

Министерство образования Республики Беларусь

Учреждение образования  
«Гомельский государственный технический  
университет имени П. О. Сухого»

Кафедра «Сельскохозяйственные машины»

**В. Б. Попов**

**СИСТЕМЫ АВТОМАТИЗИРОВАННОГО  
ПРОЕКТИРОВАНИЯ УЗЛОВ  
И АГРЕГАТОВ МАШИН**

**КУРС ЛЕКЦИЙ**

**для студентов специальности 1-36 12 01**

**«Проектирование и производство  
сельскохозяйственной техники»**

**Электронный аналог печатного издания**

**Гомель 2010**

УДК 658.512.011.56(075.8)  
ББК 30.2-5-05я73  
П58

*Рекомендовано к изданию научно-методическим советом  
механико-технологического факультета ГГТУ им. П. О. Сухого  
(протокол № 3 от 23.06.2009 г.)*

Рецензент: канд. техн. наук, доц. каф. «Информационные технологии»  
ГГТУ им. П. О. Сухого *В. И. Мисюткин*

**Попов, В. Б.**  
С12 Системы автоматизированного проектирования узлов и агрегатов машин : курс лекций для студентов специальности 1-36 12 01 «Проектирование и производство сельскохозяйственной техники» / В. Б. Попов. – Гомель : ГГТУ им. П. О. Сухого, 2010. – 187 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <http://lib.gstu.local>. – Загл. с титул. экрана.

ISBN 978-985-420-958-6.

Изложен ряд специализированных тем: этапы процесса проектирования сельскохозяйственных машин; принципы формирования и структура САПР самоходной уборочной машины; проектирующие подсистемы рабочих органов и агрегатов машин; автоматизация конструирования мобильных машин на базе геометрического моделирования деталей и узлов с использованием программных комплексов AutoCAD и КОМПАС, конструкторских баз данных и СУБД.

Для студентов специальности 1-36 12 01 «Проектирование и производство сельскохозяйственной техники».

**УДК 658.512.011.56  
ББК 30.2-5-05**

**ISBN 978-985-420-958-6**

© Попов В. Б., 2010  
© Учреждение образования «Гомельский  
государственный технический университет  
имени П. О. Сухого», 2010

## ВВЕДЕНИЕ

Системы автоматизированного проектирования (САПР) занимают особое место среди информационных технологий.

Во-первых, САПР деталей и узлов машин – синтетическая дисциплина, ее составными частями являются многие другие современные информационные технологии. Так, техническое обеспечение САПР основано на использовании вычислительных сетей и телекоммуникационных технологий, в САПР используются персональные компьютеры и рабочие станции. Математическое обеспечение САПР отличается богатством и разнообразием используемых методов вычислительной математики, статистики, математического программирования, искусственного интеллекта. Программные комплексы САПР относятся к числу наиболее сложных современных программных систем, основанных на операционных системах Unix, Windows-95/NT, языках программирования C, C++, Java и других современных CASE-технологиях, реляционных и объектно-ориентированных системах управления базами данных (СУБД), стандартах открытых систем и обмена данными в компьютерных средах.

Во-вторых, знание основ автоматизации проектирования и умение работать со средствами САПР требуется практически любому инженеру-разработчику. Компьютерами насыщены проектные подразделения, конструкторские бюро и офисы. Работа конструктора за обычным кульманом, расчеты с помощью логарифмической линейки или оформление отчета на пишущей машинке стали анахронизмом. Предприятия, ведущие разработки без САПР или лишь с малой степенью их использования, оказываются неконкурентоспособными как из-за больших материальных и временных затрат на проектирование, так и из-за невысокого качества проектов.

Появление первых программ для автоматизации проектирования за рубежом и в Белоруссии относится к началу 60-х годов. Тогда были созданы программы для решения задач строительной механики, анализа электронных схем, проектирования печатных плат. Дальнейшее развитие САПР шло по пути создания аппаратных и программных средств машинной графики, повышения вычислительной эффективности программ моделирования и анализа, расширения областей

применения САПР, упрощения пользовательского интерфейса, внедрения в САПР элементов искусственного интеллекта.

К настоящему времени создано большое число программно-методических комплексов для САПР с различными степенью специализации и прикладной ориентацией. В результате автоматизация проектирования стала необходимой составной частью подготовки инженеров разных специальностей; инженер, не владеющий знаниями и не умеющий работать в САПР, не может считаться полноценным специалистом.

Подготовка инженеров разных специальностей в области САПР включает базовую и специальную компоненты. Наиболее общие положения, модели и методики автоматизированного проектирования входят в программу данного курса, посвященного основам САПР, более детальное изучение тех методов и программ, которые специфичны для конкретных специальностей, предусматривается в специальных дисциплинах.

# Глава 1. ОСНОВЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

## *Тема 1. Системный подход к проектированию*

**Понятие инженерного проектирования.** Проектирование технического объекта – создание, преобразование и представление в принятой форме образа этого еще не существующего объекта. Образ объекта или его составных частей может создаваться в воображении человека в результате творческого процесса или генерироваться в соответствии с некоторыми алгоритмами в процессе взаимодействия человека и ЭВМ. В любом случае инженерное проектирование начинается при наличии выраженной потребности общества в некоторых технических объектах, которыми могут быть объекты строительства, промышленные изделия или процессы. Проектирование включает в себя разработку технического предложения и (или) технического задания (ТЗ), отражающих эти потребности, и реализацию ТЗ в виде проектной документации.

Обычно ТЗ представляют в виде некоторых документов и оно является *исходным (первичным) описанием объекта*. Результатом проектирования, как правило, служит полный комплект документации, содержащий достаточные сведения для изготовления объекта в заданных условиях. Эта документация и есть проект, точнее *окончательное описание объекта*. Более коротко, проектирование – процесс, заключающийся в получении и преобразовании исходного описания объекта в окончательное описание на основе выполнения комплекса работ исследовательского, расчетного и конструкторского характера.

Преобразование исходного описания в окончательное порождает ряд промежуточных описаний, подводящих итоги решения некоторых задач и используемых для обсуждения и принятия проектных решений для окончания или продолжения проектирования.

Проектирование, при котором все проектные решения или их часть получают путем взаимодействия человека и ЭВМ, называют автоматизированным, в отличие от ручного или автоматического (без участия человека на промежуточных этапах). Система, реализующая

автоматизированное проектирование, представляет собой систему автоматизированного проектирования (в англоязычном написании CAD System – Computer Aided Design System). Автоматическое проектирование возможно лишь в отдельных частных случаях для сравнительно несложных объектов. Превалирующим в настоящее время является автоматизированное проектирование.

Проектирование сложных объектов основано на применении идей и принципов, изложенных в ряде теорий и подходов. Наиболее общим подходом является системный подход, идеями которого пронизаны различные методики проектирования сложных систем.

**Принципы системного подхода.** Основные идеи и принципы проектирования сложных систем выражены в системном подходе. Для специалиста в области системотехники они являются очевидными и естественными, однако их соблюдение и реализация зачастую сопряжены с определенными трудностями, обусловливаемыми особенностями проектирования. Как и большинство взрослых образованных людей, правильно использующих родной язык без привлечения правил грамматики, инженеры используют системный подход без обращения к пособиям по системному анализу. Однако интуитивный подход без применения правил системного анализа может оказаться недостаточным для решения все более усложняющихся задач инженерной деятельности.

Основной общий принцип системного подхода заключается в рассмотрении частей явления или сложной системы с учетом их взаимодействия. Системный подход включает в себя выявление структуры системы, типизацию связей, определение атрибутов, анализ влияния внешней среды.

Системный подход рассматривают как направление научного познания и социальной политики. Он является базой для обобщающей дисциплины «Теория систем» (другое используемое название – «Системный анализ»). Теория систем – дисциплина, в которой конкретизируются положения системного подхода; она посвящена исследованию и проектированию сложных экономических, социальных, технических систем, чаще всего слабоструктурированных. Характерными примерами таких систем являются производственные системы. При проектировании систем цели достигаются в многошаговых процессах принятия решений. Методы принятия решений часто выделяют в самостоятельную дисциплину, называемую «Теория принятия решений».

В технике дисциплину, в которой исследуются сложные технические системы, их проектирование и аналогичную теорию систем, чаще называют системотехникой. Предметом системотехники являются, во-первых, организация процесса создания, использования и развития технических систем, во-вторых, методы и принципы их проектирования и исследования. В системотехнике важно уметь сформулировать цели системы и организовать ее рассмотрение с позиций поставленных целей. Тогда можно отбросить лишние и малозначимые части при проектировании и моделировании, перейти к постановке оптимизационных задач.

Системы автоматизированного проектирования и управления относятся к числу наиболее сложных современных искусственных систем. Их проектирование и сопровождение невозможны без системного подхода. Поэтому идеи и положения системотехники входят составной частью в дисциплины, посвященные изучению современных автоматизированных систем и технологий их применения. Интерпретация и конкретизация системного подхода имеют место в ряде известных подходов с другими названиями, которые также можно рассматривать как компоненты системотехники. Таковы структурный, блочно-иерархический, объектно-ориентированный подходы.

При *структурном подходе*, как разновидности системного, требуется синтезировать варианты системы из компонентов (блоков) и оценивать варианты при их частичном переборе с предварительным прогнозированием характеристик компонентов.

*Блочно-иерархический подход* к проектированию использует идеи декомпозиции сложных описаний объектов и соответственно средств их создания на иерархические уровни и аспекты, вводит понятие стиля проектирования (восходящее и нисходящее), устанавливает связь между параметрами соседних иерархических уровней.

Ряд важных структурных принципов, используемых при разработке информационных систем и прежде всего их программного обеспечения (ПО), выражен в объектно-ориентированном подходе к проектированию (ООП). Такой подход имеет следующие преимущества в решении проблем управления сложностью и интеграции ПО: 1) вносит в модели приложений большую структурную определенность, распределяя представленные в приложении данные и процедуры между классами объектов; 2) сокращает объем спецификаций, благодаря введению в описания иерархии объектов и отношений наследования между свойствами объектов разных уровней иерархии;

3) уменьшает вероятность искажения данных вследствие ошибочных действий за счет ограничения доступа к определенным категориям данных в объектах. Описание в каждом классе объектов допустимых обращений к ним и принятых форматов сообщений облегчает согласование и интеграцию ПО.

Для всех подходов к проектированию сложных систем характерны также следующие особенности:

1. *Структуризация* процесса проектирования, выражаемая декомпозицией проектных задач и документации, выделением стадий, этапов, проектных процедур. Эта структуризация является сущностью блочно-иерархического подхода к проектированию.

2. *Итерационный* характер проектирования.

3. *Типизация* и *унификация* проектных решений и средств проектирования.

**Основные понятия системотехники.** В теории систем и системотехнике введен ряд терминов, среди них к базовым нужно отнести следующие понятия.

*Система* – множество элементов, находящихся в отношениях и связях между собой.

*Элемент* – такая часть системы, представление о которой целесообразно подвергать при проектировании дальнейшему членению.

*Сложная система* – система, характеризуемая большим числом элементов и, что наиболее важно, большим числом взаимосвязей элементов. Сложность системы определяется также видом взаимосвязей элементов, свойствами *целенаправленности, целостности, членимости, иерархичности, многоаспектности*. Очевидно, что современные автоматизированные информационные системы и, в частности, системы автоматизированного проектирования, являются сложными в силу наличия у них перечисленных свойств и признаков.

*Подсистема* – часть системы (подмножество элементов и их взаимосвязей), которая имеет свойства системы.

*Надсистема* – система, по отношению к которой рассматриваемая система является подсистемой.

*Структура* – отображение совокупности элементов системы и их взаимосвязей; понятие структуры отличается от понятия самой системы также тем, что при описании структуры принимают во внимание лишь типы элементов и связей без конкретизации значений их параметров.

*Параметр* – величина, выражающая свойство или системы, или ее части, или влияющей на систему среды. Обычно в моделях систем в качестве параметров рассматривают величины, не изменяющиеся в процессе исследования системы. Параметры подразделяют на *внешние, внутренние и выходные*, выражающие свойства элементов системы, самой системы, внешней среды соответственно. Векторы внутренних, выходных и внешних параметров далее обозначаются  $X = (x_1, x_2, \dots, x_n)$ ,  $Y = (y_1, y_2, \dots, y_k)$ ,  $Q = (q_1, q_2, \dots, q_k)$  соответственно.

*Фазовая переменная* – величина, характеризующая энергетическое или информационное наполнение элемента или подсистемы.

*Состояние* – совокупность значений фазовых переменных, зафиксированных в одной временной точке процесса функционирования.

*Поведение (динамика) системы* – изменение состояния системы в процессе функционирования.

*Система без последействия* – ее поведение при  $t > t_0$  определяется заданием состояния в момент  $t_0$  и вектором внешних воздействий  $Q(t)$ . В системах с последействием, кроме того, нужно знать предысторию поведения, т. е. состояния системы в моменты, предшествующие  $t_0$ .

*Вектор переменных  $V$ , характеризующих состояние* (вектор переменных состояния) – избыточное множество фазовых переменных, задание значений которых в некоторый момент времени полностью определяет поведение системы в дальнейшем.

*Пространство состояний* – множество возможных значений вектора переменных состояния.

*Фазовая траектория* – представление процесса (зависимости  $V(t)$ ) в виде последовательности точек в пространстве состояний.

К характеристикам сложных систем, как сказано выше, часто относят следующие понятия.

*Целенаправленность* – свойство искусственной системы, выражающее назначение системы. Это свойство необходимо для оценки эффективности вариантов системы.

*Целостность* – свойство системы, характеризующее взаимосвязанность элементов и наличие зависимости выходных параметров от параметров элементов, при этом большинство выходных параметров не является простым повторением или суммой параметров элементов.

*Иерархичность* – свойство сложной системы, выражающее возможность и целесообразность ее иерархического описания, т. е. пред-

ставления в виде нескольких уровней, между компонентами которых имеются отношения целое-часть.

Составными частями системотехники являются следующие основные разделы: иерархическая структура систем, организация их проектирования; анализ и моделирование систем; синтез и оптимизация систем.

Моделирование имеет две четко различимые задачи: 1 – создание моделей сложных систем (в англоязычном написании – *modeling*); 2 – анализ свойств систем на основе исследования их моделей (*simulation*).

Синтез также подразделяют на две задачи: 1 – синтез структуры проектируемых систем (*структурный синтез*); 2 – выбор численных значений параметров элементов систем (параметрический синтез). Эти задачи относятся к области принятия проектных решений.

Моделирование и оптимизацию желательно выполнять с учетом статистической природы систем. Детерминированность – лишь частный случай. При проектировании характерны нехватка достоверных исходных данных, неопределенность условий принятия решений. Учет статистического характера данных при моделировании в значительной мере основан на методе статистических испытаний (методе Монте-Карло), а принятие решений – на использовании нечетких множеств, экспертных систем, эволюционных вычислений.

*Пример 1.* Компьютер является сложной системой в силу наличия у него большого числа элементов, разнообразных связей между элементами и подсистемами, свойств целенаправленности, целостности, иерархичности. К подсистемам компьютера относятся процессор, оперативная память, кэш-память, шины, устройства ввода–вывода.

В качестве надсистемы могут выступать вычислительная сеть, автоматизированная и (или) организационная система, к которым принадлежит компьютер. Внутренние параметры – времена выполнения арифметических операций, чтения (записи) в накопителях, пропускная способность шин и др. Выходные параметры – производительность компьютера, емкость оперативной и внешней памяти, себестоимость, время наработки на отказ и др. Внешние параметры – напряжение питания сети и его стабильность, температура окружающей среды и др.

*Пример 2.* Для ДВС подсистемами являются коленчатый вал, механизм газораспределения, поршневая группа, система смазки и охлаждения. Внутренние параметры – число цилиндров, объем каме-

ры сгорания и др. Выходные параметры – мощность двигателя, КПД, расход топлива и др. Внешние параметры – характеристики топлива, температура воздуха, нагрузка на выходном валу.

## **Тема 2. Структура процесса проектирования**

**Иерархическая структура проектных спецификаций и иерархические уровни проектирования.** При использовании блочно-иерархического подхода к проектированию представления о проектируемой системе расчлениают на *иерархические уровни*. На верхнем уровне используют наименее детализированное представление, отражающее только самые общие черты и особенности проектируемой системы. На следующих уровнях степень подробности описания возрастает, при этом рассматривают уже отдельные блоки системы, но с учетом воздействий на каждый из них его соседей. Такой подход позволяет на каждом иерархическом уровне формулировать задачи приемлемой сложности, поддающиеся решению с помощью имеющихся средств проектирования. Разбиение на уровни должно быть таким, чтобы документация на блок любого уровня была обзрима и воспринимается одним человеком.

Другими словами, блочно-иерархический подход есть декомпозиционный подход (его можно назвать также диакоптическим), который основан на разбиении сложной задачи большой размерности на последовательно и (или) параллельно решаемые группы задач малой размерности, что существенно сокращает требования к используемым вычислительным ресурсам или время решения задач.

Можно говорить не только об иерархических уровнях спецификаций, но и об иерархических уровнях проектирования, понимая под каждым из них совокупность спецификаций некоторого иерархического уровня совместно с постановками задач, методами получения описаний и решения возникающих проектных задач.

Список иерархических уровней в каждом приложении может быть специфичным, но для большинства приложений характерно следующее наиболее крупное выделение уровней:

– *системный уровень*, на котором решают наиболее общие задачи проектирования систем, машин и процессов; результаты проектирования представляют в виде структурных схем, генеральных планов, схем размещения оборудования, диаграмм потоков данных и т. п.;

– *макроуровень*, на котором проектируют отдельные устройства, узлы машин и приборов; результаты представляют в виде функциональных, принципиальных и кинематических схем, сборочных чертежей и т. п.;

– *микроуровень*, на котором проектируют отдельные детали и элементы машин и приборов.

В каждом приложении число выделяемых уровней и их наименования могут быть различными. Так, в радиоэлектронике микроуровень часто называют компонентным, макроуровень – схемотехническим. Между схемотехническим и системным уровнями вводят уровень, называемый функционально-логическим. В вычислительной технике системный уровень подразделяют на уровни проектирования ЭВМ (вычислительных систем) и вычислительных сетей. В машиностроении имеются уровни деталей, узлов, машин, комплексов.

В зависимости от последовательности решения задач иерархических уровней различают нисходящее, восходящее и смешанное проектирование (стили проектирования). Последовательность решения задач от нижних уровней к верхним характеризует *восходящее* проектирование, обратная последовательность приводит к *нисходящему* проектированию, в *смешанном* стиле имеются элементы как восходящего, так и нисходящего проектирования. В большинстве случаев для сложных систем предпочитают нисходящее проектирование. Отметим однако, что при наличии заранее спроектированных составных блоков (устройств) можно говорить о смешанном проектировании.

Неопределенность и нечеткость исходных данных при нисходящем проектировании (так как еще не спроектированы компоненты) или исходных требований при восходящем проектировании (поскольку ТЗ имеется на всю систему, а не на ее части) обуславливают необходимость прогнозирования недостающих данных с последующим их уточнением, т. е. последовательного приближения к окончательному решению (итерационность проектирования).

Наряду с декомпозицией описаний на иерархические уровни применяют разделение представлений о проектируемых объектах на аспекты.

*Аспект описания (страта)* – описание системы или ее части с некоторой оговоренной точки зрения, определяемой функциональными, физическими или иного типа отношениями между свойствами и элементами.

Различают аспекты функциональный, информационный, структурный и поведенческий (процессный). *Функциональное* описание относят к функциям системы и чаще всего представляют его функциональными схемами. *Информационное* описание включает в себя основные понятия предметной области (сущности), словесное пояснение или числовые значения характеристик (атрибутов) используемых объектов, а также описание связей между этими понятиями и характеристиками. Информационные модели можно представлять графически (графы, диаграммы сущность–отношение), в виде таблиц или списков. *Структурное* описание относится к морфологии системы, характеризует составные части системы и их межсоединения и может быть представлено структурными схемами, а также различного рода конструкторской документацией. *Поведенческое* описание характеризует процессы функционирования (алгоритмы) системы и (или) технологические процессы создания системы. Иногда аспекты описаний связывают с подсистемами, функционирование которых основано на различных физических процессах.

Отметим, что в общем случае выделение страт может быть неоднозначным. Так, помимо указанного подхода очевидна целесообразность выделения таких аспектов, как *функциональное* (разработка принципов действия, структурных, функциональных, принципиальных схем), *конструкторское* (определение форм и пространственного расположения компонентов изделий), *алгоритмическое* (разработка алгоритмов и программного обеспечения) и *технологическое* (разработка технологических процессов) проектирование систем. Примерами страт в случае САПР могут служить также рассматриваемые далее виды обеспечения автоматизированного проектирования.

**Стадии проектирования.** *Стадии проектирования* – наиболее крупные части проектирования как процесса, развивающегося во времени. В общем случае выделяют стадии научно-исследовательских работ (НИР), эскизного проекта или опытно-конструкторских работ (ОКР), технического, рабочего проектов, испытаний опытных образцов или опытных партий. Стадию НИР иногда называют предпроектными исследованиями или стадией технического предложения. Очевидно, что по мере перехода от стадии к стадии степень подробности и тщательность проработки проекта возрастают, и рабочий проект уже должен быть вполне достаточным для изготовления опытных или серийных образцов. Близким к определению

стадии, но менее четко оговоренным понятием, является понятие этапа проектирования.

Стадии (этапы) проектирования подразделяют на составные части, называемые *проектными процедурами*. Примерами проектных процедур могут служить подготовка детализованных чертежей, анализ кинематики, моделирование переходного процесса, оптимизация параметров и другие проектные задачи. В свою очередь проектные процедуры можно расчленить на более мелкие компоненты, называемые *проектными операциями*. Например, при анализе прочности детали сеточными методами операциями могут быть построение сетки, выбор или расчет внешних воздействий, собственно моделирование полей напряжений и деформаций, представление результатов моделирования в графической и текстовой формах. Проектирование сводится к выполнению некоторых последовательностей проектных процедур – *маршрутов проектирования*.

Иногда разработку ТЗ на проектирование называют внешним проектированием, а реализацию ТЗ – внутренним проектированием.

**Содержание технических заданий на проектирование.** В ТЗ на проектирование объекта указывают, по крайней мере, следующие данные:

1. Назначение объекта.

2. Условия эксплуатации. Наряду с качественными характеристиками (представленными в вербальной форме) имеются числовые параметры, называемые *внешними* параметрами, для которых указаны области допустимых значений. Примеры внешних параметров: температура окружающей среды, внешние силы, электрические напряжения, нагрузки и т. п.

3. Требования к *выходным* параметрам, т. е. к величинам, характеризующим свойства объекта, интересующие потребителя. Эти требования выражены в виде условий работоспособности

$$y_i R \in T_i,$$

где  $y_i$  –  $i$ -й выходной параметр,  $R \in \{\text{равно, меньше, больше, больше или равно, меньше или равно}\}$  – вид отношения;  $T_i$  – норма  $i$ -го выходного параметра. В случае  $R = \text{«равно»}$  нужно задать требуемую точность выполнения равенства.

*Примеры условий работоспособности:*

- расход топлива на 100 км пробега автомобиля  $< 8$  л;
- коэффициент усиления усилителя на средних частотах  $> 300$ .

**Классификация моделей и параметров, используемых при автоматизированном проектировании.** В автоматизированных проектных процедурах вместо еще не существующего проектируемого объекта оперируют некоторым *квазиобъектом* – моделью, которая отражает некоторые интересующие исследователя свойства объекта. Модель может быть физическим объектом (макет, стенд) или *спецификацией*. Среди моделей-спецификаций различают упомянутые выше функциональные, поведенческие, информационные, структурные модели (описания). Эти модели называют *математическими*, если они формализованы средствами аппарата и языка математики.

В свою очередь математические модели могут быть геометрическими, топологическими, динамическими, логическими и т. п., если они отражают соответствующие свойства объектов. Наряду с математическими моделями при проектировании используют рассматриваемые ниже функциональные IDEF0-модели, информационные модели в виде диаграмм сущность–отношение, геометрические модели-чертежи. В дальнейшем, если нет специальной оговорки, под словом «модель» будем подразумевать математическую модель.

*Математическая функциональная модель* в общем случае представляет собой алгоритм вычисления вектора выходных параметров  $Y$  при заданных векторах параметров элементов  $X$  и внешних параметров  $Q$ .

Математические модели могут быть символическими и численными. При использовании *символических* моделей оперируют не значениями величин, а их символическими обозначениями (идентификаторами). *Численные* модели могут быть *аналитическими*, т. е. их можно представить в виде явно выраженных зависимостей выходных параметров  $Y$  от параметров внутренних  $x$  и внешних  $Q$ , или *алгоритмическими*, в которых связь  $Y$ ,  $X$  и  $Q$  задана неявно в виде алгоритма моделирования. Важнейший частный случай алгоритмических моделей – *имитационные*. Они отображают процессы в системе при наличии внешних воздействий на систему. Другими словами, имитационная модель – это алгоритмическая поведенческая модель. Классификацию математических моделей выполняют также по ряду других признаков. Так, в зависимости от принадлежности к тому или иному иерархическому уровню выделяют модели уровней системного, функционально-логического, макроуровня (сосредоточенного) и микроуровня (распределенного).

По характеру используемого для описания математического аппарата различают модели лингвистические, теоретико-множественные, абстрактно-алгебраические, нечеткие и т. п.

Например, на системном уровне преимущественно применяют модели систем массового обслуживания и сети Петри, на функционально-логическом уровне – автоматные модели на основе аппарата передаточных функций или конечных автоматов, на макроуровне – системы алгебро-дифференциальных уравнений, на микроуровне – дифференциальные уравнения в частных производных. Особое место занимают геометрические модели, используемые в системах конструирования.

Кроме того, введены понятия полных моделей и макромоделей, моделей статических и динамических, детерминированных и стохастических, аналоговых и дискретных, символических и численных.

*Полная модель* объекта в отличие от макромоделей описывает не только процессы на внешних выводах моделируемого объекта, но и внутренние для объекта процессы.

*Статические модели* описывают статические состояния, в них не присутствует время в качестве независимой переменной. Динамические модели отражают поведение системы, т. е. в них обязательно используется время.

*Стохастические и детерминированные модели* различаются в зависимости от учета или неучета случайных факторов.

В *аналоговых* моделях фазовые переменные – непрерывные величины, в дискретных – дискретные, в частном случае дискретные модели являются *логическими (булевыми)*, в них состояние системы и ее элементов описывается булевыми величинами. В ряде случаев полезно применение смешанных моделей, в которых одна часть подсистем характеризуется аналоговыми моделями, другая – логическими.

*Информационные модели* относятся к информационной страте автоматизированных систем, их используют прежде всего при инфологическом проектировании баз данных (БД) для описания связей между единицами информации.

Наибольшие трудности возникают при создании моделей слабоструктурированных систем, что характерно прежде всего для системного уровня проектирования. Здесь значительное внимание уделяется экспертным методам. В теории систем сформулированы общие рекомендации по подбору экспертов при разработке модели, организации экспертизы, по обработке полученных результатов.

Обычно в имитационных моделях фигурируют фазовые переменные. Так, на макроуровне имитационные модели представляют собой системы алгебро-дифференциальных уравнений:

$$\Phi(dV/dt, V, t) = 0, \text{ при } t = 0, V = V_0, \quad (1.1)$$

где  $V$  – вектор фазовых переменных;  $t$  – время;  $V_0$  – вектор начальных условий. К примерам фазовых переменных можно отнести токи и напряжения в электрических системах, силы и скорости – в механических, давления и расходы – в гидравлических.

Выходные параметры систем могут быть двух типов. Во-первых, это параметры-функционалы, т. е. функционалы зависимостей  $V(t)$  в случае использования (1.1). Примеры таких параметров: амплитуды сигналов, временные задержки, мощности рассеивания и т. п. Во-вторых, это параметры, характеризующие способность проектируемого объекта работать при определенных внешних условиях. Эти выходные параметры являются граничными значениями диапазонов внешних переменных, в которых сохраняется работоспособность объекта.

**Типовые проектные процедуры.** Создать проект объекта (изделия или процесса) означает выбрать структуру объекта, определить значения всех его параметров и представить результаты в установленной форме. Результаты (проектная документация) могут быть выражены в виде чертежей, схем, пояснительных записок, программ для программно-управляемого технологического оборудования и других документов на бумаге или на машинных носителях информации.

Разработка (или выбор) структуры объекта есть проектная процедура, называемая *структурным синтезом*, а расчет (или выбор) значений параметров элементов  $X$  – *процедура параметрического синтеза*.

Задача структурного синтеза формулируется в системотехнике как *задача принятия решений* (ЗПР). Ее суть заключается в определении цели, множества возможных решений и ограничивающих условий.

Классификацию ЗПР осуществляют по ряду признаков. По числу критериев различают задачи одно- и многокритериальные. По степени неопределенности различают ЗПР детерминированные, ЗПР в условиях риска – при наличии в формулировке задачи случайных параметров, ЗПР в условиях неопределенности, т. е. при неполноте или недостоверности исходной информации.

Реальные задачи проектирования, как правило, являются многокритериальными. Одна из основных проблем постановки многокритериальных задач – установление правил предпочтения вариантов. Способы сведения многокритериальных задач к однокритериальным и последующие пути решения изучаются в дисциплинах, посвященных методам оптимизации и математическому программированию.

Наличие случайных факторов усложняет решение ЗПР. Основные подходы к решению ЗПР в условиях риска заключаются или в решении «для наихудшего случая», или в учете в целевой функции математического ожидания и дисперсии выходных параметров. В первом случае задачу решают как детерминированную при завышенных требованиях к качеству решения, что является главным недостатком подхода. Во втором случае достоверность результатов решения намного выше, но возникают трудности с оценкой целевой функции. Применение метода Монте-Карло в случае алгоритмических моделей становится единственной альтернативой и, следовательно, для решения требуются значительные вычислительные ресурсы.

Существуют две группы ЗПР в условиях неопределенности. Одна из них решается при наличии противодействия разумного противника. Такие задачи изучаются в *теории игр*, для задач проектирования в технике они не характерны. Во второй группе достижению цели противодействие оказывают силы природы. Для их решения полезно использовать теорию и методы *нечетких множеств*.

Например, при синтезе структуры автоматизированной системы постановка задачи должна включать в качестве исходных данных следующие сведения:

- множество выполняемых системой функций (другими словами, множество работ, каждая из которых может состоять из одной или более операций); возможно, что в этом множестве имеется частичная упорядоченность работ, что может быть представлено в виде ориентированного графа, в котором вершины соответствуют работам, а дуги – отношениям порядка;

- типы допустимых для использования серверов (машин), выполняющих функции системы;

- множество внешних источников и потребителей информации;

- во многих случаях задается также некоторая исходная структура системы в виде взаимосвязанной совокупности серверов определенных типов; эта структура может рассматриваться как обобщенная избыточная или как вариант первого приближения;

– различного рода ограничения, в частности ограничения на затраты материальных ресурсов и (или) на времена выполнения функций системы.

Задача заключается в синтезе (или коррекции) структуры, определении типов серверов (программно-аппаратных средств), распределении функций по серверам таким образом, чтобы достигался экстремум целевой функции при выполнении заданных ограничений.

Конструирование, разработка технологических процессов, оформление проектной документации – частные случаи структурного синтеза.

Задачу параметрического синтеза называют параметрической оптимизацией (или оптимизацией), если ее решают как задачу математического программирования

$$\text{extr}F(X), X \in D_x,$$

где  $F(X)$  – целевая функция;  $X$  – вектор управляемых (называемых также проектными или варьируемыми) параметров;  $D_x = \{X < \varphi(X) < 0, \psi(X) = 0\}$  – допустимая область;  $\varphi(X)$  и  $\psi(X)$  – функции-ограничения.

Следующая после синтеза группа проектных процедур – процедуры анализа. Цель анализа – получение информации о характере функционирования и значениях выходных параметров  $Y$  при заданных структуре объекта, сведениях о внешних параметрах  $Q$  и параметрах элементов  $X$ . Если заданы фиксированные значения параметров  $X$  и  $Q$ , то имеет место процедура *одновариантного анализа*, которая сводится к решению уравнений математической модели, например, такой как модель (1.1), и вычислению вектора выходных параметров  $Y$ . Если заданы статистические сведения о параметрах  $X$  и нужно получить оценки числовых характеристик распределений выходных параметров (например, оценки математических ожиданий и дисперсий), то это процедура *статистического анализа*. Если требуется рассчитать матрицы абсолютной  $A$  и (или) относительной  $B$  чувствительности, то имеет место задача *анализа чувствительности*.

Элемент  $A_{ij}$  матрицы  $A$  называют абсолютным коэффициентом чувствительности, он представляет собой частную производную  $j$ -го выходного параметра  $y_j$  по  $i$ -му параметру  $x_i$ . Другими словами,  $A_{ij}$  является элементом вектора градиента  $j$ -го выходного параметра. На практике используют безразмерные относительные коэффициенты

чувствительности  $B_{ij}$  характеризующие степень влияния изменений параметров элементов на изменения выходных параметров:

$$B_{ij} = A_{ij} x_{i\text{ном}} / y_{j\text{ном}},$$

где  $x_{i\text{ном}}$  и  $y_{j\text{ном}}$  – номинальные значения параметров соответственно.

В процедурах многовариантного анализа определяется влияние внешних параметров, разброса и нестабильности параметров элементов на выходные параметры. Процедуры статистического анализа и анализа чувствительности – характерные примеры процедур многовариантного анализа.

### **Тема 3. Системы автоматизированного проектирования и их место среди других автоматизированных систем**

**Структура САПР.** Как и любая сложная система, САПР состоит из подсистем (рис. 1.1). Различают подсистемы проектирующие и обслуживающие.

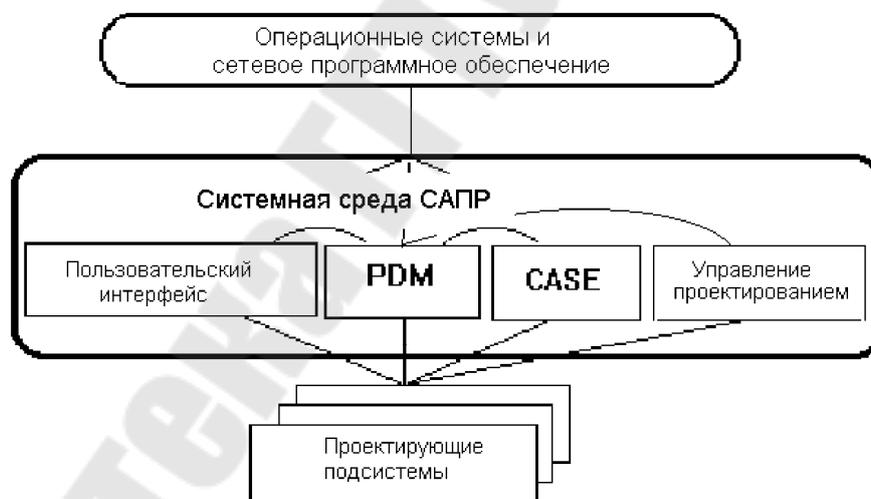


Рис. 1.1. Структура программного обеспечения САПР

Проектирующие подсистемы непосредственно выполняют проектные процедуры. Примерами проектирующих подсистем могут служить подсистемы геометрического трехмерного моделирования механических объектов, изготовления конструкторской документации, схемотехнического анализа, анализа напряженно-деформированного состояния рамной конструкции.

*Обслуживающие* подсистемы обеспечивают функционирование проектирующих подсистем, их совокупность часто называют *системной средой* (или *оболочкой*) САПР. Типичными обслуживающими подсистемами являются подсистемы управления проектными данными (PDM – Product Data Management), управления процессом проектирования (DesPM – Design Process Management), пользовательского интерфейса для связи разработчиков с ЭВМ, CASE (Computer Aided Software Engineering) для разработки и сопровождения программного обеспечения САПР, обучающие подсистемы для освоения пользователями технологий, реализованных в САПР.

Структурирование САПР по различным аспектам обуславливает появление *видов обеспечения САПР*. Принято выделять семь видов обеспечения:

– *техническое* (ТО), включающее различные аппаратные средства (ЭВМ, периферийные устройства, сетевое коммутационное оборудование, линии связи, измерительные средства);

– *математическое* (МО), объединяющее математические методы, модели и алгоритмы для выполнения проектирования;

– *программное* (ПО), представляемое компьютерными программами САПР;

– *информационное* (ИО), состоящее из баз данных (БД), систем управления базами данных (СУБД), а также других данных, используемых при проектировании; отметим, что вся совокупность используемых при проектировании данных называется *информационным фондом САПР*, а БД вместе с СУБД носит название *банка данных* (БнД);

– *лингвистическое* (ЛО), выражаемое языками общения между проектировщиками и ЭВМ, языками программирования и языками обмена данными между техническими средствами САПР;

– *методическое* (МетО), включающее различные методики проектирования, иногда к МетО относят также математическое обеспечение;

– *организационное* (ОО), представляемое штатными расписаниями, должностными инструкциями и другими документами, регламентирующими работу проектного предприятия.

**Разновидности САПР.** Классификацию САПР осуществляют по ряду признаков, например, по приложению, целевому назначению, масштабам (комплексности решаемых задач), характеру базовой подсистемы – ядра САПР.

По *приложениям* наиболее представительными и широко используемыми являются следующие группы САПР.

1. САПР для применения в отраслях общего машиностроения. Их часто называют машиностроительными САПР или MCAD (Mechanical CAD) системами.

2. САПР для радиоэлектроники. Их названия – ECAD (Electronic CAD) или EDA (Electronic Design Automation) системы.

3. САПР в области архитектуры и строительства.

Кроме того, известно большое число более специализированных САПР, или выделяемых в указанных группах, или представляющих самостоятельную ветвь в классификации. Примерами таких систем являются САПР больших интегральных схем (БИС); САПР летательных аппаратов; САПР электрических машин и т. п.

По *целевому назначению* различают САПР или подсистемы САПР, обеспечивающие разные аспекты (страты) проектирования. Так, в составе MCAD появляются CAE/CAD/CAM системы:

1) САПР функционального проектирования, иначе САПР-ФП или CAE (Computer Aided Engineering) системы;

2) *конструкторские* САПР общего машиностроения – САПР-К, часто называемые просто CAD системами;

3) *технологические* САПР общего машиностроения – САПР-Т, иначе называемые автоматизированными системами технологической подготовки производства АСТПП или системами САМ (Computer Aided Manufacturing).

По *масштабам* различают отдельные программно-методические комплексы (ПМК) САПР, например, комплекс анализа прочности механических изделий в соответствии с методом конечных элементов (МКЭ) или комплекс анализа электронных схем; системы ПМК; системы с уникальными архитектурами не только программного (software), но и технического (hardware) обеспечений.

По *характеру базовой подсистемы* различают следующие разновидности САПР:

1. *САПР на базе подсистемы машинной графики и геометрического моделирования.* Эти САПР ориентированы на приложения, где основной процедурой проектирования является конструирование, т. е. определение пространственных форм и взаимного расположения объектов. Поэтому к этой группе систем относится большинство графических ядер САПР в области машиностроения.

В настоящее время появились унифицированные графические ядра, применяемые более чем в одной САПР, это ядра Parasolid фирмы EDS Unigraphics и ACIS фирмы Intergraph.

2. *САПР на базе СУБД.* Они ориентированы на приложения, в которых при сравнительно несложных математических расчетах перерабатывается большой объем данных. Такие САПР преимущественно встречаются в технико-экономических приложениях, например, при проектировании бизнес-планов, но имеют место также при проектировании объектов, подобных щитам управления в системах автоматики.

3. *САПР на базе конкретного прикладного пакета.* Фактически это автономно используемые программно-методические комплексы, например, имитационного моделирования производственных процессов, расчета прочности по методу конечных элементов, синтеза и анализа систем автоматического управления и т. п. Часто такие САПР относятся к системам САЕ. Примерами могут служить программы логического проектирования на базе языка VHDL, математические пакеты типа MathCAD.

4. *Комплексные (интегрированные) САПР,* состоящие из совокупности подсистем предыдущих видов. Характерными примерами комплексных САПР являются САЕ/CAD/CAM-системы в машиностроении или САПР БИС. Так, САПР БИС включает в себя СУБД и подсистемы проектирования компонентов, принципиальных, логических и функциональных схем, топологии кристаллов, тестов для проверки годности изделий. Для управления столь сложными системами применяют специализированные системные среды.

**Функции, характеристики и примеры САЕ/CAD/CAM-систем.** Функции CAD-систем в машиностроении подразделяют на функции двухмерного (2D) и трехмерного (3D) проектирования. К функциям 2D относятся черчение, оформление конструкторской документации; к функциям 3D – получение трехмерных моделей, метрические расчеты, реалистичная визуализация, взаимное преобразование 2D и 3D моделей.

Среди CAD-систем различают «легкие» и «тяжелые» системы. Первые из них ориентированы преимущественно на 2D графику, сравнительно дешевы и менее требовательны в отношении вычислительных ресурсов. Вторые ориентированы на геометрическое моделирование (3D), более универсальны, дороги, оформление чертеж-

ной документации в них обычно осуществляется с помощью предварительной разработки трехмерных геометрических моделей.

Основные функции САМ-систем: разработка технологических процессов, синтез управляющих программ для технологического оборудования с числовым программным управлением (ЧПУ), моделирование процессов обработки, в том числе построение траекторий относительного движения инструмента и заготовки, генерация пост-процессоров для конкретных типов оборудования с ЧПУ (*NC* – Numerical Control), расчет норм времени обработки.

Наиболее известны (к 2000 г.) следующие САЕ/CAD/CAM-системы, предназначенные для отраслей машиностроения. «Тяжелые» системы (в скобках указана фирма, разработавшая или распространяющая продукт): Unigraphics (EDS Unigraphics); Pro/Engineer (PTC – Parametric Technology Corp.), CATIA (Dassault Systemes), EUCLID (Matra Datavision) и др.

«Легкие» системы: AutoCAD (Autodesk); Caddy (Ziegler Informatics); КОМПАС (Аскон, С.Петербург); Спрут (Sprut Technology, Набережные Челны); T-FlexCAD (Топ Системы, Москва).

Системы, занимающие промежуточное положение (среднемасштабные): Cimatron, Microstation (Bentley), Euclid Prelude (Matra Datavision), Solid Edge (Intergraph) и др. С ростом возможностей персональных ЭВМ грани между «тяжелыми» и «легкими» САД/САМ-системами постепенно стираются.

Функции САЕ-систем довольно разнообразны, так как связаны с проектными процедурами анализа, моделирования, оптимизации проектных решений. В состав машиностроительных САЕ-систем прежде всего включают программы для следующих процедур:

- моделирование полей физических величин, в том числе анализ прочности, который чаще всего выполняется в соответствии с МКЭ;
- расчет состояний и переходных процессов на макроуровне;
- имитационное моделирование сложных производственных систем на основе моделей массового обслуживания и сетей Петри.

Примеры систем моделирования полей физических величин в соответствии с МКЭ: Nastran, Ansys, Cosmos, Nisa, Moldflow.

Примеры систем моделирования динамических процессов на макроуровне: Adams и LS-Dyna – в механических системах, Spice – в электронных схемах, ПА9 – для многоаспектного моделирования, т. е. для моделирования систем, принципы действия которых основаны на взаимовлиянии физических процессов различной природы.

Для удобства адаптации САПР к нуждам конкретных приложений, для ее развития целесообразно иметь в составе САПР инструментальные средства адаптации и развития. Эти средства представлены той или иной CASE-технологией, включая языки расширения. В некоторых САПР применяют оригинальные инструментальные среды.

Примерами могут служить объектно-ориентированная интерактивная среда CAS.CADE в системе EUCLID, содержащая библиотеку компонентов, в САПР T-Flex CAD 3D предусмотрена разработка дополнений в средах Visual C++ и Visual Basic.

Важное значение для обеспечения открытости САПР, ее интегрируемости с другими автоматизированными системами (АС) имеют интерфейсы, представляемые реализованными в системе форматами межпрограммных обменов. Очевидно, что в первую очередь необходимо обеспечить связи между САЕ, САД и САМ-подсистемами.

В качестве языков – форматов межпрограммных обменов – используются IGES, DXF, Express (стандарт ISO 10303-11, входит в совокупность стандартов STEP), SAT (формат ядра ACIS) и др.

Наиболее перспективными считаются диалекты языка Express, что объясняется общим характером стандартов STEP, их направленностью на различные приложения, а также на использование в современных распределенных проектных и производственных системах. Действительно, такие форматы, как IGES или DXF, описывают только геометрию объектов, в то время как в обменах между различными САПР и их подсистемами фигурируют данные о различных свойствах и атрибутах изделий.

**Понятие о CALS-технологии.** *CALS*-технология – это технология комплексной компьютеризации сфер промышленного производства, цель которой – унификация и стандартизация спецификаций промышленной продукции на всех этапах ее жизненного цикла. Основные спецификации представлены проектной, технологической, производственной, маркетинговой, эксплуатационной документацией. В *CALS*-системах предусмотрены хранение, обработка и передача информации в компьютерных средах, оперативный доступ к данным в нужное время и в нужном месте. Соответствующие системы автоматизации назвали автоматизированными логистическими системами или *CALS* (Computer Aided Logistic Systems). Поскольку под логистикой обычно понимают дисциплину, посвященную вопросам снабжения и управления запасами, а функции *CALS* намного шире и связаны со всеми этапами жизненного цикла промышленных изделий, приме-

няют и более соответствующую предмету расшифровку аббревиатуры CALS – Continuous Acquisition and LifeCycle Support.

Применение CALS позволяет существенно сократить объемы проектных работ, так как описания многих составных частей оборудования, машин и систем, проектировавшихся ранее, хранятся в базах данных сетевых серверов, доступных любому пользователю технологии CALS. Существенно облегчается решение проблем ремонтпригодности, интеграции продукции в различного рода системы и среды, адаптации к меняющимся условиям эксплуатации, специализации проектных организаций и т. п. Ожидается, что успех на рынке сложной технической продукции будет немислим вне технологии CALS.

Развитие CALS-технологии должно привести к появлению так называемых *виртуальных производств*, при которых процесс создания спецификаций с информацией для программно управляемого технологического оборудования, достаточной для изготовления изделия, может быть распределен во времени и пространстве между многими организационно-автономными проектными отделами. Среди несомненных достижений CALS-технологии следует отметить легкость распространения передовых проектных решений, возможность многократного воспроизведения частей проекта в новых разработках.

Построение открытых распределенных автоматизированных систем для проектирования и управления в промышленности составляет основу современной CALS-технологии. Главная проблема их построения – обеспечение единообразного описания и интерпретации данных, независимо от места и времени их получения в общей системе. Структура проектной, технологической и эксплуатационной документации, языки ее представления должны быть стандартизованными. Тогда становится реальной успешная работа над общим проектом разных коллективов, разделенных во времени и пространстве и использующих разные CAE/CAD/CAM-системы. Одна и та же КД может быть использована многократно в разных проектах, а одна и та же технологическая документация адаптирована к разным производственным условиям, что позволяет сократить и удешевить цикл проектирования и производства. Кроме того, упрощается эксплуатация систем. Следовательно, информационная интеграция является неотъемлемым свойством CALS-систем. Поэтому в основу CALS-технологии положен ряд стандартов, обеспечивающих такую интеграцию.

Важные проблемы, требующие решения при создании комплексных САПР – управление сложностью проектов и интеграция ПО. Эти проблемы включают вопросы декомпозиции проектов, распараллеливания проектных работ, целостности данных, межпрограммных интерфейсов и др.

**Комплексные автоматизированные системы.** Известно, что частичная автоматизация зачастую не дает ожидаемого повышения эффективности функционирования предприятий. Поэтому предпочтительным является внедрение интегрированных САПР, автоматизирующих все основные этапы проектирования изделий. Дальнейшее повышение эффективности производства и повышение конкурентоспособности выпускаемой продукции возможно за счет интеграции систем проектирования, управления и документооборота.

Такая интеграция лежит в основе создания *комплексных систем автоматизации*, в которых помимо функций собственно САПР реализуются средства для автоматизации функций управления проектированием, документооборота, планирования производства, учета и т. п.

Проблемы интеграции лежат в основе технологии Юпитер, пропагандируемой фирмой Intergraph. Пример сращивания некоторых подсистем из САПР и АСУ – программный продукт TechnODOCS (русская фирма «Весть»). Его функции:

- интеграция программ документооборота с проектирующими пакетами (конкретно с AutoCAD, Microstation и другими программами, исполняемыми в Windows-средах и поддерживающими взаимодействие по технологиям DDE или OLE, разработанным фирмой Microsoft);

- ведение архива технической документации;

- маршрутизация работ и прохождение документации, контроль исполнения;

- управление параллельным проектированием, т. е. координацией проектных работ, выполняемых коллективно. Очевидно, что подобная интеграция является неотъемлемой чертой CALS-систем. В основу CALS-технологии положен ряд стандартов и прежде всего это стандарты STEP, а также Parts Library, Mandate, SGML (Standard Generalized Markup Language), EDIFACT (Electronic Data Interchange For Administration, Commerce, Transport) и др. Стандарт SGML устанавливает способы унифицированного оформления документов определен-

ного назначения – отчетов, каталогов, бюллетеней и т. п., а стандарт EDIFACT – способы обмена подобными документами.

