных, длительность которых определяется интервалом между возникновением двух следующих друг за другом событий. Эти принципы обычно называются «принцип Δt » и «принцип особых состояний». По принципу Δt каждое новое состояние системы рассчитывается в момент времени $t_i + \Delta t$, а по принципу особых состояний ситуация

в моделируемой системе обновляется в момент времени возникновения очередного существенного события.

Имитационные модели, использующие реальное время, подразделяют на две категории [98]: с дискретным и непрерывным временем изменения состояния.

Примерами моделей производственно-экономических систем с непрерывным изменением состояния могут служить модели работы предприятий химической промышленности, некоторых производственных процессов и черной металлургии, например доменного и сталеплавильного. В таких моделях в качестве механизма системного времени целесообразно использовать принцип Δt , который иногда используется в имитационных моделях, исследующих дискретные системы с большими потоками информации или значительным количеством случайных возмущений. Однако в большинстве моделей с дискретным изменением состояния чаще используется принцип особых состояний. Примерами таких моделей служат различные модели управления строительным производством, многостаночным цехом машиностроительного производства, а также модели управления промышленным транспортом.

Основные проблемы, решаемые в моделях с регламентацией событий по принципу Δt , — это определение длительности фиксированного интервала. В указанных моделях, особенно для дискретных систем, численная обработка процессов и состояний производится по пакетам или множествам событий, произошедших за период от t_i до $t_i + \Delta t$. С одной стороны, этот интервал должен быть как можно меньше, чтобы исключить вероятность попадания в него нескольких существенных событий и, таким образом, избежать ложного представления об одновременности их появления. С другой стороны, уменьшение Δt приводит к существенному увеличению машинного времени эксперимента, так как значительная его часть затрачивается на отслеживание и корректировку временных интервалов.

Определяется обычно интервал Δt с учетом конкретных особенностей исследуемой системы. Для систем с непрерывным изменением состояния этот интервал выбирается как можно длиннее, но таким образом, чтобы аппроксимация проинтегрированных функций — характеристик исследуемых непрерывных процессов дала результаты, близкие к реальным.

Для дискретных систем принцип Δt целесообразно применять в случае, когда события регулярны и распределены во времени относительно равномерно. При этом величину Δt можно определять как среднее значение интервалов между соседними событиями. Если события приходятся на конец интервала, целесообразно строить автоматные имитационные модели.

Модели, организованные по принципу особых состояний, имеют преимущество в том, что нет необходимости определять величину приращения времени, что влияет и на продолжительность цикла моделирования, и на точность конечных результатов. Этот принцип наиболее эффективен в моделях дискретных систем, в которых события могут длительное время не наступать, а также при их неравномерном распределении во времени и большой продолжительности.

В имитационных моделях сложных производственно-экономических систем может случиться так, что алгоритмы регламентации событий целесообразно строить на комбинации обоих принципов. Предположим, что общая структура модели строится на принципе особых состояний, а между существенными событиями возникает большое количество второстепенных (например, в систему поступает большое количество сообщений с контрольно-измерительных приборов). В этом случае поведение системы между существенными событиями можно имитировать с использованием принципа Δt . В качестве примера построения математической модели и моделируемого алгоритма рассмотрим имитационную модель погрузочно-транспортного угла, который представляется как одноканальная система массового обслуживания. Предметом теории массового обслуживания является разработка вероятностно-статистических методов решения задач обслуживания большого числа однородных объектов, которые в теории называются либо заявками, либо клиентами, либо требованиями, а станция обслуживания обычно именуется каналом.

В нашем примере заявками будут служить транспортные средства, последовательно поступающие под погрузку или разгрузку, а каналом, таким образом, является погрузочно-разгрузочный агрегат. Поскольку такая система, именуемая одноканальной системой массового обслуживания, достаточно подробно исследована вероятностно-статистическими методами, мы остановимся на этапе формулировки целей ее исследования методами имитационного моделирования. Предположим, что нам известны экономические потери от простоя транспорта в ожидании обслуживания, а также потери от простоев погрузочно-разгрузочного агрегата в ожидании транспорта. Тогда цель исследования можно сформулировать так: определить оптимальную интенсивность поступления транспортных средств на обслуживание, обеспечивающую минимум суммарных потерь от простоев.

Для построения математической модели и моделирующего алгоритма введем некоторые обозначения. Пусть t_i — время обслуживания каналом i -й заявки. В общем случае t_i — случайная величина, функция распределения которой $F(t)$. Промежуток времени между появлением двух последовательных i -й и $(i + 1)$ -й заявок обозначим τ_i , а функцию распределения случайной величины — $F(\tau)$. Будем считать t_i и τ_i входными (управляющими) переменными.

В модели выходными (исследуемыми) переменными служат потери от ожидания заявками обслуживания и потери от простоев канала в ожидании заявок. Суммарные потери в системе при обслуживании заявок можно вычислить по формуле

$$Z = c_1 \sum_{i=1}^n \Delta t_i + c_2 \sum_{i=1}^n \Delta \tau_i, \quad (5.5.1)$$

где c_1 , c_2 — потери в единицу времени из-за ожидания заявками обслуживания и простоев канала соответственно; $\Delta \tau_i$ — длительность ожидания обслуживания i -й заявки; Δt_i — длительность простоя канала в ожидании поступления i -й заявки.

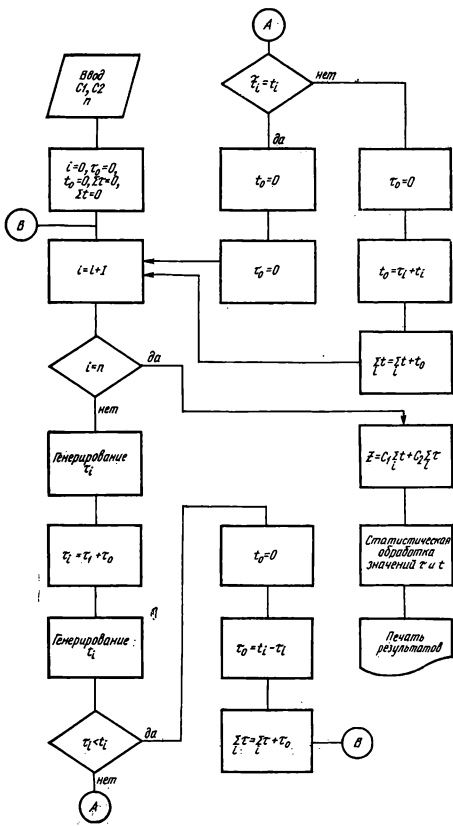


Рис. 5.5.1

На рис. 5.5.1 показана блок-схема алгоритма имитационной модели, использующего для расчетов потерь в системе массового обслуживания функциональные соотношения (5.5.1).

Представленная блок-схема алгоритма достаточно проста и наглядна, поэтому описывать подробно ее не будем. Отметим только, что в модели не предусмотрено специально организованное системное время. Очередным «особым состоянием» служит поступление на обслуживание очередной транспортной единицы, и алгоритм заканчивает работу, если обслужено заранее заданное количество n единиц, т. е. регламентация событий в модели организована по принципу «последовательной проводки заявок».

В результате имитационных экспериментов для реальных систем можно получить такие важные для практики результаты, как эффективная стратегия построения очереди поступления на погрузочно-разгрузочные работы и возможные направления совершенствования работы канала обслуживания с целью снижения потерь от простоев. С описанной моделью был проведен ряд численных экспериментов, для которых использованы реальные данные погрузочно-транспортного узла на железорудном карьере, состоящего из добычного экскаватора и автосамосвалов, поступающих на погрузку. Результаты экспериментов подробно описаны в [109].

5.6. ПОНЯТИЕ О ЯЗЫКАХ МОДЕЛИРОВАНИЯ И СОСТАВЛЕНИИ ИМИТАЦИОННЫХ ПРОГРАММ

Современное применение вычислительной техники требует не только наличия быстродействующих ЭВМ разного класса мощности, а также различной проблемной и практической ориентации, но и повышенных возможностей ускоренного составления машинных программ. Необходимость «быстрого» программирования существует практически во всех отраслях, где используется вычислительная техника: инженерных расчетах, системах обработки данных, автоматизированных системах управления производственными и технологическими процессами, автоматизации научных экспериментов, проектирования и т. д.

Для программирования задач на ЭВМ первых поколений использовались в основном так называемые машинно-ориентированные языки. Основу данных языков составляли коды, отражающие физическую структуру конкретных ЭВМ. Программирование в кодах ЭВМ — весьма трудоемкий процесс, требующий высоких профессиональных знаний программиста, и поэтому круг задач, решаемых на ЭВМ, был ограничен инженерно-физическими и математическими расчетами.

С развитием вычислительной техники и расширением сферы ее применения на смену машинно-ориентированным языкам кодирования программ пришли алгоритмические языки.

Алгоритмические языки условно разделяются на уровни, которые определяют степень их приближения к языкам естественного человеческого (профессионального) общения. Чем ниже уровень языка, тем ближе язык к машинным кодам, и наоборот, чем выше уровень, тем ближе алгоритмический язык к естественному.

Алгоритмические языки высокого уровня, в свою очередь, условно разделяются на универсальные и специализированные. Основной принцип реализации универсального алгоритмического языка — это возможность записи любого вычислительного процесса независимо от конкретной машины. Такой язык должен быть наглядным и однозначным, т. е. запись любого алгоритма, сделанная с соблюдением всех правил языка, должна читаться единственным способом и не допускать различных толкований.

Среди универсальных языков наибольшее распространение получили языки PL-1, ФОРТРАН, АЛГОЛ, КОБОЛ, БЕЙСИК и др.

Универсальность указанных языков позволяет записать программы для решения широкого круга профессиональных задач, в том числе для программирования задач имитационного моделирования по соответствующим алгоритмам. В качестве примера приведем программу на языке БЕЙСИК, составленную для персональной ЭВМ по описанному в предыдущем параграфе алгоритму моделирования работы погрузочно-транспортного узла (одноканальной системы массового обслуживания).

```
10 DIM T1 (100), T2 (100)
20 INPUT «КОЛИЧЕСТВО ТРАНСПОРТНЫХ СРЕДСТВ», N
30 INPUT «СТОИМОСТИ ПРОСТОЕВ ПОГРУЗОЧНЫХ И
   ТРАНСПОРТНЫХ СРЕДСТВ», C1, C2
40 FOR J = 1 TO N
50 GOSUB' (K) (T1 (J))
60 T1 (J) = T1 (J) + T0
70 GOSUB' (K1) (T2 (J))
80 IF T1 (J) >= T2 (J) THEN 110
90 T3 = 0 : T0 = T1 (J) + T2 (J) : Z1 = Z1 + T0
100 GOTO 150
110 IF T1 (J) = T2 (J) THEN 140
120 T0 = 0 : T3 = T1 (J) + T2 (J) : Z2 = Z2 + T3
130 GOTO 150
140 T0 = 0 : T3 = 0
150 NEXT J
160 PRINT «РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ», ...
170 END
```

Структура приведенной программы полностью совпадает с блок-схемой алгоритма моделирования.

Оператор *GOSUB* осуществляет обращение к подпрограммам с номерами $\langle K \rangle$ и $\langle K1 \rangle$, генерирующим вероятностный характер интервала поступления J -го транспортного средства ($T1 (J)$) и длительности его погрузки ($T2 (J)$).

В программе опущено не имеющее принципиального значения описание таблицы вывода результатов моделирования.

Универсальные языки фактически позволяют составить программу для решения любой проблемы из всех областей профессиональных знаний. Как видно из приведенного примера, они имеют много условностей, существенно загромождающих конкретную программу. Кроме

того, от составителя программы, наряду с глубокими профессиональными познаниями в решении поставленной проблемы, требуются знания языка программирования. По этим причинам создаются значительно более простые специализированные языки, которые по конструкции гораздо ближе к естественному языку профессионального общения.

Специализированные языки — это алгоритмические языки высокого уровня, имеющие определенную проблемную ориентацию и в которых широко используются профессиональные термины в качестве ключевых слов.

Проникновение имитационного моделирования во многие области научных исследований и технических приложений ЭВМ дало толчок к созданию различных специализированных языков имитационного моделирования. Практика применения специализированных языков показывает, что для многих проблем указанные языки позволяют создать более компактные программы. Компактность, а также использование профессиональной терминологии специалистов являются основными по исследуемой проблеме преимуществами специализированных языков. В операторы специализированных языков встроено максимально возможное число логических программ, необходимых для моделирования различных аспектов изучаемого явления. Ключевые слова операторов несут смысловую нагрузку, однозначно понимаемую исследователем (пользователем модели). Системы семантики языков соответствуют широко распространенному способу мышления специалистов определенных профессий.

Таким образом, целесообразность специализированных языков определяется, в основном, двумя признаками-причинами. Первое — это удобство программирования, и второе — проблемная ориентация. Каждая из этих причин может существенно сказаться на эффективности имитационных экспериментов и играет важную роль при составлении программ. Известный опыт имитационного моделирования [115] показывает, что многие хорошо спланированные эксперименты не удалось осуществить или хотя бы закончить вовремя из-за трудностей программирования. С другой стороны, известно много нереализованных проектов из-за просчетов, допущенных на стадии разработки имитационных моделей, усложнивших программирование настолько, что эксперимент стал практически неосуществим.

Специализированные языки во многом определили успех имитационных экспериментов с моделями конкретных систем, и два выделенных выше обстоятельства сыграли при этом решающую роль. Но следует подчеркнуть, что все языки моделирования, независимо от их концептуальной (проблемной) направленности, разработаны на базе основных понятий имитации.

Прежде всего каждый из специализированных языков имеет широкие возможности в генерировании случайных чисел и фактически обладает собственной библиотекой случайных функций. Основу такой библиотеки обычно составляет генератор последовательности псевдослучайных чисел, равномерно распределенных в интервале от 0 до 1. Датчик случайных чисел — обязательный элемент любого языка моделирования [106].

Необходимость генерировать числа в соответствие с некоторым распределением вероятностей связана со случайным характером многих изучаемых процессов.

Второй обязательный элемент специализированных языков — воспроизведение динамических свойств исследуемого процесса. Ранее мы показали, что основу любого алгоритма имитации поведения системы составляет блок ведения системного времени. Поэтому практически во все специализированные языки имитационного моделирования введено понятие часы (*time*). Данное понятие находит обязательное отражение в структурах имитационных программ производственно-экономических и транспортных систем.