Одна из наиболее известных реализаций CALS-технологии разработана фирмой Computervision. Это технология названа EPD (Electronic Product Definition) и ориентирована на поддержку процессов проектирования и эксплуатации изделий машиностроения.

В CALS-системах на всех этапах жизненного цикла изделий используется документация, полученная на этапе проектирования. Поэтому естественно, что составы подсистем в CALS и комплексных САПР в значительной мере совпадают. Технологию EPD реализуют:

- CAD – система автоматизированного проектирования;
- CAM – автоматизированная система технологической подготовки производства (АСТПП);
- CAE – система моделирования и расчетов;
- CAPE (Concurrent Art-to-Product Environment) – система поддержки параллельного проектирования (concurrent engineering);
- PDM – система управления проектными данными, представляющая собой специализированную СУБД (DBMS – Data Base Management System);
- 3D Viewer – система трехмерной визуализации;
- CADD – система документирования;
- CASE – система разработки и сопровождения программного обеспечения и методики обследования и анализа функционирования предприятий.

Основу EPD составляют системы CAD и PDM, в качестве которых используются CADD5 и Optegra соответственно. В значительной мере специфику EPD определяет система Optegra. В ней отображается иерархическая структура изделий, включающая все сборочные узлы и детали.

Важной для пользователей особенностью Optegra является работа вместе с многооконной системой визуализации 3D Viewer. Пользователь может одновременно следить за информацией в нескольких типовых окнах:

- информационный браузер, в котором высвечиваются данные, запрашиваемые пользователем, например, из почтового ящика, Internet, корпоративных ресурсов, его персональной БД;
- окно структуры изделия, представляемой в виде дерева. Можно получать ответы на запросы подсветкой деталей  $D_j$  (листьев дерева), удовлетворяющих условиям запроса;

– 3D визуализатор. В этом окне высвечивается трехмерное изображение изделия, ответы на запросы даются и в этом окне цветовым выделением деталей  $D_j$ ;

– окно пользовательского процесса, в котором в нужной последовательности в виде иконок отображается перечень задач, заданный пользователю для решения.



Рис. 1.2. Основные функции АСУП

В системе Ortega связи между объектами задаются по протоколам стандартов STEP, внешний интерфейс осуществляется через базу данных SDAI.

**Системы управления в составе комплексных автоматизированных систем.** Системы управления в промышленности, как и любые сложные системы, имеют иерархическую структуру. Если рассматривать предприятие как систему верхнего уровня, то следующими уровнями по нисходящей линии будут уровни завода, цеха, производственного участка, производственного оборудования.

Автоматизация управления реализуется с помощью автоматизированных систем управления (АСУ).

Среди АСУ различают *автоматизированные системы управления предприятием (АСУП)* и *автоматизированные системы управления технологическими процессами (АСУТП)*. АСУП охватывает уровни от предприятия до цеха, АСУТП – от цеха и ниже (на уровне цеха могут быть средства и АСУП и АСУТП).

В АСУП выделяют подсистемы, выполняющие определенные функции (рис. 1.2), типичными среди них являются: календарное планирование производства, потребностей в мощностях и материалах;

оперативное управление производством; сетевое планирование проектов; управление проектированием изделий; учет и нормирование трудозатрат; учет основных фондов; управление финансами; управление запасами (складским хозяйством); управление снабжением (статистика закупок, контракты на закупку); маркетинг (статистика и анализ реализации, контракты на реализацию, прогноз, реклама).

Процедуры, выполняющие эти функции, часто называют *бизнес-функциями*, а маршруты решения задач управления, состоящие из бизнес-функций, называют *бизнес-процессами*.

*Примечание.* Как отмечено выше, в САПР аналогичные понятия называют проектными процедурами и маршрутами проектирования.

Существуют разновидности АСУП со своими англоязычными названиями. Наиболее общую систему с перечисленными выше функциями называют ERP (Enterprise Resource Planning). Системы, направленные на управление информацией о материалах, производстве, контроле и т. п. изделий, называют MRP-2 (Manufacturing Resource Planning). В ERP, как и в САПР, важная роль отводится системам управления данными PDM. Если PDM обеспечивает управление конфигурацией проектов и относится в большей мере к проектированию, то MRP-2 управляет данными, относящимися к производству. Для таких систем иногда используют также название MES (Manufacturing Execution System).

Мировыми лидерами среди систем программного обеспечения АСУП являются системы R3 (фирма SAP) и Baan IV (Baan), широко известны также MANMAN/X (Computer Associates CIS), Series (Tecsys Inc.), Marix (IBM) и др.

Характерные особенности современных АСУП:

1. Открытость по отношению к ведущим платформам (UNIX, Windows, OS/2) и различным СУБД и прежде всего мощным СУБД типа Oracle, Ingres, Informix, Sybase; поддержка технологий типа ODBC (Open Data Base Connection), OLE (Object Linking and Embedding), DDE (Dynamic Data Exchange); поддержка архитектур клиент/сервер. Важная характеристика – возможность работы в среде распределенных вычислений.

2. Возможность сквозного выполнения всех допустимых бизнес-функций или их части, что обеспечивается модульным построением (количество функций может превышать 100).

3. Адаптируемость к конкретным заказчикам и условиям рынка.

4. Наличие инструментальных средств, в том числе языка расширения или 4GL (языка четвертого поколения). Так, в R3 используется язык АВАР/L, в Elite Series – язык Informix-4GL.

5. Техническое обеспечение АСУП – компьютерная сеть, узлы которой расположены как в административных отделах предприятия, так и в цехах.

Очевидно, что для создания и развития виртуальных предприятий необходимы распространение CALS-технологии не только на САПР, но и на АСУ, их интеграция в комплексные системы информационной поддержки всех этапов жизненного цикла промышленной продукции.

Функциями АСУТП на уровнях цеха и участка являются сбор и обработка данных о состоянии оборудования и протекании производственных процессов для принятия решений по загрузке станков, по выполнению технологических маршрутов. Программное обеспечение АСУТП на этих уровнях представлено системой диспетчерского управления и сбора данных, называемой SCADA (*Supervisory Control and Data Acquisition*), а техническое обеспечение – персональными ЭВМ и микрокомпьютерами, связанными локальной вычислительной сетью. Кроме диспетчерских функций, SCADA выполняет роль инструментальной системы разработки программного обеспечения для промышленных систем компьютерной автоматизации, т. е. роль специфической CASE-системы. Для систем АСУТП характерно использование *программируемых контроллеров* (ПЛК или PLC – Programmed Logic Controller) – компьютеров, встроенных в технологическое оборудование.

Функции SCADA: сбор первичной информации от датчиков; хранение, обработка и визуализация данных; управление и регистрация аварийных сигналов; связь с корпоративной информационной сетью; автоматизированная разработка прикладного ПО.

К разработке программ для программируемых контроллеров обычно привлекаются не профессиональные программисты, а заводские технологи. Поэтому языки программирования должны быть достаточно простыми, обычно построенными на визуальных изображениях ситуаций. Например, используются различные схемные языки. Ряд языков стандартизован и представлен в международном стандарте IEC 1131-3.

На уровне управления технологическим оборудованием в АСУТП выполняются запуск, тестирование, выключение станков,

сигнализация о неисправностях, выработка управляющих воздействий для рабочих органов программно управляемого оборудования. Для этого в составе технологического оборудования используются системы управления на базе встроенных контроллеров.

**Автоматизированные системы делопроизводства (АСД).** Информационные технологии и автоматизированные системы управления документами и документооборотом пользуются все возрастающим вниманием среди предприятий и фирм различного профиля, поскольку организация работы с документами существенно влияет на эффективность производственных и бизнес-процессов. Такие системы имеют как самостоятельное значение, так и играют важную роль в интегрированных автоматизированных системах управления и проектирования.

*Автоматизированные системы делопроизводства* по своему назначению подразделяют на системы управления документами (СУД), управления документооборотом (СДО), управления знаниями (в сфере делопроизводства) и инструментальные среды делопроизводства. В соответствии с другими критериями классификации системы делопроизводства подразделяют на специализированные и комплексные, локальные и распределенные, фактографические и документографические (полнотекстовые), заказные и тиражируемые.

Системы управления документами предназначены для обеспечения санкционированного доступа к документам. Характерные функции СУД:

- ввод документов, в частности, с помощью средств их автоматического распознавания;
- индексирование документов, например, оформление регистрационных карточек с полями для атрибутов; возможно атрибутивное индексирование – к атрибутам относятся автор документа, дата создания и ключевые слова или полнотекстовое индексирование – в индекс заносят весь текст, но без предлогов и окончаний некоторых слов.
- хранение документов;
- поиск нужных данных, который может быть атрибутивным в фактографических БД или полнотекстовым в случае слабоструктурированных документов;
- поддержка групповой работы над документами;
- разграничение прав доступа к документам;

- контроль и управление версиями документов, регламентирующие внесение в них изменений;
- сбор и анализ статистических данных по параметрам документов и функционированию системы;
- подготовка отчетов.

*Системы управления документооборотом* служат для управления деловыми процессами прохождения и обработки документов в соответствующих подразделениях и службах организации. Характерные функции СДО:

- регистрация документов при их вхождении в систему;
- маршрутизация документов, учет их движения (маршрутизация может быть жесткой при фиксированных маршрутах или свободной); управление потоками документов обеспечивает прохождение документов по заданному маршруту с контролем внесения в них резолюций, управление внесением изменений включает систему приоритетов, средства протоколирования изменений;
- контроль исполнения предписываемых документами действий;
- защита информации при ее передаче между пунктами распределенной системы;
- автоматическое уведомление соответствующих лиц о состоянии документов и содержащихся в них директив и рекомендаций;
- планирование работ, связанных с прохождением документов.

К *системам управления знаниями* в области делопроизводства относят системы, выполняющие функции, характерные для интеллектуальных систем. Примеры таких функций: классификация документов по тем или иным признакам; взаимное связывание документов, например, с помощью гипертекста; тематический отбор документов; интеграция данных, поступающих из различных источников; аналитическая обработка данных; моделирование деловых процессов.

*Инструментальные среды* в системах делопроизводства служат для формирования систем делопроизводства, адаптированных к условиям конкретных предприятий и фирм. Кроме перечня решаемых задач, выделяют следующие свойства и характеристики систем делопроизводства: открытость, программные интерфейсы и форматы данных для обмена с другими информационными системами; мобильность для инсталляции на ведущих платформах; модульное построение, что обеспечивает масштабируемость; возможность эволюционного развития, адаптируемость, возможность внедрения на предприятиях по частям; пользовательский интерфейс; быстродействие, вре-

менные затраты на выполнение задач; уровень защиты информации; соответствие стандартам информационных технологий; операционные среды и используемые СУБД, требования к аппаратным ресурсам; перенос документов по мере их устаревания на более дешевые носители.

В крупных АСД предусматривается распределенное хранение с доступом в режимах как off-line, так и on-line. В первом случае пользователь формирует запрос в виде совокупности ключевых слов и направляет его средствами электронной почты (E-mail), СДО выдает список релевантных документов, пользователь выбирает из списка нужные документы и посылает вторичный более конкретный запрос, получая по E-mail запрошенные документы. Во втором случае используется связь в реальном времени, документ вызывается на экран компьютера и пользователь может непосредственно его просматривать и редактировать.

Современные корпоративные системы делопроизводства являются распределенными, имеющими архитектуру клиент-сервер. На серверной стороне находят применение серверы баз данных, полнотекстовых документов, электронной почты, приложений, SQL- и Web-серверы. На клиентской стороне могут выделяться рабочие места пользователей, администратора и разработчиков баз данных, информационно-поисковых систем, форм документов и т. п.

## **Глава 2. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ**

### **Тема 4. Структура технического обеспечения систем автоматизированного проектирования**

**Требования к ТО САПР.** Техническое обеспечение САПР включает в себя различные технические средства (hardware), используемые для выполнения автоматизированного проектирования, а именно ЭВМ, периферийные устройства, сетевое оборудование, а также оборудование некоторых вспомогательных систем (например, измерительных), поддерживающих проектирование.

Используемые в САПР технические средства должны обеспечивать: выполнение всех необходимых проектных процедур, для которых имеется соответствующее ПО; взаимодействие между проектировщиками и ЭВМ, поддержку интерактивного режима работы; взаимодействие между членами коллектива, выполняющими работу над общим проектом. Первое из этих требований выполняется при наличии в САПР вычислительных машин и систем с достаточными производительностью и емкостью памяти.

Второе требование относится к пользовательскому интерфейсу и выполняется за счет включения в САПР удобных средств ввода-вывода данных и прежде всего устройств обмена графической информацией.

Третье требование обуславливает объединение аппаратных средств САПР в вычислительную сеть.

В результате общая структура ТО САПР представляет собой сеть узлов, связанных между собой средой передачи данных (рис. 2.1). Узлами (станциями данных) являются рабочие места проектировщиков, часто называемые *автоматизированными рабочими местами* (АРМ) или *рабочими станциями* (WS – Workstation), ими могут быть также большие ЭВМ (мейнфреймы), отдельные периферийные и измерительные устройства. Именно в АРМ должны быть

средства для интерфейса проектировщика с ЭВМ. При этом вычислительная мощность может быть распределена между различными узлами вычислительной сети.

Среда передачи данных представлена каналами передачи данных, состоящими из линий связи и коммутационного оборудования.

В каждом узле можно выделить *оконечное оборудование данных* (ООД), выполняющее определенную работу по проектированию, и *аппаратуру окончания канала данных* (АКД), предназначенную для связи ООД со средой передачи данных. Например, в качестве ООД можно рассматривать персональный компьютер, а в качестве АКД – вставляемую в компьютер сетевую плату.



Рис. 2.1. Структура технического обеспечения САПР

*Канал передачи данных* – средство двустороннего обмена данными, включающее в себя АКД и линию связи. *Линией связи* называют часть физической среды, используемую для распространения сигналов в определенном направлении, примерами линий связи могут служить коаксиальный кабель, витая пара проводов, волоконно-оптическая линия связи (ВОЛС). Близким является понятие канала (*канала связи*), под которым понимают средство односторонней передачи данных. Примером канала связи может быть полоса частот, выделенная одному передатчику при радиосвязи. В некоторой линии можно образовать несколько каналов связи, по каждому из которых передается своя информация. При этом говорят, что линия разделяется между несколькими каналами.

**Типы сетей.** Существуют два метода разделения линии передачи данных: *временное мультиплексирование* (иначе разделение по времени или TDM – Time Division Method), при котором каждому каналу выделяется некоторый квант времени, и *частотное разделение*

(FDM – Frequency Division Method), при котором каналу выделяется некоторая полоса частот.

В САПР небольших проектных организаций, насчитывающих не более единиц-десятков компьютеров, которые размещены на малых расстояниях один от другого (например, в одной или нескольких соседних комнатах), объединяющая компьютеры сеть является локальной. *Локальная вычислительная сеть* (ЛВС или LAN – Local Area Network) имеет линию связи, к которой подключаются все узлы сети. При этом топология соединений узлов (рис. 2.2) может быть шинная (bus), кольцевая (ring), звездная (star). Протяженность линии и число подключаемых узлов в ЛВС ограничены.

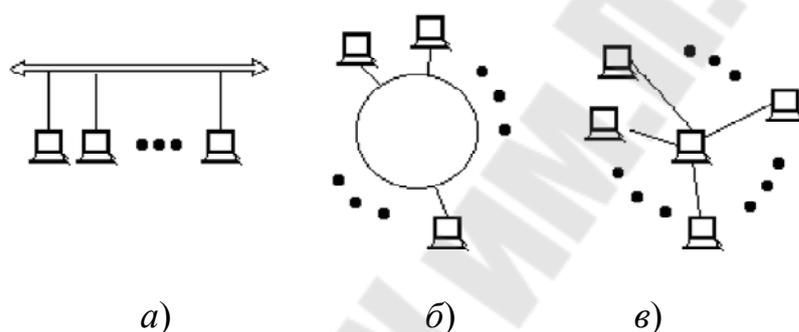


Рис. 2.2. Варианты топологии локальных вычислительных сетей:  
а – шинная; б – кольцевая; в – звездная

В более крупных по масштабам проектных организациях в сеть включены десятки-сотни и более компьютеров, относящихся к разным проектным и управленческим подразделениям и размещенных в помещениях одного или нескольких зданий. Такую сеть называют *корпоративной*. В ее структуре можно выделить ряд ЛВС, называемых *подсетями*, и средства связи ЛВС между собой. В эти средства входят коммутационные серверы (блоки взаимодействия подсетей). Если коммутационные серверы объединены отделенными от ЛВС подразделений каналами передачи данных, то они образуют новую подсеть, называемую опорной (или транспортной), а вся сеть оказывается иерархической структуры.

Если здания проектной организации удалены друг от друга на значительные расстояния (вплоть до их расположения в разных городах), то корпоративная сеть по своим масштабам становится *территориальной сетью* (WAN – Wide Area Network). В территориальной сети различают *магистральные* каналы передачи данных (магистральную сеть), имеющие значительную протяженность, и каналы

передачи данных, связывающие ЛВС с магистральной сетью и называемые *абонентской линией* или соединением «*последней мили*».

Обычно создание *выделенной* магистральной сети, т. е. сети, обслуживающей единственную организацию, обходится для нее слишком дорого. Поэтому чаще прибегают к услугам провайдера, т. е. организации, предоставляющей телекоммуникационные услуги многим пользователям. В этом случае внутри корпоративной сети связь на значительных расстояниях осуществляется через *магистральную сеть общего пользования*. В качестве такой сети можно использовать, например, городскую или междугородную телефонную сеть или территориальные сети передачи данных. Наиболее распространенной формой доступа к этим сетям в настоящее время является обращение к глобальной вычислительной сети Internet.

Для многих корпоративных сетей возможность выхода в Internet является желательной не только для обеспечения взаимосвязи удаленных сотрудников собственной организации, но и для получения других информационных услуг. Развитие виртуальных предприятий, работающих на основе CALS-технологий, с необходимостью подразумевает информационные обмены через территориальные сети, как правило, через Internet.

Структура ТО САПР для крупной организации представлена на рис. 2.3. Здесь показана типичная структура крупных корпоративных сетей САПР, называемая архитектурой *клиент-сервер*. В сетях клиент-сервер выделяется один или несколько узлов, называемых *серверами*, которые выполняют в сети управляющие или общие для многих пользователей проектные функции, а остальные узлы (рабочие места) являются терминальными, их называют *клиентами*, в них работают пользователи. Сервером называют совокупность программных средств, ориентированных на выполнение определенных функций, но если эти средства сосредоточены на конкретном узле вычислительной сети, то тогда понятие сервер относится именно к узлу сети.

Сети клиент-сервер различают по характеру распределения функций между серверами, т. е. их классифицируют по типам серверов. Различают *файл-серверы* для хранения файлов, разделяемых многими пользователями, *серверы баз данных САПР*, *серверы приложений* для решения конкретных прикладных задач, *коммутационные серверы* (называемые также блоками взаимодействия сетей или серверами доступа) для взаимосвязи сетей и подсетей, *специали-*

зированные серверы для выполнения определенных телекоммуникационных услуг, например, серверы электронной почты.

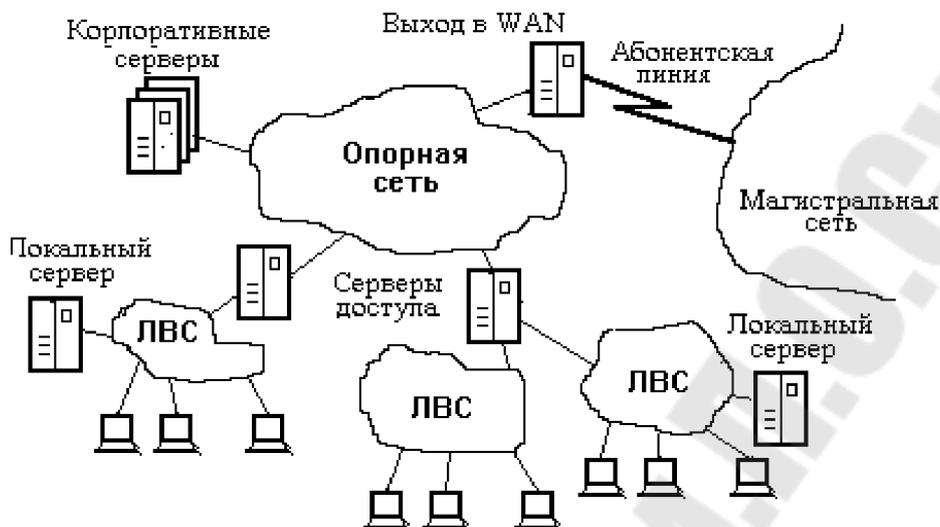


Рис. 2.3. Структура корпоративной сети САПР

В случае специализации серверов по определенным приложениям сеть называют *сетью распределенных вычислений*. Если сервер приложений обслуживает пользователей одной ЛВС, то естественно назвать такой сервер локальным. Но поскольку в САПР имеются приложения и базы данных, разделяемые пользователями разных подразделений и, следовательно, клиентами разных ЛВС, то соответствующие серверы относят к группе корпоративных, подключаемых обычно к опорной сети (рис. 2.3).

Наряду с архитектурой клиент-сервер применяют *одноранговые* сети, в которых любой узел в зависимости от решаемой задачи может выполнять как функции сервера, так и функции клиента.

В соответствии со способами коммутации различают сети с *коммутацией каналов* и *коммутацией пакетов*. В первом случае при обмене данными между узлами  $A$  и  $B$  в сети создается физическое соединение между  $A$  и  $B$ , которое во время сеанса связи используется только этими абонентами. Примером сети с коммутацией каналов может служить телефонная сеть. Здесь передача информации происходит быстро, но каналы связи используются неэффективно, так как при обмене данными возможны длительные паузы и канал «простаивает». При коммутации пакетов физического соединения, которое в каждый момент сеанса связи соединяло бы абонентов  $A$  и  $B$ , не создается. Сообщения разделяются на порции, называемые *пакетами*, которые передаются в разветвленной сети от  $A$  к  $B$  или обратно через

промежуточные узлы с возможной буферизацией (временным запоминанием) в них. Таким образом, любая линия может разделяться многими сообщениями, попеременно пропуская при этом пакеты разных сообщений с максимальным заполнением упомянутых пауз.

**Эталонная модель взаимосвязи открытых систем (ЭМВОС).** Для удобства модернизации сложных информационных систем их делают максимально открытыми, т. е. приспособленными для внесения изменений в некоторую часть системы при сохранении неизменными остальных частей. В отношении вычислительных сетей реализация концепции открытости привела к появлению ЭМВОС, предложенной международной организацией стандартизации (ISO – International Standard Organization). В этой модели дано описание общих принципов, правил, соглашений, обеспечивающих взаимодействие информационных систем и называемых *протоколами*.

В ЭМВОС информационную сеть рассматривают как совокупность функций (протоколов), которые подразделяют на группы, называемые *уровнями*. Именно разделение на уровни позволяет вносить изменения в средства реализации одного уровня без перестройки средств других уровней, что значительно упрощает и удешевляет модернизацию средств по мере развития техники.

ЭМВОС содержит семь уровней. На физическом (*physical*) уровне осуществляется представление информации в виде электрических или оптических сигналов, преобразование формы сигналов, выбор параметров физических сред передачи данных, организуется передача информации через физические среды.

На *канальном (link)* уровне выполняется обмен данными между соседними узлами сети, т. е. узлами, непосредственно связанными физическими соединениями без других промежуточных узлов. Отметим, что пакеты канального уровня обычно называют *кадрами*.

На *сетевом (network)* уровне происходит формирование пакетов по правилам тех промежуточных сетей, через которые проходит исходный пакет, и *маршрутизация* пакетов, т. е. определение и реализация маршрутов, по которым передаются пакеты. Другими словами, маршрутизация сводится к образованию логических каналов. *Логическим каналом* называют виртуальное соединение двух или более объектов сетевого уровня, при котором возможен обмен данными между этими объектами. Понятию логического канала необязательно соответствует физическое соединение линий передачи данных между связываемыми пунктами. Это понятие введено для абстрагирования

от физической реализации соединения. Еще одной важной функцией сетевого уровня после маршрутизации является контроль нагрузки на сеть с целью предотвращения перегрузок, отрицательно влияющих на работу сети.

На *транспортном (transport)* уровне обеспечивается связь между окончательными пунктами (в отличие от предыдущего сетевого уровня, на котором обеспечивается передача данных через промежуточные компоненты сети). К функциям транспортного уровня относятся мультиплексирование и демultipлексирование (сборка–разборка пакетов в конечных пунктах), обнаружение и устранение ошибок в переданных данных, реализация заказанного уровня услуг (например, заказанных скорости и надежности передачи).

На *сеансовом (session)* уровне определяются тип связи (дуплекс или полудуплекс), начало и окончание заданий, последовательность и режим обмена запросами и ответами взаимодействующих партнеров.

На *представительном (presentation)* уровне реализуются функции представления данных (кодирование, форматирование, структурирование). Например, на этом уровне выделенные для передачи данные преобразуются из одного кода в другой.

На *прикладном (application)* уровне определяются и оформляются в сообщения те данные, которые подлежат передаче по сети.

В конкретных случаях может возникать потребность в реализации лишь части названных функций, тогда соответственно сеть будет содержать лишь часть уровней. Так, в простых (неразветвленных) ЛВС отпадает необходимость в средствах сетевого и транспортного уровней. Одновременно сложность функций канального уровня делает целесообразным его разделение в ЛВС на два подуровня: *управление доступом к каналу* (MAC – Medium Access Control) и *управление логическим каналом* (LLC – Logical Link Control).

Передача данных через разветвленные сети происходит при использовании *инкапсуляции–декапсуляции* порций данных. Так, сообщение, пришедшее на транспортный уровень, делится на сегменты, которые получают заголовки и передаются на сетевой уровень. Сегментом обычно называют пакет транспортного уровня. Сетевой уровень организует передачу данных через промежуточные сети. Для этого сегмент может быть разделен на части (пакеты), если сеть не поддерживает передачу сегментов целиком. Пакет снабжается своим сетевым заголовком (т. е. происходит инкапсуляция). При передаче между узлами промежуточной ЛВС требуется инкапсуляция пакетов

в кадры с возможной разбивкой пакета. Приемник декапсулирует сегменты и восстанавливает исходное сообщение.

## **Тема 5. Аппаратура рабочих мест в автоматизированных системах проектирования и управления**

**Вычислительные системы в САПР.** В качестве средств обработки данных в современных САПР широко используют рабочие станции, серверы, персональные компьютеры. Большие ЭВМ и в том числе супер ЭВМ обычно не применяют, так как они дороги и их отношение производительность/цена существенно ниже подобного показателя серверов и многих рабочих станций. На базе рабочих станций или персональных компьютеров создают АРМ.

Типичный состав устройств АРМ: ЭВМ с одним или несколькими микропроцессорами, оперативной и кэш-памятью и шинами, служащими для взаимной связи устройств; устройства ввода-вывода, включающие в себя, как минимум, клавиатуру, мышь, дисплей; дополнительно в состав АРМ могут входить принтер, сканер, плоттер (графопостроитель), дигитайзер и некоторые другие периферийные устройства.

Память ЭВМ обычно имеет иерархическую структуру. Поскольку в памяти большого объема трудно добиться одновременно высокой скорости записи и считывания данных, память делят на сверхбыстродействующую кэш-память малой емкости, основную оперативную память умеренного объема и сравнительно медленную внешнюю память большой емкости, причем, в свою очередь, кэш-память часто разделяют на кэш первого и второго уровней.

Например, в персональных компьютерах на процессорах Pentium III кэш первого уровня имеет по 16 Кбайт для данных и для адресов, кэш-память первого и второго уровней емкостью 256 Кбайт встроены в процессорный кристалл, емкость оперативной памяти составляет десятки-сотни Мбайт.

Для связи наиболее быстродействующих устройств (процессора, оперативной и кэш-памяти, видеокарты) используется системная шина с пропускной способностью до одного-двух Гбайт/с. Кроме системной шины на материнской плате компьютера имеются шина расширения для подключения сетевого контроллера и быстрых внешних устройств (например, шина PCI с пропускной способностью

133 Мбайт/с) и шина медленных внешних устройств, таких как клавиатура, мышь, принтер и т. п.

*Рабочие станции (workstation)* по сравнению с персональными компьютерами представляют собой вычислительную систему, специализированную на выполнение определенных функций. Специализация обеспечивается как набором программ, так и аппаратно за счет использования дополнительных специализированных процессоров. Так, в САПР для машиностроения преимущественно применяют графические рабочие станции для выполнения процедур геометрического моделирования и машинной графики. Эта направленность требует мощного процессора, высокоскоростной шины, памяти достаточно большой емкости.

Высокая производительность процессора необходима по той причине, что графические операции (например, перемещения изображений, их повороты, удаление скрытых линий и др.) часто выполняются по отношению ко всем элементам изображения. Такими элементами в трехмерной (3D) графике при аппроксимации поверхностей полигональными сетками являются многоугольники, их число может превышать  $10^4$ . С другой стороны, для удобства работы проектировщика в интерактивном режиме задержка при выполнении команд указанных выше операций не должна превышать нескольких секунд. Но поскольку каждая такая операция по отношению к каждому многоугольнику реализуется большим числом машинных команд, требуемое быстродействие составляет десятки миллионов машинных операций в секунду. Такое быстродействие при приемлемой цене достигается применением наряду с основным универсальным процессором также дополнительных специализированных (графических) процессоров, в которых определенные графические операции реализуются аппаратно.

В наиболее мощных рабочих станциях в качестве основных обычно используют высокопроизводительные микропроцессоры с сокращенной системой команд (с RISC-архитектурой), работающие под управлением одной из разновидностей операционной системы Unix. В менее мощных все чаще используют технологию Wintel (т. е. микропроцессоры Intel и операционные системы Windows). Графические процессоры выполняют такие операции, как растеризация – представление изображения в растровой форме для ее визуализации, перемещение, вращение, масштабирование, удаление скрытых линий и т. п.

Типичные характеристики рабочих станций: несколько процессоров, десятки-сотни мегабайт оперативной и тысячи Мбайт внешней памяти, наличие кэш-памяти, системная шина со скоростями от сотен Мбайт/с до 1–2 Гбайт/с.

В зависимости от назначения существуют АРМ конструктора, АРМ технолога, АРМ руководителя проекта и т.п. Они могут различаться составом периферийных устройств, характеристиками ЭВМ.

В АРМ конструктора (графических рабочих станциях) используются растровые мониторы с цветными трубками. Типичные значения характеристик мониторов находятся в следующих пределах: размер экрана по диагонали 17...24 дюйма. Разрешающая способность монитора, т. е. число различимых пикселей (отдельных точек, из которых состоит изображение), определяется шагом между отверстиями в маске, через которые проходит к экрану электронный луч в электронно-лучевой трубке. Этот шаг находится в пределах 0,21–0,28 мм, что соответствует количеству пикселей изображения от 800 x 600 до 1920 x 1200 и более. Чем выше разрешающая способность, тем шире должна быть полоса пропускания электронных блоков видеосистемы при одинаковой частоте кадровой развертки.

Отметим, что чем ниже частота кадровой развертки, а это есть частота регенерации изображения, тем заметнее мерцание экрана. Желательно, чтобы эта частота была не ниже 75 Гц.

Специально выпускаемые ЭВМ как серверы высокой производительности обычно имеют структуру симметричной многопроцессорной вычислительной системы. В них системная память разделяется всеми процессорами, каждый процессор может иметь свою сверхоперативную память сравнительно небольшой емкости, число процессоров невелико (единицы, редко более десяти). Например, сервер Enterprise 250 (Sun Microsystems) имеет 1–2 процессора, его цена в зависимости от комплектации колеблется в диапазоне 24–56 тыс. долларов, а сервер Enterprise 450 с четырьмя процессорами стоит от 82 до 95 тыс. долларов.

**Периферийные устройства.** Для автоматического ввода информации с имеющихся текстовых или графических документов используют сканеры планшетного или протяжного типа. Способ считывания – оптический. В сканирующей головке размещаются оптоволоконные самофокусирующиеся линзы и фотоэлементы. Разрешающая способность в разных моделях составляет от 300 до 800 точек на дюйм (dpi). Считанная информация имеет растровую форму, про-

граммное обеспечение сканера представляет ее в одном из стандартных форматов, например TIFF, GIF, PCX, JPEG, и для дальнейшей обработки может выполнить векторизацию – перевод графической информации в векторную форму, например, в формат DXF.

Для вывода информации применяют принтеры и плоттеры. Первые из них ориентированы на получение документов малого формата (А3, А4), вторые – для вывода графической информации на широкоформатные носители.

В этих устройствах преимущественно используется растровый (т. е. построчный) способ вывода со струйной технологией печати. Печатающая система в струйных устройствах включает в себя картридж и головку. Картридж – баллон, заполненный чернилами (в цветных устройствах имеется несколько картриджей, каждый с чернилами своего цвета). Головка – матрица из сопел, из которых мельчайшие чернильные капли поступают на носитель. Физический принцип действия головки термический или пьезоэлектрический. При термопечати выбрасывание капель из сопла происходит под действием его нагревания, что вызывает образование пара и выбрасывание капелек под давлением. При пьезоэлектрическом способе пропускание тока через пьезоэлемент приводит к изменению размера сопла и выбрасыванию капли чернил. Второй способ дороже, но позволяет получить более высококачественное изображение.

Типичная разрешающая способность принтеров и плоттеров 300 dpi, в настоящее время она повышена до 720 dpi. В современных устройствах управление осуществляется встроенными микропроцессорами. Типичное время вывода монохромного изображения формата А1 находится в пределах от 2 до 7 мин, цветного – в два раза больше. Сканеры, принтеры, плоттеры могут входить в состав АРМ или разделяться пользователями нескольких рабочих станций в составе локальной вычислительной сети.

**Особенности технических средств в АСУТП.** Специфические требования предъявляют к вычислительной аппаратуре, работающей в составе АСУТП в цеховых условиях. Здесь используют как обычные персональные компьютеры, так и специализированные программируемые логические контроллеры (ПЛК), называемые промышленными компьютерами. Специфика ПЛК – наличие нескольких аналоговых и цифровых портов, встроенный интерпретатор специализированного языка, детерминированные задержки при обработке сигналов, требующих незамедлительного реагирования. Однако ПЛК в от-

личие от персональных компьютеров IBM PC рассчитаны на решение ограниченного круга задач в силу специализированности программного обеспечения.

В целом промышленные компьютеры имеют следующие особенности: работа в режиме реального времени; конструкция, приспособленная для работы ЭВМ в цеховых условиях (повышенные вибрации, электромагнитные помехи, запыленность, перепады температур); возможность встраивания дополнительных блоков управляющей, регистрирующей, сопрягающей аппаратуры; автоматический перезапуск компьютера в случае «зависания» программы; повышенные требования к надежности функционирования.

В значительной мере специализация ПК определяется программным обеспечением. Конструктивно промышленный компьютер представляет собой корзину (крейт) с несколькими гнездами (слотами) для встраиваемых плат. Возможно использование мостов между крейтами. В качестве стандартных шин в настоящее время преимущественно используются шины VME-bus (Versabus Module Europe-bus) и PCI (Peripheral Component Interconnect).

VME-bus – системная шина для создания распределенных систем управления на основе встраиваемого оборудования (процессоры, накопители, контроллеры ввода–вывода). Представляет собой расширение локальной шины компьютера на несколько гнезд объединительной платы, возможно построение многомастерных систем, т. е. систем, в которых ведущими могут быть два или более устройств. Пропускная способность шины 320 Мбайт/с.

PCI – более удобная шина для однопроцессорных архитектур, получает все большее распространение. Пропускная способность до 264 Мбайт/с, разрядность шины 2 x 32 и (или) при мультиплексировании 64, архитектура с одним ведущим устройством. Имеется ряд разновидностей шины, например шина Compact PCI, в которой унифицирован ряд геометрических и механических параметров.

Программная связь с аппаратурой нижнего уровня (датчиками, исполнительными устройствами) происходит через драйверы. Межпрограммные связи реализуются через интерфейсы, подобные OLE. Для упрощения создания систем разработан стандарт OPC (OLE for Process Control).

## **Тема 6. Методы доступа в локальных вычислительных сетях**

**Множественный доступ с контролем несущей и обнаружением конфликтов.** Типичная среда передачи данных в ЛВС – отрезок (сегмент) коаксиального кабеля. К нему через аппаратуру окончания канала данных подключаются узлы – компьютеры и, возможно, общее периферийное оборудование. Поскольку среда передачи данных общая, а запросы на сетевые обмены в узлах появляются асинхронно, то возникает проблема разделения общей среды между многими узлами, другими словами, проблема обеспечения доступа к сети.

Доступом к сети называют взаимодействие станции (узла сети) со средой передачи данных для обмена информацией с другими станциями. Управление доступом к сети – это установление последовательности, в которой станции получают доступ к среде передачи данных.

Различают случайные и детерминированные методы доступа. Среди случайных методов наиболее известен метод множественного доступа с контролем несущей и обнаружением конфликтов (МДКН/ОК). Англоязычное название метода – Carrier Sense Multiple Access/Collision Detection (CSMA/CD). Этот метод основан на контроле наличия электрических колебаний (несущей) в линии передачи данных и устранении конфликтов, возникающих в случае попыток одновременного начала передачи двумя или более станциями, путем повторения попыток захвата линии через случайный отрезок времени.

МДКН/ОК является широкоэмитальным (broadcasting) методом. Все станции при применении МДКН/ОК равноправны по доступу к сети. Если линия передачи данных свободна, то в ней отсутствуют электрические колебания, что легко распознается любой станцией, желающей начать передачу. Такая станция захватывает линию. Любая другая станция, желающая начать передачу в некоторый момент времени  $t$ , если обнаруживает электрические колебания в линии, то откладывает передачу до момента  $t + t_d$ , где  $t_d$  – задержка.

При работе сети каждая станция анализирует адресную часть передаваемых по сети кадров с целью обнаружения и приема кадров, предназначенных для нее.

Конфликтом называют ситуацию, при которой две или более станции «одновременно» пытаются захватить линию. Понятие «одновременность событий» в связи с конечностью скорости распрост-

ранения сигналов по линии конкретизируется как отстояние событий во времени не более чем на величину  $2d$ , называемую *окном столкновений*, где  $d$  – время прохождения сигналов по линии между конфликтующими станциями. Если какие-либо станции начали передачу в окне столкновений, то по сети распространяются искаженные данные. Это искажение и используют для обнаружения конфликта либо сравнением в передатчике данных, передаваемых в линию (неискаженных) и получаемых из нее (искаженных), либо по появлению постоянной составляющей напряжения в линии, что обусловлено искажением используемого для представления данных манчестерского кода. Обнаружив конфликт, станция должна оповестить об этом партнера по конфликту, послав дополнительный сигнал затора, после чего станции должны отложить попытки выхода в линию на время  $t_d$ . Очевидно, что значения  $t_d$  должны быть различными для станций, участвующих в столкновении (конфликте), поэтому  $t_d$  – случайная величина.

**Маркерные методы доступа.** Среди детерминированных методов преобладают *маркерные методы доступа*. Маркерный метод – метод доступа к среде передачи данных в ЛВС, основанный на передаче полномочий передающей станции с помощью специального информационного объекта, называемого маркером. Под полномочием понимается право инициировать определенные действия, динамически предоставляемые объекту, например станции данных в информационной сети.

Применяется ряд разновидностей маркерных методов доступа. Например, в *эстафетном методе* передача маркера выполняется в порядке очередности; в способе *селекторного опроса* (квантированной передачи) сервер опрашивает станции и передает полномочие одной из тех станций, которые готовы к передаче. В кольцевых одноранговых сетях широко применяют тактируемый маркерный доступ, при котором маркер циркулирует по кольцу и используется станциями для передачи своих данных.

## **Тема 7. Локальные вычислительные сети Ethernet**

**Состав аппаратуры.** Одной из первых среди ЛВС шинной структуры была создана сеть Ethernet, разработанная фирмой Xerox. В этой сети был применен метод доступа МДКН/ОК. Позднее Ethernet стала основой стандарта IEEE 802/3. Другой вариант шинных ЛВС

соответствует стандарту IEEE 802/4, описывающему сеть с эстафетной передачей маркера. Технология Ethernet наиболее распространена в ЛВС и по данным на 2006 г. 87 % всех компьютеров были в сетях Ethernet.

В качестве линий передачи данных в ЛВС используют коаксиальный кабель, витую пару проводов или ВОЛС. Длины отрезков коаксиального кабеля не должны превышать нескольких сотен метров, а у витой пары проводов – десятков метров. При больших расстояниях в среду передачи данных включают формирователи сигналов – повторители для сопряжения отрезков. ВОЛС позволяет существенно увеличить предельные расстояния и скорость передачи данных.

Для связи компьютеров со средой передачи данных используют сетевые контроллеры (адаптеры, сетевые карты), управляющие доступом к сети, и приемопередатчики, служащие для связи сетевого контроллера с линией связи.

*Сетевой контроллер* реализует принятый метод доступа к каналу, а также в случае метода МДКН/ОК осуществляет действия по выработке сигнала затора, задержке в передаче при наличии конфликта или при занятом моноканале, по формированию кадров, кодированию (декодированию) электрических сигналов в (из) специальный последовательный код, называемый манчестерским, распознаванию адреса в передаваемых по сети сообщениях.

После образования информационного кадра станция должна получить полномочия. Для этого контроллер прослушивает канал в ожидании его освобождения или прихода маркера. После получения полномочий осуществляется преобразование параллельного кода в последовательный, преобразование в манчестерский код и передача сигналов в кабель.

В состав приемопередатчика в шинных ЛВС с методом МДКН/ОК входят приемник сигналов от линии и передатчик сигналов от станции в линию. Назначение приемника – усиление информационных сигналов и обнаружение конфликтов путем выделения постоянной составляющей искаженных сигналов и ее сопоставления в компараторе с эталонным напряжением.

**Разновидности сетей Ethernet.** Рядом фирм на базе проекта сети Ethernet разрабатывается оборудование для ЛВС. В настоящее время унифицировано несколько вариантов сети Ethernet, различающихся топологией, особенностями физической среды передачи данных, информационной скоростью передачи данных:

1. *Thick Ethernet* (шина «с толстым» кабелем); принятое обозначение варианта 10Base-5, где первый элемент «10» характеризует скорость передачи данных по линии 10 Мбит/с, последний элемент «5» – максимальную длину сегмента кабеля (в сотнях метров), т. е. 500 м. Другие параметры сети: максимальное число сегментов 5; максимальное число узлов на одном сегменте 100; минимальное расстояние между узлами 2,5 м. Здесь под сегментом кабеля понимается часть кабеля, используемая в качестве линии передачи данных и имеющая на концах согласующие элементы (терминаторы) для предотвращения отражения сигналов.

2. *Thin Ethernet* (шина «с тонким» кабелем, *cheapernet*); принятое обозначение 10Base-2: максимальное число сегментов 5; максимальная длина сегмента 185 м; максимальное число узлов на одном сегменте 30; минимальное расстояние между узлами 0,5 м; скорость передачи данных по линии 10 Мбит/с.

3. *Twisted Pair Ethernet*; принятое обозначение 10Base – это кабельная сеть с использованием витых пар проводов и концентраторов, называемых также распределителями, или *хабами* (*hubs*). Структура сети представлена на рис. 2.4. В состав сетевого оборудования входят активные (АН) и пассивные (РН) концентраторы (*active and passive hubs*), различие между которыми заключается в наличии или отсутствии усиления сигналов и в количестве портов. Число портов в активных хабах обычно составляет 8, 12 или 16. В одной из разновидностей сети 10Base-T допускаются расстояния между активными распределителями до 600 м и между пассивными до 30 м, предельное число узлов 100. Физическая организация линий связи в 10Base-T мало напоминает шину. Однако в такой сети вполне возможна реализация метода доступа МДКН/ОК, и для пользователя (любого отдельного узла) разветвленная сеть из витых пар и концентраторов, по которой происходит широковещательная передача, есть просто среда передачи данных, такая же как шина. Поэтому по логической организации сеть 10Base-T представляет собой сеть типа Ethernet. В то же время по своей топологии 10Base-T может быть вариантом «звезда», «дерево» и т. п. В этой сети не рекомендуется включать последовательно более четырех хабов.

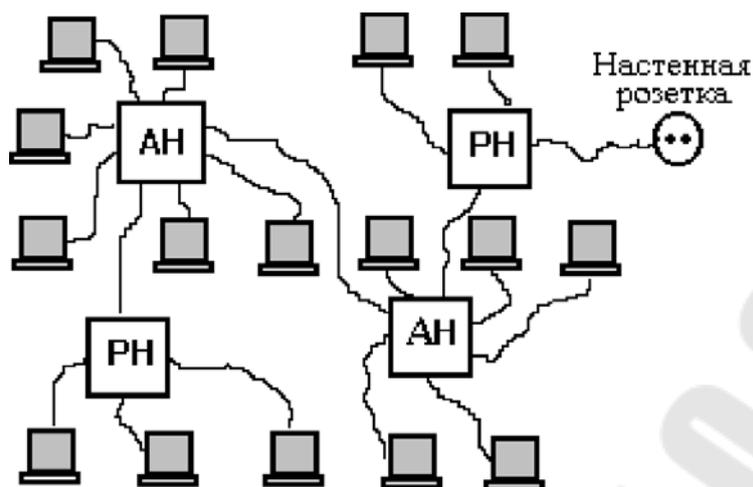


Рис. 2.4. Среда передачи данных на витой паре и концентраторах

4. *Fiber Optic Ethernet* (шина на основе оптоволоконного кабеля), обозначение 10Base-F; применяется для соединений точка–точка, например, для соединения двух конкретных распределителей в кабельной сети. Максимальные длины – в пределах 2...4 км. Цена оптоволоконного кабеля приблизительно такая же, как и медного кабеля, но у первого из них меньше габариты и масса, достигается полная гальваническая развязка. Приемопередатчик (повторитель) для волоконно-оптических линий передачи данных (световодов) состоит из частей приемной, передающей, чтения и записи данных. В приемной части имеются фотодиод, усилитель-формирователь сигналов с требуемыми уровнями напряжения, механическое контактирующее устройство для надежного контакта фотодиода со стеклянной оболочкой кабеля. Передатчик представлен светодионом или микролазером.

5. *RadioEthernet* (стандарт IEEE 802/11). Среда передачи данных – радиоволны, распространяющиеся в эфире. Структура сети может быть «постоянной» при наличии базовой кабельной сети с точками доступа от узлов по радиоканалам или «временной», когда обмены между узлами происходят только по радиоканалам. Применяется модифицированный метод МДКН/ОК, в котором вместо обнаружения конфликтов используется предотвращение конфликтов. Это осуществляется следующим образом: узел, запрашивающий связь, посылает в эфир специальный кадр запроса, а передачу информации он может начать только после истечения межкадрового промежутка времени  $T$ , если за время  $T$  после запроса в эфире не было других запросов. Иначе попытка передачи откладывается на случайное время. Любой узел может посылать кадр запроса, только если за время  $T$  перед этим в эфире не было других кадров запроса.

Предусмотрена посылка положительной квитанции от приемного узла, подтверждающая правильность приема кадра. Квитанция посылается с малой задержкой  $t$  после окончания приема. В этом интервале длительностью  $t$  конфликты невозможны, так как претенденты на передачу могут посылать кадры запроса только в том случае, если перед посылкой эфир свободен в течение интервала времени не менее  $T$  (это условие выполняется и для узлов с отложенной из-за конфликта передачей), а  $t < T$ .

6. Семь *Fast Ethernet*, иначе называемая 100BaseX, или 100Base-T (стандарт IEEE 802/30). Информационная скорость 100 Мбит/с. В этой сети применен метод доступа МДКН/ОК. Используется для построения скоростных ЛВС (последовательно включается не более двух хабов), для объединения низкоскоростных подсетей 10Base-T в единую скоростную сеть и для подключения серверов на расстояниях до 200 м. В последнем случае серверы соединяются с клиентскими узлами через шину 100 Мбит/с и коммутатор, называемый также конвертором, преобразователем или переключателем скорости 100/10. К конвертору с другой стороны подключено несколько шин 10 Мбит/с, на которые нагружены остальные узлы. Практически можно использовать до 250 узлов, теоретически – до 1024. Подсетями могут быть как *Fast Ethernet*, так и обычные *Ethernet* со скоростью 10 Мбит/с, включенные через преобразователь скорости. Различают следующие варианты: 100Base-TX, в котором применяют кабель из двух неэкранированных витых пар категории 5, 100Base-T4 – с четырьмя неэкранированными парами категории 5, 100Base-FX – на ВОЛС.

7. *Gigabit Ethernet 1000Base-X*. В этом варианте получены гигабитные скорости. В соответствии со стандартом IEEE 802.3z имеются разновидности на ВОЛС с длиной волны 830 или 1270 нм (1000Base-SX и 1000Base-LX соответственно) на расстояниях до 550 м и на витой паре категории 5 (1000Base-CX) на расстояниях до 25 м. Скорость до 1 Гбит/с. Такая скорость достигается благодаря следующим решениям.

Сеть имеет иерархическую структуру. Участки (отдельные компьютеры или подсети) по 10 Мбит/с подключаются к портам переключателей (switches) скорости 10/100, их выходы по 100 Мбит/с, в свою очередь, подключаются к портам переключателей 100/1000. В сегментах сети, имеющих 1000 Мбит/с, используются, во-первых, передача данных по ВОЛС или параллельно по четырем витым парам,

во-вторых, 5-уровневое представление данных (например, +2, +1, 0, -1, -2 В), в-третьих, кодирование 8b/10b. В результате в каждой витой паре имеем 250 Мбит/с при частоте сигналов 125 МГц, а это уже приемлемая частота для передачи по проводным соединениям.

## Тема 8. Сети кольцевой топологии

**Сеть Token Ring.** Из кольцевых ЛВС наиболее распространены сети с передачей маркера по кольцу и среди них: 1) ЛВС типа Token Ring (сеть с таким названием была разработана фирмой IBM и послужила основой для стандарта IEEE 802/5); 2) сети FDDI (Fiber Distributed Data Interface) на основе ВОЛС.

Топология сети Token Ring показана на рис. 2.5, а. Концентраторы служат для удобства управления сетью, в частности для отключения от кольца неисправных узлов. На рис. 2.5, б представлена схема подключения узлов к кольцу в концентраторах. Для отключения узла достаточно левые переключатели (рис. 2.5, б) поставить в верхнее, а правые переключатели – в нижнее положение (в нормальном режиме положение переключателей противоположное).

Типичная реализация сети Token Ring характеризуется следующими данными: максимальное число станций 96; максимальное число концентраторов 12; максимальная длина замыкающего кабеля 120 м; максимальная длина кабеля между двумя концентраторами или между концентратором и станцией 45 м; два варианта скорости передачи данных по линии 4 или 16 Мбит/с.

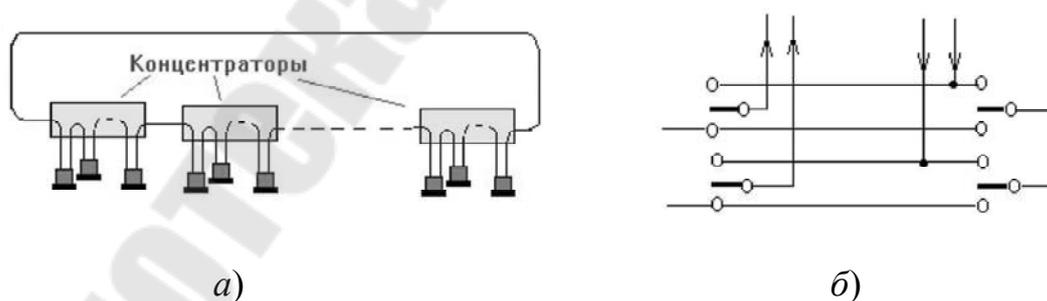


Рис. 2.5. Схема Token Ring: а – общий вид; б – схема подключения узла к кольцу

Функционирование сети заключается в следующем.

В кольцевых локальных сетях сигналы циркулируют по кольцу, состоящему из ряда отрезков линии связи, которые соединяют пары соседних узлов. Эти отрезки соединяются в узлах через повторители

сигналов, выполняющих функции приема и передачи сигналов как из кольца в кольцо, так и между АКД и линией. Повторители вносят некоторую задержку в передачу сигналов, поэтому общая задержка зависит от числа станций, включенных в кольцо.

Одним из способов взаимосвязи линии и АКД является способ вставки регистра. Станцию, получившую полномочия, называют *активной станцией*. Активная станция осуществляет вставку регистра в разрыв кольца и подключает передающий регистр, из которого в кольцо посылается передаваемый кадр.

Эти регистры являются сдвигающими. Кадр проходит через кольцо и возвращается на вставленный регистр. По пути его адресная часть проверяется остальными станциями, поскольку в них предусмотрена расшифровка адресной и управляющей информации. Если пакет предназначен данной станции, то принимается информационная часть пакета, проверяется правильность приема, и при положительном результате проверки в кольцо направляется соответствующее подтверждение. Передающая (активная) станция одновременно с передачей сформированного в ней пакета принимает пакет, прошедший по кольцу, на вставленный регистр. В каждом такте сдвига в кольцо направляется очередной бит данных, а из кольца с некоторой задержкой возвращаются переданные биты. Если подтверждена правильность передачи, то переданные данные стираются в передающей станции, которая направляет в кольцо свободный маркер, если не подтверждена, то осуществляется повторная передача пакета.

Станции, готовые к передаче собственных данных, ждут прихода свободного маркера. Станция, получившая полномочия, вставляет свой регистр в кольцо, становясь активной, а вставленный ранее регистр исключается из кольца.

Циркулирующий по сети маркер состоит из следующих частей:

$\langle \text{ограничитель} - P - \langle - M - R - \text{ограничитель} \rangle \rangle$ .

Если  $T = 0$ , то маркер свободен. Тогда если он проходит мимо станции, имеющей данные для передачи, и приоритет станции не ниже значения, записанного в  $P$ , то станция преобразует маркер в информационный кадр: устанавливает  $T = 1$  и записывает между  $R$  и конечным ограничителем адрес получателя, данные и другие сведения в соответствии с принятой структурой кадра. Информационный кадр проходит по кольцу, при этом происходит следующее: 1) каждая станция, готовая к передаче, записывает значение своего приоритета в  $R$ , если ее приоритет выше уже записанного в  $R$  значения; 2) стан-

ция-получатель, распознав свой адрес, считывает данные и отмечает в конце кадра (в бите «статус кадра») факт приема данных.

Совершив полный оборот по кольцу, кадр приходит к станции-отправителю, которая анализирует состояние кадра. Если передача не произошла, то делается повторная попытка передачи. Если произошла, то кадр преобразуется в маркер указанной выше структуры с  $T = 0$ . При этом также происходят действия:

$$P = R; R = 0,$$

где  $P$  и  $R$  – трехбитовые коды.

При следующем обороте маркер будет захвачен той станцией-претендентом, у которой на предыдущем обороте оказался наивысший приоритет (именно его значение записано в  $P$ ).

Сеть Token Ring рассчитана на меньшие предельные расстояния и число станций, чем сеть Ethernet, но лучше приспособлена к повышенным нагрузкам.

**Сеть FDDI.** Сеть FDDI (*Fiber Distributed Data Interface*) относится к высокоскоростным сетям, имеет кольцевую топологию, использует ВОЛС и специфический вариант маркерного метода доступа.

В основном варианте сети применено двойное кольцо на ВОЛС. Обеспечивается информационная скорость 100 Мбит/с. Расстояние между крайними узлами – до 200 км, между соседними станциями – не более 2 км. Максимальное число узлов 500. В ВОЛС применяются волны длиной 1300 нм.

Два кольца ВОЛС используются одновременно. Станции можно подключать к одному из колец или к обоим сразу (рис. 2.6, а). Использование конкретным узлом обоих колец позволяет получить для этого узла суммарную пропускную способность 200 Мбит/с. Другое возможное использование второго кольца – обход с его помощью поврежденного участка путем объединения колец, как показано на рис. 2.6, б.

В сети FDDI используются оригинальные код и метод доступа. Применяется код типа NRZ (без возвращения к нулю), в котором изменение полярности в очередном такте времени воспринимается как 1, отсутствие изменения полярности как 0. Чтобы код был самосинхронизирующимся, после каждых четырех битов передатчик вырабатывает синхронизирующий перепад.

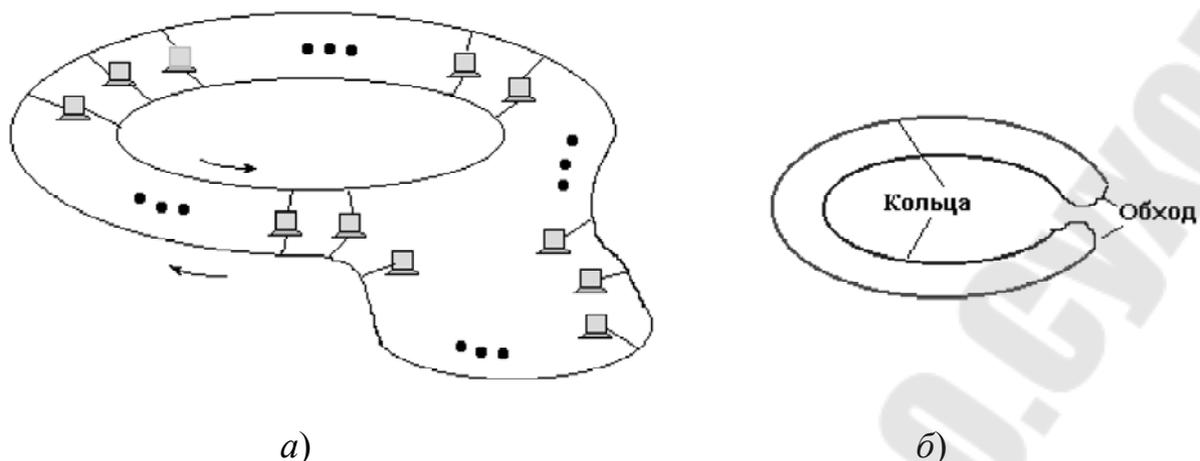


Рис. 2.6. Кольца ВОЛС в сети FDDI: а – включение узлов; б – обход поврежденного участка

Такое специальное манчестерское кодирование называют 4b/5b. Запись 4b/5b означает код, в котором для самосинхронизации при передаче четырех битов двоичного кода используется пять битов так, что не может быть более двух нулей подряд, или после четырех битов добавляется еще один обязательный перепад, что и используется в FDDI. При использовании такого кода несколько усложняются блоки кодирования и декодирования, но при этом повышается скорость передачи по линии связи, так как почти вдвое уменьшается максимальная частота переключения по сравнению с манчестерским кодом.

В соответствии с методом FDDI по кольцу циркулирует пакет, состоящий из маркера и информационных кадров. Любая станция, готовая к передаче, распознав проходящий через нее пакет, вписывает свой кадр в конец пакета. Она же ликвидирует его после того, как кадр вернется к ней после полного оборота по кольцу и при условии, что он был воспринят получателем. Если обмен происходит без сбоев, то кадр, возвращающийся к станции-отправителю, оказывается в пакете уже первым, так как все предшествующие кадры должны быть ликвидированы раньше.

Сеть FDDI обычно используют как объединяющую в единую сеть многих отдельных подсетей ЛВС. Например, при организации информационной системы крупного предприятия целесообразно иметь ЛВС типа Ethernet или Token Ring в помещениях отдельных проектных подразделений, а связь между подразделениями осуществлять через сеть FDDI.

## Глава 3. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ АНАЛИЗА ПРОЕКТНЫХ РЕШЕНИЙ

### Тема 9. Компоненты математического обеспечения

**Математический аппарат в моделях разных иерархических уровней.** К МО анализа относят математические модели, численные методы, алгоритмы выполнения проектных процедур. Компоненты МО определяются базовым математическим аппаратом, специфичным для каждого из иерархических уровней проектирования.

На *микроуровне* типичные математические модели (ММ) представлены дифференциальными уравнениями в частных производных (ДУЧП) вместе с краевыми условиями. К этим моделям, называемым *распределенными*, относятся многие уравнения математической физики. Объектами исследования здесь являются поля физических величин, что требуется при анализе прочности строительных сооружений или машиностроительных деталей, исследовании процессов в жидких средах, моделировании концентраций и потоков частиц и т. п.

Число совместно исследуемых различных сред (число деталей, слоев материала, фаз агрегатного состояния) в практически используемых моделях микроуровня не может быть большим из-за сложностей вычислительного характера. Снизить вычислительные затраты в многокомпонентных средах можно только применив иной подход к моделированию, основанный на принятии определенных допущений.

Допущение, выражаемое дискретизацией пространства, позволяет перейти к моделям *макроуровня*. Моделями макроуровня, называемыми также *сосредоточенными*, являются системы алгебраических и обыкновенных дифференциальных уравнений, поскольку независимой переменной здесь остается только время  $t$ . Упрощение описания отдельных компонентов (деталей) позволяет исследовать модели процессов в устройствах, приборах, механических узлах, число компонентов в которых может достигать до нескольких тысяч.

В тех случаях, когда число компонентов в исследуемой системе превышает некоторый порог, сложность модели системы на макро-

уровне вновь становится чрезмерной. Поэтому, принимая соответствующие допущения, переходят на функционально-логический уровень. На этом уровне используют аппарат передаточных функций для исследования аналоговых (непрерывных) процессов или аппарат математической логики и конечных автоматов, если объектом исследования является дискретный процесс, т. е. процесс с дискретным множеством состояний.

Наконец, для исследования еще более сложных объектов, примерами которых могут служить производственные предприятия и их объединения, вычислительные системы и сети, социальные системы и другие подобные объекты, применяют аппарат теории массового обслуживания, возможно использование и некоторых других подходов, например, сетей Петри. Эти модели относятся к системному уровню моделирования.

**Требования к математическим моделям и численным методам.** Основными требованиями к математическим моделям являются требования адекватности, точности, экономичности.

ММ всегда лишь приближенно отражает некоторые свойства объекта. *Адекватность* имеет место, если ММ отражает заданные свойства объекта с приемлемой точностью. Под *точностью* понимают степень соответствия оценок одноименных свойств объекта и ММ.

*Экономичность (вычислительная эффективность)* определяется затратами ресурсов, требуемых для реализации модели. Поскольку в САПР используются математические модели, далее речь пойдет о характеристиках именно математических моделей, и экономичность будет характеризоваться затратами машинного времени и памяти.

Адекватность оценивается перечнем отражаемых свойств и областями адекватности. *Область адекватности* – область в пространстве параметров, в пределах которой погрешности модели остаются в допустимых пределах. Например, область адекватности линейризованной модели поверхности детали определяется системой неравенств

$$\max |\varepsilon_{ij}| \leq \varepsilon_{\text{доп}},$$

где  $\varepsilon_{ij} = (X_{ij_{\text{ист}}} - X_{ij_{\text{мод}}}) / X_{ij_{\text{ист}}}$  и  $X_{ij_{\text{мод}}}$  –  $i$ -я координата  $j$ -й точки поверхности в объекте и модели соответственно,  $\varepsilon_{ij}$  и  $\varepsilon_{\text{доп}}$  – допущенная и предельно допустимая относительные погрешности моде-

лирования поверхности, максимум берется по всем координатам и контролируемым точкам.

Отметим, что в большинстве случаев области адекватности строятся в пространстве внешних переменных. Так, область адекватности модели электронного радиоэлемента обычно выражает допустимые для применения модели диапазоны изменения моделируемых температур, внешних напряжений, частот. Аналогичные требования по точности и экономичности фигурируют при выборе численных методов решения уравнений модели.

**Место процедур формирования моделей в маршрутах проектирования.** Вычислительный процесс при анализе состоит из этапов формирования модели и ее исследования (решения). В свою очередь, формирование модели включает две процедуры: во-первых, разработку моделей отдельных компонентов, во-вторых, формирование модели системы из моделей компонентов.

Первая из этих процедур выполняется предварительно по отношению к типовым компонентам вне маршрута проектирования конкретных объектов. Как правило, модели компонентов разрабатываются специалистами в прикладных областях, причем знающими требования к моделям и формам их представления в САПР. Обычно в помощь разработчику моделей в САПР предлагаются методики и вспомогательные средства, например, в виде программ анализа для экспериментальной отработки моделей. Созданные модели включаются в библиотеки моделей прикладных программ анализа.

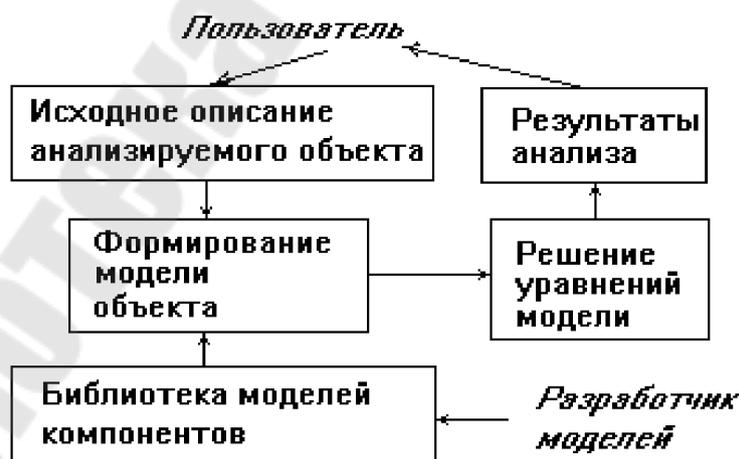


Рис. 3.1. Место процедур формирования моделей на маршрутах проектирования

На маршруте проектирования каждого нового объекта выполняется вторая процедура (рис. 3.1) – формирование модели системы с использованием библиотечных моделей компонентов. Как правило, эта процедура выполняется автоматически по алгоритмам, включенным в заранее разработанные программы анализа. Примеры таких программ имеются в различных приложениях и прежде всего в отраслях общего машиностроения и радиоэлектроники.

При применении этих программ пользователь описывает исследуемый объект на входном языке программы анализа в виде списка элементов структуры, эквивалентной схемы, эскиза или чертежа конструкции.

## **Тема 10. Математические модели анализа на макроуровне**

**Исходные уравнения моделей.** Исходное математическое описание процессов в объектах на макроуровне представлено системами обыкновенных дифференциальных и алгебраических уравнений. Аналитические решения таких систем при типичных значениях их порядков в практических задачах получить не удастся, поэтому в САПР преимущественно используются алгоритмические модели. В этом параграфе изложен обобщенный подход к формированию алгоритмических моделей на макроуровне, справедливый для большинства приложений.

Исходными для формирования математических моделей объектов на макроуровне являются компонентные и топологические уравнения. Компонентными уравнениями называют уравнения, описывающие свойства элементов (компонентов), другими словами, это уравнения математических моделей элементов (ММЭ). Топологические уравнения описывают взаимосвязи в составе моделируемой системы. В совокупности компонентные и топологические уравнения конкретной физической системы представляют собой исходную математическую модель системы (ММС).

Очевидно, что компонентные и топологические уравнения в системах различной физической природы отражают разные физические свойства, но могут иметь одинаковый формальный вид. Одинаковая форма записи математических соотношений позволяет говорить о формальных аналогиях компонентных и топологических уравнений. Такие аналогии существуют для механических поступа-

тельных, механических вращательных, электрических, гидравлических (пневматических), тепловых объектов. Наличие аналогий приводит к практически важному выводу: значительная часть алгоритмов формирования и исследования моделей в САПР оказывается инвариантной и может быть применена к анализу проектируемых объектов в разных предметных областях. Единство математического аппарата формирования ММС особенно удобно при анализе систем, состоящих из физически разнородных подсистем.

Компонентные уравнения имеют вид

$$F_k(dV/dt, V, t) = 0, \quad (3.1)$$

а топологические уравнения

$$F_m(V) = 0, \quad (3.2)$$

где  $V = (v_1, v_2, \dots, v_n)$  – вектор фазовых переменных;  $t$  – время.

Различают фазовые переменные двух типов, их обобщенные наименования – фазовые переменные типа потенциала (например, электрическое напряжение) и типа потока (например, электрический ток). Каждое компонентное уравнение характеризует связи между разнотипными фазовыми переменными, относящимися к одному компоненту (например, закон Ома описывает связь между напряжением и током в резисторе), а топологическое уравнение – связи между однотипными фазовыми переменными в разных компонентах.

Математические модели можно представлять в виде систем уравнений или в графической форме, если между этими формами установлено взаимно однозначное соответствие. В качестве графической формы часто используют эквивалентные схемы.

**Примеры компонентных и топологических уравнений. Электрические системы.** Здесь фазовыми переменными являются электрические напряжения и токи. Компонентами систем могут быть простые двухполюсные элементы и более сложные двух- и многополюсные компоненты. К простым двухполюсникам относятся следующие элементы: сопротивление, емкость и индуктивность, характеризуемые одноименными параметрами  $R$ ,  $C$ ,  $L$ . В эквивалентных схемах эти элементы обозначают в соответствии с рис. 3.2, а. Компонентные уравнения простых двухполюсников:

$$\text{для } R : u = iR \text{ (закон Ома);} \quad (3.3)$$

$$\text{для } C : i = Cdu/dt; \quad (3.4)$$

$$\text{для } L: u = L di / dt, \quad (3.5)$$

где  $u$  – напряжение;  $i$  – ток.

Эти модели лежат в основе моделей других возможных более сложных компонентов. Большая сложность может определяться нелинейностью уравнений (3.3)–(3.5) (т. е. зависимостью  $R$ ,  $C$ ,  $L$  от фазовых переменных), или учетом зависимостей параметров  $R$ ,  $C$ ,  $L$  от температуры, или наличием более двух полюсов. Однако многополюсные компоненты  $C$  могут быть сведены к совокупности взаимосвязанных простых элементов.

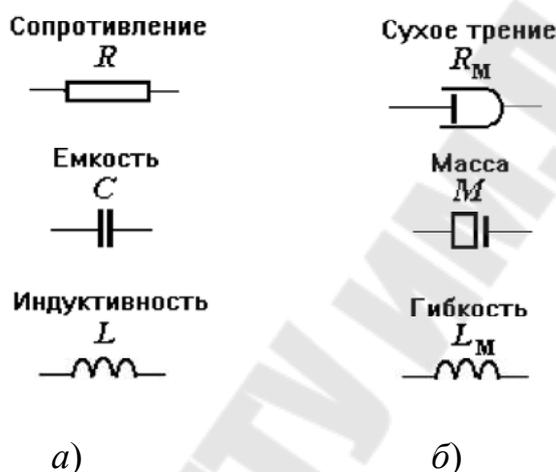


Рис. 3.2. Условные обозначения простых элементов в эквивалентных схемах: *a* – электрических, гидравлических, тепловых; *б* – механических

*Топологические уравнения* выражают законы Кирхгофа для напряжений (ЗНК) и токов (ЗТК). Согласно ЗНК, сумма напряжений на компонентах вдоль любого замкнутого контура в эквивалентной схеме равна нулю, а в соответствии с ЗТК сумма токов в любом замкнутом сечении эквивалентной схемы равна нулю:

$$\sum_{k \in K_p} u_k = 0; \quad (3.6)$$

$$\sum_{j \in J_q} i_j = 0, \quad (3.7)$$

где  $K_p$  – множество номеров элементов  $p$ -го контура;  $J_q$  – множество номеров элементов, входящих в  $q$ -е сечение.

Примером ММ сложного компонента может служить модель транзистора. На рис. 3.3 представлена эквивалентная схема биполяр-

ного транзистора, на которой зависимые от напряжений источники тока  $i_{эд} = i_{тэ} \exp(u_э / (m\phi_T))$  и  $i_{кд} = i_{тк} \exp(u_к / (m\phi_T))$  отображают статические вольтамперные характеристики  $p-n$  переходов,  $i_{тэ}$  и  $i_{тк}$  – тепловые токи переходов,  $m\phi_T$  – температурный потенциал,  $u_э$  и  $u_к$  – напряжения на эмиттерном и коллекторном переходах,  $C_э$  и  $C_к$  – емкости переходов,  $R_{уэ}$  и  $R_{ук}$  – сопротивления утечки переходов,  $R_б$  и  $R_к$  – объемные сопротивления тел базы и коллектора,  $i_r = B i_{эд} - B_i i_{кд}$  – источник тока, моделирующий усилительные свойства транзистора,  $B$  и  $B_i$  – прямой и инверсный коэффициенты усиления тока базы. Здесь  $u_э$ ,  $u_к$ ,  $i_{эд}$ ,  $i_{кд}$ ,  $i_r$  – фазовые переменные, а остальные величины – параметры модели транзистора.

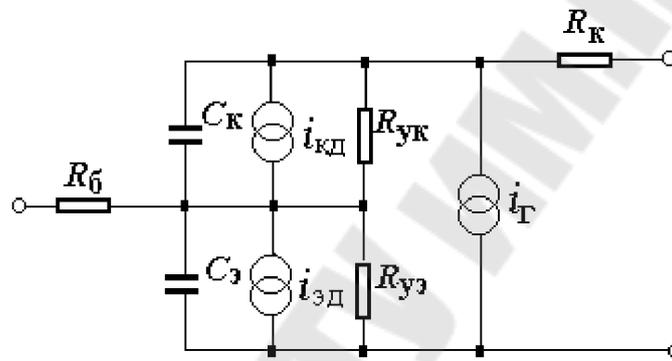


Рис. 3.3. Эквивалентная схема биполярного транзистора

*Механические системы.* Фазовыми переменными в механических поступательных системах являются силы и скорости. Используют одну из двух возможных электромеханических аналогий. В дальнейшем будем использовать ту из них, в которой скорость относят к фазовым переменным типа потенциала, а силу считают фазовой переменной типа потока. Учитывая формальный характер подобных аналогий, в равной мере можно применять и противоположную терминологию.

Компонентное уравнение, характеризующее инерционные свойства тел, в силу второго закона Ньютона имеет вид

$$F = Mdu / dt, \quad (3.8)$$

где  $F$  – сила;  $M$  – масса;  $u$  – поступательная скорость.

Упругие свойства тел описываются компонентным уравнением, которое можно получить из уравнения закона Гука. В одномерном случае (если рассматриваются продольные деформации упругого стержня)

$$G = E\varepsilon, \quad (3.9)$$

где  $G$  – механическое напряжение;  $E$  – модуль упругости;  $\varepsilon = \Delta l/l$  – относительная деформация;  $\Delta l$  – изменение длины  $l$  упругого тела под воздействием  $G$ . Учитывая, что  $G = F/S$ , где  $F$  – сила,  $S$  – площадь поперечного сечения тела, и дифференцируя (3.9), имеем

$$dF/dt = (S/l)d(\Delta l)/dt \text{ или } dF/dt = gu, \quad (3.10)$$

где  $g$  – жесткость (величину, обратную жесткости, называют гибкостью  $LM$ );  $u = d(\Delta l)/dt$  – скорость.

Диссипативные свойства в механических системах твердых тел выражаются соотношениями, характеризующими связь между силой трения и скоростью взаимного перемещения трущихся тел, причем в этих соотношениях производные сил или скоростей не фигурируют, как и в случае описания с помощью закона Ома диссипативных свойств в электрических системах.

Топологические уравнения характеризуют, во-первых, закон равновесия сил: сумма сил, приложенных к телу, включая силу инерции, равна нулю (принцип Даламбера), во-вторых, закон скоростей, согласно которому сумма относительной, переносной и абсолютной скоростей равна нулю.

В механических вращательных системах справедливы компонентные и топологические уравнения поступательных систем с заменой поступательных скоростей на угловые, сил – на вращательные моменты, масс – на моменты инерции, жесткостей – на вращательные жесткости.

Условные обозначения простых элементов механической системы показаны на рис. 3.2, б.

Нетрудно заметить наличие аналогий между электрической и механической системами. Так, токам и напряжениям в первой из них соответствуют силы (либо моменты) и скорости механической системы, компонентным уравнениям (3.4) и (3.5) и фигурирующим в них параметрам  $C$  и  $L$  – уравнения (3.8) и (3.10) и параметры  $M$  и  $LM$  – очевидна аналогия и между топологическими уравнениями. Далее параметры  $C$  и  $M$  будем называть емкостными (емкостного типа), параметры  $L$  и  $LM$  – индуктивными (индуктивного типа), а параметры  $R$  и  $R_{тр} = du/dF$  – резистивными (резистивного типа).

Имеется и существенное отличие в моделировании электрических и механических систем: первые из них одномерны, а процессы

во вторых часто приходится рассматривать в двух- (2D) или трехмерном (3D) пространстве. Следовательно, при моделировании механических систем в общем случае в пространстве 3D нужно использовать векторное представление фазовых переменных, каждая из которых имеет шесть составляющих, соответствующих шести степеням свободы. Однако отмеченные выше аналогии остаются справедливыми, если их относить к проекциям сил и скоростей на каждую пространственную ось, а при графическом представлении моделей использовать шесть эквивалентных схем – три для поступательных составляющих и три для вращательных.

*Гидравлические системы.* Фазовыми переменными в гидравлических системах являются расходы и давления. Как и в предыдущем случае, компонентные уравнения описывают свойства жидкости рассеивать или накапливать энергию.

Рассмотрим компонентные уравнения для жидкости на линейном участке трубопровода длиной  $A$  и воспользуемся уравнением Навье-Стокса в следующей его форме (для ламинарного течения жидкости):

$$\rho U / t = -P / X - 2aU,$$

где  $\rho$  – плотность жидкости;  $U$  – скорость;  $P$  – давление;  $a$  – коэффициент линеаризованного вязкого трения. Так как  $U = Q/S$ , где  $Q$  – объемный расход;  $S$  – площадь поперечного сечения трубопровода, то, заменяя пространственную производную отношением конечных разностей, имеем

$$dQ / dt = S / (\Delta l \rho) \Delta P - (2a / \rho) Q, \text{ или } \Delta P = L_{\Gamma} dQ / dt + R_{\Gamma} Q, \quad (3.11)$$

где  $\Delta P$  – падение давления на рассматриваемом участке трубопровода;  $L_{\Gamma} = \Delta l \rho / S$  – гидравлическая индуктивность, отражающая инерционные свойства жидкости,  $R_{\Gamma} = 2a / \rho$  – гидравлическое сопротивление, отражающее вязкое трение.

*Примечание.* В трубопроводе круглого сечения радиусом  $r$  удобно использовать выражение для гидравлического сопротивления при ламинарном течении:  $R_{\Gamma} = 8\nu \Delta l / (\pi r^4)$ , где  $\nu$  – кинематическая вязкость; в случае турбулентного характера течения жидкости компонентное уравнение для вязкого трения имеет вид  $\Delta P = RQ|Q|$  при  $R = 0,37(\pi r \nu / |Q|)^{1/4}$ . Пояснение уравнения (3.11) приводит к эквивалентной схеме (рис. 3.4).

Явление сжимаемости жидкости описывается компонентным уравнением, вытекающим из закона Гука:

$$\Delta P = E\Delta|L|. \quad (3.12)$$



Рис. 3.4. Эквивалентная схема участка трубопровода

Дифференцируя (3.12) и учитывая, что объемный расход  $Q$  связан со скоростью  $U = d(\Delta l)/dt$  соотношением  $Q = US$ , получаем

$$d\Delta P / dt = C_r Q,$$

где  $C_r = E / (S\Delta l)$  – гидравлическая емкость.

*Связь подсистем различной физической природы.* Используют следующие способы моделирования взаимосвязей подсистем: с помощью трансформаторной, гираторной связей и с помощью зависимости параметров компонентов одной подсистемы от фазовых переменных другой. В эквивалентных схемах трансформаторные и гираторные связи представлены зависимыми источниками фазовых переменных (на рис. 3.5). На рис. 3.5  $k$  и  $n$  – коэффициенты трансформации;  $g$  – передаточная проводимость;  $U_j$  и  $I_j$  – фазовые переменные в  $j$ -й цепи.

**Представление топологических уравнений.** Известен ряд методов формирования ММС на макроуровне. Получаемые с их помощью модели различаются ориентацией на те или иные численные методы решения и набором *базисных переменных*, т. е. фазовых переменных, остающихся в уравнениях итоговой ММС. Общим для всех методов является исходная совокупность топологических и компонентных уравнений (3.1)–(3.2).

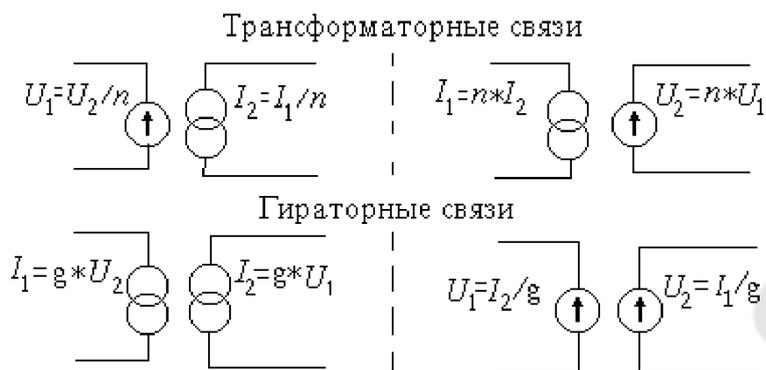


Рис. 3.5. Элементы взаимосвязи подсистем различной физической природы

При записи топологических уравнений удобно использовать промежуточную графическую форму – представление модели в виде эквивалентной схемы, состоящей из двухполюсных элементов. Общность подхода при этом сохраняется, так как любой многополюсный компонент можно заменить подсхемой из двухполюсников. В свою очередь эквивалентную схему можно рассматривать как направленный граф, дуги которого соответствуют ветвям схемы. Направления потоков в ветвях выбираются произвольно.

Пример простой эквивалентной схемы и соответствующего ей графа приведен на рис. 3.6. Для конкретности и простоты изложения на рис. 3.6 использованы условные обозначения, характерные для электрических эквивалентных схем. Очевидно, что поясненные выше аналогии позволяют при необходимости легко перейти к обозначениям и терминам, привычным для механиков.

Для получения топологических уравнений все ветви эквивалентной схемы разделяют на подмножества хорд и ветвей дерева. Имеется в виду *покрывающее* (фундаментальное) дерево, т. е. подмножество из дуг, не образующее ни одного замкнутого контура, где ( $\beta$  – число вершин графа – узлов эквивалентной схемы). На рис. 3.6, б показан граф эквивалентной схемы рис. 3.6, а, толстыми линиями выделено одно из возможных покрывающих деревьев.

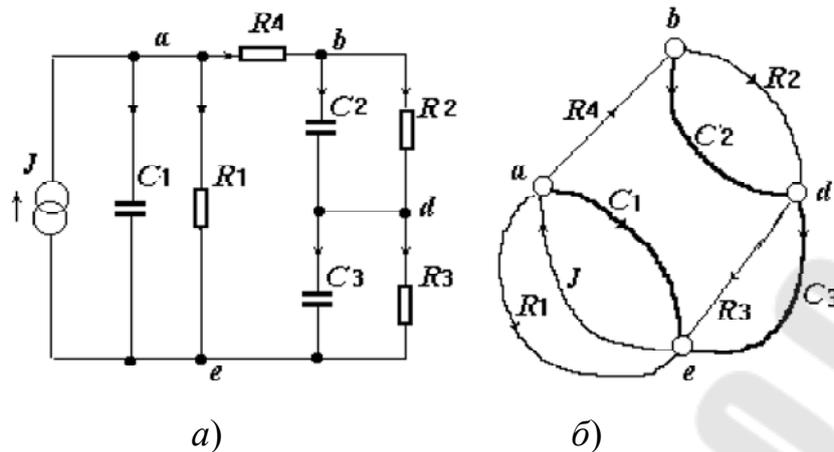


Рис. 3.6. Эквивалентная схема (а) и ее граф (б)

Выбор дерева однозначно определяет вектора напряжений  $U_x$  и токов  $I_x$  хорд, напряжений  $U_{вд}$  и токов  $I_{вд}$  ветвей дерева и приводит к записи топологических уравнений в виде:

$$U_x + MU_{вд} = 0; \quad (3.13)$$

$$I_{вд} - M^T I_x = 0, \quad (3.14)$$

где  $M$  – матрица контуров и сечений;  $M^T$  – транспонированная  $M$ -матрица.

Таблица 3.1

Хорды	Ветви дерева		
	C1	C2	C3
R1	-1	0	0
R2	0	-1	0
R3	0	0	-1
R4	-1	+1	+1
J	+1	0	0

В  $M$ -матрице число строк соответствует числу хорд, число столбцов равно числу ветвей дерева.  $M$ -матрица формируется следующим образом. Поочередно к дереву подключаются хорды. Если при подключении к дереву  $p$ -й хорды  $q$ -я ветвь входит в образовавшийся контур, то элемент  $M_{pq}$  матрицы равен +1 при совпадении направлений ветви и подключенной хорды,  $M_{pq} = -1$  при несовпадении направлений. В противном случае  $M_{pq} = 0$ .

Для схемы (рис. 3.6)  $M$ -матрица представлена в виде табл. 3.1

### Особенности эквивалентных схем механических объектов.

Для каждой степени свободы строят свою эквивалентную схему. Каждому телу с учитываемой массой соответствует узел схемы (вершина графа). Один узел, называемый базовым, отводится телу, отождествляемому с инерциальной системой отсчета.

Каждый элемент массы изображают ветвью, соединяющей узел соответствующего массе тела с базовым узлом; каждый элемент упругости – ветвью, соединяющей узлы тел, связанных упругой связью; каждый элемент трения – ветвью, соединяющей узлы трущихся тел. Внешние воздействия моделируются источниками сил и скоростей.

В качестве примера на рис. 3.7, а изображена некоторая механическая система – тележка, движущаяся по дороге и состоящая из платформы  $A$ , колес  $B1$ ,  $B2$  и рессор  $C1$ ,  $C2$ . На рис. 3.7, б приведена эквивалентная схема для вертикальных составляющих сил и скоростей, на которой телам системы соответствуют одноименные узлы, учитываются массы платформы и колес, упругость рессор, трение между колесами и дорогой; неровности дороги вызывают воздействие на систему (рис. 3.7, б) источниками силы.

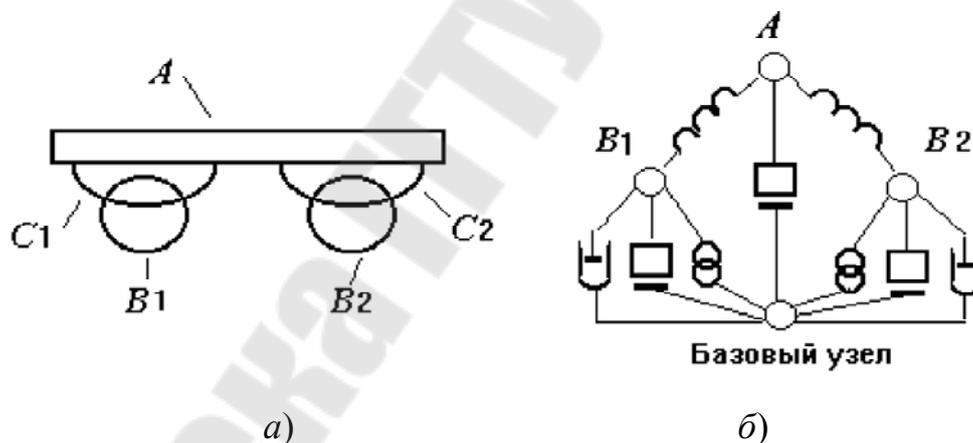


Рис. 3.7. Простая механическая система: а – эскизное изображение; б – эквивалентная схема

**Характеристика методов формирования ММС.** Исходную систему компонентных и топологических уравнений (3.1) и (3.2) можно рассматривать как окончательную ММС, которая и подлежит численному решению. Численное решение этой системы уравнений предполагает алгебраизацию дифференциальных уравнений, например, с помощью преобразования Лапласа или формул численного интегрирования. В программах анализа нелинейных объектов на макро-

уровне, как правило, применяются формулы численного интегрирования, примером которых может служить неявная формула Эйлера:

$$dV/dt|_n = (V_n - V_{n-1})/h_n,$$

где  $V_i$  – значение переменной  $v$  на  $i$ -м шаге интегрирования;  $h_n = t_n - t_{n-1}$  – шаг интегрирования. Алгебраизация подразумевает предварительную дискретизацию независимой переменной  $t$  (вместо непрерывной переменной  $t$  получаем конечное множество значений  $t_n$ ), она заключается в представлении ММС в виде системы уравнений с неизвестными  $V_n$  и  $Z_n$ :

$$\left. \begin{aligned} F_k(Z_n, V_n, t_n) &= 0, \\ F_T(V_n) &= 0, \\ Z_n &= (V_n - V_{n-1})/h_n, \end{aligned} \right\} \quad (3.15)$$

где использовано обозначение  $Z = dV/dt$ . Эту систему алгебраических уравнений, в общем случае нелинейных, необходимо решать на каждом шаге численного интегрирования исходных дифференциальных уравнений.

Однако порядок этой системы довольно высок и примерно равен  $2\alpha + \gamma$ , где  $\alpha$  – число ветвей эквивалентной схемы (каждая ветвь дает две неизвестные величины – фазовые переменные типа потока и типа потенциала, за исключением ветвей внешних источников, у каждой из которых неизвестна лишь одна фазовая переменная),  $\gamma$  – число элементов в векторе производных. Чтобы снизить порядок системы уравнений и тем самым повысить вычислительную эффективность ММС, желательно выполнить предварительное преобразование модели (в символическом виде) перед ее многошаговым численным решением. Предварительное преобразование сводится к исключению из системы части неизвестных и соответствующего числа уравнений. Оставшиеся неизвестные называют базисными. В зависимости от набора базисных неизвестных различают несколько методов формирования ММС.

Согласно методу переменных состояния (более полное название метода – метод переменных, характеризующих состояние), вектор базисных переменных  $W$  состоит из переменных состояния. Этот вектор включает неизбыточное множество переменных, характеризующих накопленную в системе энергию. Например, такими переменными могут быть скорости тел (кинетическая энергия определяется скоро-

стью, так как равна  $Mu^2/2$ ), емкостные напряжения, индуктивные токи и т. п. Очевидно, что число уравнений не превышает  $u$ . Кроме того, итоговая форма ММС оказывается приближенной к явной форме представления системы дифференциальных уравнений, т. е. к форме, в которой вектор  $dW/dt$  явно выражен через вектор  $W$ , что упрощает дальнейшее применение явных методов численного интегрирования. Метод реализуется путем особого выбора системы хорд и ветвей дерева при формировании топологических уравнений.

В классическом варианте узлового метода в качестве базисных переменных используются узловые потенциалы (т. е. скорости тел относительно инерциальной системы отсчета, абсолютные температуры, перепады давления между моделируемой и внешней средой, электрические потенциалы относительно базового узла). Число узловых потенциалов и соответственно уравнений в ММС оказывается равным  $\beta-1$ , где  $\beta$  – число узлов в эквивалентной схеме. Обычно  $\beta$  заметно меньше  $\alpha$  и, следовательно, порядок системы уравнений в ММС снижен более чем в два раза по сравнению с порядком исходной системы. Однако классический вариант узлового метода имеет ограничения на применение и потому в современных программах анализа наибольшее распространение получил модифицированный узловой метод.

*Узловой метод.* Матрицу контуров и сечений  $M$  в узловом методе формируют следующим образом. Выбирают базовый узел эквивалентной схемы и каждый из остальных узлов соединяют с базовым фиктивной ветвью. Именно фиктивные ветви принимают в качестве ветвей дерева, а все реальные ветви оказываются в числе хорд. Поскольку токи фиктивных ветвей равны нулю, а вектор напряжений фиктивных ветвей есть вектор узловых потенциалов  $\varphi$ , то уравнения (3.13) и (3.14) принимают вид:

$$U + M\varphi = 0; \quad (3.16)$$

$$M^T I = 0, \quad (3.17)$$

где  $U$  и  $I$  – векторы напряжений и токов реальных ветвей.

Компонентные уравнения алгебраизуются с помощью одной из формул численного интегрирования, линеаризуются с помощью разложения в ряд Тейлора с сохранением только линейных членов, и их представляют в виде

$$I_n = G_n U_n + A_n, \quad (3.18)$$

где  $G_n$  – диагональная матрица проводимостей, рассчитанная в точке  $t_n$ ;  $A_n$  – вектор, зависящий от значений фазовых переменных на предшествующих шагах интегрирования и потому уже известный к моменту времени  $t_n$ . Каждая ветвь (за исключением идеальных источников напряжения) имеет проводимость, которая занимает одну из диагональных клеток матрицы проводимостей. Окончательно ММС получаем, подставляя (3.18), и затем (3.16) в (3.17):

$$M^T I_n = M^T (G_n U_n + A_n) = -M^T G_n M \varphi_n + M^T A_n = 0$$

или

$$Y_n \varphi_n = B_n, \quad (3.19)$$

где  $Y_n = M^T G_n M$  – матрица Якоби,  $B_n = M^T A_n$  – вектор правых частей. Отметим, что матрица  $M$  имеет размер равен  $\alpha \times (\beta-1)$ , матрица  $G_n$  –  $\alpha \times \alpha$ , а матрица Якоби –  $(\beta-1) \cdot (\beta-1)$ .

Система (3.19) является *системой линейных алгебраических уравнений*, полученной в результате дискретизации независимой переменной, алгебраизации дифференциальных уравнений и линеаризации алгебраических уравнений. Алгебраизация приводит к необходимости пошагового вычислительного процесса интегрирования, линеаризация – к выполнению итерационного вычислительного процесса на каждом шаге интегрирования.

Рассмотрим, каким образом определяются проводимости ветвей.

Для резистивных ветвей проводимость – величина, обратная сопротивлению  $R$ .

При использовании неявного метода Эйлера проводимость емкостной ветви получается из ее компонентного уравнения следующим образом.

На  $n$ -м шаге интегрирования

$$i_n = C du/dt|_n = C(u_n - u_{n-1})/h_n,$$

проводимость  $g = \partial i_n / \partial u_n$  и при  $C = \text{const}$  получаем

$$g = C/h_n.$$

При этом в вектор правых частей входит элемент  $a_n = g u_{n-1}$ .

Проводимость индуктивной ветви можно найти аналогично:

$$u_n = L(i_n - i_{n-1})/h_n$$

и при  $L = \text{const}$

$$g = h_n / a_n = i_{n-i}.$$

Аналогично определяют проводимости и при использовании других разностных формул численного интегрирования, общий вид которых

$$dU / dt|_n = \mu_n U_n - \eta_n,$$

где  $\mu_n$  зависит от шага интегрирования;  $\eta_n$  – от значений вектора  $U$  на предыдущих шагах.

Классический вариант узлового метода имеет ограничения на применение. Так, недопустимы идеальные (с бесконечной проводимостью) источники напряжения, зависимые источники, аргументами которых являются токи, а также индуктивности, поскольку в классическом варианте токи не входят в число базисных переменных. Устранить эти ограничения довольно просто – нужно расширить совокупность базисных координат, включив в нее токи-аргументы зависимых источников, а также токи ветвей индуктивных и источников напряжения. Полученный вариант метода называют *модифицированным узловым методом*.

Таблица 3.2

Тип ветви	Фиктивные ветви	$U_{\text{ист}}(I)$	$E(t)$
неособые ветви	Ми	M12	M13
L	M21	M22	M23
$I_{\text{ист}}(I)$	M31	M32	M33

Согласно модифицированному узловому методу, в дерево при построении матрицы  $M$  включают ветви источников напряжения и затем фиктивные ветви. В результате матрица  $M$  принимает вид (табл. 3.2), где введены обозначения:  $U_{\text{ист}}(I)$  – источники напряжения, зависящие от тока;  $E(t)$  – независимые источники напряжения;  $I_{\text{ист}}(I)$  – источники тока, зависящие от тока;  $L$  – индуктивные ветви;  $M_{ij}$  – подматрица контуров хорд группы  $i$  и сечений фиктивных ветвей группы  $j$ .

Те же обозначения  $U_{\text{ист}}$ ,  $I$ ,  $E$ ,  $I_{\text{ист}}$  будем использовать и для соответствующих векторов напряжений и токов. Назовем ветви, токи которых являются аргументами в выражениях для зависимых источников, т. е. входят в вектор  $I$ , *особыми* ветвями. Остальные ветви (за исключением индуктивных) – *неособые*. Введем также обозначения:  $I_L$  –

вектор индуктивных токов;  $I_x$  и  $U_x$  – векторы токов и напряжений неособых ветвей;  $G_x$ ,  $G_L$ ,  $G_I$  – диагональные матрицы проводимостей ветвей неособых, индуктивных и особых.

Уравнение закона токов Кирхгофа (3.17) для фиктивных ветвей имеет вид

$$(M_{11})^T I_k + (M_{21})^T I_L + (M_{31})^T I_{ист} = 0.$$

Исключим вектор  $I_x$  с помощью компонентного уравнения (3.18), а вектор  $I_{ист}$  с помощью очевидного выражения

$$I_{ист} = KI,$$

где  $K = (\partial I_{ист} / \partial I)$  – матрица передаточных коэффициентов источников тока. Используем также выражение (3.16), принимающее вид:

$$U_x = -M_{11}\varphi - M_{12}U_{ист} - M_{31}E = -M_{11}\varphi - M_{12}(\partial U_{ист} / \partial I)I - M_{31}E.$$

Получаем систему из трех матричных уравнений с неизвестными векторами  $\varphi$ ,  $I$  и  $I_L$ :

$$\begin{aligned} & -(M_{11})^T G_x (M_{11}\varphi + M_{12}RI) + (M_{21})^T I_L + (M_{31})^T KI = \\ & = G_x M_{31}E + (M_{11})^T A_x; \end{aligned} \quad (3.20)$$

$$IL = -G_L (M_{21}\varphi + M_{22}RI + M_{23}E) + A_L; \quad (3.21)$$

$$I = -G_I (M_{31}\varphi + M_{32}RI + M_{33}E) + A_I, \quad (3.22)$$

где обозначено  $R = (\partial U_{ист} / \partial I)$ . Эта система и является итоговой ММ в узловом модифицированном методе.