Поскольку время, или, точнее, его представление в численном эксперименте, составляет сущность имитационного подхода, то необходимо в первую очередь правильно отобразить в ЭВМ ход реального времени. Можно утверждать, что организован имитационный эксперимент, если удалось адекватно связать выполнение программы, описывающей динамику поведения реальной системы с имитируемым системным временем.

В настоящее время насчитывается более 200 специализированных алгоритмических языков моделирования различной проблемной ориентации [115, 116].

В основу систематизации языков моделирования положены некоторые понятия и определения, используемые для описания моделируемых динамических систем.

Во-первых, динамические системы, в отличие от статических, — это системы, которые изменяются во времени. Такое определение необходимо дополнить некоторыми условиями, характерными для методов имитационного моделирования и языков моделирования. Динамические системы состоят обычно из статического каркаса или *базы*, включающих некоторые множества элементов, из которых атрибуты при экспериментах, направленных на моделирование определенной проблемы, предполагаются неизменными. Например, при моделировании перемещения транспортных средств на транспортной сети базой может служить топология сети, размещения станций, длина перегонов и т. п. Однако при решении такой проблемы, как увеличение пропускной способности сети, некоторые из перечисленных атрибутов перестают быть неизменными, так как решается задача установления их оптимальных значений. Таким образом, статичность базы условна относительно имитационных экспериментов по решению конкретной проблемы или задачи. Неизменные элементы базы имитационной модели в языках моделирования называют обычно *активностями*.

В динамических имитационных системах имеются элементы, атрибуты которых меняются во времени, и в языках моделирования такие элементы называются *транзактами*. Исходя из указанных определений динамические системы и соответственно языки моделирования делятся на два основных типа: *АТ* и *ТА* [115]. В первом случае транзакты в языке играют роль пассивных элементов, во втором роль пассивных элементов отводится активностям. Языки типа *АТ* и *ТА* спроектированы для систем с изменяющейся во времени структурой. Их использование для систем с постоянной структурой является частным

случае. Для имитаций постоянной структуры имеются более простые языки типа *A*, например *COSMO DUANA* [115].

В систематизации языков моделирования так же, как и в описанных ранее моделирующих алгоритмах, важную роль играют принципы реализации системного времени. Согласно указанным принципам языки условно делятся на языки непрерывного моделирования, языки дискретного моделирования и языки комбинированного, или непрерывно-дискретного моделирования. К языкам дискретного моделирования относятся такие, в которых каждый атрибут каждого транзакта в течение его присутствия в системе может изменять свои значения только в конечном числе моментов времени. Языки непрерывного и комбинированного моделирования допускают неограниченную область значений реальных атрибутов.

Следует отметить, что три перечисленных принципа организации системного времени могут использоваться в языках различных типов: *A*, *AT*, *TA*. Наибольший интерес представляют языки, в которых можно реализовать любой из принципов организации системного времени, и, в свою очередь, такие языки наиболее подходят для программирования имитационных моделей сложных производственно-транспортных систем.

С этой точки зрения наибольший интерес представляют языки *GPSS*, *ДИНАМО*, *СИМУЛА*, *СИМУЛЕЙТ* и др. [106].

Язык *GPSS* был создан в основном для дискретного моделирования систем массового обслуживания. Более поздние редакции проектов языка позволяют использовать его также для моделирования непрерывных систем других классов. Язык *GPSS* иногда называют языком транзактов, имея в виду, что его использование наиболее эффективно для имитации пространственного движения объектов при фиксированной структуре модели. Приведем упрощенную запись программы моделирования работы погрузочно-транспортного узла с дискретным изменением системного времени.

* *SIMPLE TRANSPORT SYSTEM*

*

* *BLOCK DEFINITION CARDS*

GENERATE t, Δt

QUEUE 1

SEIZE 1

DEPART 1

ADVANCE τ, Δτ

RELEASE 1

k TABULATE 1

TERMINATE 1

*

* *TABLE DEFINITION CARD*

1 TABLE

*

* *CONTROL CARD*

START N

Программа состоит из заглавия и нескольких блоков, главный из которых — блок описания структуры и имитации движения транзактов. В блоке генерируется поступление транспортных средств на погрузку с интервалом $t \pm \Delta t$, затем следуют команды поставить транспортное средство в очередь 1, занять погрузочную установку 1, сгенерировать длительность погрузки с интервалом $\tau \pm \Delta \tau$, освободить установку 1, запомнить все данные об ожидании и погрузке и уничтожить транзакт.

Второй блок описывает таблицу вывода результатов имитации, и последний блок состоит из команд, управляющих всем процессом имитации работы погрузочно-транспортного узла, пока через него не пройдут N транспортных единиц.

Если сравнить описанную программу с программой, составленной на языке Бейсик, то первая выглядит значительно проще и потому она предпочтительнее. Следует отметить, что программы составлены для одноканальных систем массового обслуживания.

Для многоканальных систем сложность программы на языке Бейсик значительно возрастет, в то время как программа на *GPSS* останется без изменения. Достаточно в операторах *QUEUE*, *SEIZE* и последующих вместо 1 поставить фактическое количество каналов.

Коротко охарактеризуем некоторые другие языки имитации, применимые для моделирования производственно-транспортного и экономических систем. Широко распространены языки СИМУЛЕЙТ и ДИНАМО, которые были разработаны прежде всего для моделирования больших экономических, социальных, транспортных и других систем, описываемых значительным числом уравнений (неравенств) [115]. Для имитации производственно-транспортных систем дискретного типа часто используются языки *GASP*, *SDAM* и др. [107]. Наиболее универсальными считаются языки СИМУЛА-67 и НЕДИС, снабженные всеми необходимыми средствами для введения любых имитационных понятий [115].

Следует отметить, что каждый язык отражает структуру определенных, применяющихся для описания довольно широкого класса систем.

Применение конкретного языка имитации дает исследователю мощный инструмент для формализации исследуемой проблемы, зачастую минуя этапы разработки математической модели и алгоритма имитации. Высокий уровень проблемной ориентации значительно упрощает процесс программирования и анализа результатов моделирования. Однако языки моделирования имеют и ряд недостатков, связанных прежде всего с упрощением (подготовкой) реальных систем под формальные требования и системы абстракций предусмотренных в описании языка. Поэтому не всегда удается создать пригодную имитационную модель, обладающую необходимым уровнем адекватности сложной производственно-транспортной системы.

К недостаткам специализированных языков следует отнести отсутствие гибкости, особенно при исследовании сложных систем, модели которых включают элементы различных классов. Излишняя жесткость специализированных языков, а также отсутствие трансляторов в математическом обеспечении ЭВМ, ограничивает сферу их применения.

Кроме того, за сокращение времени программирования приходится платить увеличением длительности счета на ЭВМ, а для некоторых языков — и увеличением затрат на подготовку данных согласно требованиям ко входным форматам.

Положительный опыт применения специализированных языков показал, что имеется возможность единообразно описывать значительное число задач имитационного моделирования и при этом различать элементы одного класса в разных моделях. Это позволяет с помощью универсальных или машинно-ориентированных языков составлять стандартные программные модули для элементов одного класса и включать их в виде микрокоманд в специализированные языки имитационного моделирования. По такому пути пошли создатели УАИМ [104].

Совокупность программ имитации отдельных элементов, преобразованных к стандартной форме обработки и анализа результатов моделирования, а также стандартных процедур имитации составляют УАИМ, которая может быть оформлена в виде пакета прикладных программ с собственной операционной системой, управляющей имитационным экспериментом. Единообразие математических моделей элементов некоторых классов систем, а также стандартные формы сопряжения элементов и их взаимодействие с внешней средой обеспечивают возможность построения универсальных программных модулей, включаемых в библиотеку стандартных программ УАИМ.

Использование имитационного эксперимента на этапах формулировки проблемы, разработки математической модели и моделирующего алгоритма требует выполнения процедур, связанных с преобразованием элементов модели, а также схем сопряжения к стандартному виду. Наличие пакета прикладных программ по имитации работы элементов модели и схем их сопряжения значительно упрощает этап разработки машинных программ имитационного эксперимента с моделью сложной системы.

В состав пакета прикладных программ наряду с локальной операционной системой включаются программы:

- 1) формирования базы данных об объекте и задаче;
- 2) решения задачи имитации процесса функционирования отдельных элементов системы;
- 3) обработки результатов имитации;
- 4) расширения пакета;
- 5) организации различных режимов работы пакета.

Важным достоинством УАИМ является то, что она может быть включена в состав математического обеспечения АСУ. Сопряжение ее с информационным полем (базами данных) и операционной системой АСУ позволит автоматизировать подготовку исходной информации для моделирования, что значительно повысит оперативность имитационного эксперимента. В этом случае АСУ будет располагать своеобразным полигоном для оценки вариантов плановых решений и проверки качества управляющих воздействий на производственную систему [101].

В целом, УАИМ обладает гибкостью по сравнению со специализированными языками имитационного моделирования и применима к

более широкому классу моделей сложных производственно-транспортных систем. Это подтверждается опытом ее использования для агрегатных систем, т. е. производственных систем, в которых отдельные элементы описываются в виде кусочно-линейных агрегатов. Хорошие перспективы для УАИМ имеются в автоматном моделировании, с элементами и опытом использования которого можно познакомиться в [92, 109].

5.7. ПРОВЕРКА ПРИГОДНОСТИ ИМИТАЦИОННОЙ МОДЕЛИ И ОБРАБОТКА РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТОВ

Особое место среди всех этапов имитационного эксперимента занимает проверка пригодности разработанной модели. Имитационные эксперименты с моделью сложной производственно-экономической системы весьма дорогостоящее исследование, требующее значительных усилий разработчиков, затрат материальных ресурсов и машинного времени. Поэтому, создав имитационную модель, разработчики вполне резонно могут поставить следующие вопросы: Будут ли результаты экспериментов достоверными и убедительными для целей принятия управленческих решений и выбора наилучшей стратегии развития и функционирования системы? Насколько точны предсказания имитационной модели насчет поведения реальной системы? Ответы на эти вопросы призвана дать проверка пригодности имитационной модели.

Не исключено, что в процессе построения модели разработчиками могут быть допущены как принципиальные, так и разного рода технические ошибки.

Обычный путь устранения этих ошибок — многократные визуальные проверки выполненных разработок на всех предшествующих стадиях эксперимента. Такие локальные проверки помогают устранить технические ошибки. Гораздо сложнее выявить и устранить принципиальные ошибки, которые могут существенно повлиять на пригодность имитационной модели.

На стадии обследования системы и статистического анализа такие ошибки являются следствием неверно сформулированных гипотез о законах распределения случайных величин и корреляционных связях между ними, а также неправильно построенных регрессионных зависимостей между переменными системы. Избегать указанных ошибок можно с помощью локальных проверок, многократно повторяя статистический анализ по возможности с большими объемами статистических совокупностей. Такие ошибки выявляются при проверке пригодности модели.

Следствием допущения ошибок на стадии обследования системы могут быть принципиальные ошибки на этапе разработки математической модели. На основании данных экономико-статистического анализа зачастую решаются вопросы определения необходимого круга управляющих переменных, а также вопросы установления функциональных зависимостей, связывающих переменные и параметры математической модели.

Включение в модель тех или иных переменных и функциональных зависимостей между ними имеет два аспекта. С одной стороны, они

должны обеспечить максимальную адекватность математической модели реальному процессу. С другой — чрезмерное расширение круга переменных и соотношений приводит к значительному росту размерности модели и, как следствие, существенно увеличивает затраты на подготовку и проведение имитационного эксперимента. Отсевание второстепенных переменных и зависимостей, которые несущественно сказываются на адекватности модели, обычно производится по интуитивным соображениям разработчиков.

К решению вопросов отсеивания переменных и зависимостей следует подходить осторожно, так как здесь чаще всего встречаются принципиальные ошибки. Очевидно, что если на этом этапе будут проигнорированы важные переменные и функциональные соотношения математической модели, то эти ошибки могут свести на нет все усилия разработчиков по получению достоверных данных имитационного эксперимента.

На этапе разработки моделирующего алгоритма принципиальной ошибкой может быть неправильный выбор механизма регламентации событий и процессов в модели. Такие ошибки в большинстве случаев несущественно сказываются на достоверности результатов имитации и могут лишь значительно увеличить затраты машинного времени на проведение эксперимента.

Ошибки в реализации датчиков случайных чисел на этапе разработки машинных программ принципиальными могут быть лишь в случае неправильно построенных ранее законов распределения случайных величин при экономико-статистическом анализе системы.

Итак, на этапе проверки пригодности модели, если разработчики не смогли выявить возможные технические и принципиальные ошибки, ставится задача доказать отсутствие таких ошибок и подтвердить достоверность результатов, получаемых с помощью имитационных экспериментов. Какими же критериями следует пользоваться при оценке пригодности модели? Очевидно, что модели, основанные на вымышленных данных и чисто гипотетических функциональных соотношениях, не проверенных эмпирически, могут содержать принципиальные ошибки и вряд ли пригодны для практического использования. Единственным критерием пригодности модели является то, что она верна и результаты имитации подтверждаются практикой функционирования системы. Если в результате имитационного эксперимента правильно предсказываются последствия решений, реализуемых на практике, тогда можно вынести окончательное положительное решение о пригодности модели. Следует отметить, что проверка пригодности модели сравнением результатов практической деятельности исследуемых систем с предсказаниями имитационного эксперимента зачастую не представляется возможной. Прежде всего это касается моделей, имитирующих поведение систем на длительный период времени. Ожидание конца такого периода для оценки правильности принятых решений и соответственно пригодности модели практически неприемлемо.

С помощью имитационных экспериментов могут приниматься ответственные решения, особенно в области выработки стратегии развития и функционирования систем. Если нет гарантий адекватности

модели исследуемому объекту, то существует большой риск во внедрении результатов экспериментов в область принятия ответственных стратегических решений.

Наиболее приемлемым, хотя и не окончательным, методом проверки пригодности модели в этом случае является так называемое ретроспективное предсказание, т. е. проведение экспериментов с прошлым системы [106]. В таком эксперименте в качестве исходных параметров и независимых переменных берутся отчетные данные о работе системы за прошедший период времени. Если также имеются данные о воздействиях на систему, имевших место в прошлом, то необходимо проверить, приведут ли указанные воздействия на нашу модель к аналогичным последствиям.