*Примечание:*

1. Вектор индуктивных токов нельзя исключить из итоговой системы уравнений, так как его значения входят в вектор  $A_L$  на последующих шагах численного интегрирования.

2. Источники тока, зависящие от напряжений, относятся к неособым ветвям, их проводимости  $(\partial U_{ист} / \partial U)$  входят в матрицу  $G_x$ , которая при этом может иметь недиагональный вид.

3. Источники напряжения, зависящие от напряжений, в приведенных выше выражениях не учитываются, при их наличии нужно в матрице  $M$  выделить столбец для этих ветвей, что приводит к появлению дополнительных слагаемых в правых частях уравнений (3.19)–(3.21).

## Тема 11. Методы и алгоритмы анализа на макроуровне

**Выбор методов анализа во временной области.** Анализ процессов в проектируемых объектах можно производить во временной и частотной областях. *Анализ во временной области* (динамический анализ) позволяет получить картину переходных процессов, оценить динамические свойства объекта, он является важной процедурой при исследовании как линейных, так и нелинейных систем. *Анализ в частотной области* более специфичен, его применяют как правило к объектам с линеаризуемыми ММ при исследовании колебательных стационарных процессов, анализе устойчивости, расчете искажений информации, представляемой спектральными составляющими сигналов, и т. п.

Методы анализа во временной области, используемые в универсальных программах анализа в САПР, – это численные методы интегрирования систем обыкновенных дифференциальных уравнений (СОДУ):

$$F(dV / dt, V, t) = 0.$$

Другими словами, это методы алгебраизации дифференциальных уравнений. Формулы интегрирования СОДУ могут входить в ММ независимо от компонентных уравнений, как это имеет место в (3.15), или быть интегрированными в ММ компонентов, как это выполнено в узловом методе.

От выбора метода решения СОДУ существенно зависят такие характеристики анализа, как точность и вычислительная эффективность. Эти характеристики определяются прежде всего типом и порядком выбранного метода интегрирования СОДУ.

Применяют два типа методов интегрирования – явные (иначе экстраполяционные или методы, основанные на формулах интегрирования вперед), и неявные (интерполяционные, основанные на формулах интегрирования назад). Различия между ними удобно показать на примере простейших методов первого порядка – методов Эйлера.

Формула *явного метода Эйлера* представляет собой следующую формулу замены производных в точке  $t_n$ :

$$dV / dt|_n = (V_{n+1} - V_n) / h_n,$$

где индекс равен номеру шага интегрирования;  $h_n = t_{n+1} - t_n$  – размер шага интегрирования (обычно  $h_n$  называют просто шагом интегрирования). В формуле неявного метода Эйлера использовано дифференцирование назад:

$$dV/dt|_n = (V_n - V_{n-i})/h_n,$$

где  $h_n = t_n - t_{n-1}$ .

Выполним сравнительный анализ явных и неявных методов на примере модельной задачи:

$$dV/dt = AV \quad (3.23)$$

при ненулевых начальных условиях  $V_0 \neq 0$  и при использовании методов Эйлера с постоянным шагом  $h$ . Здесь  $A$  – постоянная матрица;  $V$  – вектор фазовых переменных. При алгебраизации явным методом имеем

$$(V_{n+1} - V_n)/h = AV_n \text{ или } V_{n+1} = (E + hA)V_n,$$

где  $E$  – единичная матрица. Вектор  $V_{n+1}$  можно выразить через вектор начальных условий  $V_0$ :

$$V_{n+1} = (E + hA)^n V_0. \quad (3.24)$$

Обозначим

$$B = E + hA \quad (3.25)$$

и применим преобразование подобия для матрицы  $B$

$$B = T^{-1} \text{diag}\{\lambda_{Bj}\} T,$$

где  $T$  – преобразующая матрица;  $\text{diag}\{\lambda_{Bj}\}$  – диагональная матрица с собственными значениями  $\lambda_{Bj}$  матрицы  $B$  на диагонали. Нетрудно видеть, что  $B^n = T^{-1} \text{diag}\{\lambda_{Bj}^n\} T$ .

Из линейной алгебры известно, что собственные значения матриц, связанных арифметическими операциями, оказываются связанными такими же преобразованиями. Поэтому из (3.25) следует

$$\lambda_{Bj} = 1 + h\lambda_{Aj}.$$

Точное решение модельной задачи (3.23)  $V(t) \rightarrow 0$  при  $t \rightarrow \infty$ , следовательно, условием устойчивости процесса численного решения можно считать

$$V_{n+1} \rightarrow 0 \text{ при } n \rightarrow \infty,$$

откуда последовательно получаем  $(E + hA)^n V_0 \rightarrow 0$ , так как  $V_0 \neq 0$ , то  $(E + hA)^n \rightarrow 0$ , поскольку  $T \neq 0$ , то  $\lambda_{Bj}^n \rightarrow 0$  и условие устойчивости

$$-1 < |1 + h\lambda_{Aj}| < 1. \quad (3.26)$$

Известно, что для физически устойчивых систем собственные значения матрицы коэффициентов в ММС оказываются отрицательными. Если к тому же все  $\lambda_{Aj}$  – вещественные величины (характер процессов в ММС с моделью (3.23) апериодический), то естественно определить постоянные времени физической системы как

$$\tau_i = -1/\lambda_{Ai}$$

и условие (3.26) конкретизируется следующим образом:

$$-1 < |1 - h/\tau_j| < 1 \text{ или } 0 < h < 2\tau_{\min}, \quad (3.27)$$

где  $\tau_{\min}$  – минимальная постоянная времени. Если использовать явные методы более высокого порядка, то может увеличиться коэффициент перед  $\tau_{\min}$  в (3.27), но это принципиально не меняет оценки явных методов.

Если нарушено условие (3.27), то происходит потеря устойчивости вычислений, а это означает, что в решении задачи возникают ложные колебания с увеличивающейся от шага к шагу амплитудой и быстрым аварийным остановом ЭВМ вследствие переполнения разрядной сетки. Конечно, ни о какой адекватности решения говорить не приходится.

Для соблюдения (3.27) применяют те или иные алгоритмы автоматического выбора шага. Отметим, что в сложной модели расчет  $\tau_{\min}$  для непосредственного выбора шага по (3.27) слишком трудоемок, кроме того, однократный расчет  $\tau_{\min}$  мало чем помогает, так как в нелинейных моделях  $\tau_{\min}$  может изменяться от шага к шагу.

Условие (3.27) накладывает жесткие ограничения на шаг интегрирования. В результате вычислительная эффективность явных методов резко падает с ухудшением обусловленности ММС. В самом деле, длительность  $T_{\text{инт}}$  моделируемого процесса должна быть соизмеримой с временем успокоения системы после возбуждающего воздействия,

т. е. соизмерима с максимальной постоянной времени  $\tau_{\min}$ . Требуемое число шагов интегрирования равно  $I = T_{\text{инт}} / h \sim \tau_{\max} / \tau_{\min}$ .

Отношение  $\mathcal{C} = \tau_{\max} / \tau_{\min}$  называют разбросом постоянных времени или числом обусловленности. Чем больше это число, тем хуже обусловленность. Попытки применения явных методов к любым ММС чаще всего приводят к недопустимо низкой вычислительной эффективности, поскольку в реальных моделях  $\mathcal{C} > 10^5$  – обычная ситуация. Поэтому в настоящее время в универсальных программах анализа явные методы решения СОДУ не применяют.

Аналогичный анализ числовой устойчивости неявных методов дает следующие результаты. Вместо (3.24) имеем  $V_n = (E - hA)^{-n} V_0$  и условие числовой устойчивости принимает вид

$$-1 < |1/(1 + h/\tau_j)| < 1,$$

которое выполняется при любых  $h > 0$ . Следовательно, неявный метод Эйлера обладает так называемой А-устойчивостью.

*Примечание.* Метод интегрирования СОДУ называют А-устойчивым, если погрешность интегрирования остается ограниченной при любом шаге  $h > 0$ .

Применение А-устойчивых методов позволяет существенно уменьшить требуемые числа шагов  $m$ . В этих методах шаг выбирается автоматически не из условий устойчивости, а только из соображений точности решения.

Выбор порядка метода решения СОДУ довольно прост: во-первых, более высокий порядок обеспечивает более высокую точность, во-вторых, среди неявных разностных методов, кроме метода Эйлера, А-устойчивы также методы второго порядка и среди них – *метод трапеций*. Поэтому преобладающее распространение в программах анализа получили методы второго порядка – модификации метода трапеций.

**Алгоритм численного интегрирования СОДУ.** Одна из удачных реализаций неявного метода второго порядка, которую можно считать модификацией метода трапеций, основана на комбинированном использовании явной и неявной формул Эйлера. Рассмотрим вопрос, почему такое комбинирование снижает погрешность и приводит к повышению порядка метода.

Предварительно отметим, что в методах  $p$ -го порядка локальная погрешность, т. е. погрешность, допущенная на одном  $n$ -м шаге интегрирования, оценивается старшим из отбрасываемых членов

$$\sigma = c \|V^{(p+1)}(\tau)\| h^{p+1},$$

в разложении решения  $V(t)$  в ряд Тейлора, где  $c$  – постоянный коэффициент, зависящий от метода,  $\|V^{(p+1)}(\tau)\|$  – норма вектора  $(p+1)$ -х производных  $V(t)$ , которая оценивается с помощью конечно-разностной аппроксимации,  $\tau$  – значение времени  $t$  внутри шага.

Если  $n$ -й шаг интегрирования в комбинированном методе был неявным, т. е. выполненным по неявной формуле, то следующий шаг с тем же значением  $h$  должен быть явным. Используя разложение решения  $V(t)$  в ряд Тейлора в окрестностях точки  $t_{n+1}$ , получаем для  $(n+1)$ -го неявного шага

$$V(t_n) = V(t_{n+1}) - (dV/dt)h_H + (d^2V/dt^2)h_H^2/2! - (d^3V/dt^3)h_H^3/3! + \dots; \quad (3.28)$$

и для  $(n+2)$ -го явного шага

$$V(t_{n+2}) = V(t_{n+1}) + (dV/dt)h_Y + (d^2V/dt^2)h_Y^2/2! + (d^3V/dt^3)h_Y^3/3! \quad (3.29)$$

где  $h_H$  и  $h_Y$  – величины неявного и явного шагов, а значения производных относятся к моменту  $t_{n+1}$ . Подставляя (3.28) в (3.29), при  $h = h_Y = h_H$  получаем

$$V(t_{n+2}) = V(t_n) + 2(dV/dt)h + 2(d^3V/dt^3)h^3/3! + \dots,$$

т. е. погрешности, обуславливаемые квадратичными членами в (3.28) и (3.29) взаимно компенсируются, и старшим из отбрасываемых членов становится член с  $h^3$ . Следовательно, изложенное комбинирование неявной и явной формул Эйлера дает метод интегрирования второго порядка.

Неявные методы и, в частности, рассмотренный комбинированный метод целесообразно использовать только при переменной величине шага. Действительно, при заметных скоростях изменения фазовых переменных погрешность остается в допустимых пределах только при малых шагах, в квазистатических режимах шаг может быть во много раз больше.

Алгоритмы автоматического выбора шага основаны на сравнении допущенной и допустимой локальных погрешностей. Например, вводится некоторый диапазон (коридор) погрешностей  $\delta$ , в пределах

которого шаг сохраняется неизменным. Если допущенная погрешность превышает верхнюю границу диапазона, то шаг уменьшается, если же выходит за нижнюю границу, то шаг увеличивается.

**Методы решения систем нелинейных алгебраических уравнений.** Вычисления при решении СОДУ состоят из нескольких вложенных один в другой циклических процессов. Внешний цикл – цикл пошагового численного интегрирования, параметром цикла является номер шага. Если модель анализируемого объекта нелинейна, то на каждом шаге выполняется промежуточный цикл – итерационный цикл решения системы нелинейных алгебраических уравнений (СНАУ). Параметр цикла – номер итерации. Во внутреннем цикле решается система линейных алгебраических уравнений (СЛАУ), например, при применении узлового метода формирования ММС такой системой является (3.19). Поэтому в математическое обеспечение анализа на макроуровне входят методы решения СНАУ и СЛАУ.

Для решения СНАУ можно применять такие прямые итерационные методы, как метод простой итерации или метод Зейделя, но в современных программах анализа наибольшее распространение получил метод Ньютона, основанный на линеаризации СНАУ. Собственно модель (3.19) получена именно в соответствии с методом Ньютона. Основное преимущество метода Ньютона – высокая скорость сходимости.

Представим СНАУ в виде

$$F(X) = 0. \quad (3.30)$$

Разлагая  $F(X)$  в ряд Тейлора в окрестностях некоторой точки  $X_k$ , получаем  $F(X) = F(X_k) + (\partial F/\partial X)(X-X_k) + (X-X_k)^T(\partial^2 F/\partial X^2)(X-X_k)/2 + \dots = 0$ . Сохраняя только линейные члены, получаем СЛАУ с неизвестным вектором  $X$ :

$$Y_k(X - X_k) = -F(X_k), \quad (3.31)$$

где  $Y_k = (\partial F/\partial X)|_k$ . Решение системы (3.31) дает очередное приближение к корню системы (3.30), которое удобно обозначить  $X_{k+1}$ .

Вычислительный процесс стартует с начального приближения  $X_0$  и в случае сходимости итераций заканчивается, когда погрешность, оцениваемая как

$$|\Delta X_k| = |X_k - X_{k-1}|$$

станет меньше допустимой погрешности  $\varepsilon$ .

Однако метод Ньютона не всегда приводит к сходящимся итерациям. Условия сходимости метода Ньютона выражаются довольно сложно, но существует легко используемый подход к улучшению сходимости. Это близость начального приближения к искомому корню СНАУ. Использование этого фактора привело к появлению метода решения СНАУ, называемого *продолжением решения по параметру*.

В методе продолжения решения по параметру в ММС выделяется некоторый параметр  $\alpha$ , такой, что при  $\alpha = 0$  корень  $X_{\alpha=0}$  системы (3.30) известен, а при увеличении  $\alpha$  от 0 до его истинного значения составляющие вектора  $X$  плавно изменяются от  $X_{\alpha=0}$  до истинного значения корня. Тогда задача разбивается на ряд подзадач, последовательно решаемых при меняющихся значениях  $a$ , и при достаточно малом шаге  $\Delta a$  изменения  $\alpha$  условия сходимости выполняются.

В качестве параметра  $a$  можно выбрать некоторый внешний параметр, например, при анализе электронных схем им может быть напряжение источника питания. Но на практике при интегрировании СОДУ в качестве  $a$  выбирают шаг интегрирования  $h$ . Очевидно, что при  $h = 0$  корень СНАУ равен значению вектора неизвестных на предыдущем шаге. Регулирование значений  $h$  возлагается на алгоритм автоматического выбора шага.

В этих условиях очевидна целесообразность представления математических моделей для анализа статических состояний в виде СОДУ, как и для анализа динамических режимов.

**Методы решения систем линейных алгебраических уравнений.** В программах анализа в САПР для решения СЛАУ чаще всего применяют метод Гаусса или его разновидности. Метод Гаусса – метод последовательного исключения неизвестных из системы уравнений. При исключении  $k$ -й неизвестной  $x_k$  из системы уравнений

$$AX = B \quad (3.32)$$

все коэффициенты  $a_{ij}$  при  $i > k$  и  $j > k$  пересчитывают по формуле

$$a_{ij} = a_{ij} - a_{ik} a_{kj} / a_{kk}. \quad (3.33)$$

Исключение  $n-1$  неизвестных, где  $n$  – порядок системы (3.32), называют прямым ходом, в процессе которого матрица коэффициентов приобретает треугольный вид. При обратном ходе последовательно вычисляют неизвестные, начиная с  $x_n$ .

В общем случае число арифметических операций для решения (3.32) по Гауссу пропорционально  $n^3$ . Это приводит к значительным затратам машинного времени, поскольку СЛАУ решается многократно в процессе одновариантного анализа, и существенно ограни-

чивает сложность анализируемых объектов. Заметно повысить вычислительную эффективность анализа можно, если использовать характерное практически для всех приложений свойство высокой разреженности матрицы  $A$  в модели (3.32).

Матрицу называют *разреженной*, если большинство ее элементов равно нулю. Эффективность обработки разреженных матриц велика, потому что, во-первых, пересчет по формуле (3.33) не требуется, если хотя бы один из элементов  $a_{ik}$  или  $a_{kj}$  оказывается нулевым, во-вторых, не требуются затраты памяти для хранения нулевых элементов. Хотя алгоритмы обработки разреженных матриц более сложны, но в результате удается получить затраты машинного времени, близкие к линейным.

При использовании методов разреженных матриц нужно учитывать зависимость вычислительной эффективности от того, как представлена матрица коэффициентов  $A$ , точнее от того, в каком порядке записаны ее строки и столбцы.

Для пояснения этой зависимости рассмотрим два варианта представления одной и той же СЛАУ. В первом случае система уравнений имеет вид

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 = b_1;$$

$$a_{21}x_1 + a_{22}x_2 = b_2;$$

$$a_{31}x_1 + a_{33}x_3 = b_3;$$

$$a_{41}x_1 + a_{44}x_4 = b_4;$$

$$a_{51}x_1 + a_{55}x_5 = b_5.$$

При прямом ходе в соответствии с формулой (3.33) все элементы матрицы, которые первоначально были нулевыми, становятся ненулевыми, а матрица оказывается полностью *насыщенной*.

Таблица 3.3

+	+	+	+	+
+	+	.	.	.
+	.	+	.	.
+	.	.	+	.
+	.	.	.	+

Таблица 3.4

+				+
	+			+
		+		+
			+	+
+	+	+	+	+

Элементы, становящиеся ненулевыми в процессе гауссовых исключений, называют *вторичными нулями*. Вторичные нули в таблице 3.3 отмечены точкой.

Во втором случае меняются местами первое и пятое уравнения. Матрицы коэффициентов имеют вид таблиц 3.3 и 3.4, где ненулевые элементы представлены знаком «+». Теперь вторичные нули не появляются, матрица остается разреженной, высокая вычислительная эффективность сохраняется. Таким образом, методы разреженных матриц должны включать в себя способы *оптимального упорядочения строк и столбцов* матриц. Используют несколько критериев оптимальности упорядочения. Простейшим из них является критерий расположения строк в порядке увеличения числа первичных нулей, более сложные критерии учитывают не только первичные нули, но и появляющиеся вторичные нули.

*Методом разреженных матриц* называют метод решения СЛАУ на основе метода Гаусса с учетом разреженности матрицы коэффициентов. Метод разреженных матриц можно реализовать путем интерпретации и компиляции. В обоих случаях создаются массивы ненулевых коэффициентов матрицы (с учетом вторичных нулей) и массивы координат этих ненулевых элементов.

При этом выигрыш в затратах памяти довольно значителен. Так, при матрице умеренного размера  $200 \times 200$  без учета разреженности потребуется 320 кбайт. Если же взять характерное значение 9 для среднего числа нулей в одной строке, то для коэффициентов и указателей координат потребуется не более 28 кбайт.

В случае *интерпретации* моделирующая программа для каждой операции по (3.33) при  $a_{jk} \neq 0$  и  $a_{kj} \neq 0$  находит, используя указатели, нужные коэффициенты и выполняет арифметические операции по (3.33). Поскольку СЛАУ в процессе анализа решается многократно, то и операции поиска нужных коэффициентов также повторяются многократно, на что естественно тратится машинное время.

Способ *компиляции* более экономичен по затратам времени, но уступает способу интерпретации по затратам памяти. При компиляции поиск нужных для (3.33) коэффициентов выполняется однократно перед численным решением задачи. Вместо выполнения арифметических операций для каждой из них компилируется команда с найденными адресами ненулевых коэффициентов. Такие команды образуют рабочую программу решения СЛАУ, которая и будет решаться многократно.

Очевидно, что теперь в рабочей программе будет выполняться минимально необходимое число арифметических операций.

**Анализ в частотной области.** Анализ в частотной области выполняется по отношению к линеаризованным моделям объектов. Для линейных СОДУ справедливо применение для алгебраизации дифференциальных уравнений преобразования Фурье, в котором оператор  $d/dt$  заменяется на оператор  $j\omega$ .

Характерной особенностью получающейся СЛАУ является комплексный характер матрицы коэффициентов, что в некоторой степени усложняет процедуру решения, но не создает принципиальных трудностей. При решении задают ряд частот  $\omega_k$ . Для каждой частоты решают СЛАУ и определяют действительные и мнимые части искомым фазовых переменных. По ним определяют амплитуду и фазовый угол каждой спектральной составляющей, что и позволяет построить амплитудно-частотные, фазочастотные характеристики, найти собственные частоты колебательной системы и т. п.

**Многовариантный анализ.** Одновариантный анализ позволяет получить информацию о состоянии и поведении проектируемого объекта в одной точке пространства внутренних  $X$  и внешних  $Q$  параметров. Очевидно, что для оценки свойств проектируемого объекта этого недостаточно. Нужно выполнять *многовариантный анализ*, т. е. исследовать поведение объекта в ряде точек упомянутого пространства, которое будем далее называть *пространством аргументов*.

Чаще всего многовариантный анализ в САПР выполняется в интерактивном режиме, когда разработчик неоднократно меняет в математической модели те или иные параметры из множеств  $X$  и  $Q$ , выполняет одновариантный анализ и фиксирует полученные значения выходных параметров. Подобный многовариантный анализ позволяет оценить *области работоспособности*, степень выполнения условий работоспособности, а следовательно, степень выполнения технического задания (ТЗ) на проектирование, разумность принимаемых промежуточных решений по изменению проекта и т. п.

*Примечание.* Областью работоспособности называют область в пространстве аргументов, в пределах которой выполняются все заданные условия работоспособности, т. е. значения всех выходных параметров находятся в допустимых по ТЗ пределах.

Как изложено в главе 1, среди процедур многовариантного анализа можно выделить типовые, выполняемые по заранее составлен-

ным программам. К таким процедурам относятся анализ чувствительности и статистический анализ.

Наиболее просто *анализ чувствительности* реализуется путем численного дифференцирования. Пусть анализ проводится в некоторой точке  $x_{\text{НОМ}}$  пространства аргументов, в которой предварительно проведен одновариантный анализ и найдены значения выходных параметров  $y_{\text{НОМ}}$ . Выделяется  $N$  параметров-аргументов  $x_i$  (из числа элементов векторов  $X$  и  $Q$ , влияние которых на выходные параметры подлежит оценить, поочередно каждый из них получает приращение  $\Delta x_i$ , выполняется одновариантный анализ, фиксируются значения выходных параметров  $y_j$  и подсчитываются значения абсолютных

$$A_{ij} = (y_j - y_{\text{НОМ}}) / \Delta x_i$$

и относительных коэффициентов чувствительности

$$B_{ij} = A_{ij} x_{i\text{НОМ}} / y_{\text{НОМ}}.$$

Такой метод численного дифференцирования называют *методом приращений*. Для анализа чувствительности, согласно методу приращений, требуется выполнить  $N + 1$  раз одновариантный анализ. Результат его применения – матрицы абсолютной и относительной чувствительности, элементами которых являются коэффициенты  $A_{ji}$  и  $B_{ji}$ .

*Примечание.* Анализ чувствительности – это расчет векторов градиентов выходных параметров, который входит составной частью в программы параметрической оптимизации.

Цель статистического анализа – оценка законов распределения выходных параметров и (или) числовых характеристик этих распределений. Случайный характер величин  $y_j$  обусловлен случайным характером параметров элементов  $x_i$ , поэтому исходными данными для статистического анализа являются сведения о законах распределения  $x_i$ . В соответствии с результатами статистического анализа прогнозируют процент бракованных изделий в готовой продукции (рис. 3.8). На рис. 3.8 представлена рассчитанная плотность  $P$  распределения выходного параметра  $y_j$ , имеющего условие работоспособности  $y_j < T_j$ , затемненный участок характеризует долю изделий, не удовлетворяющих условию работоспособности параметра  $y_j$ .

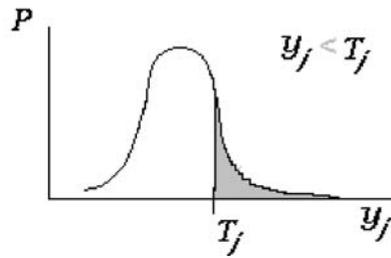


Рис. 3.8. Иллюстрация определения процента выпуска негодных изделий

В САПР статистический анализ осуществляется численным методом – *методом Монте-Карло* (статистических испытаний). В соответствии с этим методом выполняются  $N$  статистических испытаний, каждое статистическое испытание представляет собой одновариантный анализ, выполняемый при случайных значениях параметров-аргументов. Эти случайные значения выбирают в соответствии с заданными законами распределения аргументов  $x_i$ . Полученные в каждом испытании значения выходных параметров накапливают, после  $N$  испытаний обрабатывают, что дает следующие результаты: гистограммы выходных параметров (оценки математических ожиданий и дисперсий выходных параметров: оценки коэффициентов корреляции и регрессии между избранными выходными и внутренними параметрами, которые, в частности, можно использовать для оценки коэффициентов чувствительности).

Статистический анализ, выполняемый в соответствии с методом Монте-Карло, – трудоемкая процедура, поскольку число испытаний  $N$  приходится выбирать довольно большим, чтобы достичь приемлемой точности анализа. Другая причина, затрудняющая применение метода Монте-Карло, – трудности в получении достоверной исходной информации о законах распределения параметров-аргументов  $x_i$ .

Более типична ситуация, когда законы распределения  $x_i$  неизвестны, но с большой долей уверенности можно указать предельно допустимые отклонения  $\Delta x_i$  параметров  $x_i$  от номинальных значений  $x_{iном}$  (такие отклонения часто указываются в паспортных данных на комплектующие детали). В таких случаях более реалистично применять *метод анализа на наихудший случай*. Согласно этому методу, сначала выполняют анализ чувствительности с целью определения знаков коэффициентов чувствительности. Далее осуществляют  $m$  раз одновариантный анализ, где  $m$  – число выходных параметров. В каждом варианте задают значения аргументов, наиболее неблагоприятные для выполнения условия работоспособности очередного выход-

ного параметра  $y_j, j[1 : m]$ . Так, если  $y_j < T_j$  и коэффициент чувствительности положительный (т. е.  $\text{sign}(B_{ji}) = 0$ ) или  $y_j > T_j$  и  $\text{sign}(B_{ji}) = 1$ , то

$$x_i = x_{\text{ином}} + \Delta x_i, \text{ иначе } x_i = x_{\text{ином}} - \Delta x.$$

Следует отметить, что, проводя анализ на наихудший случай, можно получить завышенные значения разброса выходных параметров, и если добиваться выполнения условий работоспособности в наихудших случаях, то это часто ведет к неоправданному увеличению стоимости, габаритных размеров, массы и других показателей проектируемых конструкций, хотя и гарантирует с запасом выполнение условий работоспособности.

**Организация вычислительного процесса в универсальных программах анализа на макроуровне.** На рис. 3.9 представлена графическая схема вычислительного процесса при анализе во временной области на макроуровне. Алгоритм отражает решение системы алгебро-дифференциальных уравнений:

$$\varphi(dV / dt, V, T) = 0.$$

На каждом шаге численного интегрирования решается система нелинейных алгебраических уравнений методом Ньютона:

$$F(X) = 0.$$

На каждой итерации выполняется решение системы линейных алгебраических уравнений:

$$Y \Delta X = B.$$

Другие используемые обозначения:  $V_0(t_0)$  – начальные условия;  $h$  и  $h_{\text{нач}}$  – шаг интегрирования и его начальное значение;  $U_{\text{вн}}(t)$  – вектор внешних воздействий;  $N$  и  $N_{\text{д}}$  – число ньютоновских итераций и его максимально допустимое значение;  $\varepsilon$  – предельно допустимая погрешность решения СНАУ;  $\delta$  – погрешность, допущенная на одном шаге интегрирования;  $m1$  – максимально допустимое значение погрешности интегрирования на одном шаге;  $m2$  – нижняя граница коридора рациональных погрешностей интегрирования.

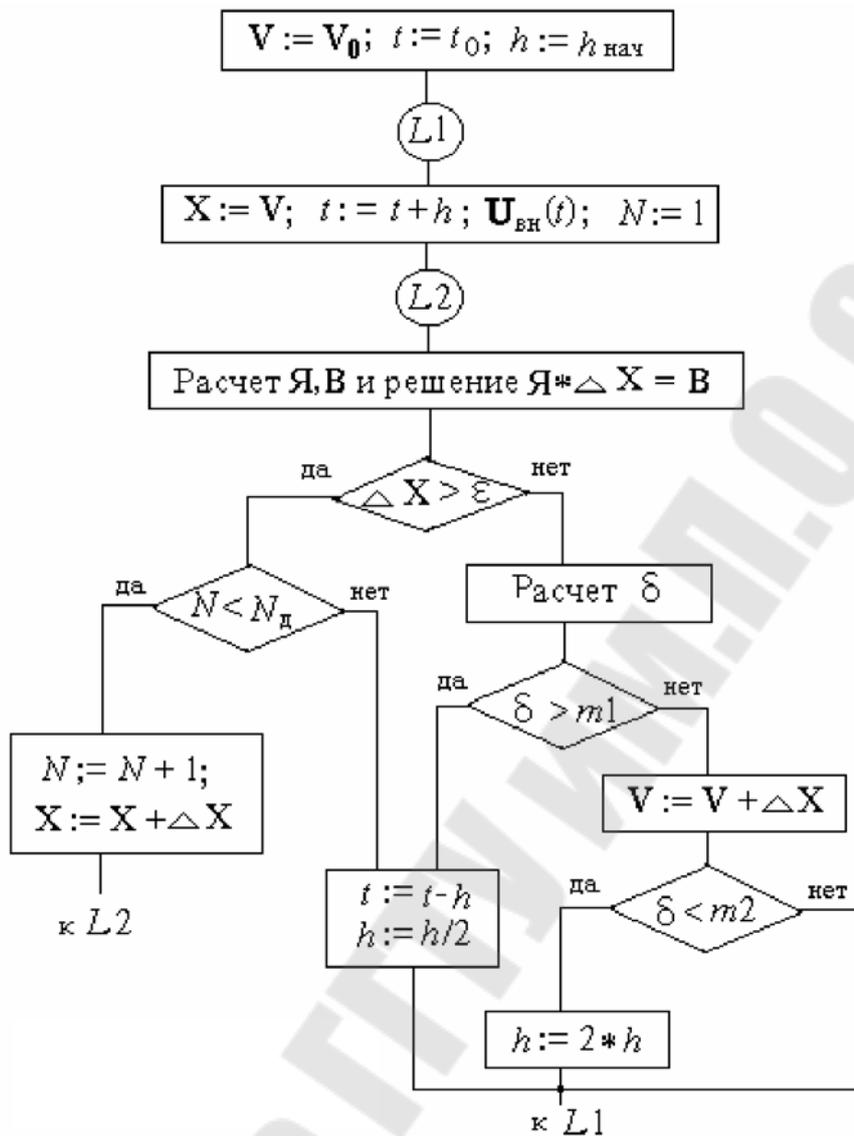


Рис. 3.9. Схема вычислительного процесса анализа на макроуровне

Из рис. 3.9 видно, что при  $N > N_d$  фиксируется несходимость ньютоновских итераций и после дробления шага происходит возврат к интегрированию при тех же начальных для данного шага условиях. При сходимости рассчитывается  $\delta$  и в зависимости от того, выходит погрешность за пределы диапазона  $[m2, m1]$  или нет, шаг изменяется либо сохраняет свое прежнее значение. Параметры  $N_d, m1, m2, \epsilon, h_{нач}$  задаются «по умолчанию» и могут настраиваться пользователем.

Матрицу Якоби  $Y$  и вектор правых частей  $V$  необходимо рассчитывать по программе, составляемой для каждого нового исследуемого объекта. Составление программы выполняет компилятор, входящий в

состав программного комплекса анализа. Общая структура такого комплекса представлена на рис. 3.10.

Исходные данные об объекте можно задавать в графическом виде или на входном языке программы анализа. Запись на таком языке обычно представляет собой список компонентов анализируемого объекта с указанием их взаимосвязей. Вводимые данные преобразуются во внутреннее представление с помощью графического и лингвистического препроцессоров, в которых предусмотрена также диагностика нарушений формальных языковых правил. Графическое представление более удобно, особенно для малоопытных пользователей.

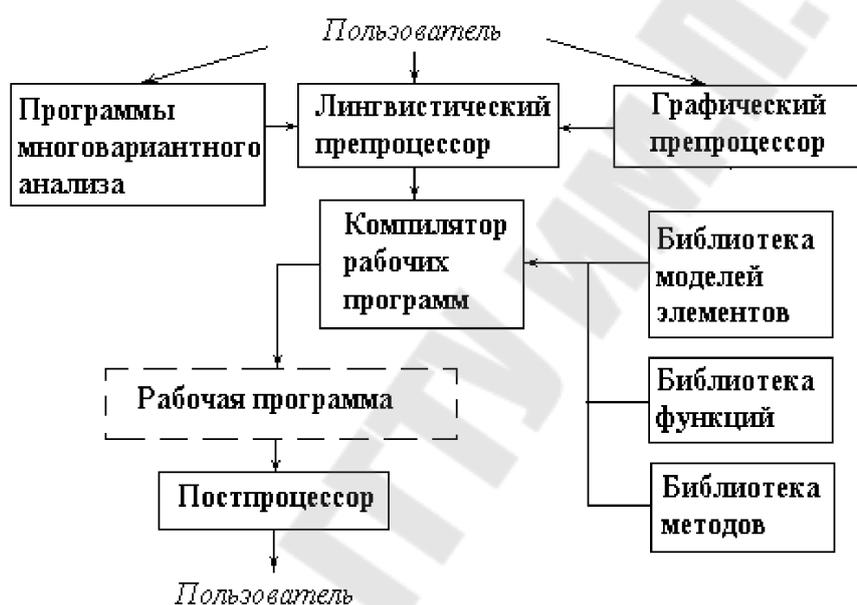


Рис. 3.10. Структура программного комплекса анализа на макроуровне

Задав описание объекта, пользователь может приступить к многовариантному анализу либо по одной из программ такого анализа, либо в интерактивном режиме, изменяя условия моделирования между вариантами с помощью лингвистического препроцессора.

Наиболее сложная часть комплекса – компилятор рабочих программ, именно в нем создаются программы расчета матрицы Якоби  $Y$  и вектора правых частей  $B$ , фигурирующих в вычислительном процессе (рис. 3.9). Собственно рабочая программа (рис. 3.10) – это и есть программа процесса, показанного на рис. 3.9. Для каждого нового моделируемого объекта составляются свои рабочие программы. При компиляции используются заранее разработанные математиче-

ские модели типовых компонентов, известные функции для отображения входных воздействий и т. п. из соответствующих библиотек.

Постпроцессор представляет результаты анализа в табличной и графической формах, это могут быть зависимости фазовых переменных от времени, значения выходных параметров-функционалов и т. п.

## **Тема 12. Математическое обеспечение анализа на микроуровне**

**Математические модели на микроуровне.** Математическими моделями на микроуровне являются дифференциальные уравнения в частных производных или интегральные уравнения, описывающие поля физических величин. Другими словами, на микроуровне используются модели с распределенными параметрами. В качестве независимых переменных в моделях могут фигурировать пространственные переменные  $x_1, x_2, x_3$  и время  $t$ .

Характерными примерами моделей могут служить уравнения математической физики вместе с заданными краевыми условиями.

Например:

1) уравнение теплопроводности

$$C\rho\partial T / \partial t = \text{div}(\lambda\text{grad}T) + g ,$$

где  $C$  – удельная теплоемкость;  $\rho$  – плотность;  $T$  – температура;  $t$  – время;  $\lambda$  – коэффициент теплопроводности;  $g$  – количество теплоты, выделяемой в единицу времени в единице объема;

2) уравнение диффузии

$$\partial N / \partial t = \text{div}(D\text{grad}N) ,$$

где  $N$  – концентрация частиц;  $D$  – коэффициент диффузии.

Краевые условия включают начальные условия, характеризующие пространственное распределение зависимых переменных в начальный момент времени, и граничные, задающие значения этих переменных на границах рассматриваемой области в функции времени.

**Методы анализа на микроуровне.** В САПР решение дифференциальных или интегро-дифференциальных уравнений с частными производными выполняется численными методами. Эти методы основаны на дискретизации независимых переменных – их представлении конечным множеством значений в выбранных узловых точках исследуемого пространства. Эти точки рассматриваются как узлы не-

которой сетки, поэтому используемые в САПР методы – это *сеточные методы*.

Среди сеточных методов наибольшее распространение получили два метода: метод конечных разностей (МКР) и метод конечных элементов (МКЭ). Обычно выполняют дискретизацию пространственных независимых переменных, т. е. используют пространственную сетку. В этом случае результатом дискретизации является система обыкновенных дифференциальных уравнений для задачи нестационарной или система алгебраических уравнений для стационарной.

Пусть необходимо решить уравнение

$$LV(z) = fz$$

с заданными краевыми условиями  $MV(z) = \psi(z)$ , где  $L$  и  $M$  – дифференциальные операторы,  $V(z)$  – фазовая переменная,  $z = (x_1, x_2, x_3, t)$  – вектор независимых переменных,  $f(z)$  и  $\psi(z)$  – заданные функции независимых переменных.

В методе конечных разностей алгебраизация производных по пространственным координатам базируется на аппроксимации производных конечно-разностными выражениями. При использовании метода нужно выбрать шаги сетки по каждой координате и вид шаблона. Под шаблоном понимают множество узловых точек, значения переменных в которых используются для аппроксимации производной в одной конкретной точке.

Примеры шаблонов для одномерных и двумерных задач приведены на рис. 3.11. На данном рисунке кружком большего диаметра обозначены узлы, в которых аппроксимируется производная. Черными точками обозначены узлы, значения фазовой переменной в которых входят в аппроксимирующее выражение. Число, записанное около узла, равно коэффициенту, с которым значение фазовой переменной входит в аппроксимирующее выражение. Так, для одномерных шаблонов в верхней части рисунка показана аппроксимация производной  $dV/dx$  в точке  $k$ , и указанным шаблонам при их просмотре слева направо соответствуют аппроксимации

$$h(\partial V / \partial x) = V_{k+1} - V_k; \quad 2h(\partial V / \partial x) = V_{k+1} - V_{k-1};$$

$$h^2(\partial^2 V / \partial x^2) = V_{k+1} - 2V_k + V_{k-1},$$

где  $h$  – шаг дискретизации по оси  $x$ .

Шаблоны для двумерных задач в нижней части (рис. 3.11) соответствуют следующим конечно-разностным операторам:

– левый рисунок

$$h^2 \nabla^2 V = C^2 (\partial^2 V / \partial x_1^2 + \partial^2 V / \partial x_2^2) = V_{k+1j} + V_{k-1j} + V_{kj+1} + V_{kj-1} - 4V_{kj};$$

– средний рисунок

$$h^2 \nabla^2 V = V_{k+1j+1} + V_{k-1j+1} + V_{k+1j-1} + V_{k-1j-1} - 4V_{kj};$$

– правый рисунок

$$4h^2 \partial^2 V / \partial x_1 \partial x_2 = V_{k+1j+1} - V_{k-1j+1} - V_{k+1j-1} + V_{k-1j-1}.$$

Здесь  $V_{kj}$  – значение  $V$  в точке  $(x_{1k}, x_{2j})$ ; приняты одинаковые значения шагов  $h$  по обеим координатам.

Метод конечных элементов основан на аппроксимации не производных, а самого решения  $V(z)$ . Но поскольку оно неизвестно, то аппроксимация выполняется выражениями с неопределенными коэффициентами  $q_t$ , где  $Q^T = (q_1, q_2, \dots, q_n)^T$  вектор-строка неопределенных коэффициентов,  $\varphi(z)$  – вектор-столбец координатных (иначе опорных) функций, заданных так, что удовлетворяются граничные условия.

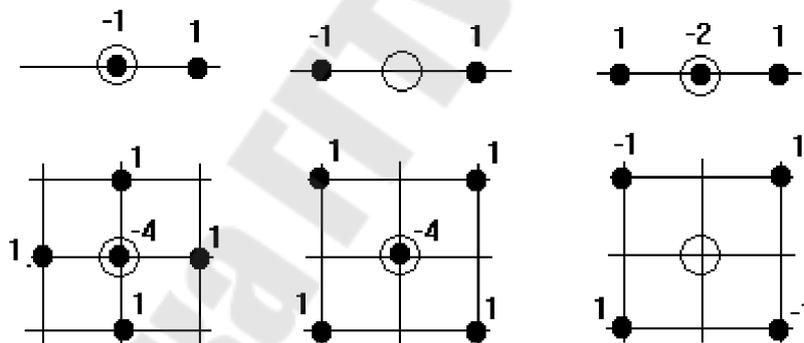


Рис. 3.11. Примеры шаблонов для метода конечных разностей

$$U(z) = Q^T \varphi(z). \quad (3.34)$$

При этом речь идет об аппроксимациях решения в пределах конечных элементов, а с учетом их малых размеров можно говорить об использовании сравнительно простых аппроксимирующих выражений  $U(z)$  (например,  $\varphi(z)$  – полиномы низких степеней). В результате подстановки  $U(z)$  в исходное дифференциальное уравнение и выполнения операций дифференцирования получаем систему невязок

$$\Delta(z, Q) = LU(z) - f(z) = L(Q^T \varphi(z)) - f(z), \quad (3.35)$$

из которой требуется найти вектор  $Q$ .

Эту задачу (определение  $Q$ ) решают одним из следующих методов: 1) *метод коллокаций*, в котором, используя (3.35), формируют  $n$  уравнений с неизвестным вектором  $Q$ :

$$L(Q^T \varphi(z_i)) - f(z_i) = 0, i = 1, 2, \dots, n,$$

где  $n$  – число неопределенных коэффициентов;

2) *метод наименьших квадратов*, основанный на минимизации квадратов невязок (3.35) в  $n$  точках или в среднем по рассматриваемой области;

3) *метод Галеркина*, с помощью которого минимизируются в среднем по области невязки со специально задаваемыми весовыми коэффициентами.

Наибольшее распространение МКЭ получил в САПР машиностроения для анализа прочности объектов. Для этой задачи можно использовать рассмотренный подход, т. е. выполнить алгебраизацию исходного уравнения упругости (уравнения Ламе). Однако более удобным в реализации МКЭ оказался подход, основанный на вариационных принципах механики.

**МКЭ в программах анализа механической прочности.** В качестве исходного положения принимают вариационный принцип Лагранжа (принцип потенциальной энергии), в соответствии с которым равновесное состояние, в которое может прийти система, характеризуется минимумом потенциальной энергии.

Потенциальная энергия  $\Pi$  определяется как разность энергии  $\mathcal{E}$  деформации тела и работы  $A$  массовых и приложенных поверхностных сил. В свою очередь

$$\mathcal{E} = 0,5 \int_R \varepsilon^T \sigma dR, \quad (3.36)$$

где  $\varepsilon^T = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{13}, \varepsilon_{23})^T$  – вектор-строка деформаций;  $\sigma = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23})$  – вектор-столбец напряжений;  $R$  – рассматриваемая область. Деформации  $\varepsilon_{ij}$  можно выразить через перемещения

$$\varepsilon_{ij} = 0,5(\partial V_i / \partial x_j + \partial W_j / \partial x_i), \quad (3.37)$$

где  $W_i$  – перемещение вдоль оси  $x_i$  или в матричной форме

$$\varepsilon = 0,5SW, \quad (3.38)$$

где  $S$  – очевидный из (3.37) оператор дифференцирования.

Деформации и напряжения связаны между собой с помощью матрицы  $D$ , характеризующей упругие свойства среды, которая представлена в табл. 3.5:

$$\sigma = D\varepsilon. \quad (3.39)$$

Коэффициенты  $\lambda$  и  $\mu$ , фигурирующие в табл. 3.5, называют постоянными Ламе, они выражают упругие свойства материала детали.

Подставляя (3.39) и (3.38) в (3.36), получаем

$$\Theta = 0,5 \int_R W^T S^T D S W dR.$$

Таблица 3.5

$\lambda + 2\mu$	$\lambda$	0	0	0	0
$\lambda$	$\lambda + 2\mu$	$\lambda$	0	0	0
0	$\lambda$	$\lambda + 2\mu$	0	0	0
0	0	0	$2\mu$	0	0
0	0	0	0	$2\mu$	0
0	0	0	0	0	$2\mu$

Решением задачи должно быть поле перемещений  $W(X)$ , где  $X = (x_1, x_2, x_3)$ . В соответствии с МКЭ это решение аппроксимируется с помощью функций (3.34), которые применительно к совокупности конечных элементов представим в матричной форме:

$$U(X) = NQ,$$

где  $N$  – матрица координатных функций;  $Q$  – вектор неопределенных коэффициентов. Заменяя  $W(X)$  на  $U(X)$ , получаем

$$\Theta = 0,5 \int_R N^T S^T D S N Q dR = 0,5 Q^T \left( \int_R (S N)^T D S N dR \right) Q = 0,5 Q^T K Q, \quad (3.40)$$

где  $K = \int (S N)^T D S N dR$  – матрица жесткости.

В соответствии с принципом потенциальной энергии в состоянии равновесия имеем

$$\partial\Pi/\partial Q = \partial\mathcal{E}/\partial Q - \partial A/\partial Q = 0$$

или, дифференцируя (3.40), находим

$$KQ = B, \quad (3.41)$$

где  $B = \partial A/\partial Q$  – вектор нагрузок. Таким образом, задача анализа прочности, согласно МКЭ, сведена к решению системы линейных алгебраических уравнений (3.41).

Матрица жесткости также оказывается сильно разреженной, поэтому для решения (3.41) применяют методы разреженных матриц.

*Примечание.* Одним из широко известных методов разреженных матриц является метод прогонки, применяемый в случае трехдиагональных матриц коэффициентов в системе алгебраических уравнений.

**Структура программ анализа по МКЭ на микроуровне.** Основными частями программы анализа по МКЭ являются библиотеки конечных элементов, препроцессор, решатель и постпроцессор.

*Библиотеки конечных элементов (КЭ)* содержат модели КЭ – их матрицы жесткости. Очевидно, что модели КЭ будут различными для разных задач (анализ упругих или пластических деформаций, моделирование полей температур, электрических потенциалов и т. п.), разных форм КЭ (например, в двумерном случае – треугольные или четырехугольные элементы), разных наборов координатных функций.

*Исходные данные для препроцессора* – геометрическая модель объекта, чаще всего получаемая из подсистемы конструирования. Основная функция препроцессора – представление исследуемой среды (детали) в сеточном виде, т. е. в виде множества конечных элементов.

*Решатель* – программа, которая ассемблирует (собирает) модели отдельных КЭ в общую систему алгебраических уравнений (3.41) и решает эту систему одним из методов разреженных матриц.

*Постпроцессор* служит для визуализации результатов решения в удобной для пользователя форме. В машиностроительных САПР это графическая форма. Пользователь может видеть исходную (до нагружения) и деформированную формы детали, поля напряжений, температур, потенциалов и т. п. в виде цветных изображений, в которых палитра цветов или интенсивность свечения характеризуют значения фазовой переменной.

Мировыми лидерами среди программ конечно-элементного анализа являются программно-методические комплексы NASTRAN, ANSYS, COSMOS.

Как правило, эти комплексы включают в себя ряд программ, родственных по МО, интерфейсам, общности некоторых используемых модулей. Эти программы различаются ориентацией на разные приложения, степенью специализации, ценой или выполняемой обслуживающей функцией. Например, в комплексе ANSYS основные решающие модули позволяют выполнять анализ механической прочности, теплопроводности, динамики жидкостей и газов, акустических и электромагнитных полей. Во все варианты программ входят пре- и постпроцессоры, а также интерфейс с базой данных. Предусмотрен экспорт (импорт) данных между ANSYS и ведущими комплексами геометрического моделирования и машинной графики.

## Глава 4. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИНТЕЗА ПРОЕКТНЫХ РЕШЕНИЙ

### Тема 13. Постановка задач параметрического синтеза

**Место процедур синтеза в проектировании.** Сущность проектирования заключается в принятии проектных решений, обеспечивающих выполнение будущим объектом предъявляемых к нему требований. Синтез проектных решений – основа проектирования; от успешного выполнения процедуры синтеза в определяющей мере зависят потребительские свойства будущей продукции. Конечно, анализ – необходимая составная часть проектирования, служащая для верификации принимаемых проектных решений. Именно анализ позволяет получить необходимую информацию для целенаправленного выполнения процедур синтеза в итерационном процессе проектирования. Поэтому синтез и анализ неразрывно связаны.

Как отмечено в главе 1, синтез подразделяют на параметрический и структурный. Проектирование начинается со *структурного синтеза*, при котором генерируется принципиальное решение. Таким решением может быть облик будущего летательного аппарата, или физический принцип действия датчика, или одна из типовых конструкций двигателя, или функциональная схема микропроцессора. Но эти конструкции и схемы выбирают в параметрическом виде, т. е. без указания числовых значений параметров элементов. Поэтому, прежде чем приступить к верификации проектного решения, нужно задать или рассчитать значения этих параметров, т. е. выполнить *параметрический синтез*. Примерами результатов параметрического синтеза могут служить геометрические размеры деталей в механическом узле или параметры режимов резания в технологической операции и т. п.

В случае если по результатам анализа проектное решение признается неокончательным, то начинается процесс последовательных приближений к приемлемому варианту проекта. Во многих приложениях для улучшения проекта удобнее варьировать значения параметров элементов, т. е. использовать параметрический синтез на базе

многовариантного анализа. При этом задача параметрического синтеза может быть сформулирована как задача определения значений параметров элементов, наилучших с позиций удовлетворения требований технического задания при неизменной структуре проектируемого объекта. Тогда параметрический синтез называют параметрической оптимизацией или просто *оптимизацией*. Если параметрический синтез не приводит к успеху, то повторяют процедуры структурного синтеза, т. е. на очередных итерациях корректируют или перевыбирают структуру объекта.

**Критерии оптимальности.** В САПР процедуры параметрического синтеза выполняются либо человеком в процессе многовариантного анализа (в интерактивном режиме), либо реализуются на базе формальных методов оптимизации (в автоматическом режиме). В последнем случае находят применение несколько постановок задач оптимизации.

Наиболее распространенной является детерминированная постановка: заданы условия работоспособности на выходные параметры  $Y$  и нужно найти номинальные значения проектных параметров  $X$ , к которым относятся параметры элементов проектируемого объекта. Назовем эту задачу оптимизации базовой. В частном случае, когда требования к выходным параметрам заданы нечетко, к числу рассчитываемых величин могут быть отнесены также нормы выходных параметров, фигурирующие в их условиях работоспособности.

Если проектируются изделия для дальнейшего серийного производства, то важное значение приобретает такой показатель, как процент выпуска годных изделий в процессе производства. Успешное выполнение условий работоспособности в номинальном режиме не гарантирует их выполнения при учете производственных погрешностей, задаваемых допусками параметров элементов. Поэтому целью оптимизации становится максимизация процента выхода годных, а к результатам решения задачи оптимизации относятся не только номинальные значения проектных параметров, но и их допуски.

Базовая задача оптимизации ставится как задача математического программирования:

$$\begin{aligned} \text{extr} F(X) \quad X \in D_x; \\ D_x = \{X | \varphi(X) > 0, y(X) = 0\}, \end{aligned} \quad (4.1)$$

где  $F(X)$  – целевая функция,  $X$  – вектор управляемых (проектных) параметров,  $\varphi(X)$  и  $\psi(X)$  – функции-ограничения;  $D_x$  – допустимая область в пространстве управляемых параметров.

Запись (4.1) интерпретируется как задача поиска экстремума целевой функции путем варьирования управляемых параметров в пределах допустимой области. Таким образом, для выполнения расчета номинальных значений параметров необходимо, во-первых, сформулировать задачу (4.1), во-вторых, решить задачу поиска экстремума  $F(X)$ .

Сложность постановки оптимизационных проектных задач обусловлена наличием у проектируемых объектов нескольких выходных параметров, которые могут быть критериями оптимальности, но в задаче (4.1) целевая функция должна быть одна. Другими словами, проектные задачи являются многокритериальными, и возникает проблема сведения многокритериальной задачи к однокритериальной.

Применяют несколько способов выбора критерия оптимальности.

В частном критерии среди выходных параметров один выбирают в качестве целевой функции, а условия работоспособности остальных выходных параметров относят к ограничениям задачи (4.1). Эта постановка вполне приемлема, если действительно можно выделить один наиболее критичный выходной параметр. Но в большинстве случаев сказывается недостаток частного критерия (рис. 4.1).



Рис. 4.1. Области Парето и работоспособности

На рис. 4.1 представлено двумерное пространство выходных параметров  $y_1$  и  $y_2$ , для которых заданы условия работоспособности  $y_1 < T_1$  и  $y_2 < T_2$ . Кривая  $AB$  является границей достижимых значений выходных параметров. Это ограничение объективное и связано с существующими физическими и технологическими условиями производства, называемыми условиями реализуемости. Область, в пределах которой

выполняются все условия реализуемости и работоспособности, называют *областью работоспособности*. Множество точек пространства выходных параметров, из которых невозможно перемещение, приводящее к улучшению всех выходных параметров, называют областью компромиссов, или *областью Парето*. Участок кривой  $AB$  (рис. 4.1) относится к области Парето.

Если в качестве целевой функции (рис. 4.1) выбрать параметр  $y_1$ , то результатом оптимизации будут параметры  $X$ , соответствующие точке  $B$ . Но это граница области работоспособности и, следовательно, при нестабильности внутренних и внешних параметров велика вероятность выхода за пределы области работоспособности. Результаты можно улучшить, если применять так называемый метод уступок, при котором в качестве ограничения принимают условие работоспособности со скорректированной нормой в виде

$$y_2 < T_2 + \Delta,$$

где  $\Delta$  – уступка.

Но возникает проблема выбора значений уступок, т. е. результаты оптимизации будут иметь субъективный характер. Очевидно, что ситуация не изменится, если целевой функцией будет выбран параметр  $y_2$ , – оптимизация приведет в точку  $A$ .

*Аддитивный критерий* объединяет (свертывает) все выходные параметры (частные критерии) в одну целевую функцию, представляющую собой взвешенную сумму частных критериев:

$$F(X) = \sum_{j=1}^m \omega_j y_j(X), \quad (4.2)$$

где  $\omega_j$  – весовой коэффициент;  $m$  – число выходных параметров. Функция (4.2) подлежит минимизации, при этом если условие работоспособности имеет вид  $y_j > T_j$ , то  $\omega_j < 0$ .

Недостатки аддитивного критерия – субъективный подход к выбору весовых коэффициентов и неучет требований ТЗ. Действительно в (4.2) не входят нормы выходных параметров.

Аналогичные недостатки присущи и *мультипликативному критерию*, целевая функция которого имеет вид

$$F(X) = \prod_{j=1}^m y_j^{\omega_j}(X), \quad (4.3)$$

где  $j = 1$ .

Нетрудно видеть, что если прологарифмировать (4.3), то мультипликативный критерий превращается в аддитивный.

Более предпочтительным является *максиминный критерий*, в качестве целевой функции которого принимают выходной параметр, наиболее неблагоприятный с позиций выполнения условий работоспособности. Для оценки степени выполнения условия работоспособности  $j$ -го выходного параметра вводят запас работоспособности этого параметра  $S_j$  и этот запас можно рассматривать как нормированный  $j$ -й выходной параметр. Например (здесь и далее для лаконичности изложения предполагается, что все выходные параметры приведены к виду, при котором условия работоспособности становятся неравенствами в форме  $y_j < T_j$ ):

$$S_j = (T_j - y_j) / T_j$$

или

$$S_j = (T_j - y_{\text{ном}j}) / \sigma_j; S_j = (T_j - y_{\text{ном}j}) / \delta_j,$$

где  $y_{\text{ном}j}$  – номинальное значение;  $\delta_j$  – некоторая характеристика рассеяния  $j$ -го выходного параметра, например, трехсигмовый допуск. Тогда целевая функция в максиминном критерии есть

$$F(X) = \min_{j \in [1:m]} Z_j(X).$$

Здесь запись  $[1:m]$  означает множество целых чисел в диапазоне от 1 до  $m$  и в максиминном критерии конкретизируется следующим образом:

$$F(X) = \max_{X \in D_x} \min_{j \in [1:m]} Z_j(x);$$

$$X \in D_x \quad j \in [1:m],$$

где допустимая область  $d_x$  определяется только прямыми ограничениями на управляемые параметры  $x_i$ :

$$x_{i \min} < x_i < x_{i \max}.$$

**Задачи оптимизации с учетом допусков.** Содержательную сторону оптимизации с учетом допусков поясняет рис. 4.2, на котором представлены области работоспособности и допусковая в двумерном пространстве управляемых параметров. Если собственно допуски заданы и не относятся к управляемым параметрам, то цель оптимизации

максимально совместить эти области так, чтобы вероятность выхода за пределы области работоспособности была минимальной.



Рис. 4.2. Области допусковая и работоспособности

Решение этой задачи исключительно трудоемко, так как на каждом шаге оптимизации нужно выполнять оценку упомянутой вероятности методами статистического анализа, а для сложных моделей объектов таким методом является метод статистических испытаний. Поэтому на практике подобные задачи решают, принимая те или иные допущения.

Например, если допустить, что цель оптимизации достигается при совмещении центров областей работоспособности Э и допусковой  $X_{ном}$ , то оптимизация сводится к задаче центрирования, т. е. к определению центра Э. Задачу центрирования обычно решают путем предварительного нормирования управляемых параметров  $x_i$  с последующим вписыванием гиперкуба с максимально возможными размерами в нормированную область работоспособности.

*Примечание.* Нормирование проводят таким образом, что допусковая область приобретает форму гиперкуба, получающегося после нормирования. Очевидно, что решение задачи центрирования позволяет не только оптимизировать номинальные значения проектных параметров, но и их допуски, если последние относятся к управляемым параметрам.

## Тема 14. Методы оптимизации

**Классификация методов математического программирования.** В САПР основными методами оптимизации являются поисковые методы. Поисковые методы основаны на пошаговом изменении управляемых параметров

$$X_{k+1} = X_k + \Delta X_k, \quad (4.5)$$

где в большинстве методов приращение  $\Delta X_k$  вектора управляемых параметров вычисляется по формуле

$$\Delta X_k = hg(X_k). \quad (4.6)$$

Здесь  $X_k$  – значение вектора управляемых параметров на  $k$ -м шаге,  $h$  – шаг, а  $g(X_k)$  – направление поиска. Следовательно, если выполняются условия сходимости, то реализуется пошаговое (итерационное) приближение к экстремуму.

Методы оптимизации классифицируют по ряду признаков.

В зависимости от числа управляемых параметров различают методы *одномерной* и *многомерной* оптимизации, в первых из них управляемый параметр единственный, во вторых размер вектора  $X$  не менее двух. Реальные задачи в САПР многомерны, методы одномерной оптимизации играют вспомогательную роль на отдельных этапах многомерного поиска.

Различают методы *условной* и *безусловной* оптимизации по наличию или отсутствию ограничений. Для реальных задач характерно наличие ограничений, однако методы безусловной оптимизации также представляют интерес, поскольку задачи условной оптимизации с помощью специальных методов могут быть сведены к задачам без ограничений.

В зависимости от числа экстремумов различают задачи одно- и многоэкстремальные. Если метод ориентирован на определение какого-либо локального экстремума, то такой метод относится к *локальным* методам. Если же результатом является глобальный экстремум, то метод называют *методом глобального поиска*. Удовлетворительные по вычислительной эффективности методы глобального поиска для общего случая отсутствуют и потому на практике в САПР используют методы поиска локальных экстремумов.

Наконец, в зависимости от того, используются при поиске производные целевой функции по управляемым параметрам или нет, различают методы нескольких порядков. Если производные не используются, то имеет место метод *нулевого порядка*, если используются первые или вторые производные, то соответственно метод *первого* или *второго порядка*. Методы первого порядка называют также градиентными, поскольку вектор первых производных  $F(X)$  по  $X$  есть градиент целевой функции

$$\text{grad}(F(X)) = (\partial F / \partial x_1, \partial F / \partial x_2 \dots \partial F / \partial x_n).$$

Конкретные методы определяются следующими факторами: способом вычисления направления поиска  $g(X_k)$  в формуле (4.6); способом выбора шага  $h$ ; способом определения окончания поиска.

Определяющим фактором является первый из перечисленных в этом списке, он подробно описан ниже. Шаг может быть или постоянным, или выбираться исходя из одномерной оптимизации – поиска минимума целевой функции в выбранном направлении  $g(X_k)$ . В последнем случае шаг будем называть оптимальным.

Окончание поиска обычно осуществляют по правилу: если на протяжении  $r$  подряд идущих шагов траектория поиска остается в малой  $\varepsilon$ -окрестности текущей точки поиска  $X_k$  то поиск следует прекратить, следовательно, условие окончания поиска имеет вид

$$|X_k - X_{k-1}| < \varepsilon.$$

**Методы одномерной оптимизации.** К методам одномерной оптимизации относятся методы дихотомического деления, золотого сечения, чисел Фибоначчи, полиномиальной аппроксимации и ряд их модификаций.

Пусть задан отрезок, на котором имеется один минимум (в общем случае нечетное число минимумов). Согласно *методу дихотомического деления* (рис. 4.3, а), отрезок делят пополам и в точках, отстоящих от центра  $C$  отрезка на величину допустимой погрешности  $q$ , рассчитывают значения целевой функции  $F(C + q)$  и  $F(C - q)$ . Если окажется, что  $F(C + q) > F(C - q)$ , то минимум находится на отрезке  $[A, C]$ , если  $F(C + q) < F(C - q)$ , то минимум – на  $[C, B]$ , если  $F(C + q) = F(C - q)$  – на  $[C - q, C + q]$ . Таким образом, на следующем шаге вместо отрезка  $[A, B]$  нужно исследовать суженный отрезок  $[A, C]$ ,  $[C, B]$  или  $[C - q, C + q]$ . Шаги повторяются, пока длина отрезка не уменьшится до величины погрешности  $q$ . Таким образом, требуется не более  $N$  шагов, где  $N$  – ближайшее к  $\log((B - A)/q)$  целое значение, но на каждом шаге целевую функцию следует вычислять дважды.

По *методу золотого сечения* (рис. 4.3, б) внутри отрезка  $[A, B]$  выделяют две промежуточные точки  $C_1$  и  $D_1$  на расстоянии  $s = aL$  от его конечных точек, где  $L = B - A$  – длина отрезка. Затем вычисляют значения целевой функции  $F(x)$  в точках  $C_1$  и  $D_1$ . Если  $F(C_1) < F(D_1)$ , то минимум находится на отрезке  $[A, D_1]$ , если  $F(C_1) > F(D_1)$ , то – на отрезке  $[C_1, B]$ , если  $F(C_1) = F(D_1)$  – на отрезке  $[C_1, D_1]$ . Следовательно,

вместо отрезка  $[A, B]$  теперь можно рассматривать отрезок  $[A, D_1]$ ,  $[C_1, B]$  или  $[C_1, D_1]$ , т. е. длина отрезка уменьшилась не менее чем в  $L/(L - aL) = 1/(1 - a)$  раз. Если подобрать значение  $a$  так, что в полученном отрезке меньшей длины одна из промежуточных точек совпадет с промежуточной точкой от предыдущего шага, т. е. в случае выбора отрезка  $[A, D_1]$  точка  $D_2$  совпадет с точкой  $C_1$ , а в случае выбора отрезка  $[C_1, B]$  точка  $C_2$  – с точкой  $D_1$ , то это позволит сократить число вычислений целевой функции на всех шагах (кроме первого) в 2 раза. Условие получения такого значения  $a$  формулируется следующим образом:  $(1 - 2a)L_k = aL_{k-1}$  откуда с учетом того, что  $L_k/L_{k-1} = 1/(1 - a)$ , имеем  $a = 0,382$ . Это значение и называют *золотым сечением*.

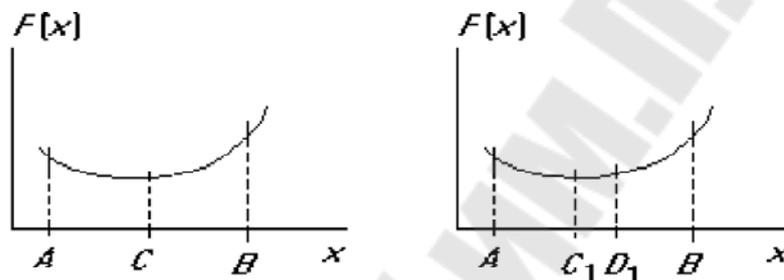


Рис. 4.3. Одномерная минимизация:  $a$  – дихотомическое деление;  $b$  – золотое сечение

Таким образом, требуется не более  $N$  шагов и  $N + 1$  вычисление целевой функции, где  $N$  можно рассчитать, используя соотношение  $(B - A)/E = (1 - a)N$  при заданной погрешности  $E$  определения экстремума.

Согласно *методу чисел Фибоначчи*, используют числа Фибоначчи  $R_i$ , последовательность которых образуется по правилу  $R_{i+2} = R_{i+1} + R_i$  при  $R_0 = R_1 = 1$ , т. е. ряд чисел Фибоначчи имеет вид 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144... Метод аналогичен методу золотого сечения с тем отличием, что коэффициент  $a$  равен отношению  $R_{i-2}/R_i$ , начальное значение  $i$  определяется из условия, что  $R_i$  должно быть наименьшим числом Фибоначчи, превышающим величину  $(B - A)/E$ , где  $E$  – заданная допустимая погрешность определения экстремума. Так, если  $(B - A)/E = 100$ , то начальное значение  $i = 12$ , поскольку  $R_{11} = 144$ , и  $a = 55/144 = 0,3819$ , на следующем шаге будет  $a = 34/89 = 0,3820$  и т. д.

По *методу полиномиальной аппроксимации* при аппроксимации  $F(x)$  квадратичным полиномом

$$P(x) = a_0 + a_1x + a_2x^2 \quad (4.7)$$

выбирают промежуточную точку  $C$  и в точках  $A, B, C$  вычисляют значения целевой функции. Далее решают систему из трех алгебраических уравнений, полученных подстановкой в (4.7) значений  $A, B, C$  вместо  $x$  и вычисленных значений функции вместо  $P(x)$ . В результате становятся известными значения коэффициентов  $a_k$  в (4.7) и, исходя из условия  $dP(x)/dx = 0$ , определяют экстремальную точку  $\mathcal{E}$  полинома. Например, если точка  $C$  выбрана в середине отрезка  $[A, B]$ , то

$$\mathcal{E} = C + (C - A)(F(A) - F(B)) / (2(F(A) - 2F(C) + F(B))).$$

**Методы безусловной оптимизации.** Среди методов нулевого порядка в САПР находят применение методы Розенброка, конфигураций (Хука–Дживса), деформируемого многогранника (Нелдера–Мида), случайного поиска. К методам с использованием производных относятся методы наискорейшего спуска, сопряженных градиентов, переменной метрики.

*Метод Розенброка* является улучшенным вариантом покоординатного спуска. Метод покоординатного спуска характеризуется выбором направлений поиска поочередно вдоль всех  $n$  координатных осей, шаг рассчитывается на основе одномерной оптимизации, критерий окончания поиска  $|X_k - X_{k-1}| < \varepsilon$ , где  $\varepsilon$  – заданная точность определения локального экстремума,  $n$  – размерность пространства управляемых параметров. Траектория покоординатного спуска для примера двумерного пространства управляемых параметров показана на рис. 4.4, где  $X_k$  – точки на траектории поиска,  $x$  – управляемые параметры. Целевая функция представлена своими линиями равного уровня, около каждой линии записано соответствующее ей значение. Очевидно, что  $\mathcal{E}$  есть точка минимума.

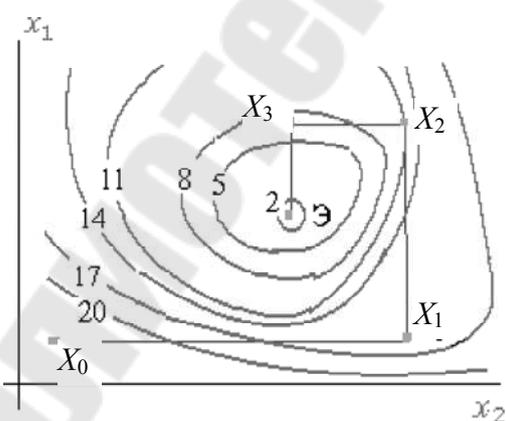


Рис. 4.4. Траектория покоординатного спуска

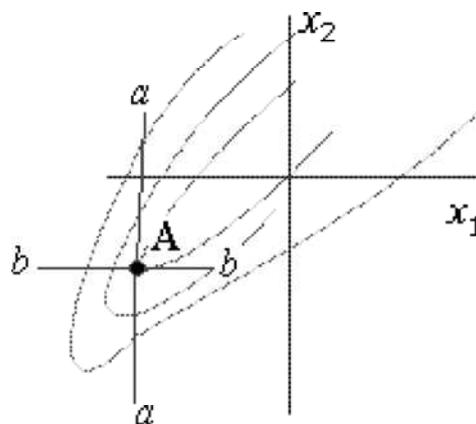


Рис. 4.5. «Застывание» покоординатного спуска на дне оврага

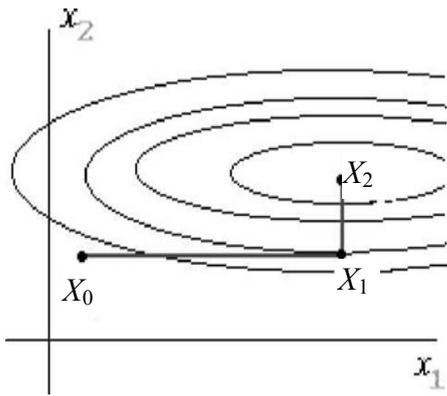


Рис. 4.6. Траектория покоординатного спуска при благоприятной ориентации координатных осей

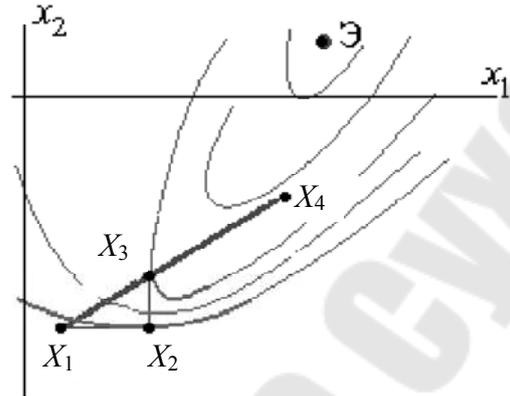


Рис. 4.7. Иллюстрация метода конфигураций

При использовании метода покоординатного спуска велика вероятность «застревания» поиска на дне оврага вдали от точки экстремума. На рис. 4.5 видно, что после попадания в точку  $A$ , расположенную на дне оврага, дальнейшие шаги возможны лишь в направлениях  $aa$  или  $bb$ , но они приводят к ухудшению целевой функции. Следовательно, поиск прекращается в точке  $A$ .

*Примечание.* Оврагом называют часть пространства управляемых параметров, в которой наблюдаются слабые изменения производных целевой функции по одним направлениям и значительные изменения с переменной знака – по некоторым другим направлениям. Знак производной меняется в точках, принадлежащих дну оврага.

В то же время при благоприятной ориентации дна оврага, а именно при положении одной из координатных осей, близком к параллельности с дном оврага, поиск оказывается весьма быстрым. Эта ситуация показана на рис. 4.6.

Метод Розенброка заключается в таком повороте координатных осей, чтобы одна из них оказалась квазипараллельной дну оврага. Такой поворот осуществляют на основе данных, полученных после серии из  $n$  шагов покоординатного спуска. Положение новых осей  $s_i$  может быть получено линейным преобразованием прежних осей  $x_i$ : ось  $s_1$  совпадает по направлению с вектором  $X_{k+n} - X_k$ ; остальные оси выбирают из условия ортогональности к  $X_1$  и друг к другу.

Другой удачной модификацией покоординатного спуска является *метод конфигураций*. В соответствии с этим методом вначале выполняют обычную серию из  $n$  шагов покоординатного спуска, затем делают дополнительный шаг в направлении вектора  $X_k - X_{k-n}$ , как

показано на рис. 4.7, где дополнительный шаг выполняют в направлении вектора  $X_3 - X_1$ , что и приводит в точку  $X_4$ .

Поиск экстремума *методом деформируемого многогранника* основан на построении многогранника с  $(n + 1)$  вершинами на каждом шаге поиска, где  $n$  – размерность пространства управляемых параметров. В начале поиска эти вершины выбирают произвольно, на последующих шагах выбор подчинен правилам метода.

Эти правила поясняются рис. 4.8 на примере двумерной задачи оптимизации. Выбраны вершины исходного треугольника:  $X_1, X_2, X_3$ . Новая вершина  $X_4$  находится на луче, проведенном из худшей вершины  $X_1$  (из вершины с наибольшим значением целевой функции) через центр тяжести ЦТ многогранника, причем рекомендуется  $X_4$  выбирать на расстоянии  $d$  от ЦТ, равном  $|\text{ЦТ} - X_1|$ . Новая вершина  $X_4$  заменяет худшую вершину  $X_1$ . Если оказывается, что  $X_4$  имеет лучшее значение целевой функции среди вершин многогранника, то расстояние  $d$  увеличивают. На рисунке именно эта ситуация имеет место и увеличение  $d$  дает точку  $X_5$ . В новом многограннике с вершинами  $X_2, X_3, X_5$  худшей является вершина  $X_2$ , аналогично получают вершину  $X_6$ , затем вершину  $X_7$  и т. д. Если новая вершина окажется худшей, то в многограннике нужно сохранить лучшую вершину, а длины всех ребер уменьшить, например, вдвое (стягивание многогранника к лучшей вершине). Поиск прекращается при выполнении условия уменьшения размеров многогранника до некоторого предела. *Случайные методы* поиска характеризуются тем, что направления поиска  $g$  выбирают случайным образом.

Особенностью *метода наискорейшего спуска* является выполнение шагов поиска в градиентном направлении

$$X_{k+1} = X_k + h \cdot \text{grad}F(X) / h \cdot \text{grad}F(X) / |\text{grad}F(X)|,$$

при этом шаг  $h$  выбирается оптимальным с помощью одномерной оптимизации.

При использовании метода наискорейшего спуска, как и большинства других методов, эффективность поиска существенно снижается в овражных ситуациях. Траектория поиска приобретает зигзагообразный вид с медленным продвижением вдоль дна оврага в сторону экстремума. Чтобы повысить эффективность градиентных методов, используют несколько приемов.

Один из приемов, использованный в *методе сопряженных градиентов* (называемом также методом Флетчера–Ривса), основан на

понятии сопряженности векторов. Векторы  $A$  и  $B$  называют  $Q$ -сопряженными, если  $A^T Q B = 0$ , где  $Q$  – положительно определенная квадратная матрица того же порядка, что и размер  $N$  векторов  $A$  и  $B$  (частный случай сопряженности – ортогональность векторов, когда  $Q$  является единичной матрицей порядка  $N$ ),  $A^T$  – вектор-строка,  $B$  – вектор-столбец.

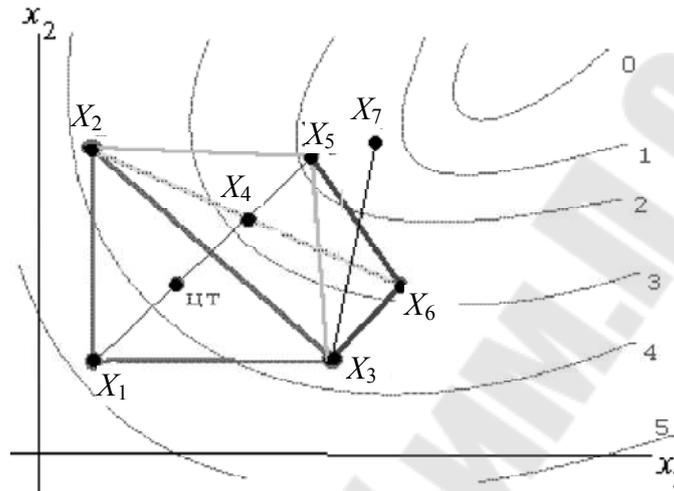


Рис. 4.8. Иллюстрация метода деформируемого многогранника

Особенность сопряженных направлений для  $Q = \Gamma$ , где  $\Gamma$  – матрица Гессе, в задачах с квадратичной целевой функцией  $F(X)$  заключается в следующем: одномерная минимизация  $F(X)$  последовательно по  $N$  сопряженным направлениям позволяет найти экстремальную точку не более, чем за  $N$  шагов.

*Примечание.* Матрицей Гессе называют матрицу вторых частных производных целевой функции по управляемым параметрам.

Основанием для использования поиска по  $\Gamma$ -сопряженным направлениям является то, что для функций  $F(X)$  общего вида может быть применена квадратичная аппроксимация, что на практике выливается в выполнение поиска более, чем за  $N$  шагов.

*Пример.* Поиск экстремума выполняют в соответствии с формулой

$$X_i = X_{i-1} + hS_i. \quad (4.8)$$

Направление  $S_{i+1}$  поиска на очередном шаге связано с направлением поиска  $S_i$  на предыдущем шаге соотношением

$$S_{i+1} = -\text{grad}F(X_i) + w_i S_i, \quad (4.9)$$

где  $w_i$  – коэффициент. Кроме того, учитывают условие сопряженности

$$S_{i+1}^T \Gamma S_i = 0 \quad (4.10)$$

и линейную аппроксимацию  $\text{grad}F(X)$  в окрестностях точки  $X_i$

$$\text{grad}F(X_{i+1}) = \text{grad}F(X) + \Gamma(X_{i+1} - X_i). \quad (4.11)$$

Поскольку шаг  $h$  рассчитывается исходя из условия одномерной оптимизации, то, во-первых, справедливо соотношение

$$S_i^T \text{grad}F(X_i) = 0, \quad (4.12)$$

во-вторых, имеем

$$X_i = X_{i-1} + hw_{i-1}S_{i-1} - h\text{grad}F(X_{i-1}),$$

откуда получаем

$$\partial F / \partial h = (\partial F(X) / \partial X)(\partial X / \partial h) = \text{grad}F(X_i) \text{grad}F(X_{i-1}) = 0. \quad (4.13)$$

Алгоритм поиска сводится к применению формулы (4.9), пока не будет выполнено условие окончания вычислений

$$|\text{grad}F(X_k)| < \varepsilon.$$

Чтобы определить коэффициент  $w_i$ , решают систему уравнений (4.8)–(4.13) путем подстановки в (4.10) величин  $S_{i+1}$  из (4.9) и  $S_i$  из (4.8):

$$\begin{aligned} S_{i+1}^T \Gamma S_i &= (w_i S_i - \text{grad}F(X_i))^T \Gamma (X_i - X_{i-1}) / h = \\ &= (w_i S_i - \text{grad}F(X_i))^T \Gamma \Gamma^{-1} (\text{grad}F(X_i) - \text{grad}F(X_{i-1})) / h = 0 \end{aligned}$$

или

$$(w_i S_i - \text{grad}F(X_i))^T (\text{grad}F(X_i) - \text{grad}F(X_{i-1})) = 0,$$

откуда

$$\begin{aligned} w_i S_i^T - (\text{grad}F(X_i) - \text{grad}F(X_{i-1})) - \text{grad}F(X_i)^T \text{grad}F(X_i) + \\ + \text{grad}F(X_i)^T + \text{grad}F(X_{i-1}) = 0, \end{aligned}$$

с учетом (4.12) и (4.13)

$$w_i S_i^T \text{grad}F(X_{i-1}) + \text{grad}F(X_i)^T \text{grad}F(X_i) = 0.$$

Следовательно,

$$w_i = \text{grad}F(X_i)^T \text{grad}F(X_i) / S_i^T \text{grad}F(X_{i-1}). \quad (4.14)$$

На первом шаге поиска выбирают  $S_1 = -\text{grad} F(X_0)$  и находят точку  $X_1$ . На втором шаге по формуле (4.14) рассчитывают  $w_1$ , по формулам (4.9) и (4.8) определяют  $S_2$  и  $X_2$  и т. д.

*Метод переменной метрики* (иначе метод Девидона–Флетчера–Пауэлла) можно рассматривать как результат усовершенствования метода второго порядка – метода Ньютона. *Метод Ньютона* основан на использовании необходимых условий безусловного экстремума целевой функции  $F(X)$ :

$$\text{grad}F(X) = 0. \quad (4.15)$$

Выражение (4.15) представляет собой систему алгебраических уравнений, для решения которой можно применить известный численный метод, называемый методом Ньютона. Корень системы (4.15) есть стационарная точка, т. е. возможное решение экстремальной задачи. Метод Ньютона является итерационным, он основан на линеаризации (4.15) в окрестности текущей точки поиска  $X_k$ :

$$\text{grad}F(X) = \text{grad}F(X_k) + \Gamma(X - X_k) = 0. \quad (4.16)$$

Выражение (4.16) – это система линейных алгебраических уравнений. Ее корень есть очередное приближение  $X_{k+1}$  к решению

$$X_{k+1} = X_k - \Gamma^{-1}(X_k) \text{grad}F(X_k).$$

Если процесс сходится, то решение достигается за малое число итераций, окончанием которых служит выполнение условия

$$|X_{k+1} - X_k| < \varepsilon.$$

Недостаток метода – высокая трудоемкость вычисления и обращения матрицы  $\Gamma$ , к тому же ее вычисление численным дифференцированием сопровождается заметными погрешностями, что снижает скорость сходимости. В методе переменной метрики вместо трудно вычисляемой обратной матрицы Гессе используют некоторую более легко вычисляемую матрицу  $N$ , т. е.

$$X_{k+1} = X_k N \text{grad}F(X_k).$$

Введем обозначения:

$$dg_k = \text{grad}F(X_k) - \text{grad}F(X_{k-1}); \quad dX_k = X_k - X_{k-1};$$

$E$  – единичная матрица. Начальное значение матрицы  $N_0 = E$ . Матрицу  $N$  корректируют на каждом шаге, т. е.

$$N_{k+1} = N_k + A_k^T B_k,$$

где

$$A_k = dX_k dX_k^T / (dX^T dg_k), \quad B_k = N_k dg_k dg_k^T N_k^T / (dg_k^T N_k dg_k).$$

Поэтому

$$N_{k+1} = E + \sum_{i=0}^k A_i - \sum_{i=0}^k B_i.$$

Можно показать, что  $A_i$  стремится к  $\Gamma^{-1}$ ,  $B_i$  – к  $E$  при  $k \rightarrow n$ , где  $n$  метров. Спустя  $n$  шагов, нужно снова начинать с  $N_{n+1} = E$ .

**Необходимые условия экстремума.** В задачах безусловной оптимизации необходимые условия представляют собой равенство нулю градиента целевой функции

$$\text{grad}F(X) = 0.$$

В общей задаче математического программирования (4.1) необходимые условия экстремума, называемые условиями Куна–Таккера, формулируются следующим образом.

Для того чтобы точка  $\mathcal{E}$  была экстремальной точкой выпуклой задачи математического программирования (ЗМП), необходимо наличие неотрицательных коэффициентов  $u_i$ , таких, что

$$u_i \varphi_i(\mathcal{E}) = 0, \quad i = 1, 2, \dots, m; \quad (4.17)$$

при этом соблюдались ограничения задачи и выполнялось условие

$$\text{grad}F(Q) + \sum_{i=1}^m u_i \text{grad}\varphi_i(\mathcal{E}) + \sum_{j=1}^L a_j \psi_j(\mathcal{E}) = 0, \quad (4.18)$$

где  $m$  – число ограничений типа неравенств;  $L$  – то же равенств, коэффициенты  $a_j > 0$ .

За приведенной абстрактной формулировкой условий скрывается достаточно просто понимаемый геометрический смысл. Действительно, рассмотрим сначала случай с ограничениями только типа не-

равенств. Если максимум находится внутри допустимой области  $R$ , то, выбирая все  $u_i = 0$ , добиваемся выполнения (4.17); если точка максимума  $\Theta$  лежит на границе области  $R$ , то, как видно из левой части рис. 4.9, эту точку всегда соответствующим подбором неотрицательных  $u_i$  можно поместить внутрь оболочки, натянутой на градиенты целевой функции  $F(X)$  и функций ограничений  $\varphi_i(X)$ .

Наоборот, если точка не является экстремальной, то (4.17) нельзя выполнить при любом выборе положительных коэффициентов  $u_i$  (рис. 4.9, б), где рассматриваемая точка  $X$  лежит вне выпуклой оболочки, натянутой на градиенты). Учет ограничений типа равенств очевиден, если добавляется последняя из указанных в (4.18) сумма.

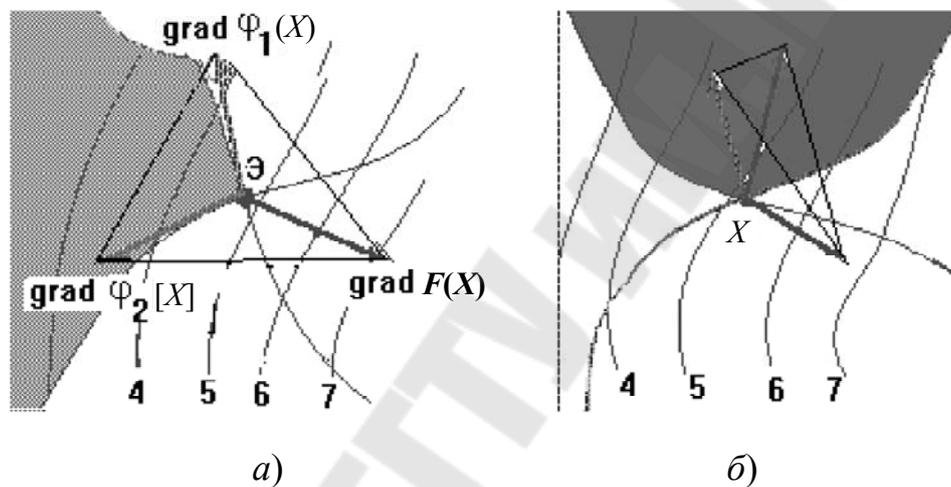


Рис. 4.9. К пояснению условий Куна–Таккера

**Методы поиска условных экстремумов.** Широко известен метод множителей Лагранжа, ориентированный на поиск экстремума при наличии ограничений типа равенств  $\psi(X) = 0$ , т. е. на решение задачи

$$\text{extr} F(X), \quad X \in R, \quad (4.19)$$

где  $R = \{X | \psi(X) = 0\}$ .

Суть метода заключается в преобразовании задачи условной оптимизации (4.19) в задачу безусловной оптимизации с помощью образования новой целевой функции

$$\Phi(X, V) = F(X) + \sum_{i=1}^L \lambda_i \psi_i(X),$$

где  $\Lambda = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_h)$  – вектор множителей Лагранжа;  $L$  – число ограничений.

Необходимые условия экстремума функции  $\Phi(X)$ :

$$\partial\Phi(X, \Lambda)/\partial X = \partial F(X)/\partial X + \sum_{i=1}^L \lambda_i \partial\psi_i(X)/\partial X = 0; \quad (4.20)$$

$$\partial\Phi(X, \Lambda)/\partial\Lambda = \psi(X) = 0.$$

Система (4.20) содержит  $n + L$  алгебраических уравнений, где  $n$  – размерность пространства управляемых параметров, ее решение дает искомые координаты экстремальной точки и значения множителей Лагранжа. Однако при численном решении (4.20), что имеет место при использовании алгоритмических моделей, возникают те же трудности, что и в методе Ньютона. Поэтому в САПР основными методами решения ЗМП являются методы штрафных функций и проекции градиента.

Основная идея *методов штрафных функций* – преобразование задачи условной оптимизации в задачу безусловной оптимизации путем формирования новой целевой функции  $\Phi(X)$  введением в исходную целевую функцию  $F(X)$  специальным образом выбранной функции штрафа  $S(X)$ :

$$\Phi(X) = F(X) + rS(X),$$

где  $r$  – множитель, значения которого можно изменять в процессе оптимизации.

Среди методов штрафных функций различают методы внутренней и внешней точки. Согласно методам внутренней точки (иначе называемыми методами *барьерных функций*) исходную для поиска точку можно выбирать только внутри допустимой области, а для методов внешней точки как внутри, так и вне допустимой области (важно лишь, чтобы в ней функции целевая и ограничений были бы определены). Ситуация появления барьера у целевой функции  $\Phi(x)$  и соотношение между условным в точке  $x_2$  и безусловным в точке  $x_1$  минимумами  $F(x)$  в простейшем одномерном случае иллюстрируется рис. 4.10.

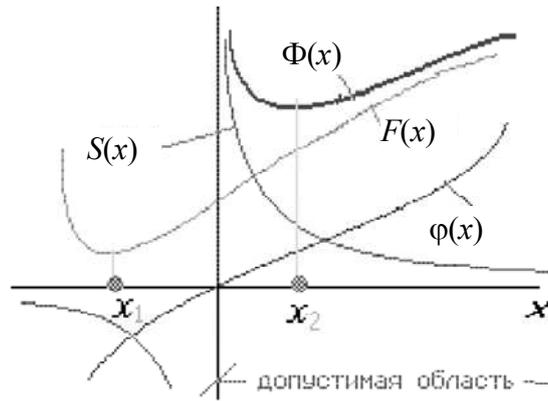


Рис. 4.10. Пояснение метода штрафных функций

Примеры штрафных функций:

1) для метода внутренней точки при ограничениях  $\varphi_i(X) > 0$

$$S(X) = \sum_{i=1}^m (1/\Phi_i(X)),$$

где  $m$  – число ограничений типа неравенств;

2) для метода внешней точки при таких же ограничениях

$$S(X) = \sum_{i=1}^m (\min\{0, \Phi_i(X)\})^2.$$

Здесь штраф сводится к включению в  $\Phi(X)$  суммы квадратов активных (т. е. нарушенных) ограничений;

3) в случае ограничений типа равенств  $\psi_i(X) = 0$ :

$$S(X) = \sum_{i=1}^L (\psi_i(X))^2.$$

Чем больше коэффициент  $r$ , тем точнее решение задачи, однако при больших  $r$  может ухудшаться ее обусловленность. Поэтому в начале поиска обычно выбирают умеренные значения  $r$ , увеличивая их в окрестностях экстремума.

Основной вариант *метода проекции градиента* ориентирован на задачи математического программирования с ограничениями типа равенств.

Поиск при выполнении ограничений осуществляется в подпространстве  $(n - m)$  измерений, где  $n$  – число управляемых параметров,  $m$  – число ограничений, при этом движение осуществляется в на-

правлении проекции градиента целевой функции  $F(X)$  на гиперплоскость, касательную к гиперповерхности ограничений (точнее к гиперповерхности пересечения гиперповерхностей ограничений).

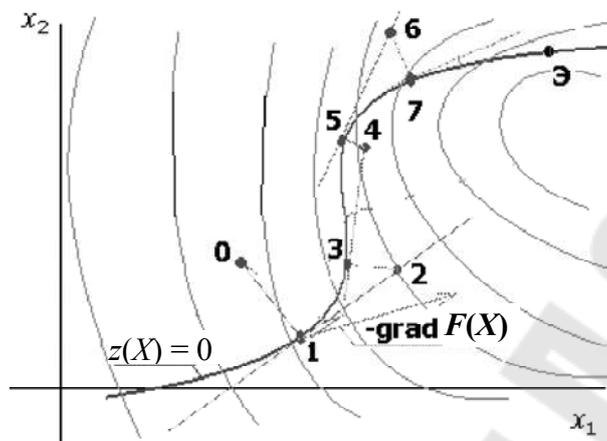


Рис. 4.11. Траектория поиска в соответствии с методом проекции градиента: Э – условный экстремум; 0, 1, 2, 3 – точки на траектории поиска

Поиск минимума начинают со спуска из исходной точки на гиперповерхность ограничений. Далее выполняют шаг в указанном выше направлении (шаг вдоль гиперповерхности ограничений). Поскольку этот шаг может привести к заметному нарушению ограничений, вновь повторяют спуск на гиперповерхность ограничений и т. д. Другими словами, поиск заключается в выполнении пар шагов, каждая пара включает спуск на гиперповерхность ограничений и движение вдоль гиперповерхности ограничений.

Идею метода легко пояснить для случая поиска в двумерном пространстве при одном ограничении  $\psi(X) = 0$ . На рис. 4.11 это ограничение представлено жирной линией, а целевая функция – совокупностью более тонких линий равного уровня. Спуск обычно осуществляют по нормали к гиперповерхности ограничений (в данном случае к линии ограничения). Условие окончания поиска основано на сопоставлении значений целевой функции в двух последовательных точках, получаемых после спуска на гиперповерхность ограничений.

Рассмотрим вопрос, касающийся получения аналитических выражений для направлений спуска и движения вдоль гиперповерхности ограничений.

*Спуск.* Необходимо из текущей точки поиска  $B$  попасть в точку  $A$ , являющуюся ближайшей к  $B$  точкой на гиперповерхности ограничений, т. е. решить задачу

$$\min |B - A|$$

при условии  $y(X) = 0$ , которое после линеаризации в окрестностях точки  $B$  имеет вид

$$\psi(B) + (\text{grad}\psi(B))^T (A - B) = 0.$$

Используя метод множителей Лагранжа, обозначая  $A - B = U$  и учитывая, что минимизация расстояния равнозначна минимизации скалярного произведения  $U$  на  $U$ , запишем:

$$\begin{aligned} \Phi(A) &= U^T U + \lambda(\psi(B) + (\text{grad}\psi(B))^T U); \\ \partial\Phi / \partial A &= 2U + \lambda(\text{grad}\psi(B)) = 0; \end{aligned} \quad (4.21)$$

$$\partial\Phi / \partial X = \psi(B) + (\text{grad}\psi(B))^T U = 0. \quad (4.22)$$

Тогда из (4.21) получаем выражение

$$U = -0,5\lambda(\text{grad}\psi(B)), \text{ подставляя его в (4.22), имеем}$$

$$\psi(B) - 0,5\lambda(\text{grad}\psi(B))^T \text{grad}\psi(B) = 0,$$

откуда

$$\lambda = (0,5(\text{grad}\psi(B))^T \text{grad}\psi(B))^{-1} \psi(B),$$

и окончательно, подставляя  $X$  в (4.21), находим

$$\lambda = -\text{grad}\psi(B)(\text{grad}\psi(B))^T \text{grad}\psi(B))^{-1} \psi(B).$$

*Движение вдоль гиперповерхности ограничений.* Шаг в гиперплоскости  $D$ , касательной к гиперповерхности ограничений, следует сделать в направлении вектора  $S$ , на котором целевая функция уменьшается в наибольшей мере при заданном шаге  $h$ . Уменьшение целевой функции при переходе из точки  $A$  в новую точку  $C$  подсчитывают, используя формулу линеаризации  $F(X)$  в окрестностях точки  $A$ :

$$F(C) - F(A) = h(\text{grad}F(A))^T S,$$

где  $\text{grad}F(A)^T S$  – приращение  $F(X)$ , которое нужно минимизировать, варьируя направления  $S$ :

$$\min F(C) = \min((\text{grad}F(A))^T S), \quad (4.23)$$

где вариация  $S$  осуществляется в пределах гиперплоскости  $D$ ;  $\text{grad}\psi(A)$  и  $S$  – ортогональные векторы. Следовательно, минимизацию (4.23) необходимо выполнять при ограничениях

$$(\text{grad}F(A))^T S = 0, \quad S^T S = 1.$$

Последнее ограничение говорит о том, что при поиске направления движения, вектор  $S$  должен лишь указывать это направление, т. е. его длина несущественна (пусть  $S$  – единичный вектор). Для решения (4.23) используем метод множителей Лагранжа

$$\Phi(S, \lambda, q) = (\text{grad}F(A))^T S + X(\text{grad}\psi(A))^T S + q(S^T S - 1),$$

где  $\lambda$  и  $q$  – множители Лагранжа;

$$\partial\Phi / \partial S = \text{grad}F(A) + X\text{grad}\psi(A) + qS = 0; \quad (4.24)$$

$$\partial\Phi / \partial X = (\text{grad}\psi(A))^T S = 0; \quad (4.25)$$

$$\partial\Phi / \partial q = S^T S - 1 = 0. \quad (4.26)$$

Из (4.24) следует, что

$$S = -(\text{grad}F(A) + X\text{grad}\psi(A))/q;$$

подставляя  $S$  в (4.25), получаем

$$(\text{grad}\psi(A))^T \text{grad}F(A) + X(\text{grad}\psi(A))^T \text{grad}\psi(A) = 0,$$

откуда

$$\begin{aligned} \lambda &= -[(\text{grad}\psi(A))^T \text{grad}\psi(A)]^{-1} (\text{grad}\psi(A))^T \text{grad}F(A), \quad S = \\ &= -\left\{ \text{grad}F(A) - \text{grad}\psi(A)[(\text{grad}\psi(A))^T \text{grad}\psi(A)]^{-1} \right. \\ &\quad \left. (\text{grad}\psi(A))^T \text{grad}F(A) / q \right\} = \\ &= -\left\{ E - \text{grad}\psi(A)[(\text{grad}\psi(A))^T \text{grad}\psi(A)]^{-1} (\text{grad}\psi(A))^T \right\} \text{grad}F(A) / q. \quad (4.27) \end{aligned}$$

Таким образом, матрица

$$P = E - \text{grad}\psi(A)[(\text{grad}\psi(A))^T \text{grad}\psi(A)]^{-1} (\text{grad}\psi(A))^T$$

представляет собой проектирующую матрицу, а вектор  $S$ , рассчитанный по (4.27), – проекцию градиента  $\text{grad}F(A)$  на гиперповерхность ограничений.