Таким образом, количественно сравнивается поведение (траектория) исследуемых выходных переменных модели при заданных параметрах и известных внешних воздействиях с фактическим поведением переменных по отчетным результатам функционирования системы.

Для оценки степени совпадения траекторий выходных переменных модели с фактическими данными используются обычно различные вероятностно-статистические методы и коэффициенты, такие, как дисперсионный и факторный анализ, коэффициент несовпадения Тейла [114].

С помощью ретроспективного предсказания можно лишь приблизительно оценить, в какой степени имитационная модель соответствует своему назначению. Это объясняется тем, что в прошлом рассматривалась только одна стратегия управления, результатом воздействия которой являлась единственная траектория и связанные с ней данные за прошлые годы. Поэтому невозможно сравнить эти данные с результатами расчетов по другим возможным траекториям. Особенно это важно при применении имитации в долгосрочном планировании, где таким образом мы не сможем проверить, способна ли наша модель указать лучшую стратегию. В этом случае имеет смысл дополнить имитационную модель планирования оптимизационным блоком, позволяющим из множества вариантов планов выбрать оптимальный. Окончательную оценку пригодности модели можно получить только экспериментальной проверкой.

Главным и конечным этапом имитационного моделирования является проведение непосредственно на ЭВМ имитационного эксперимента, который сопровождается, с одной стороны, планированием, а с другой — обработкой результатов эксперимента. Теория планирования экспериментов и методов анализа его результатов — один из важнейших разделов современной математической статистики. Ранее теория создавалась и развивалась применительно к натурным (физическим) экспериментам, а в последнее время все более широкое применение находит в имитационном моделировании.

С помощью методов теории планирования экспериментов можно добиться значительного сокращения объема экспериментальных расчетов без ущерба для достоверности результатов моделирования. Методы теории позволяют оценить значимость влияния на результаты моделирования тех или иных воздействий внешней среды и целенаправ-

ленных изменений экспериментатором различных параметров модели.

Следует отметить, что многие аспекты применения теории планирования экспериментов в имитационном моделировании освещены недостаточно полно. Большое внимание проблеме планирования эксперимента уделено в работах [98, 106]. Различные стороны этого вопроса, а также некоторые практические результаты планирования, обработки результатов имитационных экспериментов и оценки их значимости приведены в [106, 109]. Отметим, что все перечисленные работы основаны на классических методах планирования эксперимента, а также дисперсионного и регрессионного анализа.

В теории планирования эксперимента есть два важных понятия: фактор и реакция [106]. В зависимости от роли переменной в эксперименте она может быть как фактором, так и реакцией. Если цель эксперимента — изучение влияния какой-нибудь переменной X на переменную Y , то X — фактор, а Y — реакция. В экспериментах с моделями экономических систем факторами являются входные (управляющие) переменные, а реакцией — выходные (управляемые). Факторы, в свою очередь, называются управляемыми, если их уровни можно целенаправленно выбирать в процессе эксперимента. Факторы называются наблюдаемыми, если их уровни измеряются или регистрируются. Довольно часто в эксперименте наблюдаемые факторы совпадают с управляемыми, однако среди первых могут быть и неуправляемые, введенные в модель для повышения ее достоверности. Факторы называются количественными, если их уровни являются числами, влияющими на реакцию. В противном случае факторы называются качественными. Примером последних могут служить правила принятия решений, стратегии управления, выбранные из множества альтернатив, и т. п. Под уровнем фактора понимают его конкретное числовое или качественное значение. При многофакторном эксперименте исследуется обычно зависимость реакции от факторов при различных сочетаниях уровней.

Зачастую количество таких сочетаний может оказаться весьма значительным. Так, в 4-факторном эксперименте с моделью предприятия, имеющей 6 уровней фактора и заказы на производство 5 уровней поставок сырья, 5 возможных стратегий запасов сырья и 10 стратегий замены погрузочно-транспортного оборудования, для полного факторного эксперимента потребуется $6 \times 5 \times 5 \times 10 = 1500$ комбинаций уровней фактора. Каждая из данных комбинаций требует экспериментальных исследований, т. е. проведения серии имитационных расчетов и последующей обработки. Поэтому важным вопросом в планировании эксперимента является уменьшение исследуемых сочетаний уровней факторов. Это достигается исключением из исследуемой модели второстепенных факторов и уменьшением необходимого для рассмотрения количества уровней факторов. Если таким путем достичь сокращения исследуемых сочетаний уровней факторов без ущерба для адекватности модели не представляется возможным, то имеет смысл рассматривать неполные факторные планы, требующие намного меньших затрат машинного времени и не приводящие к значительным

Т а б л и ц а 5.7.1

Капиталовложения	Затраты на ремонт			
	Z (1)	Z (2)	Z (3)	Z (4)
V (1)	P ₁ (1), P ₁ (2), ..., P ₁ (N)			
V (2)	...			
V (3)	...			P ₁₂ (1), P ₁₂ (2),, P ₁₂ (N)

потерям информации о поведении функции реакции эксперимента. Неполные факторные планы чаще всего используются для выявления переменных, наиболее сильно влияющих на реакцию. При планировании эксперимента возникает необходимость как можно точнее определить его цель, что дает возможность наилучшим образом построить план эксперимента. Можно выделить две основные цели эксперимента: 1) найти такую комбинацию уровней факторов, при которой переменная реакция принимает минимальное и максимальное значения; 2) выяснить зависимость реакции от факторов, чтобы понять механизм, лежащий в основе изучаемого процесса.

Первая цель чаще всего преследуется при количественных уровнях факторов, когда возникает необходимость составить оптимальный план работы предприятия по максимуму производства, прибыли, реализации или минимуму издержек и т. п. Вторая цель имеет место при выборе стратегий развития предприятия, оценке критериев управления процессом.

После того как составлен план эксперимента и эксперимент выполнен, надо обработать его результаты. Классические методы планирования эксперимента используются при обработке результатов дисперсионного и регрессионного анализов. Покажем применение дисперсионного анализа на примере возможного эксперимента на ЭВМ с моделью производственно-транспортной системы. В этой модели исходными переменными, т. е. факторами, являются объемы капиталовложений на расширение транспортной сети $V(1)$, $V(2)$, $V(3)$ и затраты на оборудование $Z(1)$, $Z(2)$, $Z(3)$, $Z(4)$, а реакцией — прибыль производства. Исходный план эксперимента состоит в получении данных (генерировании на ЭВМ) для всех комбинаций каждого из трех уровней объема капиталовложений и четырех уровней затрат на реконструкцию оборудования. Такой план называется полным факторным планом для двух факторов и представляется в виде двухмерной таблицы (табл. 5.7.1).

Каждая из 12 клеток таблицы содержит статистическую совокупность N наблюдений одной из 12 комбинаций объема капвложений и затрат на реконструкцию. Величина N определяется потребностью достаточно представительной выборки.

При исследовании влияния рассматриваемых факторов на реакцию возникает главный вопрос: воздействуют ли выбранные факторы на

реакцию, т. е. различаются ли 12 статистических распределений в построенном плане эксперимента. Методы дисперсионного анализа позволяют ответить на эти вопросы путем сравнения выборочных средних значений и дисперсий. Для оценки значимости различий между выборочными средними обычно используется t -критерий Стьюдента, а сравнение осуществляется с помощью F -критерия. Некоторые примеры обработки результатов имитационных экспериментов будут описаны в практических реализациях имитационных исследований с моделями реальных производственно-транспортных систем.

Глава 6

ИМИТАЦИОННЫЕ МОДЕЛИ В НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ УПРАВЛЕНИЯ ПРОИЗВОДСТВЕННО-ТРАНСПОРТНЫМИ СИСТЕМАМИ

6.1. ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ ПРОМЫШЛЕННОГО ТРАНСПОРТА ПРЕДПРИЯТИЙ МЕТАЛЛУРГИЧЕСКОГО КОМПЛЕКСА

Большинство моделей управления экономическими и производственными процессами опирается на балансовые соотношения, связывающие различные технико-экономические показатели функционирования систем с расходами сырья, материалов, топлива и прочих ресурсов, используемых в производстве. Модели, основой которых являются указанные балансовые соотношения, часто описываются как производственно-транспортные экономико-математические модели.

В таких моделях можно выделить два относительно независимых блока ограничений. В первом посредством функциональных соотношений связываются переменные, отражающие производственно-экономическую сущность функционирования системы: объемы производства, мощности, нормы расхода сырья, топлива, материалов, нормативы использования трудовых, финансовых и других ресурсов. Второй блок является непосредственно транспортным и отражает связи поставщиков с потребителями, транспортные издержки и специфику функционирования транспорта, направленную на обеспечение производственной деятельности исследуемой системы. Здесь должны быть сбалансированы объемы поставок и потребления грузов с выделенными для перевозок транспортными средствами и пропускными способностями транспортных сетей.

Следует отметить, что специфика транспортного процесса в производственно-транспортных моделях зачастую отражена недостаточно полно, хотя условия работы транспорта сложны и могут оказать существенное влияние на конечный результат моделирования. С другой стороны, построение моделей управления транспортом в отрыве от моделей управления сопряженными экономическими системами делает первые недостаточно адекватными и пригодными для управления реальными транспортно-технологическими системами.

Поэтому первая и основная особенность моделирования транспортно-технологических систем — это необходимость тесного взаимодействия моделей или блоков (подмоделей), описывающих производственно-экономическую деятельность систем с процессами их транспортного обслуживания. Это в полной мере относится к детерминированным экономико-математическим моделям, и к моделям, учитывающим неопределенность функционирования систем, и экономико-статистическим моделям, пригодным для имитационного моделирования. Дан-

ная особенность одинаково должна учитываться на отраслевом уровне, и на уровне управления предприятием и цехом.

Сам по себе транспорт, вне связи с сопряженными производственно-экономическими объектами, представляет собой достаточно сложную технико-экономическую систему, моделирование которой обычно осуществляют с помощью комплекса взаимосвязанных экономико-математических блоков [109]:

блока определения необходимой исходной информации о потребных перевозках и эксплуатационно-технических и экономических характеристиках элементов транспортной сети;

блока оптимального планирования и управления потоков по транспортной сети (одного или нескольких видов транспорта), включая определение загрузок отдельных элементов сети и распределение перевозок по видам транспорта;

блока оптимального планирования развития транспортной сети, включая при этом определение этапов и уровней строительства или реконструкции отдельных элементов сети;

блока планирования численности и структуры парка подвижного состава и других технических средств транспорта;

блока управления потребностью транспорта в ресурсах различных видов и их использованием.

Особенность перечисленных блоков заключается в том, что результаты решения соответствующих задач направлены на оптимизацию управления и планирования развития технологического транспорта. Исходная информация для решения задач в перечисленных блоках формируется при решении задач управления сопряженными производственно-экономическими системами (отраслями, подотраслями, предприятиями, участками и т. п.).

Стохастический характер переменных в условиях неопределенности и неполноты исходной информации, высокая динамика транспортно-технологических работ на многих народнохозяйственных объектах зачастую не позволяют использовать для решения отдельных технико-экономических задач известные методы математического программирования. В этом случае весьма эффективными и надежными становятся методы имитационного моделирования, применяемые на основе новых информационных технологий управления промышленным транспортом.

Широкое распространение методы имитационного моделирования получили в решении разнообразных задач управления транспортно-технологическими системами общего назначения. Здесь имитационные эксперименты особенно интенсивно применяются при проектировании АСУ промышленным транспортом, а также при решении различных задач комплексного планирования организации и управления перевозочным процессом и его стыковки с сопряженными технологическими системами, т. е. практически во всех перечисленных выше блоках.

Принимая во внимание сложность транспортно-технологических процессов, высокую информационную нагрузку на диспетчерский персонал, случайный характер многих воздействий на систему, можно сказать, что вопросы создания АСУ промышленным транспортом на

различных предприятиях народного хозяйства весьма актуальны. В условиях функционирующих АСУ, когда система снабжена достоверной информацией о состоянии транспортно-технологических работ, имитационное моделирование может стать одним из важнейших способов выбора альтернативных управляющих воздействий при различных отклонениях от нормы перевозочного процесса.

Важное место имитационные методы занимают при проектировании транспортных систем, выборе стратегии построения и развития АСУ, состава ее технических средств, информационного обеспечения, комплекса задач и критериев управления.

Здесь рассмотрим принципы построения АСУ промышленным транспортом металлургического предприятия, применения новых информационных технологий и некоторые примеры использования имитационного моделирования в проектировании и управлении транспортно-технологическими системами.

Характерно, что при исследовании различные транспортно-технологические системы, зачастую рассматриваются как разновидности систем массового обслуживания. Интересно, что в таких системах потоком заявок могут служить потоки грузов в конечных пунктах транспортной сети, а транспортные средства и коммуникации являются каналами обслуживания.

С другой стороны, поток транспортных средств часто рассматривается как поток заявок на обслуживание, а технологические агрегаты и узлы в конечных пунктах транспортной сети служат каналами обслуживания [110]. Таким образом, для моделирования транспортно-технологического процесса в принципе возможно использовать язык типа *GPSS*. В целом отметим, что транспортно-технологические комплексы выглядят организационно гораздо более сложными по сравнению с классическими системами массового обслуживания, и исследование их с помощью имитационных экспериментов в каждом конкретном случае требует индивидуального подхода.

Основной задачей транспорта металлургического предприятия является обеспечение и продолжение технологических процессов по выплавке металла и производства готовой металлургической продукции. Поэтому работа транспорта подчинена общей цели металлургического производства, и от четкости работы транспорта зависят качественные и количественные показатели основного производства. Кроме определяющих факторов основной технологии при управлении железнодорожным транспортом, должен учитываться ряд собственных специфических особенностей. Они касаются правильности формирования-расформирования поездов, выбора маршрутов и пропуски поездов, оформления перевозок и выполнения коммерческих операций, распределения трудовых ресурсов, обеспечения подвижным составом технологических операций и др.

Для управления работой транспорта на металлургическом заводе созданы соответствующие службы, отделы и сектора, обладающие собственной организационной структурой, методами и средствами управления. Регламентирующими документами для управления промышленным транспортом является Устав железных дорог, единый технологи-

ческий процесс, инструкции по сигнализации, централизации, блокировки (СЦБ) и связи, технике безопасности, различные технико-распорядительные акты и служебные инструкции.

Основным средством сбора и передачи оперативной и учетной информации являются документированные и недокументированные сообщения по каналам связи.

В железнодорожном цехе можно выделить непосредственно производство с его обслуживанием и аппарат управления. Если под аппаратом управления понимать не только коллектив работающих, но и средства, методы и используемую информацию, то с точки зрения теории управления производство является объектом, а аппарат — системой [1:09].

Железнодорожному транспорту металлургического завода как объекту управления присущи свои формы организации технологического процесса, своя производственная структура и система материально-производственных связей. Система управления работой транспорта характеризуется своей структурой органов управления, соответствующей системой функций и перечнем информации, используемых в управлении транспортно-технологическим процессом традиционными методами.

Железнодорожный цех как система является одновременно технологическим и экономическим, социальным и психологическим объектом. Поэтому аппарат управления должен располагать различными методами воздействия на объект управления (физическими, экономическими и административными).

В целом исследуемый объект, как показывает опыт, практика и специальная литература, сложный по своему поведению, состоит из огромного количества разнообразных элементов со значительным многообразием и разнородностью количественных и качественных связей между ними, с наличием большого количества стоящих перед ними частных и зачастую разноречивых целей. Причем совокупность последних должна вести к достижению общей цели функционирования транспортно-технологического комплекса.

Между объектом управления и системой управления происходит непрерывный обмен информацией. От объекта к системе идет информация о ходе технологического и транспортного процесса, а система передает на объект информацию об управляющих воздействиях. Эти встречные потоки циркулируют по определенным законам и в совокупности представляют собой информационное обеспечение исследуемого комплекса.

Темпы известительной и управляющей информации при традиционных, неавтоматизированных методах управления транспортом металлургического предприятия таковы, что операторы и учетные работники, а тем более диспетчеры и начальники служб получают только ограниченную агрегированную информацию и зачастую с отставанием от хода технологического процесса. Указанные факты, а также значительная сложность эксплуатационных процессов, многовариантность управленческих решений и высокие требования к оперативности их принятия требуют совершенствования существующих систем управ-

ления, опираясь на применение современных средств вычислительной техники и математических методов, в том числе и имитационного моделирования, а также системного подхода при реализации этих мероприятий. Системный подход характеризуется рассмотрением с единых позиций всех элементов процесса управления объектом, как внутренних, так и внешних.

Ввиду особой роли транспорта в технологии металлургического производства представляется целесообразным выделение в общей системе автоматизации управления завода автономной системы автоматизированного управления железнодорожным транспортом [109]. Такого рода системы должны рассматриваться как составная часть системы комплексной автоматизации диспетчерского управления основными производственными процессами металлургического завода.

АСУ промышленным транспортом представляют собой сложный программно-технический комплекс. Как и большинство подобных систем, они имеют иерархическую многофункциональную структуру. На низшем уровне автоматизации осуществляется управление технологическим (перевозочным) процессом работы транспорта. Этот процесс является основным источником первичной информации. На более высоких уровнях информация агрегируется и на ее основе решаются соответствующие каждому из уровней комплексы функциональных задач.

Известно, что с точки зрения получения максимального экономического эффекта наиболее приемлема система комплексной автоматизации, охватывающая практически все стороны деятельности системы управления транспортом. Однако в силу ее сложности имеет смысл выделить ряд основных этапов ее разработки и внедрения. В решении вопросов этапности создания систем, а также необходимого или эффективного уровня комплексной автоматизации транспортных работ широкое применение находят методы имитационного моделирования. В работах [108, 116] освещены различные аспекты использования имитационных экспериментов с моделями транспортных систем для проектирования АСУ промышленного транспорта.

В начале разработки систем обеспечивается автоматизированный сбор информации, которая используется для решения разного рода информационно- и учетно-статистических задач, определяющих качественную и количественную оценку управления работой транспорта. Возможности информационных систем ограничиваются автоматизацией отдельных учетных функций объекта, что затрудняет решение оптимизационных задач и создание имитационных моделей в системном режиме. Отсутствие эффективной системы сбора информации в разрабатываемых АСУ приводит к разрыву во времени между событием и его отражением в системе, в результате чего значительно снижается ценность первичной информации.

Однако приобретенный опыт конкретных разработок информационных систем, современные достижения в развитии вычислительной техники и средств автоматизации позволили устранить ряд недостатков и приступить к созданию более прогрессивных информационно-управляющих систем [109].

Такие системы используются для проведения расчетов, связанных с планированием и управлением работой транспорта, и имеют принципиальные отличия от информационных систем по различным аспектам. Основные из них — решение задачи в реальном масштабе времени, значительное расширение комплекса функциональных задач, включая задачи планирования и управления работой транспорта, решаемые методами математического программирования и имитационного моделирования, высокая надежность системы автоматического сбора информации, объединяющей большое количество датчиков и устройств, современные управляющие вычислительные комплексы, построенные по иерархическому принципу и обеспечивающие работу системы в диалоговом режиме на всех уровнях управления.

Действенным инструментом в руках проектировщика новых информационных технологий становится имитация работы транспорта и эксперименты с моделями информационных потоков, вариантами каналов связи, средств сбора и обработки информации и возможными структурами программных комплексов.

При реализации общей алгоритмической структуры информационного обеспечения системы наиболее перспективным подходом является создание модели ведения транспортной ситуации на объекте. Такая модель информационная основа для решения задач планирования и управления транспортно-технологическими процессами.

В соответствии с новыми идеями современных информационных технологий такая модель может иметь иерархическую структуру. На нижнем уровне модели, реализуемом на ЛС, персональных ЭВМ низовых звеньев диспетчерской службы железнодорожного цеха и сопряженных технологических участков, ведется обработка текущей оперативной информации и накопление статистических данных. Агрегированная информация передается на ЭВМ центра обработки данных, где формируется динамическая модель транспортной ситуации.

Такую модель можно представить в виде графа $G = (U, T)$, где U — локальные подмодели транспортной ситуации на конечных и промежуточных участках транспортной сети; T — множество дуг, связывающих локальные подмодели.

Последние формируются на персональных ЭВМ низовых диспетчерских звеньев и сопряженных технологических участков. Элементы множества T (дуги графа G) фактически соответствуют внутривозовым транспортным коммуникациям, связывающим конечные и промежуточные пункты транспортной сети. Причем $g_{ij} \in T$, если пункты i и j связаны рельсовой цепью и изменение ситуации в подмодели $u_i \in U$ может повлечь за собой изменение ситуации в $u_j \in U$.

Транспортная ситуация на металлургическом предприятии имеет в некоторой степени стохастический характер. Ввиду значительного количества агрегатов металлургического передела, требующих подвоза и вывоза продукции, а также разнотипности транспортных средств, участвующих в технологическом процессе, задача составления оперативных графиков внутривозовых перевозок является далеко не тривиальной.

Общую задачу можно разбить на ряд подзадач по составлению графиков межцеховых перевозок на участках: рудный двор — доменные

печи; доменные печи — сталеплавильные цехи; сталь — прокат, прокатные станы — станция примыкания МПС.

Каждая из перечисленных подзадач также остается достаточно сложной, так как при решении необходимо учесть ряд вероятностных переменных и согласовывать графики перемещения транспортных единиц с графиками работы технологических агрегатов, ввиду чего размерности задач становятся катастрофически огромны. По этим причинам их практически невозможно решить известными методами математического программирования, а наиболее приемлемым методом решения нам представляется имитационное моделирование как в чистом виде, так и в сочетании с методами оптимизации.

6.2. МОДЕЛИ УПРАВЛЕНИЯ ЗАПАСАМИ И ТРАНСПОРТИРОВКОЙ ЖЕЛЕЗОРУДНОГО СЫРЬЯ

В данном параграфе рассмотрим модели управления запасами железорудного сырья и задач составления графиков движения транспортных средств на участках рудный двор — доменные печи и сталь — прокат, а также некоторые результаты имитационных экспериментов с ними.

Модели управления запасами железорудного сырья на рудном дворе доменного цеха тесно связаны с моделями планирования производства чугуна, составления доменной шихты, регулирования поставок сырья извне и его подачи в доменные печи. Реальную задачу управления запасами сырья и составления шихты для доменного процесса можно описать следующим образом.

На рудном (шихтовом) дворе металлургического комбината имеется некоторое множество M пунктов (фронтов) погрузки-выгрузки различных видов металлургического сырья: руды, агломерата, окатышей, некоторых металлосодержащих добавок. Сырье на каждом складе характеризуется химическим составом, и прежде всего содержанием железа α_i ($i \in M$), а также стоимостью единицы веса.

Потребителей железорудного сырья — доменные печи условно объединим в множество N . Экономико-математическую модель данной задачи можно записать в следующем виде:

$$Z = \sum_{i \in M} \sum_{i \in M} c_i x_{ij} \rightarrow \min; \quad (6.2.1)$$

$$\sum_{j \in N} x_{ij} \leq \min \{W_j, p_{ij}\}, \quad i \in M; \quad (6.2.2)$$

$$\sum_{i \in M} x_{ij} \geq Q_j, \quad j \in N; \quad (6.2.3)$$

$$\beta_j - \Delta\beta \leq \frac{\sum_{i \in M} x_{ij} \alpha_i}{\sum_{i \in M} x_{ij}} \leq \beta_j + \Delta\beta; \quad j \in N; \quad (6.2.4)$$

$$x_{ij} \geq 0; \quad i \in M; \quad j \in N, \quad (6.2.5)$$

где x_{ij} — искомый объем поставок сырья с i -го пункта в j -ю доменную печь; W_j — запас j -го вида сырья; p_{ij} — производительность погру-

зочно-транспортного оборудования на участке i -й пункт погрузки j -я доменная печь; Q_j — потребность в сырье для j -й печи; $\beta_j, \Delta\beta$ — содержание железа в шихте j -й печи и допустимые отклонения.

Решив задачу в постановке (6.2.1) — (6.2.5), мы составим план погрузки и транспортировки сырья с рудного двора в доменные печи. Исходя из составленного плана, можно определить объем производства чугуна по каждой доменной печи. Для j -й печи он составит

$$A_j = \sum_{i \in M} x_{ij} \alpha_j \frac{1 - \Delta_j}{k_j}, \quad j \in N, \quad (6.2.6)$$

где Δ_j — потери железа при выплавке чугуна в j -й печи; k_j — содержание железа в чугуне, выплаваемом в j -й печи.

Объем выплавки чугуна в r -е сутки по цеху составит

$$A_r = \sum_{j \in N} A_j, \quad (6.2.7)$$

а суточные объемы в плановый период T должны удовлетворять плановому заданию или госзаказу (A) на данный период $\sum_{r=1}^T A_r = A$.

Для определения дефицитности ресурсов и соответственно моментов для заказа поставок сырья извне можно решить двойственную к (6.2.1) — (6.2.5) задачу. Двойственные переменные, соответствующие ресурсным неравенствам, являются мерой дефицитности ресурсов, находящихся на складах сырья. Из теории двойственности задач линейного программирования известно, что если двойственная переменная $y_i = 0$, то соответствующий ей i -й ресурс является недефицитным, а чем больше значение y_i , тем выше его дефицитность и условная цена.

Таким образом, при имитационных экспериментах по установлению оптимального уровня запасов на i -ом складе двойственная переменная y_i как мера дефицитности ресурсов может служить ценой потерь при отсутствии ресурсов и сигналом для возобновления поставок.

Задача обеспечения железорудными ресурсами металлургических предприятий относится к классу задач отраслевого планирования. В новых условиях хозяйствования, когда предприятию предоставлена значительная самостоятельность в решении производственных и планово-экономических задач, а на передний план вместо директивных заданий выходит госзаказ, на отраслевой орган управления ложатся ответственные задачи по подкреплению госзаказов ресурсным обеспечением.

В комплексе задач ресурсного обеспечения предприятий металлургического комплекса наиболее важны задачи, связанные с планированием добычи, переработки и транспортировки железорудного сырья. Поскольку добыча и подготовка железорудного сырья осуществляется на горнообогатительных комбинатах и рудниках отрасли, а потребителями являются в основном доменные цехи металлургического производства, то первые олицетворяют поставщиков, а вторые — потребителей сырья.

Задачи планирования перевозок и управления запасами железорудного сырья в отраслевом масштабе решаются на транспортной сети,

связывающей конечные пункты. Такую сеть можно представить в виде графа $G = (M, S)$, где M множество станций на сети, состоящее из $M_1 \subset M$ — подмножества станций примыкания горнообогатительных комбинатов (поставщиков); $M_2 \subset M$ — подмножества станций примыкания металлургических предприятий (потребителей) и $M_3 = M \setminus (M_1 \cup M_2)$ — подмножества промежуточных узловых станций транспортной сети. Дуги соответствуют транспортным магистралям, связывающим конечные и промежуточные пункты транспортной сети.

Задача отраслевого планирования и прогнозирования поставок железорудного сырья на сети G решается в два этапа. На первом определяются объемы перевозок сырья от поставщиков к потребителям исходя из его потребностей, наличия транспортных средств, сырья у поставщиков и т. д. Целевой функцией такой задачи может служить минимизация транспортных или приведенных затрат при ограничениях, которые строятся аналогично модели (6.2.2) — (6.2.5).

В такой системе складываются традиционные технологические и транспортные связи между отдельными поставщиками и потребителями, которые легко формализуются с помощью матрицы $R = \|r_{ij}\|$, где элементы r_{ij} принимают следующие значения:

$$r_{ij} = \begin{cases} 1, & \text{если } i\text{-й поставщик прикреплен к } j\text{-му потребителю;} \\ 0 & \text{в противном случае.} \end{cases}$$

Экономико-математическая постановка данной задачи, так же, как и описанная ранее модель задачи (6.2.1) — (6.2.5), относятся к классу производственно-транспортных задач линейного программирования, и их решение принципиальных трудностей не встречает.

Сложнее выглядит стыковка указанных задач с имитационными моделями маршрутизации движения железнодорожных составов, с сырьем на транспортной сети G . Здесь привлекаются последние разработки в области новых информационных технологий и в некоторой степени математического аппарата экспертных систем.

Согласно описанной методике проведения имитационных экспериментов с моделями производственно-транспортных систем требуется выполнения ряда необходимых процедур. Наиболее важными из них являются разработка математической модели объекта, моделирующего алгоритма, который реализует системное время, соответствующее реальному, а также проведение непосредственно на ЭВМ экспериментов и обработка их результатов.