Частным случаем применения метода проекции градиента являются задачи оптимизации с максиминным критерием. Действительно, для поиска экстремума функции минимума

$$\max_x \min_j Z(X),$$

где  $Z_j$  – нормированная величина  $j$ -го выходного параметра  $y$ , удобно применять метод проекции градиента. В качестве ограничений задачи в исходной постановке фигурируют только прямые ограничения

$$x_{\max i} > x_i > x_{\min i}.$$

Здесь  $x_{\max i}$  и  $x_{\min i}$  – граничные значения допустимого диапазона варьирования параметра  $x_i$ . В процессе поиска, если минимальной является функция  $Z_q(X)$  и траектория поиска пересекает гребень

$$Z_q(X) - Z_k(X) = 0, \quad (4.28)$$

то поиск продолжается в направлении проекции градиента функции  $Z_q(X)$  на гиперповерхность гребня (4.28).

## **Тема 15. Постановка задач структурного синтеза**

**Процедуры синтеза проектных решений.** Принятие проектных решений охватывает широкий круг задач и процедур – от выбора вариантов в конечных и обозримых множествах до задач творческого характера, не имеющих формальных способов решения.

Соответственно в САПР применяют как средства формального синтеза проектных решений, выполняемого в автоматическом режиме, так и вспомогательные средства, способствующие выполнению синтеза проектных решений в интерактивном режиме. К вспомогательным средствам относятся базы типовых проектных решений, системы обучения проектированию, программно-методические комплексы верификации проектных решений, унифицированные языки описания ТЗ и результатов проектирования.

Задачи синтеза структур проектируемых объектов относятся к наиболее трудно формализуемым. Существует ряд общих подходов к постановке этих задач, однако практическая реализация большинства из них неочевидна.

Именно по этой причине структурный синтез, как правило, выполняют в интерактивном режиме при решающей роли инженера-

разработчика, а ЭВМ играет вспомогательную роль: предоставление необходимых справочных данных, фиксация и оценка промежуточных и окончательных результатов.

Однако в ряде приложений имеются и примеры успешной автоматизации структурного синтеза в ряде приложений; среди них заслуживают упоминания в первую очередь синтеза технологических процессов и управляющих программ для механообработки в машиностроении и некоторые другие.

Структурный синтез заключается в преобразовании описаний проектируемого объекта: исходное описание содержит информацию о требованиях к свойствам объекта, об условиях его функционирования, ограничениях на элементный состав и т. п., а результирующее описание должно содержать сведения о *структуре*, т. е. о составе элементов и способах их соединения и взаимодействия.

Постановки и методы решения задач структурного синтеза в связи с трудностями формализации не достигли степени обобщения и детализации, свойственной математическому обеспечению процедур анализа. Достигнутая степень обобщения выражается в установлении типичной последовательности действий и используемых видов описаний при их преобразованиях в САПР. Исходное описание, как правило, представляет собой ТЗ на проектирование, по нему составляют описание на некотором формальном языке, являющемся входным языком используемых подсистем САПР. Затем выполняют преобразования описаний и получаемое итоговое для данного этапа описание документируют – представляют в виде твердой копии или файла в соответствующем формате для передачи на следующий этап.

Важное значение для развития подсистем синтеза в САПР имеют разработка и унификация языков представления описаний (спецификаций). Каждый язык, поддерживая выбранную методику принятия решений, формирует у пользователей САПР – разработчиков технических объектов – определенный стиль мышления; особенности языков непосредственно влияют на особенности правил преобразования спецификаций. Примерами унифицированных языков описания проектных решений являются язык VHDL для радиоэлектроники, он сочетает в себе средства для функциональных, поведенческих и структурных описаний, или язык Express – универсальный язык спецификаций для представления и обмена информацией в компьютерных средах.

**Задача принятия решений.** Имеется ряд подходов для обобщенного описания задач принятия проектных решений в процессе структурного синтеза.

Задачу принятия решений (ЗПР) формулируют следующим образом:

$$\text{ЗПР} = \langle A, K, \text{Мод}, \Pi \rangle,$$

где  $A$  – множество альтернатив проектного решения;  $K = (K_1, K_2, \dots, K_m)$  – множество критериев (выходных параметров), по которым оценивается соответствие альтернативы поставленным целям; Мод:  $A \rightarrow K$  – модель, позволяющая для каждой альтернативы рассчитать вектор критериев;  $\Pi$  – решающее правило для выбора наиболее подходящей альтернативы в многокритериальной ситуации.

В свою очередь, каждой альтернативе конкретного приложения можно поставить в соответствие значения упорядоченного множества (набора) атрибутов  $X = \langle x_1, x_2, \dots, x_n \rangle$ , характеризующих свойства альтернативы. При этом  $x_i$  может быть величиной типа real, integer, boolean, string (в последнем случае величину называют *предметной* или *лингвистической*). Множество  $X$  называют записью (в теории баз данных), фреймом (в искусственном интеллекте) или *хромосомой* (в генетических алгоритмах). Модель Мод называют структурно-критериальной, если среди  $x_i$  имеются параметры, характеризующие структуру моделируемого объекта.

Основными проблемами в ЗПР являются:

- компактное представление множества вариантов (альтернатив);
- построение модели синтезируемого устройства, в том числе выбор степени абстрагирования для оценки значений критериев;
- формулировка предпочтений в многокритериальных ситуациях (т. е. преобразование векторного критерия  $K$  в скалярную целевую функцию);
- установление порядка (предпочтений) между альтернативами в отсутствие количественной оценки целевой функции (что обычно является следствием неколичественного характера всех или части критериев);
- выбор метода поиска оптимального варианта (сокращение перебора вариантов).

Присущая проектным задачам неопределенность и нечеткость исходных данных, а иногда и моделей, диктуют использование спе-

специальных методов количественной формулировки исходных нечисленных данных и отношений. Эти специальные методы либо относятся к области построения измерительных шкал, либо являются предметом теории нечетких множеств.

Измерительные шкалы могут быть: абсолютными; номинальными (классификационными), значения шкалы представляют классы эквивалентности, примером может служить шкала цветов; такие шкалы соответствуют величинам нечисленного характера; порядковыми, если между объектами  $A$  и  $B$  установлено одно из следующих отношений: простого порядка, гласящее, что если  $A$  лучше  $B$ , то  $B$  хуже  $A$ , и соблюдается транзитивность; или слабого порядка, т. е. либо  $A$  не хуже  $B$ , либо  $A$  не лучше  $B$ ; или частичного порядка. Для формирования целевой функции  $F(X)$  производится оцифровка порядковой шкалы, т. е. при минимизации, если  $A$  предпочтительнее  $B$ , то  $F(X_a) < F(X_b)$ , где  $X_a$  и  $X_b$  – множества атрибутов объектов  $A$  и  $B$  соответственно; интервальными, отражающими количественные отношения интервалов: шкала единственна с точностью до линейных преобразований, т. е.  $y = ax + b$ ,  $a > 0$ ,  $-\infty < b < \infty$ , или  $y = ax$  при  $a \neq 0$ , или  $y = x + b$ .

В большинстве случаев структурного синтеза математическая модель в виде алгоритма, позволяющего по заданному множеству  $X$  и заданной структуре объекта рассчитать вектор критериев  $K$ , оказывается известной. Например, такие модели получаются автоматически в программах анализа типа Spice, Adams или ПА-9 для объектов, исследуемых на макроуровне. Однако в ряде других случаев такие модели неизвестны в силу недостаточной изученности процессов и их взаимосвязей в исследуемой среде, но известна совокупность результатов наблюдений или экспериментальных исследований. Тогда для получения моделей используют специальные *методы идентификации и аппроксимации* (модели, полученные подобным путем иногда называют феноменологическими).

Среди методов формирования моделей по экспериментальным данным наиболее известны *методы планирования экспериментов*. Не менее популярным становится подход, основанный на использовании *искусственных нейронных сетей*.

Если же математическая модель  $X \rightarrow K$  остается неизвестной, то стараются использовать подход на базе *систем искусственного интеллекта (экспертных систем)*.

Возможности практического решения задач *дискретного математического программирования* (ДМП) изучаются в теории сложности задач выбора, где показано, что задачи даже умеренного размера, относящиеся к классу *NP*-полных задач, в общем случае удается решать только приближенно.

Поэтому большинство практических задач структурного синтеза решают с помощью приближенных (эвристических) методов. Это методы, использующие специфические особенности того или иного класса задач и не гарантирующие получения оптимального решения. Часто они приводят к результатам, близким к оптимальным, при приемлемых затратах вычислительных ресурсов.

Если все управляемые параметры альтернатив, обозначаемые в виде множества  $X$ , являются количественными оценками, то используют *приближенные методы* оптимизации. Если в  $X$  входят также параметры неколичественного характера и пространство  $X$  неметризуемо, то перспективными являются *эволюционные методы* вычислений, среди которых наиболее развиты *генетические методы*. Наконец, в отсутствие обоснованных моделей их создают, основываясь на экспертных знаниях в виде системы искусственного интеллекта.

**Представление множества альтернатив.** Решению проблем упорядочения и описания множества альтернатив и связей между ними в конкретных приложениях посвящена специальная область знания, которую по аналогии с наукой описания множеств животных и растений в биологии можно назвать *систематикой*.

Простейший способ задания множества  $A$  – явное перечисление всех альтернатив. Семантика и форма описания альтернатив существенно зависят от приложения. Для представления таких описаний в памяти ЭВМ и доступа к ним используют *информационно-поисковые системы* (ИПС). Каждой альтернативе в ИПС соответствует поисковый образ, состоящий из значений атрибутов  $x_i$  и ключевых слов вербальных характеристик.

Явное перечисление альтернатив при представлении множества альтернатив возможно лишь при малой мощности  $A$ . Поэтому в большинстве случаев используют неявное описание  $A$  в виде способа (алгоритма или набора правил  $P$ ) синтеза проектных решений из ограниченного набора элементов  $\mathcal{E}$ . Поэтому здесь  $A = \langle P, \mathcal{E} \rangle$ , а типичный процесс синтеза проектных решений состоит из следующих этапов: формирование альтернативы  $A_i$  (это может быть выбор из базы данных ИПС по сформированному поисковому предписанию или

генерация из Э в соответствии с правилами  $P$ ); оценка альтернативы по результатам моделирования с помощью модели Мод; принятие решения (выполняется ЛПР – лицом, принимающим решение, или автоматически) относительно перехода к следующей альтернативе или прекращения поиска.

Для описания множеств  $P$  и Э используют следующие подходы:

1. Морфологические таблицы и альтернативные И-ИЛИ-деревья.
2. Представление знаний в интеллектуальных системах – фреймы, семантические сети, продукции.
3. Генетические методы.
4. Базы физических эффектов и эвристических приемов, применяемые при решении задач изобретательского характера.

**Морфологические таблицы.** Морфологическая таблица ( $M$ ) представляет собой обобщенную структуру в виде множества функций, выполняемых компонентами синтезируемых объектов рассматриваемого класса, и подмножеств способов их реализации. Каждой функции можно поставить в соответствие одну строку таблицы, каждому способу ее реализации – одну клетку в этой строке. Следовательно, в морфологических таблицах элемент  $M_{ij}$  – означает  $j$ -й вариант реализации  $i$ -й функции в классе технических объектов, описываемом матрицей  $M$ .

Другими словами, множество альтернатив можно представить в виде отношения  $M$ , называемого морфологической таблицей

$$M = \langle X, R \rangle,$$

где  $X$  – множество свойств (характеристик или функций), присущих объектам рассматриваемого типа;  $n$  – число этих свойств;  $r = \langle r_1, r_2, \dots, r_n \rangle$ ,  $r_i$  – множество значений (способов реализации)  $i$ -го свойства, мощность этого множества далее обозначена  $N_i$ . При этом собственно множество альтернатив  $A$  представлено композицией множеств  $R_i$ , т. е. каждая альтернатива включает по одному элементу (значению) из каждой строки морфологической таблицы. Очевидно, что общее число альтернатив  $k$ , представляемых морфологической таблицей, равно

$$k = \prod_{i=1}^n N_i.$$

Морфологические таблицы обычно считают средством неавтоматизированного синтеза, помогающим человеку просматривать компактно представленные альтернативы, преодолевать психологическую инерцию. Последнее связано с тем, что внимание ЛПР обращается на варианты, которые без морфологической таблицы оставались бы вне его поля зрения.

Собственно таблица  $M$  не содержит сведений о способе синтеза. Однако на базе  $M$  возможно построение методов синтеза с элементами алгоритмизации. В таких методах вводится метризация морфологического пространства. Морфологическое пространство составляют возможные законченные структуры, принимается, что расстояние между структурами  $S_1$  и  $S_2$  есть число несовпадающих элементов (каждая клетка  $M$  есть один элемент). Поэтому можно говорить об окрестностях решений. Далее исходят из предположения о компактности «хороших» решений, которое позволяет вместо полного перебора ограничиваться перебором в малой окрестности текущей точки поиска. Таким образом, гипотеза о «компактности» и метризация пространства решений фактически приводят к построению математической модели, к которой можно применить методы дискретной оптимизации, например локальные методы.

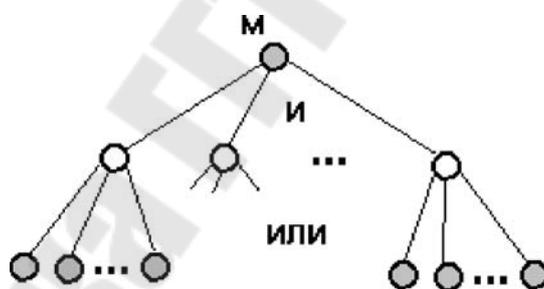


Рис. 4.12. Дерево, соответствующее морфологической таблице

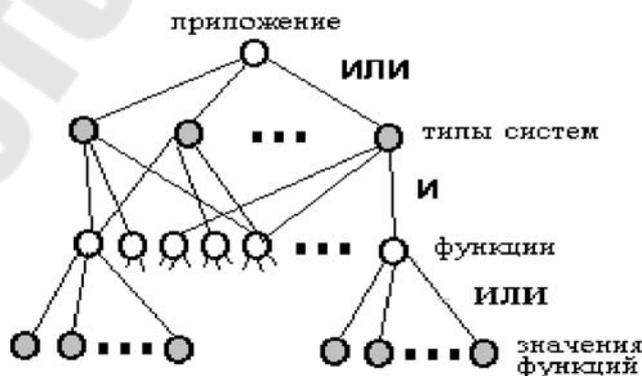


Рис. 4.13. И-ИЛИ-граф

К недостаткам  $M$  относятся неучет запрещенных сочетаний элементов в законченных структурах и отражение состава элементов в структурах без конкретизации их связей. Кроме того, морфологические таблицы строят в предположении, что множества  $R_i$  взаимно независимы, т. е. состав способов реализации  $i$ -й функции не меняется при изменении значений других функций. Очевидно, что предположение о взаимной независимости множеств  $R_i$  оправдано лишь в сравнительно простых структурах. Последний недостаток устраняется путем обобщения метода морфологических таблиц И-ИЛИ-графов при использовании метода альтернативных (И-ИЛИ) графов.

*Альтернативные графы.* Любую морфологическую таблицу можно представить в виде дерева (рис. 4.12). На рис. 4.12 функции представлены вершинами И (темные кружки), значения функций – вершинами ИЛИ (светлые кружки). Очевидно, что таблица представляет множество однотипных объектов, поскольку все они характеризуются одним и тем же множеством функций.

Для разнотипных объектов применяют многоярусные альтернативные графы. Например, на рис. 4.13 показан двухъярусный граф, в котором для разных типов объектов предусмотрены разные подмножества функций.

Если допустить некоторую избыточность при изображении И-ИЛИ-графа, то его можно превратить в И-ИЛИ-дерево, что ведет к определенным удобствам.

Очевидно, что И-ИЛИ-дерево можно представить как совокупность морфологических таблиц. Каждая И вершина дерева соответствует частной морфологической таблице, т. е. множеству функций так, что  $i$ -я выходящая ветвь отображает  $i$ -ю функцию. Каждая ИЛИ вершина, инцидентная  $i$ -й ветви, соответствует множеству вариантов реализации  $i$ -й функции, при этом  $j$ -я исходящая из ИЛИ вершины ветвь отображает  $j$ -й вариант реализации.

Алгоритмизация синтеза на базе И-ИЛИ-деревьев требует введения правил выбора альтернатив в каждой вершине ИЛИ. Эти правила чаще всего имеют эвристический характер, связаны с требованиями ТЗ, могут отражать запреты на сочетания определенных компонентов структур. Трудности эффективного решения задачи существенно возрастают при наличии ограничений, типичными среди которых являются ограничения на совместимость способов реализации разных функций, т. е. ограничения вида

$$C_{ij} \text{ and } C_{pq} = \text{false}, \quad (4.29)$$

где  $C_{ij} = \text{true}$ , если в оцениваемый вариант вошел элемент  $\mathcal{E}_{ij}$ , иначе  $C_{ij} = \text{false}$ . Условие (4.29) означает, что в допустимую структуру не могут входить одновременно элементы  $\mathcal{E}_{ij}$  и  $\mathcal{E}_{pq}$ . Совокупность ограничений типа (4.29) можно представить как систему логических уравнений с неизвестными  $C_j$ . Тогда задачу синтеза можно решать эволюционными методами, если предварительно или одновременно с ней решать систему логических уравнений (задачу о выполнимости).

**Исчисления.** Очевидно, что в большинстве случаев структурного синтеза вместо нереализуемого явного представления всего множества проектных решений задают множество элементов и совокупность правил объединения этих элементов в допустимые структуры (проектные решения).

Эти множества элементов и правил часто представляют в виде *формальной системы (исчисления)*, т. е. задача синтеза имеет вид:

$$3C = \langle \mathcal{E}; \text{HT}; \text{AK}; \text{П} \rangle,$$

где  $\mathcal{E}$  – алфавит исчисления (алфавит представлен базовыми элементами, из которых синтезируется структура); HT – множество букв, не совпадающих с буквами алфавита  $\mathcal{E}$  и служащих для обозначения переменных; АК – множество аксиом исчисления, под которыми понимаются задаваемые исходные формулы (слова) в алфавите  $\mathcal{E}$  (например, соответствия функций и элементов); П – множество правил вывода новых формул в алфавите  $\mathcal{E}$  из аксиом и ранее выведенных корректных формул. Каждую формулу можно интерпретировать как некоторую структуру, поэтому синтез – это процесс вывода формулы, удовлетворяющей исходным требованиям и ограничениям.

Другие примеры компактного задания множества альтернатив  $A$  через множества  $\mathcal{E}$  и П связан с использованием систем искусственного интеллекта, в которых  $\mathcal{E}$  есть база данных, П – база знаний, или эволюционных методов, в которых  $\mathcal{E}$  – также база данных, П – множество эвристик, последовательность применения которых определяется эволюционными и генетическими принципами.

## **Глава 5. ДРУГИЕ ВИДЫ ОБЕСПЕЧЕНИЙ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ**

### **Тема 16. Программное обеспечение систем автоматизированного проектирования**

**Состав программного обеспечения САПР.** Программное обеспечение САПР представляет собой совокупность программ на машинных носителях с необходимой программной документацией, предназначенных для выполнения автоматизированного проектирования. ПО САПР подразделяют на базовое, общесистемное и специализированное.

Базовое ПО не является объектом разработки при создании ПО САПР (например, операционная система – ОС).

Основные функции общесистемного ПО САПР: ввод, вывод и обработка инструкций пользователей; управление процессом вычислений; хранение, анализ, поиск, модификация данных; контроль и диагностика в процессе решения задач проектирования.

В состав общесистемного ПО входят: мониторинговая диалоговая система; СУБД; информационно поисковые системы; геометрические и графические процессоры; средства формирования графической и текстовой информации; средства для выполнения общетехнических расчетов.

Специализированное ПО функционирует в операционной среде, которая состоит из общесистемного и базового ПО. Основной функцией специализированного ПО является получение проектных решений. Специализированное ПО создается с учетом организации и возможностей общесистемного ПО САПР.

**Функциональное назначение ПО САПР.** По функциональному назначению ПО САПР можно разделить на ряд программных комплексов, предназначенных для выполнения заданных функций. Можно выделить проектирующие, обслуживающие и инструментальные программные комплексы (ПК). Проектирующие ПК предназначены для получения законченного проектного решения и подразделяются

на *проблемно-ориентированные* и *объектно-ориентированные*. Проблемно-ориентированные комплексы выполняют унифицированные проектные процедуры, не зависящие от объекта проектирования. Объектно-ориентированные ПК используются лишь для проектирования объектов определенного класса. Проектирующие программные комплексы входят в состав специализированного ПО. Обслуживающие ПК входят в состав общесистемного ПО и предназначены для поддержания работоспособности проектирующих ПК. Инструментальные ПК представляют собой технологические средства, предназначенные для разработки, развития и модернизации ПО САПР. Инструментальные ПК можно разделить на инструментальные средства, являющиеся частью САПР и используемые в процессе ее работы, и инструментальные средства, используемые только во время разработки САПР.

Инструментальные средства, используемые только во время работы САПР: СУБД; языковые процессоры для обеспечения взаимодействий с пользователями; средства машинной графики; пакет математических подпрограмм.

Инструментальные средства, применяемые при разработке САПР, включают в себя: средства построения спецификаций, тестирования, документирования; инструментальные средства проектирования программ, препроцессоры, отладчики.

**Принципы проектирования ПО САПР.** Проектирование ПО САПР осуществляется на основе принципов системного единства, развития, совместимости и стандартизации.

*Принцип системного единства.* При создании, функционировании и развитии ПО САПР связи между компонентами должны обеспечивать ее целостность.

*Принцип развития.* ПО САПР должно создаваться и функционировать с учетом накопления, совершенствования и обновления ее компонент.

*Принцип совместимости.* Языки, символы, коды, информация и связи между компонентами должны обеспечивать их совместное функционирование и сохранять открытую структуру системы в целом.

*Принцип стандартизации.* При проектировании ПО САПР необходимо унифицировать, типизировать и стандартизировать ПО, инвариантное к проектируемым объектам.

### **Пользователи САПР:**

– разработчики САПР (специалисты способные разрабатывать базовые методы, средства и оснащение САПР, осуществлять генерацию и настройку САПР);

– прикладные программисты;

– проектировщики;

**Общие требования к ПО САПР.** В соответствии с общими принципами создания САПР ПО должно удовлетворять следующим требованиям:

– адаптируемость – приспособленность ПО САПР к функционированию в различных условиях проектирования. Это связано с изменением самих объектов проектирования;

– гибкость, т. е. возможность легко вносить изменения, дополнения, исправление в ПО при сохранении всей системы в целом;

– компактность, т. е. потребление минимальных ресурсов ЭВМ (памяти, времени ЦП), что позволяет улучшить эксплуатационные характеристики САПР;

– мобильность, способность функционирования ПО САПР на различных технических средствах;

– надежность – обеспечение получения достоверных результатов проектирования;

– реактивность – обеспечение быстрого решения задачи, при ориентации на пользователя, не являющегося специалистом в области вычислительной техники и программирования;

– эволюционируемость – пополнение САПР новыми программами, реализующими возможность всей системы в целом.

Все вышеперечисленные требования должны обеспечивать высокое качество функционирования САПР: получение проектных решений в форме, принятой в организации; полноту диагностических сообщений; удобство освоения и сопровождения ПО САПР и т. д.

**Этапы проектирования ПО САПР.** Согласно ГОСТу устанавливаются следующие стадии разработки ПО: техническое задание, эскизный, технический и рабочий проекты, внедрение.

*Техническое задание.* На стадии ТЗ выполняются следующие виды работ: постановка задачи; сбор исходных материалов; выбор и обоснование критериев эффективности и качества разрабатываемой программы; обоснование необходимости проведения научно-исследовательских работ; предварительный выбор методов решения задач;

определение требований к техническим средствам; согласование и утверждение ТЗ.

*Эскизный проект.* На стадии эскизного проектирования выполняются следующие виды работ: внешнее проектирование программного изделия; уточнение методов решения задачи; предварительное проектирование внутренних структур данных; разработка общего алгоритма решения задачи.

*Технический проект.* Этапы технического проектирования: проектирование архитектуры программного изделия и структур данных; проектирование модулей и модульной структуры изделий; разработка пояснительной записки.

На этапе проектирования архитектуры программного изделие разбивают на составные части: определяют функции каждой компоненты, способы взаимодействия между ними, разрабатывают схемы распределения оперативной и внешней памяти вычислительной системы.

На этапе проектирования структур данных определяют способы представления, хранения и преобразования входных, выходных и внутренних данных.

На этапе проектирования модульной структуры программного изделие разбивается на модули (процедуры), определяются их функции и способы взаимодействия.

На этапе проектирования модулей составляется описание всех модулей программного изделия, которое включает в себя: имя модуля функции, выполняемые модулем, описание входных и выходных параметров, их форматов; взаимосвязи между входными и выходными параметрами и т. д., описание алгоритма модуля и т. д.

*Рабочий проект.* На стадии рабочего проекта выполняются следующие виды работ: кодирование, тестирование и отладка программ; разработка программных документов в соответствии с требованиями ЕСПД; проведение различных видов испытаний; корректировка программ и документации по результатам испытаний.

*Стадия внедрения.* Осуществляется подготовка и передача программ и программной документации для сопровождения и изготовления.

**Оценка показателей качества ПО.** В зависимости от назначения, условий создания и эксплуатации ПО могут быть выбраны свойства для формирования соответствующих показателей качества: безотказность – способность выполнять заданные функции без прояв-

ленных отказов; корректность, т. е. соответствие заданным функциональным требованиям; мобильность – возможность приспособления к работе в другой программно-технической среде; модифицируемость; совместимость; сопровождаемость, т. е. приспособленность к устранению ошибок в ходе создания и сопровождения; унифицированность – приспособленность для применения в качестве составной части для другой системы; устойчивость – способность сохранять работоспособность системы в условиях проявления отказов и сбоев технических средств; целостность – защищенность программ и данных от несанкционированного доступа.

**Принцип модульного программирования ПО.** Заключается в разделении программ на функционально самостоятельные части (модули), обеспечивающие заменяемость, модификацию, удаление и дополнение составных частей (модулей).

Методы проектирования программ, основанные на модульном, делятся на три группы: *методы нисходящего проектирования, методы расширения ядра, методы восходящего проектирования.*

На практике обычно применяют различные сочетания этих методов.

В зависимости от задач, решаемых разработчиками и от использования ими методов проектирования, модульная программа может иметь одну из следующих основных структур.

#### ***МОДУЛЬНО-ПОСЛЕДОВАТЕЛЬНАЯ СТРУКТУРА***

Такая структура программы включает в себя несколько последовательно передающих друг другу управление программных модулей. Структура проста и наглядна, но может быть реализована лишь для относительно простых задач.

#### ***МОНОЛИТНО-МОДУЛЬНАЯ СТРУКТУРА***

Она включает в себя большой программный модуль, реализующий основную часть возложенных на программу функций. Из этой части имеется незначительное число обращений к другим программным модулям небольшого размера. Подобная программа сложна как для написания, так и для сопровождения. При проектировании ПО подобных структур следует избегать.

#### ***МОДУЛЬНО-ИЕРАРХИЧЕСКАЯ СТРУКТУРА***

Структура программы включает в себя программные модули, располагаемые на нескольких уровнях иерархии. Модули верхних уровней управляют работой модулей нижних уровней. Вышестоящий модуль передает управление модулю более низкого уровня, а когда

тот отработает, он возвращает управление вызвавшему его модулю. Подобная структура достаточно проста и позволяет решать очень сложные задачи.

### *МОДУЛЬНО-ХАОТИЧЕСКАЯ СТРУКТУРА*

Эта структура программы включает в себя программные модули, которые связаны между собой таким образом, что они не образуют в явном виде не одну из перечисляемых выше структур. Такие программы сложны для проверки и сопровождения.

Порядок перечисленных видов связей соответствует увеличению силы связности от наименьшей (через простой параметр—данные) до наибольшей (через содержимое):

1. Связь через *простой параметр—данные* возникает тогда, когда все необходимые данные модуль принимает и возвращает в форме параметров вызова, а эти данные являются простыми (неструктурированными) переменными.

2. Связь через *структурный параметр—данные* возникает когда в списке параметров вызова имеются структурные данные. В этом случае модуль, который их получает, зависит не только от самих данных, но и от их структуры. Если модулю нужны только отдельные элементы данных, ему следует передавать всю структуру.

3. Связь через *управляющий параметр* возникает тогда, когда в списке параметров вызова присутствует управляющий признак, (флаг), который анализируется получающим его модулем.

Сила связи таких модулей еще более возрастает, так как модуль, передающий признак, должен быть осведомлен о внутренней структуре модуля, получающего этот признак.

4. Связь через *общий блок данных* возникает между модулями тогда, когда они ссылаются на один и тот же блок данных с целью получения и передачи параметров.

5. Связь через *внешние переменные* возникает, когда в модулях объявляется внешняя переменная, значение которой устанавливается в одном модуле, а затем используется в другом.

6. Связь через *содержимое* возникает тогда, когда один модуль передает управление другому модулю не по имени, а непосредственно в тело модуля.

**Цельность модулей ПО.** Модуль с функциональной цельностью выполняет одну четко определенную функцию. Одним из критериев для формирования функционально цельного модуля является возможность формирования назначения модуля в виде одного предложе-

ния в повелительном наклонении без использования слов типа: если, тогда, вначале, затем и т. п.

*Модуль с последовательно-функциональной цельностью* выполняет несколько последовательных функций, при этом данные на выходе какой-либо функции целиком являются входными для следующей функции.

*Модуль с коммуникационной цельностью* выполняет несколько функций, для которых входные и выходные данные расположены в одном и том же хранилище данных.

*Модуль с алгоритмической цельностью* выполняет функции, соответствующие отдельным блокам схемы алгоритма программы. Алгоритмические цельные модули получаются, когда разработчик создает программу на основе анализа логической структуры процесса обработки данных без учета связей по данным между частями программы.

*Модуль с временной цельностью* реализует функции, выполняемые на каком-либо одном этапе обработки данных. При этом этапы обработки следуют друг за другом строго определенным образом. Так же как и алгоритмически цельный модуль, модуль с временной цельностью формируется на основе анализа логической структуры программы без учета связей по данным между блоками.

*Модуль с логической цельностью* выполняет множество функций, не связанных между собой ни по управлению, ни по данным. Единственное, что между ними общего – это сходство характера выполняемых функций. Например, модуль, реализующий все имеющиеся в программе функции обработки ошибок.

*Модуль, цельный по совпадению*, обладает самой слабой цельностью. В такие модули выделяются фрагменты программ, содержащие одинаковые последовательности команд, без учета контекста, в котором он может использоваться.

Во всех случаях следует стремиться к получению модулей, обладающих функциональной, функционально-последовательной, коммуникационной и, возможно, алгоритмической цельностью; избегать получения модулей с временной и логической цельностью. Цельность по совпадению является недопустимой.

## **Тема 17. Организационное и методическое обеспечение систем автоматизированного проектирования**

**Организационное обеспечение** – совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

Организационное обеспечение реализует следующие функции:

– анализ существующей системы управления организацией, где будет использоваться САПР, и выявление задач, подлежащих автоматизации;

– подготовку задач к решению на ЭВМ, включая ТЗ на проектирование САПР и технико-экономическое обоснование ее эффективности;

– разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

**Основные положения.** Организационное обеспечение целесообразно рассматривать в четырех основных аспектах:

– организационное обеспечение САПР в проектной организации, представляющее совокупность компонентов (положений, должностных инструкций, штатных расписаний и других документов), регламентирующих организационную структуру, функции подразделений и порядок их взаимодействия в условиях функционирования САПР;

– деятельность по организации создания и развития САПР в проектной организации, включая определение состава, функций и порядка взаимодействия лиц и подразделений проектной организации, создающей и развивающей САПР, а также порядка взаимодействия со сторонними организациями, участвующими в создании и развитии системы;

– деятельность по организации создания САПР на отраслевом уровне, включая определение состава, функций и порядка взаимодействия лиц, подразделений и организаций, обеспечивающих создание и развитие САПР в организациях отрасли;

– деятельность по организации создания САПР отрасли, включая определение состава, функций и порядка взаимодействия лиц, подразделений и организаций, обеспечивающих создание и развитие

САПР отрасли. САПР проектных организаций в данном случае являются подсистемами САПР отрасли.

Организационное обеспечение САПР во многом определяется составом и функциями взаимодействующих при автоматизированном проектировании подразделений в конкретной проектной организации. Наряду с подразделениями, присущими традиционным формам, при проектировании объектов в условиях функционирования САПР появляются группы специалистов с новыми функциями. В зависимости от выполняемых функций специалисты делятся на три группы: проектирующая, обеспечивающая и организующая.

В проектирующую группу входят проектировщики (проектные подразделения), непосредственно осуществляющие проектные процедуры и операции при взаимодействии с комплексом средств автоматизированного проектирования (КСАП). Обеспечивающая группа состоит из специалистов, осуществляющих функции по поддержанию работоспособности КСАП в процессе эксплуатации. При проектировании КСАП основными функциями группы являются: обеспечение согласованной работы между службами организации и утверждение «Технического задания на создание САПР». Она совместно с другими службами и отделами предприятия принимает участие в подготовке плана-графика создания САПР; в разработке технико-экономического обоснования создания САПР, в рассмотрении и обсуждении всех проектных документов на создание САПР; в разработке отдельных проектных документов на создание САПР; в разработке индивидуальных компонентов САПР с учетом специфики конкретного объекта проектирования; обеспечивает: составление и подачу заказной документации на вычислительную технику и серийно выпускаемые технические устройства для систем; комплектацию документации на изготовление не выпускаемых промышленностью технических устройств и размещение заказов на их изготовление; контроль приобретения серийно выпускаемых компонентов технических средств, а также изготовление не выпускаемых промышленностью технических устройств; участие в подготовке соответствующими подразделениями организации и размещение среди соответствующих «исполнителей» технических заданий на разработку строительной, сантехнической, энергетической и других смежных и сопряженных частей проекта, а также их передачу генпроектировщику на согласование и разработку, координацию, учет и контроль за ходом работ по созданию САПР и их приемку в организации, а также принятие мер по устранению нару-

шений сроков выполнения работы; участие в разработке программ, отладке и испытании САПР; оформление и утверждение гарантированного экономического эффекта от внедрения САПР и его уточнение на последующих стадиях создания САПР.

При вводе САПР в действие группа контролирует и обеспечивает проведение необходимых работ по реконструкции (строительству) производственных помещений, необходимых для функционирования САПР; обеспечивает заключение договоров на разработку проектов производства монтажных и наладочных работ и на выполнение этих работ контроль проведения монтажа и наладки компонентов САПР и КСАП в целом, формирование штатного коллектива специалистов-пользователей, а также обучение обеспечивающего персонала САПР проведение опытного функционирования САПР в организации под руководством разработчика.

Организации, в зависимости от степени их участия в создании САПР, могут выполнять функции заказчика, разработчика, пользователя или совмещать некоторые из них. Приведем основные функции, выполняемые заказчиком, разработчиком и пользователем.

При создании САПР заказчик предъявляет разработчику исходные требования к системе, в том числе и технико-экономические; осуществляет финансирование работ по созданию САПР, контроль за ходом выполнения работ и их приемку; обеспечивает утверждение ТЗ и согласовывает проектную документацию на создание САПР на всех стадиях системы; обеспечивает своевременное получение комплектующих технических средств САПР, контролирует ход создания САПР; осуществляет совместно с разработчиком и пользователем приемку САПР при вводе ее в действие.

Министерство может поручать выполнение функций заказчика организациям-пользователям. В этом случае организации должно быть предоставлено право заключать договор со сторонними организациями.

Для организации и обеспечения управления работами по созданию и развитию САПР заказчик назначает ответственных специалистов. Ответственность за выполнение функций заказчика несет руководство организации, выполняющей эти функции. При создании САПР разработчик проводит предпроектное обследование проектной организации-пользователя; разрабатывает ТЗ на создание САПР; разрабатывает проектную документацию, необходимую для изготовления компонентов и ввода в действие САПР; осуществляет со-

вместно с пользователем взаимную увязку компонентов САПР, а также стыковку типовых комплексов средств и компонентов САПР с индивидуальными компонентами по видам обеспечения; обеспечивает изготовление компонентов САПР (кроме технических средств) и отладку компонентов программного обеспечения; осуществляет научно-методическую помощь по подготовке пользователя к вводу в действие САПР; участвует в опытном функционировании САПР у пользователя; участвует в приемочных испытаниях САПР; осуществляет при необходимости на договорной основе в течение года авторский надзор за реализацией требований, изложенных в проектной документации на создание САПР, при функционировании системы; оказывает методическую помощь пользователю при подготовке и переподготовке кадров.

Разработчик может привлекать соисполнителей, по отношению к которым он выступает координатором их деятельности. О привлечении соисполнителей и работах по созданию САПР разработчик обязан уведомить заказчика. Основанием для заключения договора с соисполнителями и их финансирования является оформленный в установленном порядке договор заказчика с разработчиком на выполнение работ по созданию САПР в целом. При привлечении соисполнителей рекомендуется поручать им выполнение самостоятельных частей (разделов) или законченных этапов работ (стадий разработки).

При создании САПР пользователь подготавливает и предоставляет разработчику необходимую для создания САПР информацию и документацию, участвует в проведении предпроектного обследования и отвечает за обоснованность выдаваемых исходных данных; реализует привязку (адаптацию) типовых комплексов средств и компонентов САПР к своим условиям в соответствии с проектной документацией разработчика; разрабатывает индивидуальные компоненты САПР с учетом специфики конкретного объекта проектирования; разрабатывает, по согласованию с разработчиком, проектные документы на создание САПР; обеспечивает проведение необходимых работ по реконструкции (строительству) производственных помещений, необходимых для функционирования САПР; обеспечивает приобретение технических средств, а также изготовление не выпускаемых промышленностью технических устройств; осуществляет подготовку информации, необходимой для функционирования САПР, и ее перевод на машинные носители; организует и обеспечивает проведение монтажа, наладки компонентов САПР и КСАП в целом, а также про-

водит вместе с разработчиком опытное функционирование САПР; обеспечивает подготовку и переподготовку кадров, а также формирование штатного коллектива специалистов-пользователей; организует и обеспечивает проведение приемочных испытаний САПР со штатным коллективом специалистов-пользователей; осуществляет наиболее полное использование технических возможностей КСАП и функционирование САПР – в соответствии с технической документацией на систему; обеспечивает оформление и утверждение расчета фактической экономической эффективности от создания САПР; осуществляет контроль за ходом выполнения работ и их приемку; подготавливает организацию к вводу в действие системы; готовит предложения по планомерному совершенствованию и развитию системы.

Создание, функционирование и развитие САПР в проектной организации обеспечивает служба САПР. Эта служба привлекает к участию в создании САПР в качестве соисполнителей проектные и конструкторские подразделения – пользователей САПР организации.

Для практического применения можно рекомендовать следующие основные функции службы САПР в проектной организации.

В предпроектный период: участвует в проведении предпроектного обследования организации и обеспечивает участие в нем соответствующих служб и отделов организации; участвует в подготовке технико-экономического обоснования создания САПР и обеспечивает его согласование и утверждение; обеспечивает подготовку и предоставляет разработчику необходимую для создания САПР информацию и документацию и отвечает за обоснованность выдаваемых исходных данных; обеспечивает оформление и заключение соответствующими службами организации договоров с организациями-соисполнителями.

**Методическое обеспечение САПР.** Методическое обеспечение САПР представляет собой документы, характеризующие состав, правила отбора и эксплуатации средств автоматизированного проектирования. Допускается более широкое толкование понятия методического обеспечения, при котором под методическим обеспечением подразумевают совокупность *математического, лингвистического, программного обеспечения* и перечисленных документов, реализующих правила использования средств проектирования.

*Лингвистическое обеспечение САПР* – совокупность языков, используемых в процессе разработки и эксплуатации САПР для обмена информацией между человеком и ЭВМ.

Разработчик САПР использует алгоритмические языки для записи программ. Он же создает языки проектирования, предназначенные для общения инженера-пользователя САПР с ЭВМ и служащие для описания исходных данных и результатов решения проектных задач. Обращаясь к САПР, пользователь должен познакомиться со способами описания задач и объектов проектирования – с языками проектирования используемой САПР. Лингвистическое обеспечение САПР можно разделить: на *языки программирования* – на них записывается специальное программное обеспечение САПР; *языки проектирования* – языки для обмена информацией об объектах и процессе проектирования между проектировщиком и ЭВМ. Иногда в отдельную группу выделяют *языки управления* технологическим оборудованием, устройствами документирования, периферийными устройствами ЭВМ.

К средствам математического обеспечения относятся: средства моделирования процессов управления; типовые задачи управления; методы математического программирования, математической статистики, теории массового обслуживания и др.

## Глава 6. АВТОМАТИЗАЦИЯ КОНСТРУИРОВАНИЯ

### **Тема 18. Автоматизация конструирования машин на основе геометрического моделирования**

Аббревиатуру САПР (система автоматизированного проектирования, англ. – CAD, Computer-Aided Design) впервые использовал основоположник этого научного направления Айвен Сазерленд.

Широкое распространение САПР обусловлено появлением микропроцессорной техники, предоставившей возможности создавать, модифицировать и обрабатывать сложные графические изображения на экране.

В настоящее время термином «САПР» обозначают процесс проектирования с использованием сложных средств машинной графики, поддерживаемых пакетами программного обеспечения, для решения на компьютерах аналитических, экономических и эргономических проблем, связанных с проектной деятельностью.

*Достоинства автоматизированного проектирования.* Более быстрое выполнение чертежей. Конструктор, использующий систему САПР, может выполнять чертежи в среднем в три раза быстрее, чем работая за кульманом. Такая дисциплина работы ускоряет процесс проектирования в целом, позволяет в более сжатые сроки выпускать продукцию и быстрее реагировать на рыночную конъюнктуру.

*Повышение точности выполнения чертежей.* Точность чертежа, выполненного вручную, определяется остротой зрения конструктора и толщиной грифеля-карандаша. На чертеже, построенном с помощью системы САПР, место любой точки определено точно, а для более детального просмотра его элементов имеется средство, называемое «наезд» (zooming), позволяющее увеличить любую часть данного чертежа. Итак, все чертежи деталей и сборок, выполненные с помощью системы САПР, являются точными. Отдельные эскизные чертежи можно выполнять и без использования САПР.

*Повышение качества выполнения чертежей.* Качество изображения на обычном чертеже полностью зависит от мастерства конструктора, тогда как плоттер системы САПР рисует высококачественные линии и тексты независимо от индивидуальных способностей че-

ловека, работающего на этой системе. Кроме того, большинство обычных чертежей имеют следы ластика, что придает им неряшливый вид. Система САПР позволяет быстро стереть любое число линий без каких-либо последствий для конечного чертежа.

*Возможность многократного использования чертежа.* Построение изображения всего чертежа или его части можно сохранить для дальнейшей работы. Обычно это полезно, когда в состав чертежа входит ряд компонентов, имеющих одинаковую форму. Запомненный чертеж может быть использован повторно для проектирования, например, кондукторов и хомутов, анализа траектории движения инструмента. Раньше для решения каждой такой задачи требовался отдельный чертеж. Память компьютера, является также идеальным средством хранения скомпилированных библиотек символов, стандартных компонентов и геометрических форм.

*Ускорение расчетов и анализа при проектировании.* В настоящее время существует большое разнообразие программного обеспечения, которое позволяет выполнять на компьютерах часть проектных расчетов заранее.

*Высокий уровень проектирования.* Мощные средства компьютерного моделирования (такие как метод конечных элементов) освобождают конструктора от использования традиционных формул и позволяют проектировать нестандартные геометрические формы. Эти формы можно быстро модифицировать и оптимизировать, что позволяет снизить общие затраты в такой степени, которая была раньше недостижима, поскольку требовала слишком много времени.

*Сокращение затрат на усовершенствование.* Средства имитации и анализа, имеющиеся в системе САПР, позволяют резко сократить затраты времени и денег на тестирование и усовершенствование прототипов, которые являются дорогостоящими этапами процесса проектирования. Интегрированная вычислительная сеть с высококачественными средствами коммуникаций обеспечивает системе САПР более тесное взаимодействие с инженерными подразделениями по сравнению со старой организацией проектирования.

В основе любой САПР лежит графическое ядро (graphics kernel) – система хранения и обработки графической информации. Возможно несколько подходов к компьютерному геометрическому моделированию – каркасный, поверхностный, твердотельный. Кроме того, САПР может работать с плоскими проекциями (2D) или с объемной моделью изделия (3D). В настоящее время все современные

системы автоматизации проектирования – Pro/Engineer, SolidWorks, Unigraphics – работают с трехмерными моделями. Следует отметить, что распространенная система AutoCAD до 12-й версии вообще не поддерживала работу в 3D.

*Каркасная модель* полностью описывается в терминах точек и линий. Каркасное моделирование представляет собой моделирование самого низкого уровня и имеет ряд серьезных ограничений, большинство из которых возникает из-за недостатка информации о гранях, заключенных между линиями, и невозможности выделить внешнюю и внутреннюю область изображения твердого объемного тела. Однако каркасная модель требует гораздо меньше компьютерной памяти, чем две другие модели, и может оказаться вполне пригодной для решения некоторых задач.

На рис. 6.1 показан пример вида  $XYZ$  простого трехмерного каркасного изображения. Так как это форма с постоянным поперечным сечением, ее следовало бы строить следующим образом: сначала создать вид  $XY$ , а затем к каждой точке «приписать» два значения координаты  $Z$ , характеризующие «глубину» изображения. Формы, имеющие непостоянную толщину или поперечное сечение, уже довольно трудны для визуального восприятия, если работать только с одним видом. В таких случаях необходимо определять точки в трехмерном пространстве на всех видах  $XY$ ,  $XZ$ ,  $YZ$ ,  $XYZ$ .

Ограничения каркасных моделей:

1. *Неоднозначность.* Одним из основных недостатков рассматриваемых моделей является возможность неоднозначно интерпретировать ориентацию и видимость граней каркасного изображения. Например, трехмерное изображение (рис. 6.1) можно было бы интерпретировать и как вид сверху, и как вид снизу. Этот эффект, обусловленный природой каркасной модели, может привести к непредсказуемым результатам. В отличие от твердотельной модели в каркасной модели нельзя отличить «видимые» грани геометрической формы от «скрытых» (невидимых).

Операцию по удалению скрытых линий можно выполнить только вручную с применением команд редактирования к каждой отдельной линии. Но результат этой утомительной работы будет поистине катастрофическим, равносильным «разрушению» всей созданной каркасной конструкции, потому что линии, невидимые на одних видах, видимы на других видах. Удаление же любой линии на каком-либо виде неизбежно повлечет за собой удаление ее на всех других видах (рис. 6.2).