Остановимся подробнее на разработке имитационной математической модели применительно к поставленной задаче.

Объединим железнодорожные составы, осуществляющие перевозки сырья на сети G , в множество $B = \{1, 2, \dots, b\}$. Обычно для перевозки железорудного сырья формируются специализированные составы, которые довольно редко подвергаются расформированию и потому характеризуются относительно постоянной грузоподъемностью q_i ($i \in B$). Скорость движения состава в нашей модели является случайной величиной с заданными параметрами закона распределения вероятностей. Для учета зависимостей параметров закона распределения V_i от ха-

характеристик перегона (загруженность ветки, топология, качество потока и т. п.), по которому движется состав, задается матрица $D = \|d_{ij}\|$; $j, f \in M$; $j \neq f$, где d_{ij} — эмпирический коэффициент, с помощью которого простым умножением корректируется средняя скорость движения состава на перегоне между j -м и f -м пунктами транспортной сети.

Аналогично задается матрица расстояний $S = \|s_{ij}\|$; $j, f \in M$; $j \neq f$, где s_{ij} — расстояние между j -ом и f -ом пунктами сети G .

Маршрут движения i -го состава ($i \in B$) характеризуется конечными пунктами, т. е. станциями примыкания поставщиков и потребителей железорудного сырья, а также промежуточными узловыми станциями транспортной сети. Таким образом, маршрут движения можно представить в виде упорядоченного множества $P = \{p_1, p_2, \dots, p_k\}$, где p_1 и p_k — начальный и конечный пункты маршрута; p_1, p_2, \dots, p_{k-1} — упорядоченная последовательность промежуточных узловых станций.

Длина маршрута движения i -го состава вычисляется по формуле

$$Z_i = \sum_{p_i, p_{i+1} \in P} s_{p_i, p_{i+1}}$$

Отметим, что начальный и конечный пункты маршрута (поставщик и потребитель) определяются в результате решения задачи отраслевого планирования поставок железорудного сырья. Промежуточные станции, а также временные характеристики маршрута будут определены в процессе имитации перемещения i -го состава по транспортной сети.

Таким образом, маршрут движения состава и его характеристики являются управляемыми переменными модели. Управляющими (независимыми) переменными являются скорости движения и соответственно длительности перемещения составов на отдельных участках транспортной сети.

Выходной (исследуемой) переменной может служить какой-либо технико-экономический показатель, например суммарные транспортные издержки, грузооборот составов, доля прибыли металлургического комплекса, полученная в результате повышения эффективности системы управления запасами. Указанная переменная является функцией входных (управляемых и управляющих) переменных и параметров модели

$$Z = F(p_i, z_i, D, V). \quad (6.2.8)$$

В результате имитационных экспериментов необходимо построить графики перевозок сырья на транспортной сети таким образом, чтобы за период времени T были обеспечены поставки согласно отраслевому плану и при этом достигался оптимум выходной переменной согласно (6.2.8).

На временные характеристики маршрутов, переменные и параметры имитационной модели накладываются ограничения, учитывающие длительность обработки составов у поставщиков и потребителей. Кроме того, для упрощения экспериментов в модели не имитируются обратные перевозки грузов.

Минимально необходимое время возврата состава, окончившего разгрузку, к поставщику задается в виде ограничения следующим образом. К началу моделирования в множество B включаются все составы, занятые на перевозке железорудного сырья на сети G . Каждому из них соответствует ограничение на минимальное время поступления состава к поставщику. Указанные моменты для всех составов записываются в виде вектора $t = (t_1, t_2, \dots, t_b)$. Если к началу моделирования нет данных о фактическом состоянии транспортной ситуации, то компонентам вектора t присваивается значение 0. Если в процессе моделирования некоторый состав i закончил маршрут у потребителя и длительность его продвижения по маршруту составила θ_i ($\theta_i \approx z_i/v_i$), то соответствующая компонента вектора t принимает значение

$$t_i := t_i + 2\theta_i. \quad (6.2.9)$$

Таким образом, i -й состав может участвовать в перевозке железорудного сырья начиная с момента времени t_i .

Рассмотрим более подробно моделирующий алгоритм, реализующий основные соотношения имитационной модели (6.2.8) — (6.2.9) при наличии высокоорганизованной информационной базы для описания транспортной ситуации в отраслевом масштабе. Системное время в алгоритме организовано по принципу «особых состояний», под которыми здесь подразумевается окончание одной из текущих операций, а именно окончание погрузки или разгрузки состава в конечном пункте, прибытие состава на промежуточный или конечный пункт и т. д. Этот принцип наиболее подходит к имитации функционирования систем на моделях подобного класса.

Опишем процесс корректировки системного времени на одной промежуточной итерации. Каждая итерация начинается с просмотра подмножества $(M \setminus M_1) \subset G$, т. е. просматриваются все пункты сети, кроме поставщиков сырья. Из просматриваемого множества выбирается пункт, в котором находится i -й состав с минимальным временем прибытия:

$$\tau_i = \min_{i \in (M \setminus M_1)} \{\tau_i\}. \quad (6.2.10)$$

Множество элементов $\{\tau_i\}$; $i \in (M \setminus M_1)$ совместно с компонентами вектора t в модели образуют информационный массив особых состояний. После нахождения пункта с минимальным значением τ_i последняя величина сравнивается с элементами вектора t и строится множество D номеров j компонент вектора t , для которых $t_j \leq \tau_i$; $D = \{j/j \in B; t_j \leq \tau_i\}$.

Рассмотрим три возможных ситуации.

1. $D \neq \emptyset$, т. е. имеется хотя бы один состав, готовый к погрузке сырья раньше, чем какой-либо груженный состав прибывает на i -ю промежуточную станцию. В этой ситуации выбирается состав с номером r , для которого $t_r = \min_{j \in D} \{t_j\}$, и осуществляется условная погрузка и отправка r -го состава. Для упрощения алгоритма составы могут быть заведомо прикреплены к поставщикам. Если реальные условия не позволяют этого сделать, то моделируется выбор поставщика по специальному алгоритму.

Условная погрузка и отправка r -го состава от станции примыкания l -го горнообогатительного комбината к ближайшей k -й узловой станции заключается в пересчете времени прохождения состава по данному маршруту с учетом времени его пребывания у поставщика. Момент времени прибытия состава на k -ю станцию рассчитывается по формуле

$$\tau_k^{(r)} = t_r + d_{lr} + \frac{s_{ik}}{v_r} d_{ik}, \quad (6.2.11)$$

где d_{lr} — суммарная длительность прохождения технологических операций r -ым составом у l -го поставщика.

Полученное согласно (6.2.11) новое значение $\tau_k^{(r)}$ заносится в информационный массив особых состояний в специальный раздел, где формируется транспортная ситуация, соответствующая k -й узловой станции.

При расчете по формуле (6.2.11) величины d_{lr} и v_r генерируются как случайные с известными параметрами законов распределения вероятностей. После пересчетов компонента t_r должна быть исключена из рассмотрения на последующих итерациях, для чего ей присваивается значение достаточно большого числа, например $t_r = \Delta$ ($\Delta > T$), и затем осуществляется переход к поиску нового минимального значения τ_l .

2. $D = 0$. Проверяется принадлежность соответствующего пункта множеству поставщиков $i \in M_2$. Если последнее условие не выполняется, т. е. выбранный пункт является промежуточной узловой станцией, то осуществляется выбор следующей станции, к которой необходимо переадресовать r -й состав. При этом следующая станция должна находиться на допустимом для данного состава маршруте перевозки сырья, т. е. маршрут должен связывать конечные пункты согласно прикреплению, полученному в результате решения задачи отраслевого планирования. Если возможных станций назначения состава окажется несколько, то выбор адреса производится по критерию, обеспечивающему минимизацию длительности прохождения всего маршрута. При этом учитывается расстояние от выбираемой станции до конечного пункта назначения. В некотором смысле реализуется принцип оптимальности динамического программирования, а также рассматриваются возможные ситуации, возникающие к моменту прибытия состава на станцию. Здесь можно генерировать задержки составов на промежуточных станциях при наличии статистических параметров распределения фактических задержек.

После выбора оптимального j -го адреса для перемещения r -го состава производится пересчет времени освобождения на j -й станции и занесение его в массив особых состояний:

$$\tau_j^r = \tau_l^{(r)} + \frac{s_{lj}}{v_{lj}} d_{lj} + e_{rj}. \quad (6.2.12)$$

Здесь e_{rj} — случайная величина задержки r -го состава на j -й станции, а v_{lj} генерируется как случайная величина в (6.2.12) по аналогии с (6.2.11).

3. $D = 0$ и $i \in M_2$, т. е. i -й пункт является станцией примыкания потребителя металлургического предприятия, и r -й состав прибыл

на разгрузку. В этом случае фиксируется весь маршрут продвижения r -го состава в виде упорядоченного множества R_r , подсчитывается длина маршрута L_r и длительность нахождения состава на маршруте:

$$\sum_{\substack{i, j \in R_r \\ i=j+1}} (r_i - \tau_i).$$

На основании этих данных ведется расчет текущего значения выходной переменной Z , которая суммируется по всем составам, прибывающим в пункты назначения. Например, если в качестве выходной переменной выбрали суммарный грузооборот, то подсчет ведется по формуле

$$Z := Z + L_r q_r. \quad (6.2.13)$$

Моделирование заканчивается в тот момент, когда минимальное значение τ_i превысит длительность планового периода T .

Как указывалось выше, основной целью имитационных экспериментов с приведенной моделью является построение графиков перевозки железорудного сырья на сети G , обеспечивающего оптимальное значение комплексного критерия (6.2.13), а также необходимый уровень запасов железорудного сырья на складах металлургических предприятий. Имитационные эксперименты с предложенной моделью позволяют также исследовать данную производственно-транспортную систему для определения оптимальных значений некоторых технологических и технико-экономических параметров. Здесь можно установить зависимости выходной переменной от грузоподъемности составов, скорости их продвижения на отдельных участках транспортной сети, длительности погрузочно-разгрузочных работ и задержек составов на промежуточных станциях.

Обработка результатов имитационных экспериментов осуществляется с помощью методов планирования экспериментов, дисперсионного и регрессионного анализа. Данные, полученные при имитации, а также результаты их обработки становятся информационной основой для объективных экспертных оценок эффективности функционирования производственно-транспортной системы, прогнозирования ее развития и совершенствования информационных технологий в системе управления объектом.

6.3. РЕЗУЛЬТАТЫ ИМИТАЦИОННЫХ ЭКСПЕРИМЕНТОВ С МОДЕЛЬЮ ТРАНСПОРТИРОВКИ ЖЕЛЕЗОРУДНОГО СЫРЬЯ

Описанные в предыдущем параграфе модели и алгоритмы отражают наиболее общие положения проблемы управления запасами и транспортировкой железорудного сырья в отраслевом масштабе. При машинной реализации моделей авторам пришлось отказаться от сложной системы формализации критерия оптимальности функционирования системы и сформулировать конкретные цели, а на их основе — достаточно простые критерии моделирования. В качестве управляемых (выходных) переменных модели поочередно выбирались два технико-экономических показателя: 1) суммарное время простоев (тихих ходов)

доменных цехов ввиду отсутствия железорудного сырья; 2) объемы сверхнормативных запасов сырья.

В модель были введены соотношения, которые связывают систему управления транспортировкой сырья с системами управления запасами сырья у металлургических предприятий. Основной задачей серии экспериментов служило составление оптимальных графиков поставок сырья, обеспечивающих минимальное суммарное время простоев доменных печей или минимальную сумму сверхнормативных запасов сырья.

Проверка пригодности модели и оценка эффективности стратегии изменения управляющих переменных производилась на данных, отражающих реальные производственно-транспортные связи между горнорудными и металлургическими предприятиями. С другой стороны, была принята некоторая условность исходных данных, которая обеспечивала необходимую простоту и доступность машинного эксперимента.

На первой стадии экспериментов были выбраны 4 условных поставщика железорудного сырья (горнообогатительных комбинатов), 5 условных потребителей (металлургических предприятий) и 5 составов (вертушек), приспособленных для перевозки железорудного сырья на сети, связывающей условных поставщиков и потребителей. Отметим, что условные поставщики и потребители выбраны из множества реальных, а сеть построена исходя из реальных железнодорожных магистралей, но без промежуточных станций. Запрет проезда от i -го к j -у пункту задается путем присвоения $l_{ij} = M$, где M — достаточно большое число.

Разработанный для данного эксперимента моделирующий алгоритм несколько отличается от описанного ранее. В частности, в качестве особого состояния на каждой итерации служило минимальное для всех потребителей время достижения критического уровня запасов сырья. Выбрав металлургическое предприятие с наименьшим (ниже критического) уровнем запаса, алгоритм осуществляет поиск состава с минимально возможным уровнем прибытия состава к данному предприятию. Для него же производится пересчет новой временной точки достижения критического уровня запаса, а для состава — момент его освобождения после разгрузки. Укрупненная блок-схема алгоритма моделирования показана на рис. 6.3.1.

Управляющими переменными в модели используются скорости движения составов, длительности их технологического обслуживания у поставщиков и потребителей, а также количество составов, используемых для транспортировки сырья в системе. Первые две переменные имеют вероятностный характер. На основе реальных данных были построены законные распределения вероятностей и оценены параметры законов. В процессе имитационных экспериментов последовательно с участием ЛПР изменялись уровни переменных (математические ожидания). Для каждого уровня генерировались конкретные значения скоростей движения составов и длительности их технологического обслуживания в пределах законов их распределения.

В ряде экспериментов исследовались зависимости суммарных простоев технологического оборудования у потребителей — доменных

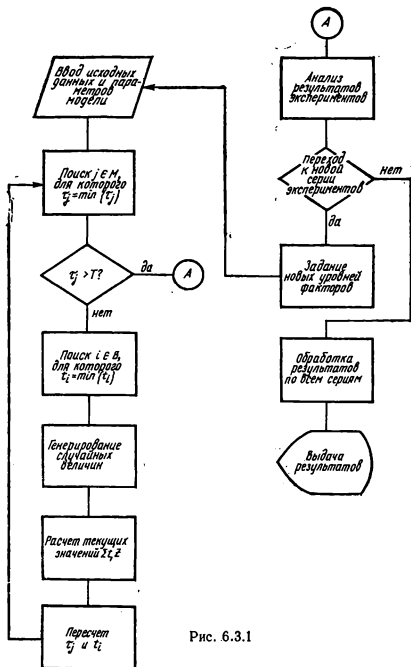


Рис. 6.3.1

печей (Σt) и сверхнормативных запасов сырья в доменных цехах (Z) от количества составов, используемых для транспортировки сырья на сети. Для двухфакторных экспериментов построены графики зависимости выходных переменных (Σt , Z) от управляющих — количества составов K и уровня скорости их движения V (рис. 6.3.2, 6.3.3).

Пример результатов имитационных экспериментов при четырех составах и средней скорости $V = 60$ км/ч приведен в табл. 6.3.1. Длительность условного отрезка моделирования здесь равна 100 часам.

Невзирая на то что приведенные результаты получены для условной производственно-транспортной системы, они оказались полезны

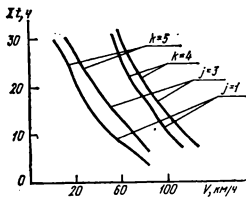


Рис. 6.3.2

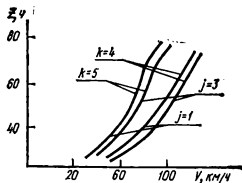


Рис. 6.3.3

для управления реальным процессом поставок железорудного сырья и управления его запасами на предприятиях металлургического комплекса республики. В частности, построенная имитационная модель позволяет выбрать эффективные критерии принятия управленческих решений в отраслевой АСУ по составлению оптимальных графиков перемещения составов на транспортной сети и по прогнозированию их прибытия на конечные пункты. Результаты экспериментов можно использовать для выбора некоторых рациональных параметров системы, таких, как критические уровни запасов сырья, грузоподъемность по-

Таблица 6.3.1

Номер состава	Номер пункта		Момент времени		Момент возникновения критического уровня запаса	Простой поездов на тихом ходу
	погрузки	разгрузки	погрузки	разгрузки		
4	4	3	8.3	20.6	21.0	0.0
2	5	1	7.0	16.0	24.0	0.0
1	2	2	11.0	20.0	27.0	0.0
3	4	4	12.0	21.0	32.0	0.0
2	5	3	23.0	37.0	37.0	1.0
3	5	1	29.6	38.6	39.0	0.0
1	2	2	27.0	36.0	42.0	0.0
4	4	4	30.9	39.9	47.0	0.0
1	2	3	43.0	58.6	58.6	7.6
3	5	1	45.6	54.6	54.6	0.6
2	2	2	30.6	59.6	59.6	2.6
4	4	4	46.9	55.9	62.0	0.0
3	5	1	61.6	70.6	70.6	1.6
4	4	3	62.9	75.3	75.3	9.3
2	2	2	66.6	75.6	75.6	3.6
1	4	4	69.0	78.0	78.0	1.0

Номер состава	Номер пункта		Момент времени		Момент возникновения критического уровня запаса	Простой печей на тихом ходу
	погруз-ки	разгруз-ки	погрузки	разгрузки		
3	5	1	77.6	86.6	86.6	2.6
1	4	3	85.0	97.3	97.3	16.3
2	2	2	82.6	91.6	91.6	4.6
4	4	4	85.6	94.6	94.6	2.6
3	5	1	93.6	102.6	102.6	3.6
2	2	2	98.6	107.6	107.6	5.6
4	4	4	101.6	110.6	110.6	3.6
1	4	3	107.6	120.0	120.0	24.0
3	5	1	109.6	118.6	118.6	4.6

движного состава и необходимое количество составов, которые обеспечивают требуемые объемы транспортировки сырья в течение всего планового периода.

С помощью имитационных экспериментов исследованы две ветви модели, соответствующие двум целям моделирования: 1) установлению зависимости простоев оборудования Σt от количества работающих на сети составов и уровней скорости их движения; 2) сумме сверхнормативных запасов сырья Z от тех же управляющих переменных.

При использовании данной имитационной модели в информационно-управляющей системе для прогнозирования транспортно-технологического процесса каждая из двух ветвей может быть дополнена блоками рекомендаций ЛПР по принятию управленческих решений в случае ожидаемого срыва выполнения плановых заданий. Например, если прогноз покажет значительное превышение величины над плановым, т. е. прогнозируются значительные недопоставки сырья потребителям, указанный блок должен дать рекомендации по задействованию на транспортировке для сырья дополнительного количества составов, либо предложить увеличить в пределах допустимого магистральные скорости движения, либо другие приемлемые для ЛПР и системы управленческие решения. Затем по модели отраслевого планирования следует пересчитать план поставок железорудного сырья и выдать ЛПР новые плановые установки для принятия окончательных управленческих решений.

Описанная модель была реализована на языке Бейсик и имитационные эксперименты проводились на персональной ЭВМ ЕС-1840. Средства поддержки диалога в данной ЭВМ позволили достаточно оперативно в интерактивном режиме корректировать уровни факторов и проследить направления изменения управляемых переменных.

В процессе экспериментов, кроме поставленных задач исследования, конкретной транспортно-технологической системы методами имита-

ционного моделирования выполнялась проверка возможности использования персональных ЭВМ в качестве технического оснащения автоматизированного рабочего места транспортного диспетчера. Получены положительные результаты, показавшие, что эффективность диспетчерского управления будет значительно повышена при использовании ПЭВМ, ЛС и распределенных баз данных на ПЭВМ для организации информационного обслуживания управленческих звеньев исследуемой системы.

Реализованные на технической информационной основе методы имитационного моделирования и оптимизации составляют новую информационную технологию управления производственно-транспортным комплексом.

Результаты экспериментов показали, что такая информационная технология дает мощный инструмент для ЛПР в принятии качественно новых управленческих решений и может значительно повысить эффективность функционирования системы обеспечения железорудным сырьем металлургического производства.

6.4. МОДЕЛИРОВАНИЕ МЕЖЦЕХОВЫХ ПЕРЕВОЗОК

Имитационное моделирование может значительно повысить эффективность управления промышленным транспортом в реальном масштабе времени. Методы имитации в информационных технологиях управления межцеховыми перевозками позволяют точно и быстро решать сложные задачи по оптимальной адресации транспортных средств к сопряженным технологическим агрегатам, обеспечивая минимальные простои как агрегатов, так и железнодорожных составов. При этом обеспечивается безусловное выполнение оперативных графиков выпуска полуфабрикатов на промежуточных участках технологических процессов и поставка их на последующие переделы в заданные моменты времени.

Транспортно-технологическая система крупного металлургического предприятия подвержена различным, зачастую случайным возмущениям, и многие переменные, характеризующие межцеховые перевозки, носят стохастический характер.

По этой причине вопросы разработки математических моделей и алгоритмов оперативного планирования и управления транспортом металлургического предприятия далеко не тривиальны. Повышение адекватности функциональных задач реальному процессу требует учета стохастического характера переменных математических моделей, а это делает практически невозможным использование известных методов математического программирования. Большое значение имеет здесь правильная организация информационного и технического обеспечения, которые следует создавать на принципах новой информационной технологии, используя распределенные базы данных и локальные сети мини-ЭВМ.

Наиболее приемлемым методом решения функциональных задач становится имитационное моделирование как в чистом виде, так и в сочетании с методами оптимизации и стохастического программирования.

Рассмотрим некоторые алгоритмы решения задач оперативного планирования и управления межцеховыми перевозками на примере составления оперативных графиков и оптимальной адресации транспортных средств на участке сталеплавильные цехи — прокатные станы (сталь — прокат).

Задачу решаем, предполагая, что имеются графики выпуска стали из сталеплавильных агрегатов и графики загрузки слитков в нагревательные печи прокатных станов, т. е. известны объемы взаимных межцеховых поставок, где поставщиками выступают сталеплавильные агрегаты, а потребителями — прокатные станы. Объемы межцеховых поставок на смену можно задать в виде матрицы $Y = \|y_{ij}\|$, где y_{ij} — плановый объем поставок стальных слитков от i -го агрегата (мартеновской печи, конвертора, электросталеплавильной печи) к j -у прокатному стану. Маршрут движения каждой транспортной единицы не имеет постоянного, например челночного, характера и определяется в определенные моменты времени на основании заявок цехов. Заявки, в свою очередь, возникают в моменты окончания плавки на сталеплавильных агрегатах или в соответствии с графиками загрузки слитков в нагревательные печи.

Следует подчеркнуть такую особенность промышленного транспорта металлургического предприятия, как специализация вагонов и составов, используемых для перевозки металлургического сырья и переработанного металла. В частности, на транспортном участке, связывающем сталеплавильные цехи и прокатные станы, используются так называемые слитковозы — составы из платформ, приспособленных для перевозки стальных слитков. Аналогично на участке, связывающем доменные и сталеплавильные агрегаты, работают чугуновозы, а на участке шихтовый двор — доменные печи используются вагоны, приспособленные для перевозки железорудного сырья и кокса. На указанных участках выполняются и другие виды перевозок, однако их объемы столь незначительны, что из общей задачи управления промышленным транспортом металлургического предприятия можно без ущерба для достоверности расчетов выделить три относительно независимых блока управления, в том числе и блок оперативного управления транспортом на участке сталь — прокат.

Остановимся подробнее на задачах составления графика движения транспорта и выбора оптимальных адресов для подачи слитковозов под погрузку. Для разработки математической имитационной модели и моделирующего алгоритма введем некоторые обозначения. Пусть $N = \{1, 2, \dots, n\}$ — множество номеров слитковозов, работающих на участке. Каждому номеру $i \in N$ соответствуют q_i — грузоподъемность i -го слитковоза; $V_i^{\text{гр}}$ — скорость движения груженого состава; $V_i^{\text{пор}}$ — скорость движения порожнего состава.

Предположим, что известны законы распределения случайных величин $V_i^{\text{гр}}$ и $V_i^{\text{пор}}$ и заданы параметры законов распределений.

Множество номеров сталеплавильных агрегатов обозначим через $M = \{1, 2, \dots, m\}$. Каждому номеру $j \in M$ соответствует вес плавки q_j , который на период оперативного планирования можно считать ве-

личиной условно постоянной; длительность разливки стали одной плавки с j -го агрегата в i -й слитковоз — τ_{ij}^{pas} , случайная величина с известными законами распределения.

Согласно оперативному плану производства по каждому сталеплавильному агрегату имеется график начала и окончания плавки, т. е. заданы моменты начала разливки стали, фактически совпадающие с окончаниями плавки. Указанные моменты запишем в виде вектора

$$T^{(p)} = (t_{1j}^{(p)}, t_{2j}^{(p)}, \dots, t_{lj}^{(p)}),$$

где l — общее количество плавки на плановый период. Компоненты вектора $T^{(p)}$ в моделирующем алгоритме будут служить точками особых состояний, причем в векторе они расположены в порядке неубывания ($t_{ip}^{(p)} \geq t_{is}^{(p)}$, если $p > s$).

Нагревательные печи прокатных станов объединим в множество $K = \{1, 2, \dots, k\}$. Каждому номеру $k \in K$ соответствует τ_{ik}^a — длительность выгрузки слитков из i -го слитковоза в k -ю печь, являющаяся случайной величиной с известным законом распределения; β_k — часовая производительность k -го прокатного стана в пересчете на стальные слитки. В дальнейшем будет использоваться нормативное значение последней.

С учетом топологии транспортной сети и технологических связей между сталеплавильными агрегатами и прокатными станами на участке сталь — прокат задается матрица $\|\alpha_{jk}\|$, элементы которой принимают значения

$$\alpha_{jk} = \begin{cases} 1, & \text{если } j\text{-й сталеплавильный агрегат имеет транспортную коммуникацию и прикреплен к } k\text{-му стану;} \\ 0 & \text{в противном случае.} \end{cases}$$

Основным функциональным соотношением имитационной математической модели является формула для расчета момента освобождения слитковоза, т. е. расчета момента окончания разгрузки слитков в нагревательные печи прокатного стана. Указанный расчет выполняется при возникновении очередного особого состояния, определяемого очередным r -м моментом начала разливки стали на j -м агрегате — $t_{rj}^{(p)}$. Для приема стали здесь выбираются несколько свободных слитковозов, способных вместить всю плавку. Метод выбора последних мы рассмотрим несколько позже.

Момент освобождения i -го слитковоза, выбранного для приема стали на j -ом агрегате, рассчитывается по формуле

$$t_{ik} = t_{rj}^{(p)} + \tau_{ij}^{(pas)} + L_{jk}/V_i^{rp} + \max\{\tau_{jk}^{(mar)}; q_i/\beta_k\}. \quad (6.4.1)$$

Для последней составляющей формулы (6.4.1) выбирается большее из двух чисел: длительность выгрузки слитков на входе прокатного стана в нагревательные печи или длительность проката слитков, если прокат производится, минуя нагревательные печи. Более точный учет данного показателя можно обеспечить, моделируя работу прокатного стана и нагревательных печей в отдельном блоке моделей.

Момент времени t_{ik} является определяющим для очередной производственной операции i -го слитковоза. Если момент возникновения

очередного особого состояния окажется больше t_{ik} , то i -й слитковоз может быть выбран для приема стали на соответствующем (j') сталеплавильном агрегате при условии, что между последним и k -м прокатным станом имеется транспортная связь, т. е. $\alpha_{j'k} = 1$.

Основная часть моделирующего алгоритма.

Шаг 1. Ввод исходных данных.

Шаг 2. Формирование вектора моментов возникновения особых состояний.

Шаг 3. Выбор j -го сталеплавильного агрегата $j \in M$ с минимальным временем окончания плавки $t_{rj}^{(p)} = \min_{S \in M} \{t_{rj}^{(p)}\}$.

Шаг 4. Вычеркиваем элемент t_{rj} из вектора моментов возникновения особых состояний и проверяем его на окончание процесса моделирования $t_{rj}^{(p)} > T$. Если момент возникновения очередного особого состояния больше длительности моделирования (например, длительности смены), то переход к шагу 14.

Шаг 5. Формируем подмножество слитковозов, готовых к разливке стали на j -м агрегате, т. е. тех, которые могут прибыть к j -му агрегату раньше момента $t_{rj}^{(p)}$:

$$N' = \{i | i \in N; t_{ik} + L_{jk}/V_i^{(nop)} \leq t_{rj}^{(p)}; \alpha_{jk} = 1\}.$$

Шаг 6. Проверка условия $\sum_{i \in N'} q_i > p_j$. Если условие удовлетворяется, то переход к шагу 8.