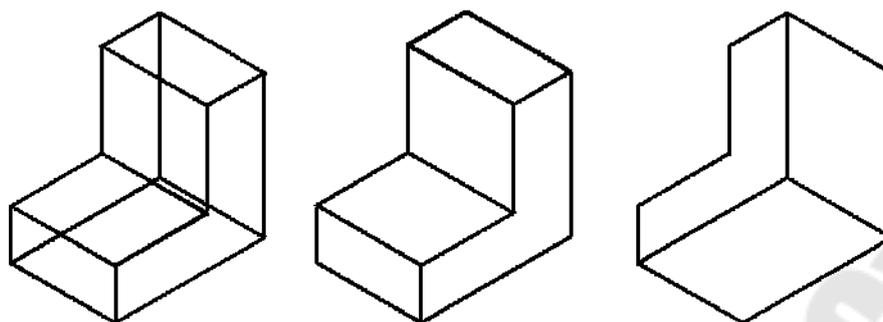


Рис. 6.1. Неоднозначность каркасной модели

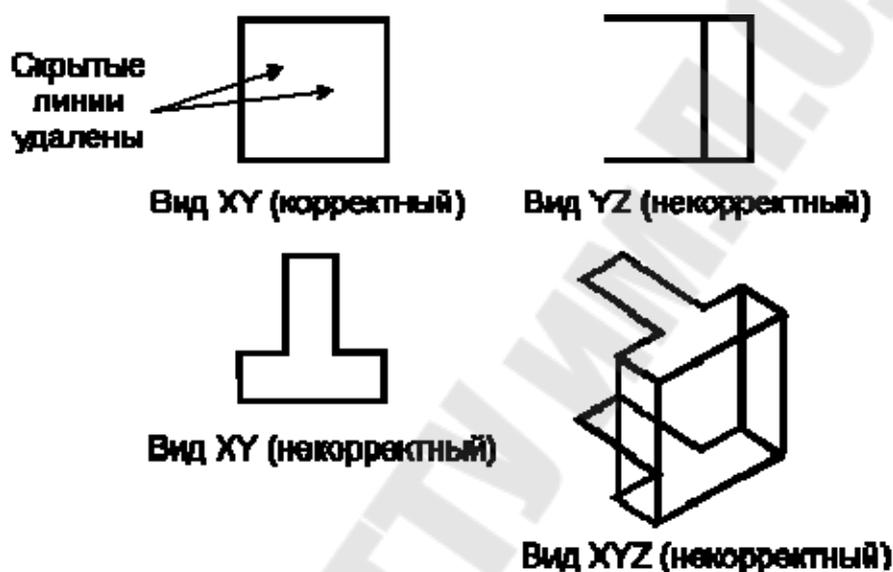


Рис. 6.2. Результат удаления скрытых линий на одном виде

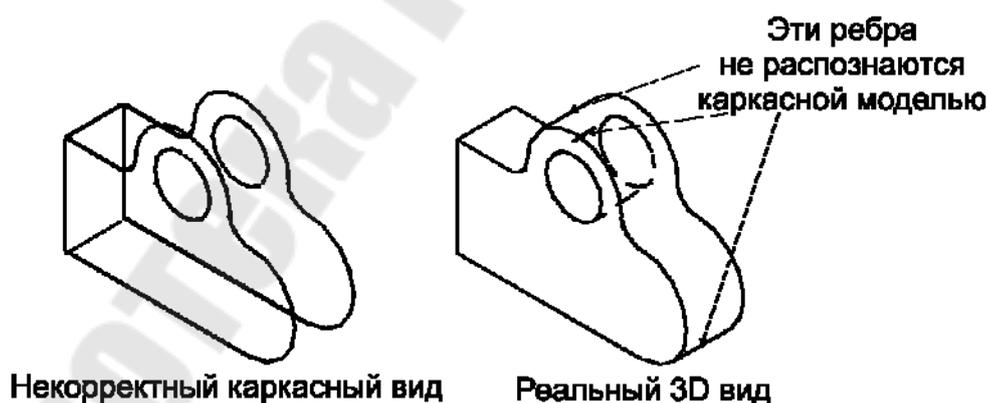


Рис. 6.3. Реальный трехмерный вид и каркасный вид

2. *Невозможность распознавания криволинейных граней.* Боковые поверхности цилиндрических форм реально не имеют ребер, хотя на изображении показываюся силуэты неких мнимых ребер, которые ограничивают такие поверхности. Расположение этих «ребер» в про-

странстве меняется в зависимости от направления вида (точки зрения), поэтому эти силуэты не распознаются как элементы каркасной модели и не изображаются на ней. На рис. 6.3 показана форма с криволинейной гранью и сквозным отверстием, которые можно увидеть на трехмерном изображении. На каркасной же модели мы имеем совершенно другое, некорректное представление данной формы.

Можно попытаться представить криволинейные грани продольными «теневыми линиями», расположенными через регулярные интервалы. Однако наличие этих несуществующих линий может внести еще больше путаницы в чертеж, который и так уже полон неоднозначностей. Эти проблемы возникают также при реализации представлений форм с непостоянным поперечным сечением (рис. 6.4).

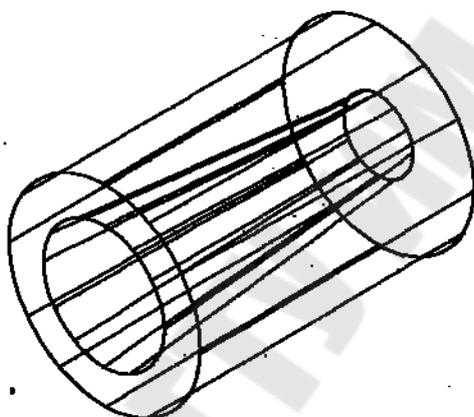


Рис. 6.4. Попытка отметить криволинейную грань отрезками

3. *Невозможность обнаружения взаимного влияния компонентов.* Каркасная модель не несет информации о поверхностях, ограничивающих форму, что обуславливает невозможность обнаружения нежелательных взаимодействий между гранями объекта и существенно ограничивает применение каркасного моделирования при трехмерном кинетическом анализе механизмов, имитации функционирования роботов, проектировании планов размещения заводского оборудования и выполнении сборочных чертежей сложных трубопроводов. Ограничено использование каркасных моделей и в пакетах, имитирующих траекторию движения инструмента, потому что при таком моделировании не могут быть выявлены на стадии проектирования многие коллизии, возникающие при механической обработке.

4. *Трудности, возникающие при вычислении физических характеристик,* также являются следствием недостатка данных о поверхностях. Существует вероятность того, что корректно построенная гео-

метрическая форма, а следовательно, и объем тела, отличающегося от базовых стандартных компонентов, могут быть определены неточно. Таким образом, значения физически характеристик (например, масса, площадь поверхности, центр тяжести или моменты инерции) будут недостоверными.

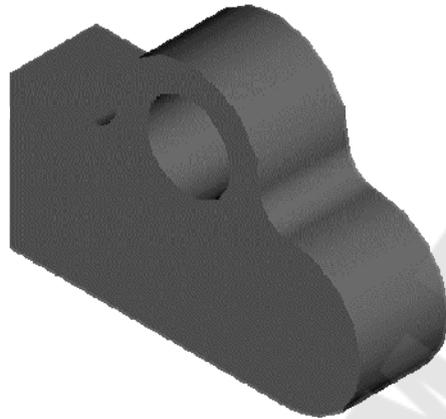


Рис. 6.5. Полутоновое изображение

5. *Отсутствие средств выполнения тоновых изображений.* Обеспечение плавных переходов различных цветов и нанесение светотени (рендеринг) составляют ту необходимую технику, которой должен владеть не только каждый художник, но и пакет трехмерного моделирования. Например, на рис. 6.5 показано полутоновое изображение, полученное в пакете AutoCAD 2000. Основным принципом этой техники является то, что «затенению» подвергаются грани, а не ребра. Таким образом, она не может быть применена к моделям, не имеющим поверхностей.

*Поверхностная модель* определяется с помощью точек, линий и поверхностей. Таким образом, ее можно рассматривать как модель более высокого уровня, чем каркасная модель, а следовательно, как более гибкую и многофункциональную.

Поверхностное моделирование имеет следующие преимущества по сравнению с каркасным: способность распознавать и изображать сложные криволинейные грани; способность распознавать грани и таким образом обеспечивать средство получения тоновых трехмерных изображений; способность распознавать особые построения на поверхностях, например отверстия; возможность получения качественного изображения и обеспечение удобного производственного интерфейса со станками с ЧПУ при имитации траектории движения инст-

румента в трехмерном пространстве для цикла обработки деталей сложных форм по нескольким осям.

### Типы поверхностей

1. *Базовые геометрические поверхности.* К ним относятся плоские поверхности, которые можно получить, начертив сначала отрезок прямой, а затем введя такую команду, которая разворачивает в трехмерном пространстве образ этого отрезка на заданное расстояние. Подобным образом (разверткой окружностей или дуг) могут быть сгенерированы цилиндрические и конические поверхности (рис. 6.6).

Области поверхностей также могут быть развернуты в трехмерные объекты (рис. 6.6). Следует отметить, что системы поверхностного моделирования не распознают такие формы, как твердые объемные тела; они представляют их просто как поверхности (на рис. 6.7 – семь плоских граней), соединенные между собой неким образом в пространстве и ограничивающие «пустой» объем.

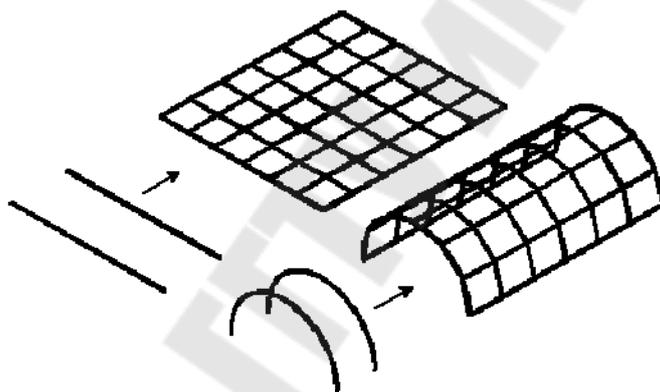


Рис. 6.6. Формирование развернутых плоских и криволинейных поверхностей

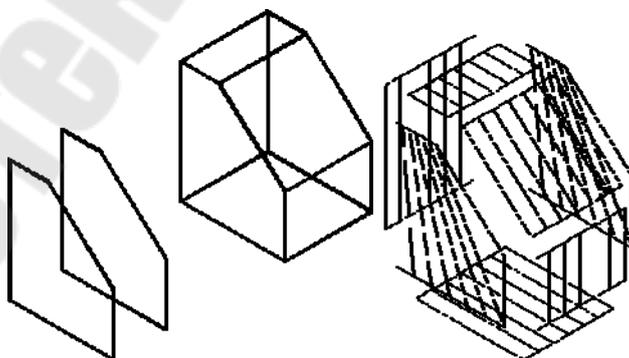


Рис. 6.7. Формирование объемного объекта разверткой плоской области

2. *Поверхности вращения* могут быть легко получены по команде, создающей поверхность вращением плоской грани вокруг определенной оси (эту процедуру можно трактовать как «круговую развертку»). При этом опять необходимо помнить, что создается не объемное тело, а генерируется только поверхность (рис. 6.8).

3. *Поверхности сопряжений и пересечений*. Возможность построения плавного сопряжения одной поверхности с другой является наиболее мощным и часто используемым на практике средством поверхностного моделирования. Кроме этого, может быть доступно средство определения пересечения поверхностей.

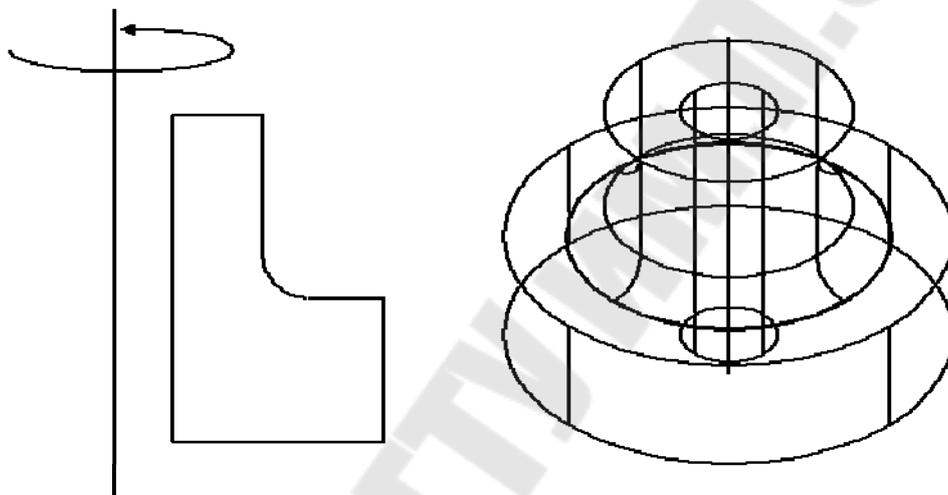


Рис. 6.8. Получение поверхности вращения

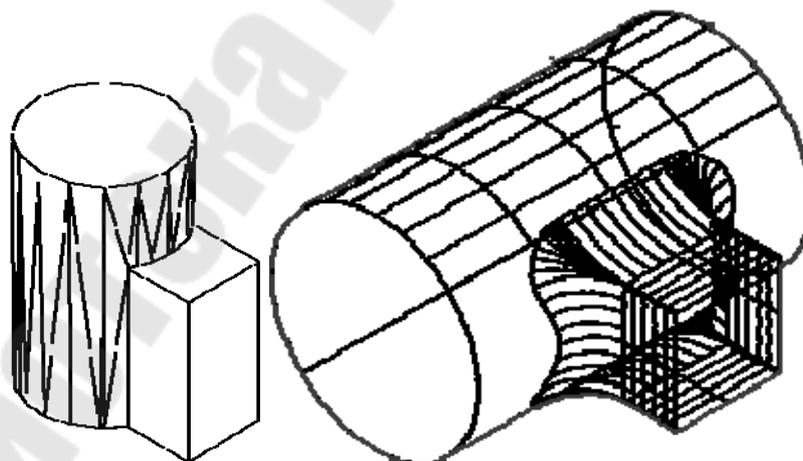


Рис. 6.9. Поверхности сопряжения

На рис. 6.9 показано, как в трехмерном пространстве можно построить обычное и плавное сопряжения боковых поверхностей параллелепипеда и цилиндра. Проблема порождения результирующей по-

верхности в данном случае сводится к задаче построения методом сплайн-интерполяции особых кривых в трехмерном пространстве, «выходящих» из сторон квадрата и «входящих» в автоматически генерируемую кривую на поверхности цилиндра, по которой заданные поверхности должны пересекаться.

4. *Аналитические поверхности.* Каждая такая поверхность определяется одним математическим уравнением с неизвестными  $X$ ,  $Y$  и  $Z$  (неизвестные обозначают координаты поверхности). Иначе говоря, чтобы изобразить любую аналитическую поверхность, необходимо знать математическое уравнение, которым она описывается.

Составную поверхность можно полностью определить, покрыв ее сеткой четырехугольных кусков. Каждый такой кусок носит название поверхности Кунца (Coons patch) по имени профессора Стивена Кунца, разработавшего данный метод поверхностного моделирования в 1963 году (рис. 6.10).

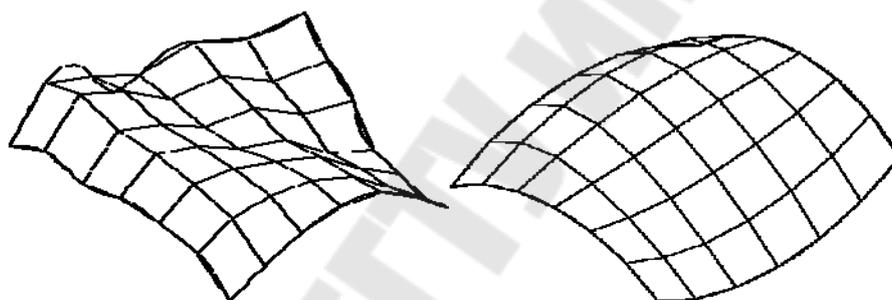


Рис. 6.10. Поверхности Кунца

Каждый кусок имеет геометрическую форму топологического прямоугольника. Границы кусков представляют собой непрерывные кривые, что обеспечивает гладкость поверхности, натянутой на сетку. Внутренняя область каждого куска определяется методом интерполяции. Изображение составной поверхности, реализованное указанным способом, может быть получено на экране либо с помощью построения по точкам сплайновых кривых, либо путем создания многогранного каркаса, на который система будет автоматически аппроксимировать натяжение гладкой криволинейной поверхности.

Хотя методы поверхностного моделирования обладают многими достоинствами, существует ряд ограничений на их использование. Основными являются следующие ограничения: возникновение неоднозначности при попытке моделирования реального твердого тела; недостаточность точности представления некоторых поверхностных моделей для обеспечения надежных данных о трехмерных объемных

телах; сложность процедур удаления скрытых линий и отображения внутренних областей.

*Твердотельная модель* описывается в терминах того трехмерного объема, который занимает определяемое ею тело. Таким образом, твердотельное моделирование является единственным средством, которое обеспечивает полное однозначное описание трехмерной геометрической формы. Этот способ моделирования представляет собой самый современный и наиболее мощный из трех разработанных методов.

Преимущества твердотельных моделей:

- полное определение объемной формы с возможностью разграничения внешней и внутренней областей объекта, что необходимо для обнаружения нежелательных взаимовлияний компонентов;
- обеспечение автоматического удаления скрытых линий;
- автоматическое построение трехмерных разрезов компонентов, что особенно важно при анализе сложных сборочных изделий;
- применение перспективных методов анализа с автоматическим получением изображения точных весовых характеристик и эффективных конструкций методом конечных элементов;
- наличие разнообразной палитры цветов, управление цветовой гаммой, получение тоновых эффектов манипуляцией источником света – всего того, что способствует реализации качественных изображений форм, компонентов и сечений;
- повышение эффективности имитации динамики механизмов, процедур генерации траектории движения инструмента и функционирования роботов.

Методы твердотельного моделирования делятся на два класса: метод конструктивного представления (метод C-Rep) и метод граничного представления (метод B-Rep).

**Метод конструктивного представления C-Rep.** Данный метод состоит в построении твердотельных моделей из базовых составляющих элементов, называемых твердотельными примитивами и определяемых формой, размерами, точкой привязки и ориентацией. К ним могут относиться призма, сфера, цилиндр, конус и др. Булевы операции являются существенным инструментарием для построения модели C-Rep при определении взаимоотношений между соседними примитивами. Булевы операции базируются на понятиях алгебраической теории множеств и имеют обычный смысл, когда применяются к

твердотельным объектам. Рассмотрим три булевы операции: объединение, разность и пересечение (рис.6.11).

*Операция объединения* ( $\cup$ ) определяет пространство внутри внешней границы составной формы, полученной из двух тел с общей областью. Объединение двух произвольных кругов  $A$  и  $B$  (рис. 6.11) представляет собой заштрихованную область  $A \cup B$ . Таким образом, операция объединения определяет результирующую составную форму как один элемент. На том же рисунке показано применение эквивалентной операции для двух твердотельных примитивов (цилиндра и кубоида).

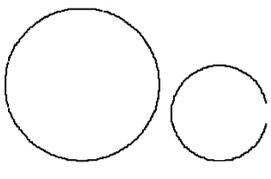
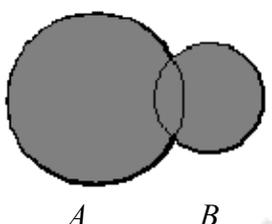
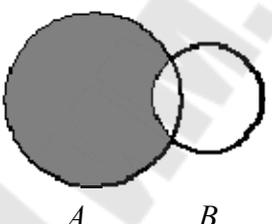
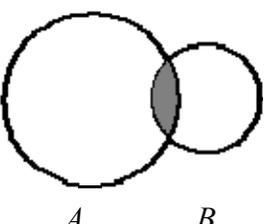
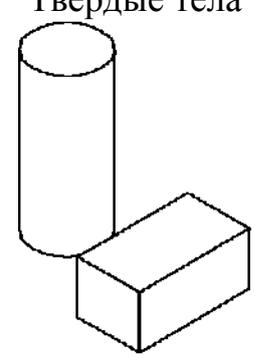
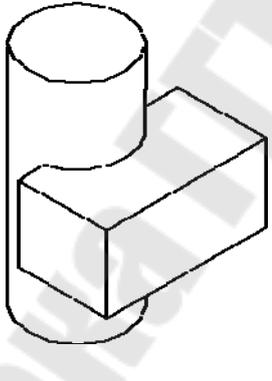
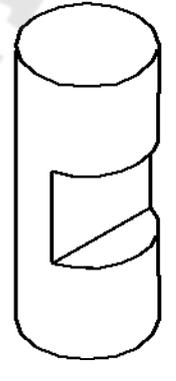
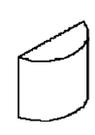
	Объединение $A \cup B$	Разность $A - B$	Пересечение $A \cap B$
Множества 			
Твердые тела 			

Рис. 6.11. Булевы операции

*Операция разности* ( $-$ ) определяет пространство, ограниченное поверхностью, оставшейся от одной формы, и внешней границей общей области двух форм. Заштрихованной областью  $A - B$  показан результат действия операции булевой разности на круги  $A$  и  $B$ . Ниже на этом же рисунке изображен цилиндр с пазом, являющийся разностью двух твердых тел.

*Операция пересечения* ( $\cap$ ) определяет пространство внутри границ общей области объектов. Пересечение кругов  $A$  и  $B$  представлено заштрихованной областью  $A \cap B$ , а пересечение тел образует сегмент

цилиндра. Обратите внимание, что все части обоих тел, не являющиеся общими, отсекаются.

**Построение модели C-Rep.** Твёрдотельные примитивы могут быть созданы с помощью развертки двумерных областей в трехмерном пространстве (как при поверхностном моделировании), только в данном случае образуется именно «твёрдый объем», а не пустое пространство, ограниченное несколькими поверхностями.

Природа рассматриваемых моделей такова, что на них можно автоматически удалять скрытые линии и получать тоновые изображения поверхностей (рис. 6.12).

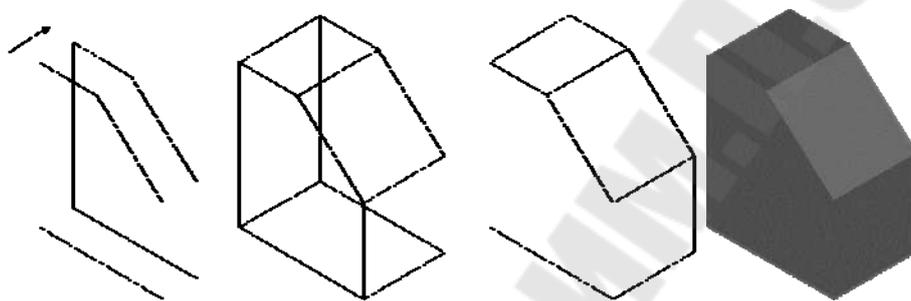


Рис. 6.12. Этапы создания твердотельной модели операцией выдавливания

В основе каждого объемного тела лежит *плоский профиль*, к которому применяется операция выдавливания – параллельного перемещения профиля вдоль заданной направляющей с образованием твердого тела.

Другой способ создания твердотельного примитива – вращение профиля вокруг заданной оси (рис. 6.13).

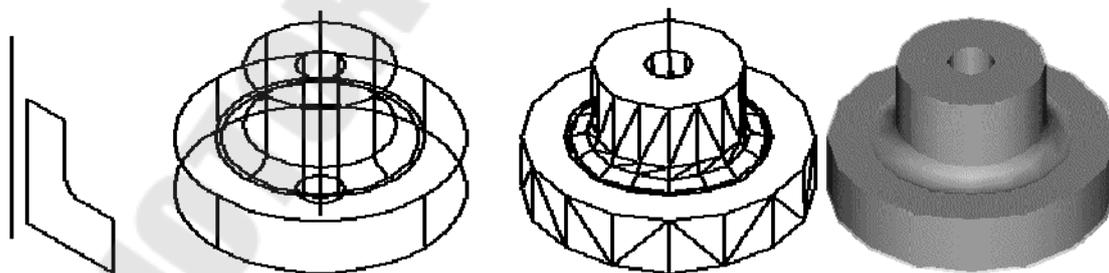


Рис. 6.13. Этапы создания твердотельной модели операцией вращения

**Моделирование методом B-Rep.** Моделирование производится теми же средствами, какими пользуются при создании компонентов методом C-Rep, т. е. вначале строятся примитивы с помощью линей-

ной или круговой развертки, а затем выполняется построение из них с помощью булевых операций составных форм.

Однако пакеты моделирования C-Rep распознают сложную форму в терминах составляющих ее твердотельных примитивов, тогда как пакеты анализа B-Rep оперируют с моделью в терминах ребер и граней, которые образуют трехмерную граничную поверхность объемного тела. Данные об этой поверхности структурируются в терминах ее топологии (которая описывает число граней) и ее геометрии (описание формы, а также размещения в пространстве вершин, ребер и граней).

Основным достоинством моделирования методом B-Rep является возможность более легкой модификации граничной поверхности. Однако соответствующие системы моделирования методом граничного представления требуют большего объема компьютерной памяти.

Различия между рассматриваемыми типами твердотельного моделирования невелики, и в некоторых системах применяется комбинация этих двух методов.

### ***Применения булевых операций***

*Вычисление объемных и весовых характеристик.* Как уже показано выше, булевы операции сложения и разности используются для построения составной твердотельной модели. После каждого преобразования модели автоматически переопределяется ее объем и сразу же корректируются ее весовые характеристики.

*Расчет напряжений.* Применяя булевы операции разности так же, как они использовались для получения разрезов (см. предыдущий раздел), можно разбить трехмерную модель на твердотельные конечные элементы для проведения расчета напряжений методом МКЭ.

*Имитация операций механической обработки* является одним из наиболее важных применений булевых операций при твердотельном моделировании. Операции резки металла, такие как точение, фрезерование и сверление, могут быть естественно сопоставлены булевым операциям сложения и разности.

*Обнаружение взаимовлияний и коллизий* является естественным приложением булевой теории для проверки корректности конструкции.

Для практической реализации твердотельного моделирования разработаны специальные библиотеки функций, которые называют графическими ядрами. Наиболее широко применяются два ядра: Parasolid и ACIS. На них основаны последние версии AutoCAD, SolidWorks, SolidEdge, Unigraphics, Pro/Engineer. Каждая из этих систем

имеет свою область применения и свои преимущества и недостатки. Рассмотрим коротко их характеристики.

AutoCAD'2000 и выше фирмы Autodesk Inc. – удобен для 2D-черчения и разработки на его базе специализированных САПР. Твёрдотельное моделирование остается трудоемким и в какой-то степени бесполезным, так как нет возможности получения проекций примитива. Облегчает ситуацию надстройка AutoCAD Mechanical Desktop или новая система Autodesk Inventor.



SolidWorks фирмы SolidWorks Corp. и SolidEdge – новые (с 1997) системы 3D-моделирования, отличающиеся удобным интерфейсом, соответствием требованиям ЕСКД, легкостью редактирования моделей, возможностью создания сложных сборок. В последнее время SolidWorks считается лучшей в своем классе.



Pro/Engineer фирмы Parametric Technology Corporation – «тяжелая» система, в основе которой лежит тщательная математическая модель, известная как feature-based design. Отличается очень большими возможностями, реализован на разных платформах, но страдает очень сложным и неудобным пользовательским интерфейсом.



## **Тема 19. Современные программные САД/САМ продукты для проектирования изделий машиностроения**

До середины 90-х годов пользователям САД/САМ систем были доступны лишь два пути выполнения проектируемого изделия:

– использование мощной, полностью интегрированной системы например, CATIA, широко известной и популярной в мире, разработчик DASSAULT SYSTEMES во Франции;

– работа с более «легкими», решающими только определенный круг задач.

Оба этих варианта были по-своему хороши, но за возможность использовать преимущества первого из них приходилось выкладывать *порядочную сумму денег*, а второй, при всей доступности и простоте, мог подойти для решения только одной локальной задачи.

Пользователям специализированных САПР, предназначенных для использования на персональных компьютерах, остается только самостоятельно дописывать интерфейсы обмена или программы декодирования из одного формата данных в другой.

Но появляются новые технологии, и то, что раньше было доступно лишь пользователям сверхмощных графических станций, теперь доступно и для конструктора, работающего на обычном современном персональном компьютере.

Мощность и быстродействие современных персональных компьютеров позволяют поручать им задачи, еще несколько лет назад являвшиеся исключительной прерогативой дорогостоящих рабочих станций. Пользователям требуется *комплексное решение* ряда проблем проектирования. К ним относятся моделирование изделий, их расчет и разработка технологического процесса изготовления.

Современные CAD/CAM системы можно подразделить на следующие уровни: системы высокого уровня – UNIGRAPHICS, CATIA; САПР среднего уровня – SOLID EDGE, SOLID WORKS, AUTODESK AUTOCAD, AUTODESK MECHANICAL DESKTOP, AUTO-DESK INVENTOR; САПР нижнего уровня – АСКОН КОМПАС-3D, T-FLEX 3D.

Система CAD/CAM/CAE компании Unigraphics Solutions – система современных средств *трехмерного моделирования*, формирования *сложных сборок*, электронного макетирования, воплощения визуальных проектов в реальное изделие. Она содержит в себе технологию, ориентированную на процессы производства сложных изделий (автомобилей, самолетов, медицинских протезов и инструментов). Unigraphics предоставляет компаниям возможность строить полную *цифровую модель* будущего изделия, необходимую для проектирования и конструирования, инженерного анализа и изготовления.

AutoCAD (фирма AUTODESK Inc., USA) – первая получившая широкое распространение система автоматизированного проектирования (САПР) на персональных компьютерах; удерживает лидерство и до настоящего времени.

Графический редактор AutoCAD распространен повсеместно, в том числе и в Белоруссии. До сих пор многие пользователи воспри-

нимают его как единственное средство для выпуска чертежей. На самом деле сегодня AutoCAD является базовой САПР, которую можно использовать в самых разных целях: как электронный кульман для создания двумерных чертежей и рисунков и как систему 3D моделирования.

Mechanical Desktop – система твердотельного и поверхностного трехмерного моделирования, проектирования сборок, создания чертежей. 1995 год стал переломным для мирового рынка систем CAD/CAM массового применения. Впервые за долгое время пакеты твердотельного параметрического моделирования с промышленными возможностями стали доступны пользователям персональных компьютеров. Одно из лучших решений такого уровня смогла предложить американская компания SolidWorks Corporation. Созданная в 1993 году, эта фирма уже через два года (ноябрь 1995 года), выпустила на базе геометрического ядра Parasolid свой первый программный продукт. Пакет твердотельного параметрического моделирования SolidWorks 95 сразу занял ведущие позиции среди продуктов этого класса, буквально ворвавшись в мировую «табель о рангах» систем CAD/CAM.

К середине 90-х годов многие конструкторы и технологи во всем мире практически одновременно пришли к одинаковому выводу – для того, чтобы повысить эффективность своего труда и качество разрабатываемой продукции, необходимо срочно переходить от работы в смешанной среде двумерной графики и трехмерного моделирования к *использованию объемных моделей*, в качестве основных объектов проектирования.

Создатели системы SolidWorks учли все эти требования, и, таким образом, дали возможность десяткам тысяч конструкторов использовать на своих персональных рабочих местах новейшие достижения науки в области технологий CAD/CAM.

Еще одна современная система Solid Edge является принципиально новой системой автоматизированного конструирования (CAD), которая предназначена для разработки *сборочных узлов* и *геометрического моделирования* отдельных деталей. Разработанный с использованием передовой технологии трехмерного моделирования, Solid Edge обеспечивает настоящий прорыв в области интерактивного конструирования изделий машиностроения и позволяет значительно сократить время разработки изделия.

Компания Unigraphics Solutions имеет уникальную для отрасли степень интеграции между своими продуктами САПР среднего уровня Solid Edge, САПР верхнего уровня Unigraphics и системой введения проекта iMAN. В то время как другие могут предложить только передачу геометрических моделей, Unigraphics обеспечивает сохранение ассоциативности и возможности автоматического обновления деталей и сборок в обоих направлениях. Для того, чтобы это стало возможно, системы должны иметь общими не только *геометрическое ядро* (Parasolid), но и систему именования топологии и систему идентификации изменения топологии.

Новый продукт фирмы AutoDesk – AutoDesk Inventor – предназначен для твердотельного моделирования и работы с *большими сборками*. Это принципиально новый программный продукт, *не основанный* на платформе AutoCAD, ориентированный на пользователей САД/САМ систем высокого уровня. В нем реализованы принципиально новые технологические подходы, инструменты и приемы проектирования.

КОМПАС-3D – новый модуль известного программного комплекса. Осенью 1999 года компания «АСКОН», на протяжении уже целого десятилетия известная как поставщик чертежно-графического редактора КОМПАС-ГРАФИК и семейства продуктов под маркой КОМПАС, выпускает на рынок свою систему твердотельного трехмерного моделирования КОМПАС-3D. Ввиду сравнительно невысокой цены КОМПАС-3D его можно рекомендовать для эксплуатации в комплексе с «тяжелыми» и «средними» САПР (например, с пакетом SolidWorks). Подготовленные в КОМПАС-3D модели деталей можно затем передать в смежную систему для последующей их сборки. Как и было запланировано, основным новшеством стало появление в КОМПАС-3D средств *моделирования сборок*.

Сборка является новым типом документа КОМПАС. Принципы моделирования сборки позволяют пользователю получить *объемную модель изделия в целом*, с учетом всей его структуры.

Для создания трехмерной модели проектируемого изделия фирма «Топ Системы» предлагает систему параметрического трехмерного твердотельного моделирования T-FLEX CAD 3D, которая является закономерным развитием системы T-FLEX CAD 2D. T-FLEX CAD 3D построена на геометрическом ядре Parasolid фирмы Unigraphics Solutions, которое сегодня считается лучшим ядром для трехмерного твердотельного моделирования.

Моделирование в T-FLEX CAD 3D может осуществляться как непосредственно в 3D пространстве, так и на основе данных двумерного чертежа. Проектировщик может выбрать любой способ работы в T-FLEX CAD 3D или их комбинацию: от трехмерной модели к чертежам изделия; от двухмерного чертежа к трехмерной модели.

## **Тема 20. Интеллектуальные подсистемы систем автоматизированного проектирования**

**1. Введение в системы искусственного интеллекта.** *Искусственный интеллект (ИИ)* – это программные системы, имитирующие на компьютере мышление человека. Для создания такой системы необходимо изучить процесс мышления человека, решающего определенные задачи или принимающего решения в конкретной области, выделить основные шаги этого процесса и разработать программные средства, воспроизводящие их на компьютере. Следовательно, методы ИИ предполагают простой структурный подход к разработке сложных программных систем принятия решений.

Искусственный интеллект, как следует из самого названия, придает компьютеру черты разума. Методы ИИ упрощают объединение программ и дают возможность заложить в систему ИИ способность к самообучению и накоплению новой, полезной информации.

**Применение систем искусственного интеллекта.** Обработка естественного языка – понимание и синтез речи, создание моделей коммуникаций между человеком и искусственной системой, создание систем интеграции сообщений разной модальности в единый образ.

Основными функциями анализа языка являются: лексический анализ (анализ слов); синтаксический анализ (анализ порядка слов в предложении с учетом правил грамматики); семантический анализ (анализ значения предложения самого по себе и в его связи с другими предложениями).

**Создание экспертных систем.** Экспертная система – это система обработки данных, основанная на знаниях и опыте человека в конкретной узкой области.

Знания: формальные (отражены в справочниках, законах); неформальные (принадлежат одному человеку, нигде не отражаются);

Автоматическое программирование – возможность описания алгоритмов на языке высокого уровня.

Обработка визуальной информации: обработка изображения; анализ изображения; синтез изображения.

Искусственный интеллект и САПР.

Основные принципы интеллектуализации САПР: необходимость использования накопленного опыта специалистов-проектировщиков; необходимость решения трудноформализуемых задач; необходимость улучшения интерфейса между пользователем и системой.

Задачи САПР, решение которых целесообразно методами ИИ – структурный и параметрический анализ. Структурный анализ – определение структуры проектируемого объекта. Параметрический анализ – определение численных параметров элементов при заданной структуре и требованиях к работоспособности.

## **2. Экспертные системы**

**2.1. Экспертная система** – это система ИИ, созданная для решения задач в конкретной проблемной области. Источником знаний для наполнения экспертных систем служат эксперты в соответствующей сфере человеческой деятельности. При создании экспертной системы группа, состоящая из эксперта и инженера по знаниям, собирает факты, правила и эвристические правила, относящиеся к проблемной области, а затем включает их в программу ИИ. *Эвристическим* называется правило, формулирующееся не на основании обычных, признанных знаний, а на основании практических знаний эксперта.

В основе человеческой деятельности лежит мышление, которое можно представить в виде следующих составляющих:

**Цель.** Целью называется конечный результат, на который направлены мыслительные процессы человека. Цель заставляет человека думать.

**Факты и правила.** Человеческий мозг хранит огромное число фактов и правил их использования. Для достижения определенной цели надо только обратиться к нужным фактам и правилам.

*Пример*

Факт 1. Тихие, темные улицы опасны.

Факт 2. Пожилые люди обычно не совершают преступлений.

Факт 3. Полиция защищает людей от преступников.

*Правило 1.* ЕСЛИ на тихой, темной улице встретится пожилой человек, ТО можно не очень беспокоиться.

*Правило 2.* ЕСЛИ на тихой, темной улице вы видите полицейского, ТО можно чувствовать себя в безопасности. Упрощение.

Упрощением называется выбор правильной реакции на конкретную ситуацию. Механизм упрощения быстро и эффективно выбирает факты и правила, нужные для достижения ближайшей цели.

Механизм логического вывода (МЛВ). Механизм вывода завершает мыслительный процесс, выполняя заключения на основании правил, отобранных механизмом, *упрощения*, и генерируя новые факты, которые добавляются к знаниям человека.

**2.2. Структура и состав экспертной системы.** Типовая экспертная система включает в себя: базу знаний (БЗ); механизм логического вывода (МЛВ); модуль извлечения знаний; систему объяснений.

*Структура базы знаний.* База знаний включает факты и правила, по которым в зависимости от входной информации принимается то или иное решение. Факты представляют краткосрочную информацию. Они могут не быть в БЗ и могут изменяться. Правила – долгосрочная информация о том, как манипулировать знаниями для получения решений.

*Механизм логического вывода.* МЛВ решает две задачи: дополнение и изменение БЗ на основе анализа БЗ и исходной информации; управление порядком обработки БЗ.

Механизм логического вывода функционирует циклически. В каждом цикле решаются следующие задачи: сопоставление, выбор, действие. Сопоставляется имеющаяся условная часть с имеющимися фактами. В случае наличия нескольких правил с одинаковой условной частью выбирается одно правило для срабатывания действия. Выполняется действие, определенное следственной частью сработавшего правила.

В настоящее время существуют две основные стратегии логического вывода: прямая цепочка рассуждений (основана на сопоставлении исходных данных со всеми правилами в БД и получении некоторого результата); обратная цепочка рассуждений (выдвигается гипотеза о предполагаемом решении задачи и путем анализа БЗ ищется подтверждение этой гипотезы путем сравнения БЗ и исходных данных; если подтверждение не найдено, то выдвигается новая гипотеза).

Назначение модуля извлечения знаний состоит в предоставлении экспертных знаний в структурированное в виде, пригодном для использования в компьютере. Система объяснения предназначена для показа пользователю всего процесса рассуждения, в результате которого было найдено или не найдено решение.

**2.3. Виды экспертных систем.** Выделяют группы: оболочка экспертной системы (экспертная система, не содержащая никаких знаний); прототип экспертной системы (демонстрационно-экспертная система, которая показывает возможность использования методов искусственного интеллекта для решения задач в данной конкретной области).

В разработке экспертной системы работают специалисты трех специальностей:

- эксперт (определяет знания, которые характеризуют экспертную систему, и обеспечивает их правильность и правоту);
- инженер по знаниям (помогает эксперту структурировать знания, определяет способ представления знаний и выбирает инструментальные средства для разработки экспертной системы);
- программист.

Экспертная система работает в двух режимах: в режиме приобретения знаний и режиме решения задач.

### **3. Представление знаний в экспертной системе**

#### **3.1. Представление с помощью правил.**

ЕСЛИ <условная часть> ТО <следственная часть>.

Правила выражены условным отношением ЕСЛИ–ТО, т. е. когда условная часть правила удовлетворяет фактам, выполняется следственная часть. Действия, предусмотренные следственной частью, могут состоять: в модификации фактов; во взаимодействии с внешней средой. Механизм логического вывода в этом случае работает в цикле: сопоставить и выполнить.

*Достоинства:* естественность, единообразие структуры, гибкость изменения логического вывода.

*Недостатки:* процесс трудно поддается управлению, трудно представить иерархию понятий, малоэффективен процесс поиска решений.

**3.2. Представление знаний в виде фреймов.** Фреймом называется структура для описания стереотипной ситуации, которая включает характеристики этих ситуаций и их значения. Характеристики называются *слотами*, а их значения – *заполнителями слотов*. Заполнители слотов могут содержать не только значения, но и процедуру определения этого значения, в том числе и правило для поиска этого значения.

Процедуры, которые входят в состав слотов, бывают трех видов:

– «Если добавлено». Процедура выполняется, если добавлена новая информация в слот.

– «Если удалено». Процедура выполняется, если удаляется информация из слота.

– «Если необходимо». Процедура выполняется, если запрашивается информация из слота, а он пустой.

Процедуры, расположенные в слоте, называются связанными.

Фрейм представляет собой иерархическую структуру для моделирования конкретной предметной области. На верхнем уровне иерархии находится фрейм, содержащий информацию, истинную для всех остальных фреймов. Фреймы обладают способностью наследовать характеристики фреймов, находящихся на более высоком уровне, но они могут быть изменены на более низком уровне. Это позволяет обрабатывать изменения в знаниях.

**3.3. Представление знаний в виде семантических сетей.** При разработке экспертных систем для графического представления знаний используются семантические сети. Метод основан на сетевой структуре. Семантические сети состоят из узлов, изображаемых окружностями, и соединяющих их линий со стрелками. Узлы сети обозначают некоторые порции информации, а соединяющие их линии – взаимосвязи между ними.

К данному виду представления знаний относятся следующие отношения: «Является» Объект входит в состав данного класса. «Имеет» Позволяет задавать свойства объектов. «Является следствием». «Имеет значение».

Достоинства – легкость обработки иерархии отношений.

Недостаток – сложность обработки исключений.

**3.4. Представление знаний с помощью коэффициентов уверенности.** Коэффициенты уверенности (КУ) используются в области математики, называемой *нечеткой логикой*. Поскольку эвристические правила ЕСЛИ-ТО основываются исключительно на человеческом опыте, с полной определенностью никогда нельзя сказать, что они верны. Пользователь экспертной системы также не может быть полностью уверен что значения, которые он присваивает переменным, абсолютно корректны. КУ может иметь значение от  $-1$  до  $1$ . Отрицательное значение КУ показывает степень уверенности в том, что правило неверно, а положительное значение – что правило верно.

Сформулируем общие принципы.

Выбрать максимальное значение КУ из КУ для условий правила, разделенных логическим оператором. Если в правиле есть оператор ИЛИ, выбрать максимальное значение из КУ для всех условий правила, разделенных оператором И для всех условий, связанных оператором ИЛИ умножить выбранный КУ на КУ правила. Если существует несколько правил с одинаковым логическим выводом, выбрать из всех полученных КУ максимальных. Если граничные коэффициента уверенности заданы, то логический вывод считается верным когда его КУ превышает заранее заданные граничные значения. Работа с БЗ продолжается до тех пор, пока значение коэффициента уверенности логического вывода больше граничного значения.

#### **4. Нечеткие множества**

Методы построения математических моделей часто основаны хотя и на неточной, но в целом объективной информации об объекте. Однако возможны ситуации, когда при построении моделей решающее значение имеют сведения, полученные от эксперта, обычно качественного характера. Они отражают содержательные особенности изучаемого объекта и формулируются на естественном языке. Описание объекта в таком случае носит нечеткий характер. Например, экспертом предоставлена следующая информация:

*ЕСЛИ концентрация кислоты мала и температура раствора невысокая, то выбирается первый вариант конструкции аппарата.*

*ЕСЛИ концентрация кислоты мала и температура раствора высокая, то выбирается второй вариант конструкции аппарата.*

*ЕСЛИ концентрация кислоты большая и температура раствора высокая, то выбирается третий вариант конструкции аппарата.*

Пусть ставится задача определить конструкцию аппарата, если концентрация кислоты – 25 %, а температура раствора 30–40 °С.

Каким образом работать с такой информацией?

Существуют такие понятия, как «высокая», «невысокая», «большая», «малая». Применительно к понятию «концентрация» значение «малая» может быть интерпретирована, как любой, из значений от 2 до 20 %. Для понятия «температура» значение «высокая» может интерпретироваться в пределах от 40 до 100 °С.

В данном случае, «высокая», «малая», «невысокая», «большая» являются нечеткими переменными. Каждой нечеткой переменной соответствуют определенные значения в некотором интервале. Использование нечетких переменных для построения и анализа правил назы-

вают нечеткой логикой, в основе которой лежит понятие нечеткого множества.

### **5. Интеллектуальные САПР и принципы их построения**

Совершенствование САПР должно основываться на успехах развития программного обеспечения (ПО), реализующего методы искусственного интеллекта. Создаются новые предпосылки для человеко-машинного взаимодействия, в условиях которого конструктор-проектировщик освобождается от необходимости описания процесса решения задачи и основное внимание сосредотачивает на формировании задачи процесса проектирования. При этом он формулирует проблему, дав описание ее области, и цели, которые должны быть достигнуты. При неудовлетворительном варианте решения пользователь может модифицировать формулировку проблемы. ПО, основанное на элементах ИИ, осуществляет выбор шагов процесса и их использование, а также выбор между различными альтернативными шагами путем сопоставления их ожидаемых последствий с целями, которые необходимо достигнуть. Помимо этого, ПО должно корректировать свой выбор и обучаться на ошибках. Существующие САПР ориентированы на выполнение трех основных функций: автоматизация проектирования; человеко-машинное взаимодействие; управление базами данных.

САПР относится к классу систем организационного типа. Под организационной понимается система, предусматривающая организацию деятельности коллектива людей для достижения определенных целей. Такие системы характеризуются присущим им типом поведения. Известны трудности описания моделей поведения (функционирования) подобного типа систем. Это касается, прежде всего, сложности описания целей управления в виде количественных соотношений между параметрами системы, трудностью установления количественных зависимостей (соотношений), многошаговостью процесса управления таких систем, причем содержание каждого шага однозначно заранее не определено. Выделяют следующие особенности подобных систем, усложняющие формализацию задач управления и поведения САПР: слабая конструкция задач; многокритериальность, приводящая к оптимизации показателей отдельного параметра за счет ухудшения других (это свойство характерно для слабо структурированных задач, в которых формулировка целевой установки в виде единого критерия труднодостижима); условие неопределенности решения задачи.

Эти особенности и создают основные сложности формализации процессов поведения и управления в таких системах. Поэтому в основу исследования и описания систем организованного типа целесообразно положить лингвистическую концепцию, из которой следует, что естественный язык и его математические модели играют исключительно важную роль как среда для моделирования процессов мышления и для обеспечения общения на естественном языке. В ней центральное место занимает проблема построения семиотических моделей, так как от способа ее решения зависит качество выдаваемых решений по управлению моделью. В настоящее время семиотические модели строятся на основе неформальных методов, что предъявляет высокие требования к квалификации ее разработчиков и вызывает существенные трудности в использовании созданных моделей для описания других объектов или при необходимости их корректировки. В связи с этим вызывает научный и практический интерес задача построения аксиоматических моделей систем и объектов проектирования основе формально-логического подхода, базирующегося на аппарате логики предикатов первого порядка.

При автоматизации управления в практике использовались методы аналитического и имитационного моделирования, но эти методы не позволяют организовать структурную адаптацию моделей алгоритмов. Для преодоления этого недостатка обращаются к логико-лингвистическим моделям, основанными на логической формализации управления в САПР. При этом процессы автоматизации проектирования могут основываться на семиотических моделях.

Семиотическая модель представляет собой лингвистическую модель среды с присущими ей законами, в которой функционирует система. Такая модель содержит в себе все знания о среде, управлении, объектах, процессах и способах трансформации при выработке управляющих воздействий. Поведение САПР и ее «интеллектуальность» зависят от ее целей и объема накопленных знаний: актуальных (совокупность истинных фактов о внешнем мире и законов, ему присущих) и (набор универсальных и специализированных методов достижения целей системы относительно внешнего мира). САПР осуществляет целенаправленное поведение, если она ищет план решения поставленной задачи, отраженных в них, в соответствии со своими знаниями и возможностями и выполняет действия, адекватные плану.

Общепризнано, что для выполнения возложенных на семиотическую модель функций она должна обеспечить: структурированное

представление знаний о внешнем мире и способах деятельности в ней разрабатываемой системы; способ описания знаний (декодированный или процедурный), не зависящий от их вида; представление информационных моделей управления; контекстуальное управление выбором операторов языка манипулирования знаниями при выработке плана решения сформулированной задачи проектирования. Известны четыре типа языков, используемых для построения семиотических моделей управления: языки традиционной математической логики, языки сетевого типа, реляционные языки и языки типа PLANNER.

#### **6. Интеллектуальная подсистема САПР для принятия решений на трудноформализуемых этапах решения**

Одним из основных путей повышения эффективности производства и качества выпускаемой продукции является использование и постоянное развитие САПР, цель которых – симбиоз конструктора и ЭВМ в единую команду, способную приходить к поставленным целям в задачах проектирования более эффективно, чем каждый из них, работая в отдельности.

Традиционно процесс проектирования представляется в виде отдельных, функционально независимых этапов, взаимосвязь между которыми определяется конкретной технологией проектирования. В зависимости от характера принятия решения (ПР) этапы проектирования классифицируются на полностью детерминированные, алгоритмически-определенные и трудноформализуемые (т. е. с определенным, нечетким алгоритмом ПР). К последнему классу относятся этапы проектирования, для которых не существует четкой модели ПР, в явном виде устанавливающей взаимосвязь между значениями выходных и входных параметров процесса проектирования (в виде формул, систем уравнений или другим образом). Процесс автоматизации, как правило, охватывает этапы, относящиеся к первым двум классам, тогда как трудноформализуемые этапы проектирования традиционно остаются за конструктором – пользователем САПР.