Шаг 7. Формируем подмножество слитковозов, готовых к разливке стали после момента окончания загрузки транспортных единиц подмножества N' :

$$N'' = \{i | i \in (N \setminus N'); t_{ik} + L_{jk}/V_i^{(nop)} < t_{rj}^{(p)} + \sum_{i \in N'} \tau_i^{(pass)}; \alpha_{jk} = 1\}.$$

Шаг 8. Из N' выбираем i -й слитковоз с минимальным временем прибытия на разливку $t_{ik} + L_{jk}/V_i^{(nop)} = \min_{S \in N'} \{t_{sk} + L_{jk}/V_s^{(nop)}\}$. Полагаем $N' = N' \setminus \{i\}$; $p_j = p_j - q_i$.

Шаг 9. Если $p_j > 0$, то переход к шагу 10. В противном случае к шагу 12.

Шаг 10. Если $N' = 0$, то переход к шагу 8 с заменой подмножества N' на N'' .

Шаг 11. Выбор k -го прокатного стана для разгрузки i -го слитковоза при условии $\alpha_{jk} = 1$. Пересчет момента времени освобождения i -го слитковоза согласно (6.4.1). Переход к шагу 8.

Шаг 12. Печать графиков движения транспортных средств, погруженных на j -ом сталеплавильном агрегате.

Шаг 13. Переход к шагу 3.

Шаг 14. Печать окончательных результатов моделирования.

Шаг 15. Конец.

Предложенный алгоритм позволяет составить близкий к оптимальному оперативный график движения слитковозов на участке сталь — прокат. Кроме того, отдельные шаги (шаг 5 — шаг 11) алгоритма могут быть использованы для реализации управления движением транспорта в реальном масштабе времени.

Весь комплекс задач управления решается на информационной модели, описывающей производственно-транспортную ситуацию на участке сталь — прокат. При составлении программ для описанных модулирующих алгоритмов элементами информационной технологии служат векторы-строки, в которых по каждой технологической и транспортной единице содержатся стационарные данные (номер, грузоподъемность, производительность и т. д.), а также текущие данные, описывающие динамику состояния объекта в любой момент времени. Некоторые данные состояния накапливаются и на их основе формируются конечные учетно-аналитические показатели работы производственно-транспортной системы. Указанные векторы-строки в файле информации и под управлением соответствующей СУБД представляют собой распределенную базу данных о транспортно-технологической ситуации на объекте.

В процессе моделирования и решения задач управления в реальном масштабе времени генерируются случайные переменные $V_{i\text{пор}}$, $V_{i\text{р}}$, $\tau_{ij}^{(\text{рас})}$, $\tau_{ik}^{(\text{внг})}$ и др., параметры законов распределения которых к началу моделирования содержатся в базе данных в виде условно-постоянных величин. В процессе функционирования информационно-управляющей системы промышленного транспорта производится накопление фактических данных, их статистическая обработка и постоянная актуализация информации о параметрах законов распределения.

Аналогично описанной может быть реализована информационная технология и построены имитационные модели управления транспортом на участке «чугун — сталь» и на других участках транспортной сети металлургического предприятия.

Следует отметить, что для более адекватного описания транспортно-технологических процессов на указанных участках необходимо также моделировать движение транспортных средств, перевозящих не только стальные и чугунные слитки, но и другие виды переделного металла, а также сырье и топливо. Особенно важно учитывать сопутствующие перевозки на участке чугун — сталь, где до 30 % транспортных работ составляют перевозки шихтовых материалов для производства стали. Более подробно указанные модели описаны в работе [109].

Результаты имитационных экспериментов с описанными моделями в условиях современных АСУ технологическим транспортом дают близкие к оптимальным решения сложных задач оперативного планирования и управления транспортом и сопряженными технологическими процессами в реальном масштабе времени. Они могут быть использованы для проектирования и совершенствования организации транспортно-технологических систем. Например, при вводе в строй на металлургическом предприятии нового сталеплавильного цеха или прокатного стана возникает задача спроектировать оптимальные транспортные коммуникации, связывающие новые технологические агрегаты со смежными, а также определить рациональное количество дополнительных транспортных единиц для перевозки новой продукции с минимальными потерями.

В системах автоматизированного проектирования такую задачу можно решить с помощью имитационных экспериментов на аналогич-

ной модели с организованной соответствующим образом новой информационной технологией. В информационное обеспечение имитационных экспериментов должны вводиться дополнительные количественные и качественные факторы, например, такие, как варианты расположения транспортных коммуникаций, увеличения парка транспортных средств и т. п. Для всех возможных сочетаний вариантов факторов проводятся серии имитационных экспериментов и исследуется поведение реакции. Реакцией (выходным показателем) исследований служит какой-либо обобщающий экономический показатель, например транспортно-технологические издержки функционирования системы. В результате экспериментальных исследований проектировщики могут обоснованно выбрать лучший вариант проекта развития транспортно-технологического комплекса.

В заключение отметим, что применение имитационного моделирования существенно дополняет новые информационные технологии, внедряемые в управление производственно-транспортным комплексом металлургического производства и повышает их эффективность.

СПИСОК ЛИТЕРАТУРЫ

1. Михалевич В. С., Скурихин В. И., Гриценко В. И., Каныгин Ю. М. Некоторые подходы к разработке концепции информации общества.— Киев, 1988.— 20 с.— (Препринт / АН УССР. Ин-т кибернетики, им. В. М. Глушкова; № 88—66).
2. Гриценко В. И., Панышин Б. Н. Информационная технология: вопросы развития и применения.— Киев : Наук. думка, 1988.— 272 с.
3. Михайлов А. И., Черный А. И., Гиляревский Р. С. Научные коммуникации и информатика.— М. : Наука, 1976.— 205 с.
4. Поздняков А. И. Информатика как комплексная научно-техническая дисциплина // *Вопр. философии*.— 1986.— № 5.— С. 62—70.
5. Михайлов А. И., Черный А. И., Гиляревский Р. С. Информатика // *БЭС*.— 3-е изд.— М. : Сов. энцикл., 1972.— Т. 10.— 348 с.
6. Глушков В. М. Основы безбумажной информатики.— М. : Наука, 1982.— 552 с.
7. Самарский А. А. Проблемы использования вычислительной техники и развитие информатики // *Вестн. АН СССР*.— 1985.— № 3.— С. 57—69.
8. Михалевич В. С., Каныгин Ю. М. Кибернетика в жизни общества.— Киев : Политиздат Украины, 1985.— 199 с.
9. Чешев В. В. Взаимосвязь инженерной деятельности и научного знания // *Вопр. философии*.— 1986.— № 6.— С. 53—60.
10. Моисеев Н. Н., Фролов И. Т. Высокое соприкосновение : Общество, человек и природа в век микроэлектроники, информатики и биотехнологии // *Вопр. философии*.— 1984.— № 9.— С. 24—41.
11. Михалевич В. С., Каныгин Ю. М., Гриценко В. И. Информатика : Общие положения.— Киев, 1983.— 45 с.— (Препринт / АН УССР. Ин-т кибернетики им. В. М. Глушкова; № 83—31).
12. Дородницын А. А. Информатика: предмет и задачи // *Вестн. АН СССР*.— 1983.— № 2.— С. 86—89.
13. Кочетков Т. Б. Автоматизация конторского труда в США (теория и практика «Офиса будущего»).— М. : Наука, 1985.— 224 с.
14. Самарский А. А. Математическое моделирование и вычислительный эксперимент // *Вестн. АН СССР*.— 1979.— № 5.— С. 38—49.
15. Сифоров В. И. Методологические вопросы науки об информации // *Вопр. философии*.— 1974.— № 7.— 109 с.
16. Глушков В. М. Автоматизированные системы управления сегодня и завтра.— М. : Мысль, 1976.— 64 с.
17. Шмудевич М. И. АСУ промышленным транспортом.— М. : Транспорт, 1974.— 410 с.
18. Венда В. Ф., Ломов Б. Ф. Взаимодействие человека с ЭВМ и проблемы познавательного прогресса // *Филос. вопр. техн. знания*.— М. : Наука, 1984.— С. 186—210.
19. Цикритзис Д., Лоховски Ф. Модели данных / Пер. с англ.— М. : Финансы и статистика, 1985.— 344 с.
20. Ульман Дж. Основы систем баз данных / Пер. с англ.— Там же.— 1983.— 334 с.
21. Дейт К. Введение в системы баз данных.— М. : Наука, 1980.— 464 с.
22. Codd E. F. A relational model of data for large shared data banks // *Commun. ACM*.— 1970.— N 6.— P. 377—387.

23. *Codd E. F.* Extending the databases relational model to capture more meaning // ACM Trans. on Database Syst.— 1979.— N 4.— P. 397—434.
24. *Carey M., Dewitt D., Richardson J., Shkita E.* Object and file management in the EXODUS extensible database system. // Proc. 12-th Int. conf. on VLDB.— Kyoto, 1986.— P. 91—100.
25. *Manola F., Orenstein J.* Toward a general spatial data model for an Object—Oriented DBMS // Ibid.— P. 328—335.
26. *Snodgrass R.* The temporal query language TQuel // ACM Trans. on Database Syst.— 1987.— N 2.— P. 247—299.
27. *Adiba M., Bui Quang N.* Historical multi-media databases. // Proc. 12-th Int. conf. on VLDB.— Kyoto, 1986.— P. 63—70
28. *Mc-Kenzie E.* Bibliography: Temporal databases // ACM SIGMOD.— 1986.— N 4.— P. 40—52.
29. *Snodgrass R., Ahn I.* Temporal databases // Computer.— 1986.— N 9, P. 35—42.
30. *Stonebraker M., Wong E., Kreps P., Held G.* The design and implementation of INGRES // ACM Trans. on Database Syst.— 1976.— N 3.— P. 189—222
31. *IBM.* SQL/Data-System, concepts and facilities // Tech. Rep. GH24-5013-0, IBM.— 1981.— 84 p.
32. *Ahn I., Snodgrass R.* Performance evaluation of a temporal database management system // Proc. ACM SIGMOD Int. conf. on Management of Data.— Washington, 1986.— P. 96—107.
33. *Chang S. K., Kunit T. L.* Pictorial data-base system // Computer.— 1981.— N 11.— P. 13—21.
34. *Александров В. В., Горский Н. Д.* Базы видеоданных: проблемы и перспективы (материалы по программному обеспечению).— Л.: Изд-во АН СССР.— 1985.— 72 с.
35. *Александров В. В., Горский Н. Д.* Представление и обработка изображений: Рекурсивный подход.— Л.: Наука, 1985.— 192 с.
36. *ISO/TC97/SC5/WG2.* Draft international standard ISO/DIS, Version 7.0, GKS.— 1982.— N 117.— 240 p.
37. *Williams R., Giddings G.* A pictorial building system // IEEE Trans. Software Eng.— 1976.— N 1.— P. 62—66.
38. *Chang S. K., Lin B. S., Walser R.* A generalized zooming technique for pictorial database system.— In: Pict. Inform. Syst Springer-Verlag.— Berlin, 1980.— P. 257—287.
39. *Becerril J., Casajuana R., Lorie R.* GYSR—a relational database interface for graphics.— In: Data Base Tech. for Pict. Appl.— Ibid.— P. 158—182.
40. *Chang N. S., Fu K. S.* Picture query by pictorial example // IEEE Trans. Software Eng.— 1980.— N 6.— P. 519—524.
41. *Гилой В.* Интерактивная машинная графика: Структуры данных, алгоритмы, языки.— М.: Мир, 1981.— 384 с.
42. *Ramakrishnan R., Zilberschatz A.* The MR-diagram — a model for conceptual database design // Proc. 11-th Int. conf. on VLDB.— Stockgolm, 1985.— P. 376—393.
43. *Lecluse C., Richard P., Velez F. O.* an object-oriented data model // Proc. Int. conf. on Extend: Database Tech.— Venice, 1988.— p. 556—562.
44. *Sernadas A., Sernadas C.* Object-oriented specification of database: an algebraic approach // Proc. 13-th Int. conf. on VLDB.— Brighton, 1987.— P. 107—116.
45. *Калиниченко Л. А.* Методы и средства интеграции неоднородных баз данных.— М.: Наука, 1983.— 424 с.
46. *Valduriez P., Khoshafian S., Copeland G.* Implementation techniques of Complex Objects // Proc. 12-th Int. conf. on VLDB.— Kyoto, 1986.— P. 101—110.
47. *Chou H.— T., Kim W.* A unifying framework for version control in a CAD environment // Ibid.— P. 336—344.
48. *Klahold P., Schlageter G., Wilkes W.* A general model for version management in databases // Ibid.— P. 319—327.
49. *Beech D.* A foundation for evolution from relational to object database // Proc. Int. conf. on Extend. Data Base Tech.— Venice, 1988.— p. 251—270.
50. *Buchman A.* A architecture and data model for CAD databases // Proc. 11-th Int. conf. on VLDB.— Stockgolm, 1985.— P. 174—182.

51. *Богемский В. А.* Концепции построения базы данных, отображающей динамику предметной области, для систем управления железнодорожным транспортом // Управ. системы и машины.— 1988.— № 2.— С. 77—82.
52. *Гриценко В. И., Богемский В. А., Труш А. Н.* Модель базы данных для графических приложений // Банки данных. Секция 1: Теоретич. основы постро. баз данных: Тез. докл. Всесоюз. науч.-техн. конф. (г. Ташкент, 1983).— Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1983.— С. 50—52.
53. *Гриценко В. И., Богемский В. А.* Графическая база данных для представления и обработки схем транспортных объектов // Управ. системы и машины.— 1984.— № 2.— С. 85—89.
54. *Нахитин А. С., Скриган Н. И.* Иерархические графовые структуры и программные средства их обработки // Там же.— 1982.— № 2.— С. 31—34.
55. *Богемский В. А., Труш А. Н.* Оптимизация формирования и вывода графических изображений // Там же.— 1984.— № 5.— С. 74—78.
56. *Уокер Б. С., Гурд Дж. Р., Дроник Е. А.* Интерактивная машинная графика.— М.: Машиностроение, 1980.— 171 с.
57. *Вискирский В. А., Додонов С. Б., Похвалинский А. Б.* Основы построения диалоговых интегрированных САПР // Упр. системы и машины.— 1988.— № 6. С. 77—80.
58. *Рябов А. Н.* Использование одной особенности графопостроителя для повышения скорости регистрации // Там же.— 1982.— № 5.— С. 25—26.
59. *Leipala T., Nevalainen O.* A plotting sequencing system // Comput.— J.— 1979.— N 4.— P. 313—316.
60. *Edmonds J.* Maximum matching and polyhedron with 0,1 Vertices // J. Res. Nat Bur Stand.— 1965.— N 12.— P. 123—130.
61. *Майника Э.* Алгоритмы оптимизации на сетях и графах.— М.: Мир, 1981.— 323 с.
62. *Кристофидес Н.* Теория графов: Алгоритмический подход.— М.: Мир, 1978.— 432 с.
63. *Карзанов А. В.* Экономные реализации алгоритмов Эдмондса нахождения паросочетаний максимальной мощности и максимального веса // Исследования по дискретной оптимизации.— М.: Наука, 1976.— С. 306—327.
64. *Вичес С. А.* Экономный алгоритм построения кратчайшего обхода одноцветного связанного чертежа // Автоматика и телемеханика.— 1982.— № 12.— С. 85—96.
65. *Jri M., Mulola K., Matsui S.* Linear-time approximation algorithms for finding the minimum-weight perfect matching on a plane // Inform. Proc. Letters.— 1981.— N 4.— P. 206—209.
66. *Флинт Д.* Локальные сети ЭВМ: архитектура, принципы построения, реализация / Пер. с англ.— М.: Финансы и статистика, 1986.— 359 с.
67. *A local-area communication network based on a reliable token—ring system / W. Bux, F. Closs, P. Jansen, K. Kummerle, H. Muller // Proc. IFIP Int. Symp. Florence.— 1982.— P. 69—82.*
68. *Банк данных подсистемы управления транспортом отраслевой АСУ Минчермета УССР / В. И. Гриценко, В. А. Богемский, Т. М. Кучеренко, Н. Е. Павленко, А. Н. Труш // Управ. системы и машины.— 1986.— № 5.— С. 62—65.*
69. *Богемский В. А., Труш А. Н.* Об автоматизации функций администратора банка данных по контролю технологии обработки информации // Средства информационного обеспечения научных исследований.— Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1984.— С. 54—57.
70. *Тиори Т., Фрай Дж.* Проектирование структур баз данных: Пер. с англ.— М.: Мир, 1985.— С. 287 и 320.
71. *Хаббард Дж.* Автоматизированное проектирование баз данных / Пер. с англ.— М.: Мир, 1984.— 296 с.
72. *ANSI/X3/SPARC study group on data base management systems, Interim report // Bul. of ACM SIGMOD.— 1975.— N 2.— 77 p.*
73. *Богемский В. А., Павленко Н. Е., Труш А. Н.* Программный комплекс для ведения банка данных подсистемы «Транспорт» ОАСУ Минчермета УССР // Информационные и математические модели планирования и управления транспортом.— Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1983.— С. 42—50.

74. *Поспелов Г. С.* Искусственный интеллект — основа новой информационной технологии.— М.: Наука, 1988.— 280 с.
75. *Экспертные системы: Принципы работы и примеры / Пер. с англ. под ред. Р. Форсайта.*— М.: Радио и связь, 1987.— 224 с.
76. *Полов Э. В.* Экспертные системы: решение неформализованных задач в диалоге с ЭВМ.— М.: Наука, 1987.— 288 с.
77. *Хейес—Ротт Ф., Уотермат Д., Ленат Д.* Построение экспертных систем / Пер. с англ.— М.: Мир, 1987.— 430 с.
78. *Богемский В. А., Кучеренко Т. М.* Особенности реализации комплекса задач планирования ремонтов подвижного состава в подсистеме «Транспорт» ОАСУ МЧМ УССР // Экономико-математические модели и технологии обработки данных в планировании и управлении транспортом.— Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1985.— С. 43—48.
79. *Котов В. Е.* Сети Петри.— М.: Наука, 1984.— 160 с.
80. *Питерсон Дж.* Теория сетей Петри и моделирование систем.— М.: Мир, 1984.
81. *Gricenko V. I., Bogemskij V. A., Lara A. P., Truchin S. L.* Construction principles of problem-oriented simulation system for optimization of the life cycles of large-scale transport-technological complexes // Proc. 5-th IFAC (IFIP) (IFORS Int. conf. on Control in Transportation System).— Wien, 1986.— P. 41—43.
82. *Гриценко В. И., Богемский В. А., Соловицкий И. А., Труш А. Н.* Диалоговая система проектирования оптимальной организации транспортного обслуживания промышленных предприятий // Управ. системы и машины.— 1990.— № 2.— С. 75—89.
83. *Методические указания по выбору форм транспортного обслуживания предприятий и организаций отраслей народного хозяйства промышленным железнодорожным транспортом при взаимодействии его с магистральным: Отчет о НИР Украин. филиала ин-та комплекси. транспортн. пробл. Госплана СССР, 1982.— Инв. № 02860068719 УкрНИИИИТІ Госплана УССР.*
84. *Михалевич В. С., Волкович В. Л.* Вычислительные методы исследования и проектирования сложных систем.— М.: Наука, 1982.— 288 с.
85. *Бакаев А. А., Костина Н. И., Яровицкий Н. В.* Имитационные модели в экономике.— Киев: Наук. думка, 1978.— 307 с.
86. *Багриновский К. А., Егорова Е.* Имитационная система в планировании экономических объектов.— М.: Наука, 1980.— 237 с.
87. *Лобзов А. В.* Системный подход к проблеме согласования экономических интересов в рамках одной имитационной системы.— М.: Центр. эконом.- мат. ин-т АН СССР, 1987.— 176 с.
88. *Castek D. M., Lichsfeld R. A.* How simulation techniques can support tactical and operational decisions // Proc. 2-nd Int. conf. Simul. Manuf.— Chicago, 1986.— P. 169—176.
89. *Perkins J. D.* Survey of existing systems for the dynamic simulation on industrial processes // Mod. Identif. and contr.— 1986.— N 2.— P. 71—81.
90. *Берлин А. А.* Использование имитационного моделирования при анализе резервов производства // Имитационное моделирование в анализе и синтезе социально-экономических систем.— М.: Моск. ин-т управления им. С. Орджоникидзе, 1987.— С. 11—16.
91. *Ahadial Nasrollah.* Solving inventory problems by simulation // J. Syst. Manag.— 1986.— N 9.— P. 29—35.
92. *Kelth A., Porter K.* Resent development in simulation techniques for assessing levels of automation // Proc. 6-th Inf. conf. Autom. Warehouse.— Stockholm, 1985.— P. 69—76.
93. *Golden B. L., Bodin L.* A general simulation program for factory transportation — application to AGVS/ASRS // Ibid.— P. 57—68.
94. *Golden B. L., Bodin L.* Microcomputer based vehicle routing and sheduling software // Comput. and Oper. Reseach.— 1986.— N 2—3.— P. 277—285.
95. *Shivnan J., Browne J.* All based Simulation of advanced manufacturing systems // Proc. 2-nd Int. conf. Simul. Manuf.— Chicago, 1986.— P. 23—33.
96. *Lee R. M., Miller L. W.* A logic programming framework for planning and simulation // Decis. support Syst.— 1986.— N 1.— P. 15—25.
97. *Панченко А. А., Нападайло В. А.* Имитационное моделирование производственно-экономических систем.— Днепропетровск: ДГУ, 1985.— 77 с.

98. Шеннон Р. Имитационное моделирование систем — искусство и наука.— М. : Мир, 1978.— 413 с.
99. Перминов С. В. Имитационное моделирование процессов управления в экономике.— Новосибирск : Наука, 1981.— 212 с.
100. Discens I. H., Fraiser G. D. Linear programming in corporate models // Times college on simulation on corporate Simulation models.— Washington : Univer. press, 1979.— P. 145—150.
101. Бусленко Н. П., Калашников В. В., Коваленко И. Н. Лекции по теории сложных систем.— М. : Сов. радио, 1973.— 440 с.
102. Бакаев А. А., Костина Н. И., Яровицкий Н. В. Имитационные модели в экономике.— Киев : Наук. думка, 1978.— 297 с.
103. Моисеев Н. Н. Математические задачи системного анализа.— М. : Наука, 1981.— 487 с.
104. Бусленко В. Н. Автоматизация имитационного моделирования сложных систем.— М. : Наука, 1977.— 143 с.
105. Имитационное моделирование производственных систем / Под ред. А. А. Вавилова.— М. : Машиностроение, 1983.— 234 с.
106. Нейлор Т. Машинные имитационные эксперименты с моделями экономических систем.— М. : Мир, 1975.— 500 с.
107. Форрестер Дж. Основы кибернетики предприятий.— М. : Прогресс, 1971.— 380 с.
108. Мартин Ф. Моделирование на вычислительных машинах.— М. : Сов. радио, 1972.— 284 с.
109. Гриценко В. И., Панченко А. А., Лана А. И. Проблемно-ориентированное моделирование производственно-транспортных систем.— Киев : Наук. думка, 1987.— 158 с.
110. Гриценко В. И., Панченко А. А. Модели и алгоритмы решения транспортно-технологических задач в АСУ карьерным транспортом.— Киев : ИК АН УССР, 1982.— 68 с.
111. Гриценко В. И., Панченко А. А., Назаренко Н. А. Имитационные модели в условиях новой информационной технологии управления производственно-транспортными комплексами.— Киев : Ин-т кибернетики им. В. М. Глушкова АН УССР, 1986.— 46 с.
112. Нападайко В. А., Панченко А. А. Вариантная система планирования агломерационного производства в условиях ОАСУ // Анализ и моделирование процессов управления в экономике.— Днепропетровск : ДГУ, 1981.— С. 26—32.
113. Дородницын А. А. Проблемы математического моделирования в описательных науках // Кибернетика.— 1983.— № 4.— С. 6—10.
114. Бусленко Н. П. Моделирование сложных систем.— М. : Наука, 1978.— 340 с.
115. Киндлер Е. Языки моделирования.— М. : Энергоатомиздат, 1985.— 288 с.
116. Шрайбер Т. Дж. Моделирование на GPSS / Пер. с англ. под ред. М. А. Файберг.— М. : Машиностроение, 1980.— 592 с.
117. Иващилов Ю. П., Лотов А. В. Математические модели в экономике.— М. : Наука, 1979.— 302 с.

ОГЛАВЛЕНИЕ

Предисловие	5
Список условных обозначений	7
Введение	8
Глава 1. Информатика и информационная технология	12
1.1. Информатика как наука	12
1.2. Прикладная информатика и новая информационная технология	20
1.3. Методы новой информационной технологии в задачах управления промышленным транспортом	28
Глава 2. Базы данных и основные направления их развития	32
2.1. Данные, модели данных, базы данных	32
2.2. Временные базы данных	39
2.3. Базы данных в графических информационных системах	44
2.4. Объектные базы данных	51
Глава 3. Специфика построения моделей данных в системах управления транспортом	55
3.1. Отображение динамики объектов предметной области	55
3.1.1. Структуры данных	60
3.1.2. Ограничения целостности	66
3.1.3. Принципы манипулирования данными	69
3.2. Представление и обработка схем транспортных объектов	77
3.2.1. Структуры данных	78
3.2.2. Ограничения целостности	83
3.2.3. Принципы манипулирования данными	83
3.2.4. Интерактивная обработка моделей и изображений	89
3.3. Оптимизация вывода графических изображений	91
Глава 4. Информационные технологии в решении задач управления транспортом	98
4.1. Локальные сети — техническая основа построения систем управления промышленным транспортом	98
4.1.1. Определения, принципы построения, требования	98
4.1.2. Классификация локальных сетей	100
4.1.3. Межсетевое взаимодействие	104
4.1.4. Модель локальной сети для управления транспортом на уровне производственного объединения и отрасли	110
4.2. Банки данных — основная форма информационного обеспечения систем управления	116
4.3. Использование знаний в принятии решений	116
4.3.1. Управление решением сложных комплексов задач	118

4.3.2. Принципы построения экспертной системы реального времени для управления перевозками	124
4.4. Диалоговая технология автоматизированного проектирования	126
Глава 5. Имитационное моделирование производственно-транспортных систем	135
5.1. Имитационное моделирование в новых информационных технологиях управления промышленным транспортом	135
5.2. Основные этапы имитационного моделирования	137
5.3. Изучение моделируемой системы и формулировка проблемы	139
5.4. Математическая модель	144
5.5. Разработка моделирующих алгоритмов	149
5.6. Понятие о языках моделирования и составлении имитационных программ	155
5.7. Проверка пригодности имитационной модели и обработка результатов экспериментов	162
Глава 6. Имитационные модели в новых информационных технологиях управления производственно-транспортными системами	168
6.1. Особенности моделирования промышленного транспорта предприятий металлургического комплекса	168
6.2. Модели управления запасами и транспортировкой железорудного сырья	174
6.3. Результаты имитационных экспериментов с моделью транспортировки железорудного сырья	180
6.4. Моделирование межцеховых перевозок	185
Список литературы	191

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ НА ТРАНСПОРТЕ

Научное издание

Гриценко Владимир Ильич
Богемский Виктор Александрович
Панченко Александр Андреевич

ПРОМЫШЛЕННЫЙ ТРАНСПОРТ

Художественный редактор *И. П. Антолюк*
Технический редактор *И. А. Ратнер*
Корректоры *С. А. Доценко, Т. Я. Чорная, Л. М. Тищенко*

ИБ № 410492

Сдано в набор 21.04.90. Подп. в печ. 14.09.90. Формат 84×108/32. Бум. тип. № 1.
Лит. гарн. Вис. печ. Усл. печ. л. 12,5. Усл. кр.-отг. 12,5. Уч.-изд. л. 14,16.
Тираж 700 экз. Заказ 0—1483. Цена 3 р. 20 к.

Издательство «Наукова думка». 252601 Киев 4, ул. Репина, 3.

Отпечатана с матриц Головного предприятия республиканского производственного объединения «Полиграфкинига». 252057 Киев, ул. Довженко, 3 в Киевской книжной типографии научной книги. 252004 Киев 4, ул. Репина, 4. Зак. 0-750.