Большую часть изделий машиностроения составляют типовые изделия, при проектировании которых у конструктора имеется определенный опыт. Поэтому одним из путей повышения эффективности САПР является включение в нее подсистемы, которая бы имитировала поведение конструктора-эксперта на трудноформализуемых этапах. Использование интеллектуальной подсистемы в САПР позволяет: уменьшая время ПР на трудноформализуемых этапах проектирования, ускорить проектирование типовых изделий в целом; улучшить

качество проектирования за счет накопленного опыта; уменьшить влияние субъекта (пользователя САПР) на процесс проектирования.

Работа интеллектуальной подсистемы осуществляется в режиме диалога «конструктор-ЭВМ» и функционально разделяется на два этапа: обучение и собственно работа подсистемы. На этапе обучения происходит ввод информации от эксперта о процессе проектирования (ППР) в типовых ситуациях. На этапе работы для вводимых значений входных параметров ППР определяется множество рекомендуемых значений выходного параметра, или список изделий, являющихся аналогами проектируемого изделия.

Можно выделить следующие составные части интеллектуальной подсистемы: информационную базу подсистемы, представляющую собой совокупность библиотечных наборов данных и содержащую информацию о ППР; диалоговый интерфейс в виде совокупности программ, обеспечивающих диалог между экспертом и подсистемой во время обучения и пользователем и подсистемой в процессе проектирования; программу выбора предпочтительных параметров, определяющую значения параметров на трудноформализуемых этапах проектирования; программу выбора аналогов проектируемых изделий.

# Глава 7. КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ПОДГОТОВКА ПРОИЗВОДСТВА В СИСТЕМАХ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

## Тема 21. Понятие о конструкторско-технологической подготовке

Производственный процесс представляет собой ряд процедур, между которыми существуют связи. Именно наличие связей свойств материалов и размерных связей обеспечивает качество изготавливаемой продукции. При этом можно выделить два вида связей: внутри технологической системы предприятия и связи «предприятие–внешняя среда» (рис. 7.1).



Рис. 7.1. Материальные и информационные связи в системе «Предприятие–внешняя среда»

Объекты производства связаны между собой связями свойств материалов, временными, размерными и информационными (рис. 7.2). Установление связей происходит на этапе технологической подготовки производства, которая, согласно ГОСТу Р 50995.3.1–96,

представляет собой «...вид производственной деятельности предприятия, обеспечивающий технологическую готовность производства к изготовлению изделий, отвечающих требованиям заказчика или рынка данного класса изделий».

Поскольку разработка технологии изготовления изделия связывает между собой его конструкцию и производственный процесс, целесообразно говорить о *конструкторско-технологической подготовке производства* (КТПП, production planning). КТПП изделия как проектная процедура должна решать задачи, направленные на обеспечение выпуска изделия установленного качества и количества.

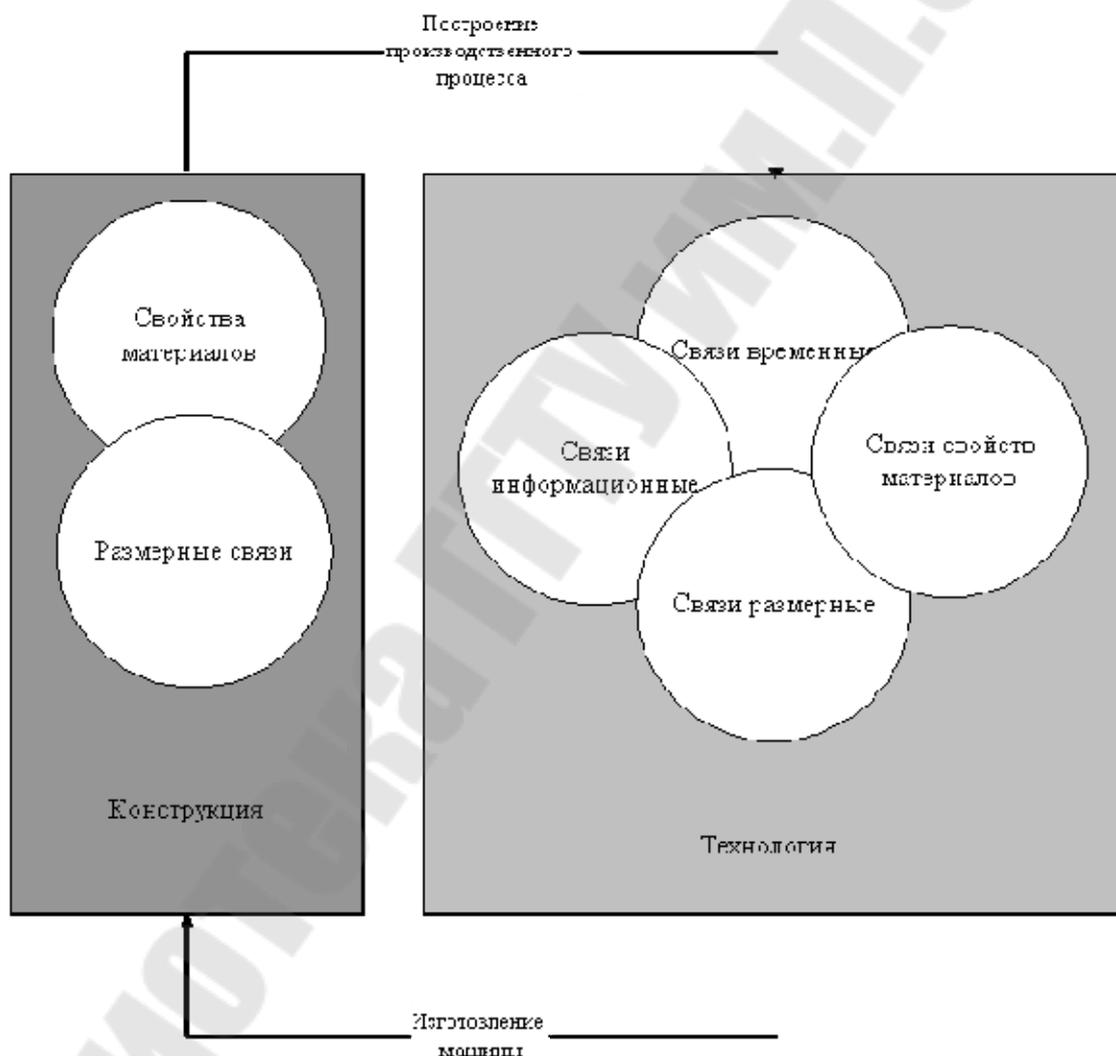


Рис. 7.2. Связи в производственном процессе

К данным задачам относятся: разработка конструкторской документации; разработка технологических процессов; проектирование средств технологического оснащения; временное планирование произ-

водственного процесса. Разработка ТП, включает в себя выбор исходных заготовок, выбор технологических баз, поиск типового техпроцесса, задание последовательности операций и их нормирование и т. д.

Для выполнения каждой из проектных процедур исполнитель (конструктор, технолог) привлекает значительный объем информации, которую можно подразделить на три вида (табл. 1): нормативная информация, общая для всей отрасли; информация, относящаяся к конкретному предприятию; информация, относящаяся к изделию, для которого выполняется КТПП.

Таблица 7.1

**Виды информации, используемой в ходе КТПП**

<b>Проектная процедура</b>	<b>1</b>	<b>2</b>	<b>3</b>
Разработка конструкторской документации	Справочники свойств материалов; требования ЕСКД; классификаторы продукции	Каталог конструкторских проектных решений	Технические требования к проектируемому изделию
Разработка техпроцессов и проектирование оснастки	Требования ЕСТД; каталоги стандартных оборудования и оснастки; нормы времени	Каталоги типовых техпроцессов; каталоги имеющихся оборудования и оснастки	Рабочие чертежи деталей
Маршрутизация	—	Каталоги имеющегося оборудования	Техпроцессы изготовления деталей
Календарное планирование	Нормы времени	Каталоги имеющихся складов для хранения заделов	Маршруты прохождения деталей по цехам

На рис. 7.3 отображена схема основных информационных потоков в ходе КТПП, наличие которых необходимо для выполнения соответствующих проектных процедур конструкторами и технологами. Главным фактором получения предприятием прибыли от производства изделия является уровень производственных затрат, в структуре которых затраты на материалы составляют до 70 %.

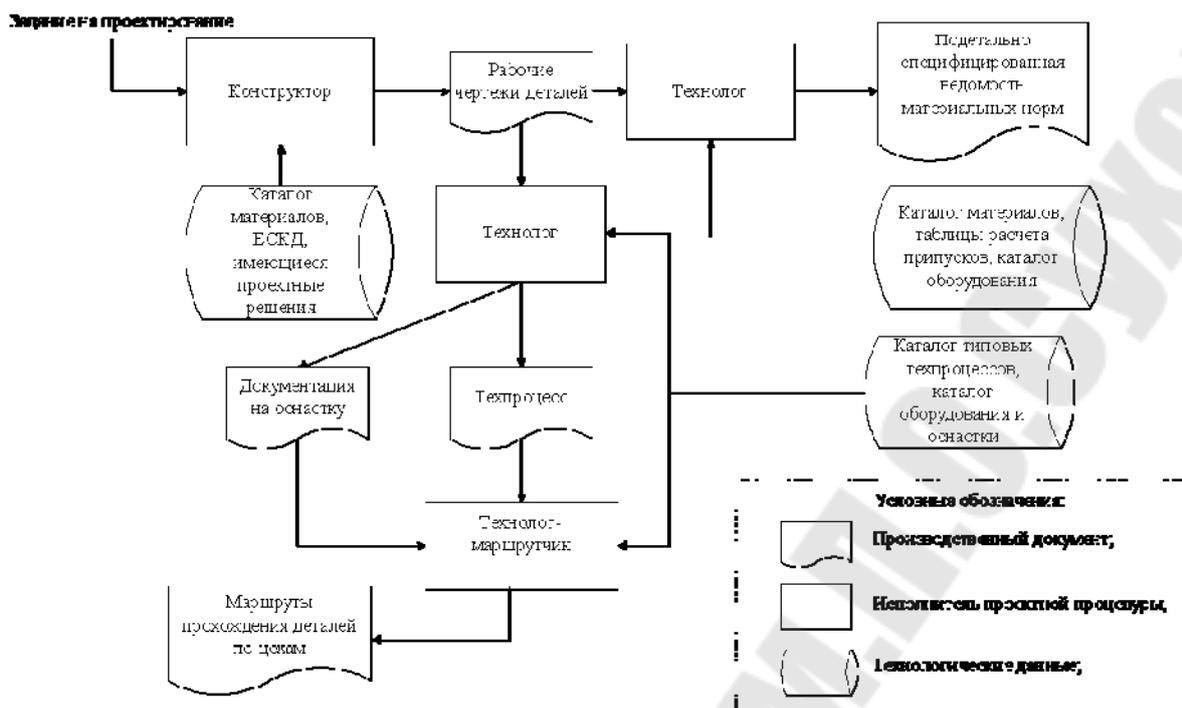


Рис. 7.3. Основные информационные потоки в процессе КТПП

Поэтому исключительно важное значение приобретает процедура материального нормирования, выполняемая всегда при проведении КТПП и от результатов выполнения которой зависит решение о целесообразности дальнейшей подготовки и самого производства изделия в имеющихся экономических условиях и о необходимости внесения изменений в принятые проектные решения.



Рис. 7.4. Заимствование проектных решений

Рассмотрим общую последовательность прохождения информационных потоков при КТПП. На входе мы имеем техническое задание (ТЗ) на проектирование. ТЗ делятся на два вида: ориентированные на модификацию ранее созданного проектного решения (рис. 7.4) и ориентированные на достижение заданных пользователем свойств, что предполагает создание новой конструкции. В первом случае задача значительно облегчается за счет заимствования конструкторских и технологических проектных решений из архива.

Конструктор создает новое изделие либо вносит изменения в модифицируемое. Затем в ряде случаев выполняется прочностной расчет методом конечных элементов. Часто в экспериментальном производстве изготавливается *прототип* изделия для испытаний.

Готовый комплект конструкторской документации поступает в бюро материальных норм предприятия, где выполняется процедура оценки себестоимости изготовления изделия и рассчитывается потребность в материалах с учетом всех технологических отходов и потерь.

В случае принятия решения о постановке изделия на производство КД передается в технологическую службу, где проводится разработка техпроцессов и их временное нормирование. По данным временного нормирования бюро маршрутизации назначает план-график движения детали по цехам. Кроме того, по заявкам технологов специальный конструкторский отдел выполняет проектирование приспособлений и оснастки, требуемых для выпуска изделия.

Для начала производства отдел снабжения закупает необходимые материалы и инструменты в рассчитанном количестве. В ходе производства решаются разнообразные задачи диспетчирования – оптимизация раскроя имеющихся материалов, оптимизация перемещения деталей и заготовок, обеспечение качества на этапе производства. Полученная продукция проходит испытания и отгружается потребителю.

Как видно, весь процесс КТПП довольно сложен и объемен. В то же время его нужно проводить в максимально сжатые сроки. Требование оперативности КТПП вызвано общей переориентацией промышленности на так называемое вариантное производство. При вариантном производстве каждое выпускаемое изделие изготавливается в соответствии с конкретными требованиями конкретного заказчика. Например, John Deere и Claas отказались от конвейерной сборки и перевели производство на стапели. Современные гибкие технологические комплексы позволяют добиться высокой экономической эффективности вариантного способа. В то же время фактически на каждое выпускаемое изделие нужен новый комплект конструкторской и технологической документации, который необходимо подготовить в крайне сжатые сроки. Это вступает в противоречие с двумя характеристиками процедур КТПП: во-первых, каждая процедура требует для своего выполнения поиска требуемых данных в каталогах, справочниках и т. д.; во-вторых, генерация окончательного проектного

решения является итерационным процессом, когда получение неудовлетворительных по каким-либо параметрам (себестоимость, технологичность) промежуточных проектных решений приводит к внесению изменений в проектные решения, принятых на более ранних стадиях подготовки производства.

Трудоемкость КТПП составляет 70 % в общей трудоемкости выпуска изделия. За последние 30 лет период нахождения изделия в производстве сократился в 2–3,5 раза, а срок подготовки производства увеличился вдвое и составляет от 0,5 до 2 лет. Это связано с постоянным усложнением изделий машиностроения, повышением требований к себестоимости, экономичности, безопасности. Поэтому большую актуальность приобретает вопрос быстрого и эффективного выполнения тех процедур КТПП, которые, во-первых, выполняются в обязательном порядке, а во-вторых, вносят максимальный вклад в себестоимость изделия. Сюда следует отнести конструкторское проектирование изделия (материалоемкость детали) и процедуру материального нормирования (принятие проектного решения, определяющего технологические отходы и потери и полную норму расхода материала на деталь). Конструкторы и технологи, выполняющие КТПП, должны иметь средства оперативного и точного выполнения проектных процедур, определения текущей материалоемкости изделия и влияния на нее принимаемых ими проектных решений. Инженеры должны иметь возможность оценивать проектные решения, исходя из критерия максимального повышения вероятности достижения основной цели производственного процесса – выпуска изделия в требуемом количестве в требуемые сроки с установленными показателями качества.

Оперативное и достоверное выполнение других проектных процедур в ходе КТПП требует перехода к новой информационной технологии обработки конструкторско-технологической информации.

**Информационная технология** – комплекс методов и процедур, реализующих функции сбора, передачи, обработки, хранения и доведения до пользователей информации.

Важная особенность информационной технологии – прямое привлечение знаний и производственного опыта конструкторов и технологов, заключающееся в рациональной организации взаимодействия человека с автоматизированной системой. Существуют несколько способов взаимодействия человека с САПР: передача задачи более опытному исполнителю; решение задачи совместными усилиями

ми; совершенствование знаний и умений (обучение партнера). Общая схема построения интегрированной модульной системы автоматизации конструкторско-технологических процедур представлена на рис. 7.5.

САПР ТП повышает производительность труда технологов в 2,5–3 раза, сокращает время на разработку маршрутных ТП в 5–6 раз, сокращает потребность в инженерах-технологах в 3 раза.

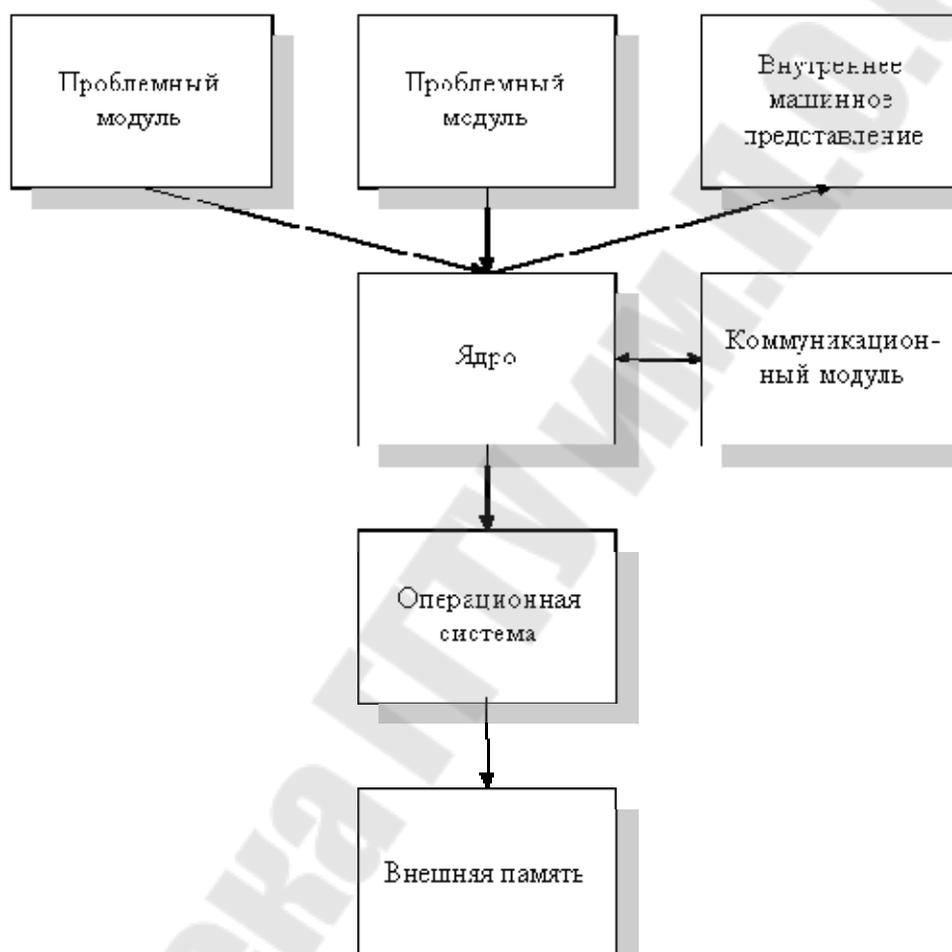


Рис. 7.5. Модуль интегрированной системы автоматизации КТПП

В табл. 7.2 приведена информация о пригодности различных видов системы автоматизации для решения основных задач КТПП. Как видно, все применяемые системы автоматизации можно разделить на три типа по их функциональному назначению:

1. Системы CAD/CAM – автоматизация конструкторского и технологического труда.
2. Системы конечноэлементного анализа (FEM) – автоматизация прочностных расчетов.
3. Специализированные программные средства.

Таблица 7.2

**Решение задач конструкторско-технологической подготовки производства с помощью средств информационного и программного обеспечения**

Задачи конструкторско-технологической подготовки производства		Средства		
		1	2	3
Конструкторские	Конструирование (объектов производства и технологического оснащения)	+	+	-
	Художественное конструирование	+	-	+
	Внесение изменений в конструкцию	+	-	+
	Контроль движений элементов конструкции	+	-	-
	Эмуляция сборки	+	-	-
	Оформление конструкторской документации	+	-	-
	Контроль конструкторской документации	+	-	-
	Конструкторские расчеты, в том числе:			
	расчеты размерных цепей	-	-	+
	прочностные, тепловые и т. п.	+	+	-
	Оптимизация конструкций	±	±	+
	Принятие решений	-	-	+
	Использование баз данных конструкций и их элементов	+	-	+
Технологические	Выбор заготовки	-	-	+
	Определение последовательности обработки	-	-	+
	Выбор технологических баз	-	-	+
	Расчет припусков	-	-	+
	Расчеты технологических размерных цепей	-	-	+
	Выбор режущих инструментов и определение их параметров	±	-	+
	Определение режимов обработки	-	-	+
	Оформление технологической документации	+	-	+
	Изменение технологической документации	+	-	+
	Подготовка управляющих программ для изготовления и контроля на оборудовании с ЧПУ	+	-	-
	Контроль обработки	+	-	+
	Выбор средств контроля и определение их параметров	-	-	+
	Принятие решений	-	-	+
	Использование баз данных производственной системы	+	-	-
	Нормирование технологического процесса	-	-	+

Задачи конструкторско-технологической подготовки производства		Средства		
		1	2	3
Организа- ционные	Организация взаимодействия специалистов	+	–	+
	Документооборот	+	–	+
	Защита информации	±	–	+
	Диспетчирование, планирование	–	–	+
<p><i>Примечание:</i>  1 – системы CAD/CAM;  2 – системы конечноэлементного анализа;  3 – специализированные программные средства;  «+» – достаточно полное решение задачи данными средствами;  «–» – задача данными средствами не решается;  «±» – задача данными средствами решается не в полной мере.</p>				

## Тема 22. Конструкторские базы и банки данных

После того как при помощи САПР получен комплект КД на изделие, сразу возникает вопрос ведения архива конструкторских проектных решений, их поиска и заимствования. Конструкторский банк данных – неотъемлемая часть интегрированной системы автоматизации КТПП. Поэтому следует рассмотреть вопросы проектирования конструкторских баз и банков данных.

В любой базе данных существуют три уровня: физический, логический и окончательного пользователя. Можно также выделить два варианта доступа к физическим и логическим ресурсам системы – локальный и телекоммуникационный (рис. 7.6).

Поддержание компактности, организации, структуры, актуальности и надежности данных входит в функции всех, кто работает с базой данных. Существуют различные стратегии поддержания такого состояния системы.

Как известно, в интерактивных САПР хранящиеся в базе данных объекты используются в качестве опытного образца разрабатываемого изделия. В силу этого:

1) доступ к таким объектам должен обеспечивать малое время реакции системы на запрос пользователя;

2) информационная модель должна развиваться и совершенствоваться в соответствии с нуждами всех работающих с СУБД пользователей: конструкторов, технологов, снабженцев и др.

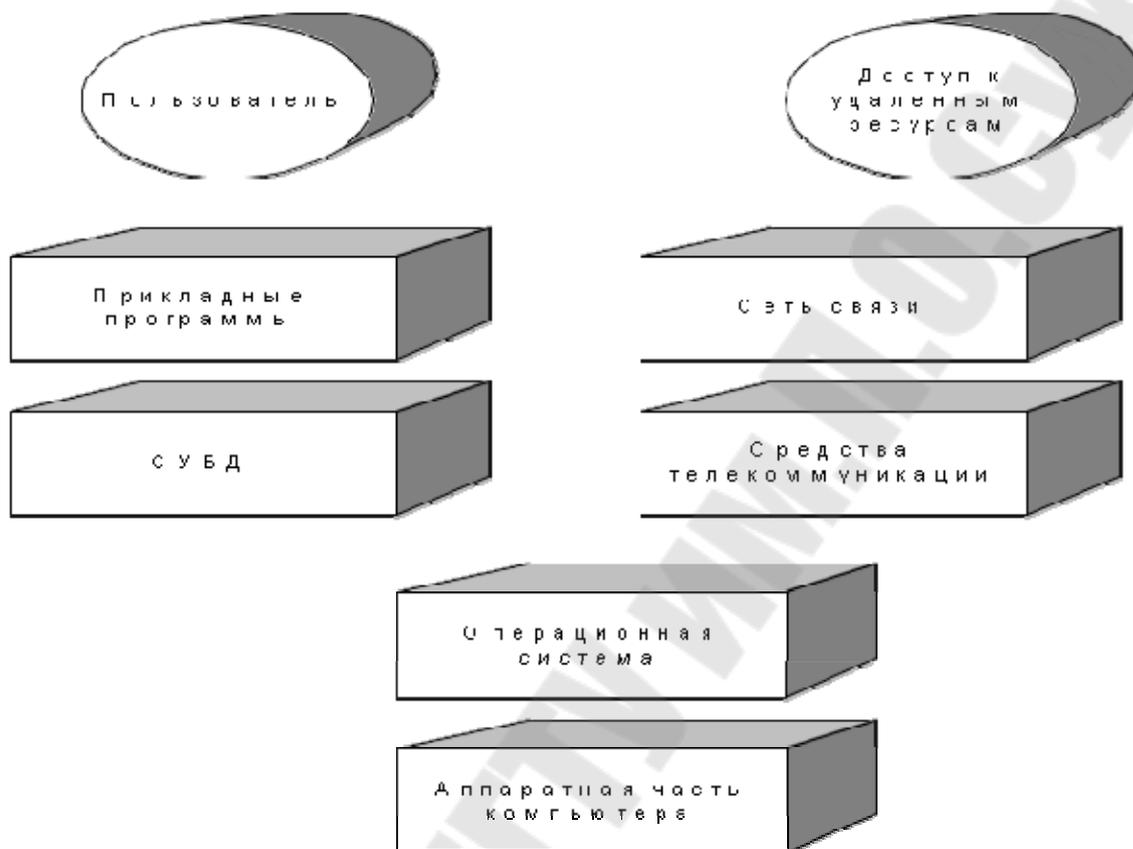


Рис. 7.6. Три уровня в базе данных

Все информационные объекты строятся на основе общей базы данных и служат источником информации, выбранной в соответствии с профессиональными интересами того или иного пользователя и преобразованной в форму, пригодную для решения его задач. Возможность работы с такими информационными объектами – основное требование к организации конструкторской базы данных.

Способы хранения данных, организация доступа к ним, алгоритмы изменения структур файлов и прочие технические подробности не должны отвлекать внимание пользователя. Более того, он может даже не подозревать об их существовании. Независимость данных упрощает их разделение. Она позволяет предоставлять одни и те же данные разным пользователям таким образом, что каждому из них будет казаться, что он работает с данными, организованными согласно его специфическим требованиям. При этом все необходимые преобразования данных выполняются средствами СУБД.

Основное требование к конструкторским базам данных состоит в том, чтобы пользователи и программисты могли разрабатывать прикладные программы, *не зависящие от хранящихся в базе данных сведений*: их номенклатуры, синтаксиса или формата, семантики, способа доступа к ним. Однако все перечисленные характеристики данных должны быть «известны» прикладной программе. Это возможно тогда, когда элемент данных может быть найден по имени, а его атрибуты имеются в словаре данных. Такие элементы данных называются *самоопределяющимися*.

В словаре данных хранятся сведения обо всех находящихся в базе элементах данных. Описания элементов включают в себя их имена, типы данных, длины и семантику. Помимо прочего, такие описания нужны для административных целей.

При проектировании архитектуры конструкторской базы данных приходится решать много сложных задач. К ним, в частности, относятся: определение требований к организации данных; определение и регулярная ревизия требований к обработке данных; разработка архитектуры системы на основе интеграции требований первых двух типов.

Требования к организации данных – объект особого внимания. При их определении следует учесть не только существующие в момент проектирования условия, но и их возможное изменение в связи с изменением условий обработки данных.

При разработке проекта конструкторской базы данных не следует забывать об основных правилах, справедливых при проектировании любой системы. Главное из них – система должна быть простой.

С ростом сложности системы растут затраты на ее разработку и обслуживание. Эти затраты могут свести на нет все преимущества, которые дает эксплуатация системы. Чтобы проектируемая конструкторская база данных не стала громоздкой, разработчик должен придерживаться определенной последовательности действий.

Практика показывает, что для успешного завершения любого проекта в коммерческой сфере необходимо следующее:

1. Определить основную цель проекта, достижение которой обеспечит компенсацию затрат и прибыль, достаточную для дальнейшего развития. Применительно к проектированию конструкторской базы данных основную цель можно сформулировать следующим образом: создать недорогой продукт, отвечающий требованиям рынка

и достаточно гибкий для того, чтобы приспосабливаться к изменению этих требований.

2. Выделить группу факторов, играющих определяющую роль в успехе проекта. В области сбыта таким фактором может быть, например, связь с потенциальными покупателями, побуждающая их приобрести именно ваш товар. В области разработок и исследований эти факторы определяют насколько система автоматизации увеличивает возможности пользователя поставлять на рынок новые конкурентоспособные товары повышенного качества.

3. Определить метод численной оценки достигнутых результатов, позволяющий вносить необходимые поправки. Для этой цели в системах управления используются системы обеспечения принятия решений. Эффективность использования таких систем многократно увеличивается, если они имеют доступ к правильно организованной базе данных.

Разработчик конструкторской базы данных должен использовать в своей работе не только системы обеспечения принятия решений, но, по возможности, экспертные системы и другие методы искусственного интеллекта. Это даст ему возможность анализировать факты, оценивать альтернативы, быстро находить и документировать правильные решения. Экспертные системы позволяют не только получить ответ на вопрос «А что будет, если...?», но и объяснение типа «Потому, что...», и даже рекомендации типа «В этом случае следует поступать так: ...».

В любой коммерческой деятельности, как финансовой, так и промышленной, можно выделить критические факторы – небольшое число аспектов этой деятельности, определяющие ее успех.

Если оценки критических факторов удовлетворительны, конкурентоспособность и успех коммерческой деятельности гарантированы.

Каждому из критических факторов соответствуют определенные задачи, и решению этих задач должно уделяться самое пристальное внимание. В сфере инженерной деятельности использование баз данных должно в первую очередь способствовать именно решению таких задач. Главная цель внедрения конструкторской базы данных – улучшение конечной продукции и повышение эффективности конструирования. При этом недопустимо усложнение рабочих операций до такой степени, когда по каждому пустяку приходится консультироваться со специалистами по информатике.

Рассмотрим теперь вопросы практической реализации конструкторского банка данных (рис. 7.7).

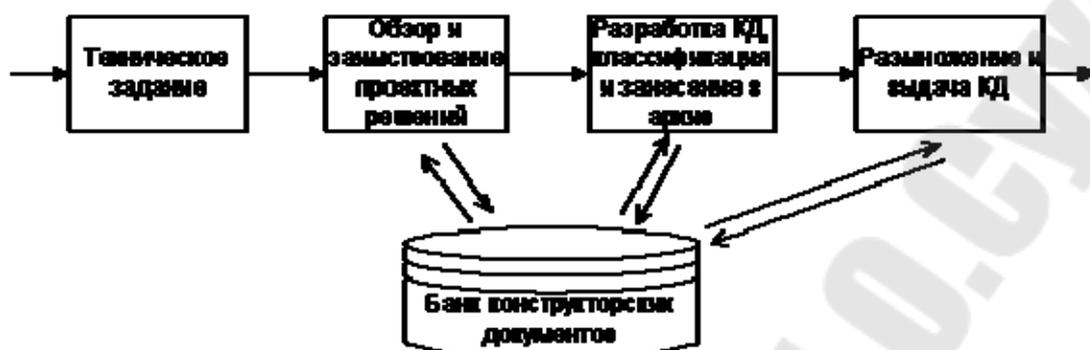


Рис. 7.7. Схема работы с банком конструкторских документов

Банк конструкторских документов должен представлять собой систему, содержащую КД, информацию о составе изделия и средства обеспечения доступа к файлам документов.

ГОСТ 2.101–68 устанавливает классификацию изделий по видам (рис. 7.8). Такой же классификации следует придерживаться и в конструкторском банке данных. ГОСТ 2.201–80 устанавливает единую классификационную систему обозначения изделий основного и вспомогательного производства и их КД для всех отраслей промышленности при разработке, изготовлении, эксплуатации и ремонте изделий. В соответствии с ним каждому изделию должно быть присвоено обозначение, которое является одновременно обозначением и его основного конструкторского документа (чертежа детали, спецификации).

Конструкторская документация – это графическая и текстовая информация. Она хранится в файлах различных форматов, поддерживаемых графическими и текстовыми редакторами. Доступ к этим файлам осуществляется через банк данных, содержащий иерархически упорядоченный (в соответствии с ГОСТ 2.101–68) перечень документации на изделие.

Наиболее эффективно рассмотреть банк данных в терминах объектно-ориентированного подхода. Информация о составе изделия хранится в спецификациях по ГОСТу 2.108–68, следовательно, основным объектом конструкторской базы данных будет *спецификация* (рис. 7.9).

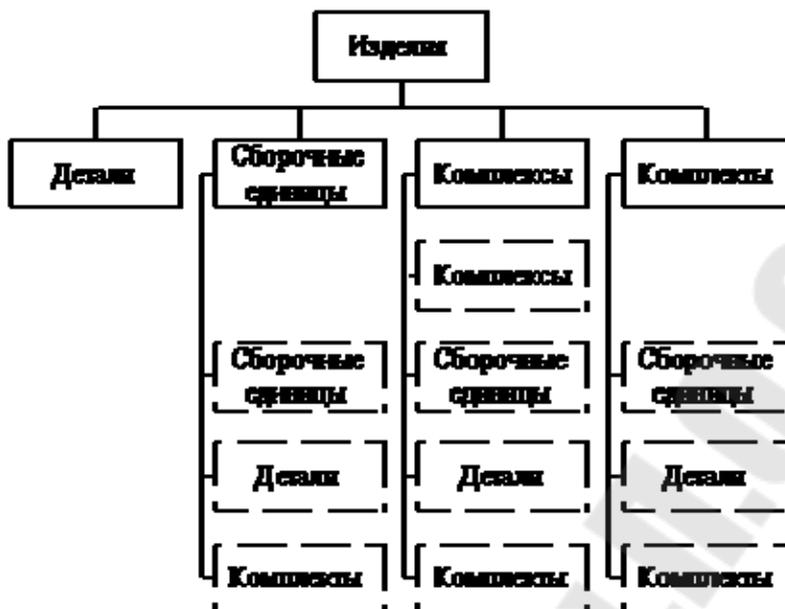


Рис. 7.8. Виды изделий

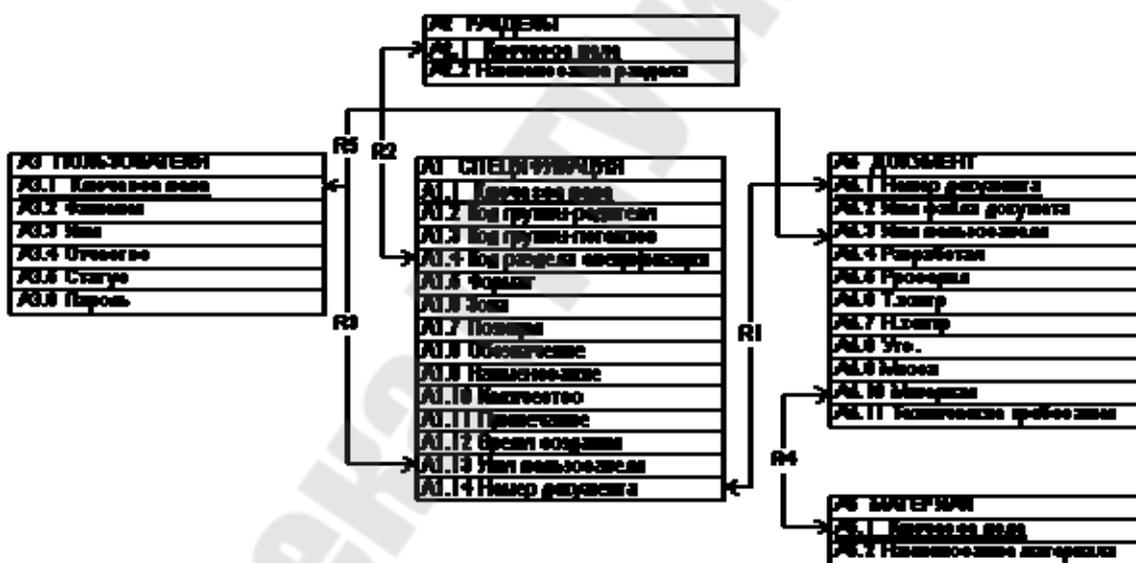


Рис. 7.9. Информационная модель банка конструкторских документов

Атрибуты А1.2 «Код группы-родителя» и А1.3 «Код группы-потомков» обеспечивают построение иерархической структуры каталога. Пунктам спецификации сопоставлен документ. Связь с объектом *документ* осуществляется через атрибут А1.14 «Номер документа». Наименования разделов хранятся в отдельном – объекте *разделы* А2. Для организации журнала работы и разграничения доступа, введен объект *пользователи* А3, содержащий информацию о пользова-

телях системы. Справочник материалов представлен в виде отдельного объекта *материал* A5.

Для разработки каталога используется активная реляционная БД. Операции работы с ней могут быть формализованы при помощи аппарата реляционной алгебры. Реляционная модель базы данных  $R$  может быть представлена в виде следующих отношений:

$$R = \begin{cases} R_1(\underline{A_{1.1}}, A_{1.2}, \dots, A_{1.4}) - \text{СПЕЦИФИКАЦИЯ} \\ R_2(\underline{A_{2.1}}, A_{2.2}) - \text{РАЗДЕЛЫ} \\ R_3(\underline{A_{3.1}}, A_{3.2}, \dots, A_{3.6}) - \text{ПОЛЬЗОВАТЕЛИ} \\ R_4(\underline{A_{4.1}}, A_{4.2}, \dots, A_{4.11}) - \text{ДОКУМЕНТ} \\ R_{51}(\underline{A_{5.1}}, A_{5.2}) - \text{МАТЕРИАЛ} \end{cases} \quad (7.1)$$

Для обеспечения информационной целостности данных на них наложены следующие ограничения:

1. Атрибут A1.4 «Код раздела спецификации» отношения  $R1$  может принимать только значения, содержащиеся в проекции атрибута A2.1 «Ключевое поле» отношения  $R2$  или, множество значений атрибута A1.4 «Код раздела спецификации» отношения  $R1$  содержится или совпадает с проекцией атрибута A2.1 «Ключевое поле» отношения  $R2$ .

$$\text{dom}(A1.4) \subseteq \pi(R2; 1). \quad (7.2)$$

A1.14 «Номер документа» может принимать только значения атрибутов A4.1 «Номер документа» отношения  $R4$  *документ*:

$$\text{dom}(A1.14) \subseteq \pi(R4; 1). \quad (7.3)$$

Атрибут A1.13 «Имя пользователя» может принимать только значения атрибутов A3.1 «Ключевое поле» отношения  $R3$  *пользователи*:

$$\text{dom}(A1.13) \subseteq \pi(R3; 1). \quad (7.4)$$

Атрибут A4.10 «Материал» отношения  $R4$  может принимать только значения, содержащиеся в проекции атрибута A5.1 «Ключевое поле» отношения  $R5$  *материал*:

$$\text{dom}(A4.10) \subseteq \pi(R5; 1). \quad (7.5)$$

Данные ограничения поддерживаются активной БД. Любое действие, направленное на их нарушение, будет отменено. Для получе-

ния информации об изделиях первой группы необходимо произвести выборку

$$\sigma(r_1; A1.2) = 0, \quad (7.6)$$

где A1.2 – «Код группы-родителя».

Для определения следующего уровня по отношению к текущему или вхождения в сборку необходимо над отношением  $r_1$  провести выборку по условию

$$\sigma(r_1; A1.2) = N, \quad (7.7)$$

где  $N$  – значение атрибута A1.3 «Код группы-потомков» текущего уровня.

Упорядочение производится по атрибутам: «Код группы-родителя» A1.2, «Код раздела спецификации» A1.4 и «Позиция» A1.7, что соответствует стандартным требованиям.

У одной группы может быть несколько родителей (рис. 7.10). Это позволяет при заимствовании сборочного чертежа не набирать заново всю сборку, а лишь указать обозначение сборочного чертежа. Рассмотрим операции над БД, производимые при выполнении конструктором ряда повседневных задач. При заполнении спецификации конструктор заносит в банк документов формат документа, наименование, позицию и количество деталей в сборке. Далее банк данных выполняет ряд операций, ранее проводимые конструктором: проверяет отсутствие в данном разделе детали с таким же обозначением; выясняет, свободна ли данная позиция, если условие не выполняется, то необходимо раздвинуть позиции; проверяет, есть ли в БД данная деталь или сборка с таким обозначением. Если деталь или сборка найдена ( $|T| > 0$ ), то нужно запомнить значения соответствующих атрибутов; собственно добавляет позицию в спецификацию. Аналогичные проверки проводятся и при редактировании пункта спецификации.

Рассмотрим вопрос эффективности внедрения описываемого электронного банка. При ведении бумажного архива конструкторской документации затраты на его ведение и поиск информации с увеличением количества чертежей будут постоянно возрастать (рис. 7.11, а). Внедрение такого архива приводит к повышению производительности труда конструкторов на 35–40 %.

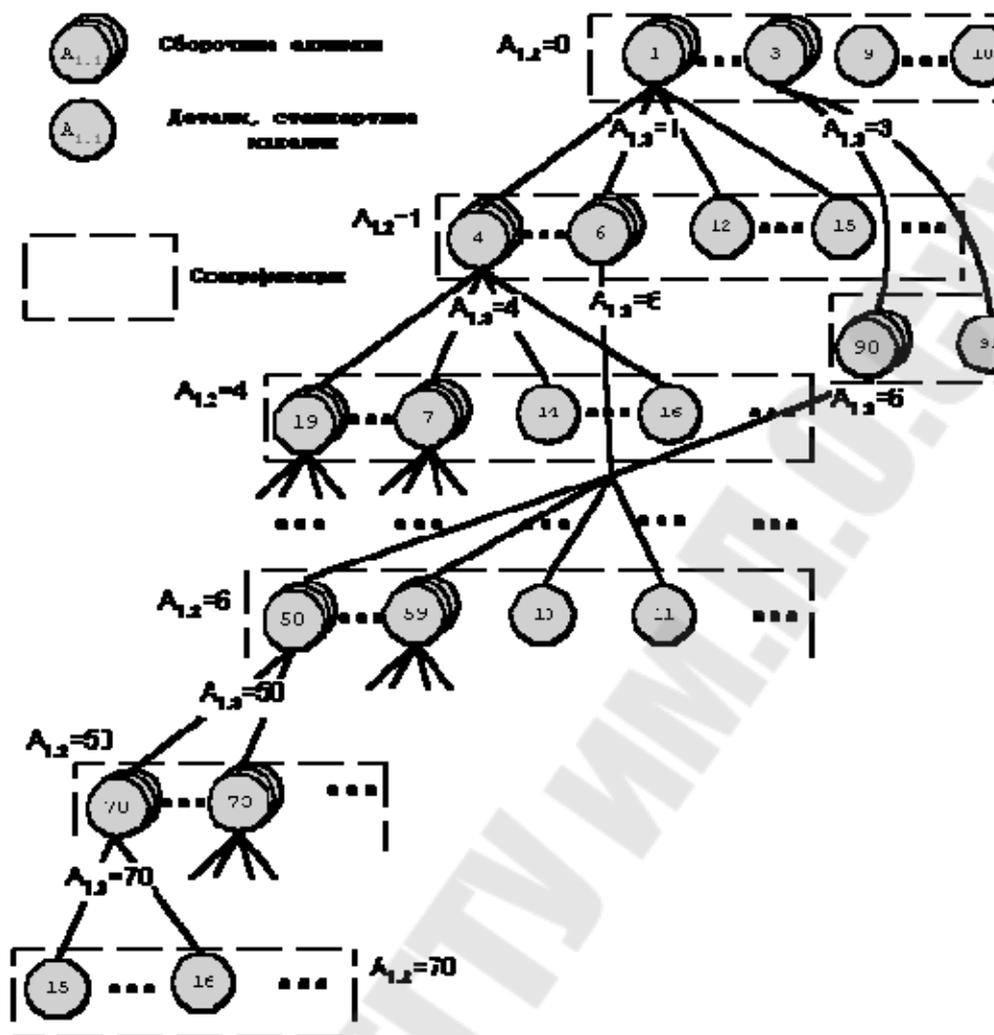


Рис. 7.10. Дерево каталога конструкторских документов

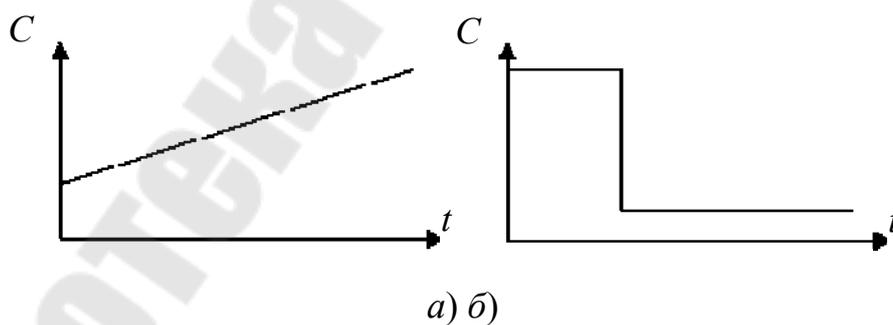


Рис. 7.11. Затраты труда на ведение архива: а – при ручном ведении архива; б – при внедрении электронного архива

При внедрении же электронного архива затраты возрастают на этапе его приобретения, внедрения и занесения документов в архив, после чего они резко падают и остаются практически постоянными (рис. 7.11, б).

## ЛИТЕРАТУРА

1. Красильникова, Г. Автоматизация инженерно-графических работ / Г. Красильникова, В. Самсонов, С. Тарелкин. – Санкт-Петербург, 2001. – 256 с. : ил.
2. Дементьев, Ю. В. САПР в автомобиле- и тракторостроении : учеб. для студентов высш. учеб. заведений / Ю. В. Дементьев, Ю. С. Щетинин. – Москва : Издат. центр «Академия», 2004. – 224 с.
3. Норенков, И. П. САПР. Принципы построения и структура / И. П. Норенков. – Минск : Выш. шк., 1987. – Кн. 1. – 1236 с.
4. Прохоров, А. Ф. Конструктор и ЭВМ / А. Ф. Прохоров. – Москва : Машиностроение, 1987. – 300 с.
5. Практическое руководство к лабораторным работам по курсу «САПР узлов и агрегатов машин» для студентов специальности Т.05.09.00 «Тракторы и сельскохозяйственные машины» / авт.-сост. В. Б. Попов. – Гомель : ГГТУ им. П.О. Сухого, 2004. – 55 с.
6. САПР узлов и агрегатов машин : практ. рук. по выполнению контрол. работ для студентов специальности 1-36 12 01 «Проектирование и производство сельскохозяйственной техники» / авт.-сост. В. Б. Попов. – Гомель : ГГТУ им. П. О. Сухого, 2005. – 63 с.
7. Хокс, Б. Автоматизированное проектирование и производство : пер. с англ. / Б. Хокс. – Москва : Мир, 1991. – 296 с. : ил.
8. САПР технологических процессов, приспособлений и режущих инструментов : учеб. пособие для вузов / В. И. Аверченко [и др.]. – Минск : Выш. шк. 1993. – 228 с. : ил.
9. Аветисян, Д. А. Системы автоматизированного проектирования, типовые элементы, методы и процессы / Д. А. Аветисян. – Москва : Изд-во стандартов, 1985. – 400 с.
10. Босой, Е. С. Теория, конструкции и расчет сельскохозяйственных машин / Е. С. Босой. – Москва : Машиностроение, 1978. – 560 с.
11. Тракторы. Проектирование, конструирование и расчет / И. П. Ксенович [и др.]. – Москва : Машиностроение, 1991. – 544 с.
12. Информатика : учебник / под ред. проф. Н. В. Макаровой. – Москва : Финансы и статистика, 1997. – 768 с. : ил.
13. Половинкин, А. И. Основы инженерного творчества / А. И. Половинкин. – Москва : Машиностроение, 1988. – 368 с.
14. AutoCAD 14 / Э. Т. Романычева [и др.]. – Москва : Радио и связь, 1997. – 480 с. : ил.
15. Энкарначчо, Ж. Автоматизированное проектирование. Основные понятия и архитектура систем / Ж. Энкарначчо, Э. Шлехтендаль. – Москва : Радио и связь, 1986. – 288 с.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
Глава 1. ОСНОВЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ.....	5
Тема 1. Системный подход к проектированию.....	5
Тема 2. Структура процесса проектирования .....	11
Тема 3. Системы автоматизированного проектирования и их место среди других автоматизированных систем .....	20
Глава 2. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ.....	35
Тема 4. Структура технического обеспечения систем автоматизированного проектирования.....	35
Тема 5. Аппаратура рабочих мест в автоматизированных системах проектирования и управления.....	42
Тема 6. Методы доступа в локальных вычислительных сетях.....	47
Тема 7. Локальные вычислительные сети Ethernet .....	48
Тема 8. Сети кольцевой топологии.....	53
Глава 3. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ АНАЛИЗА ПРОЕКТНЫХ РЕШЕНИЙ.....	57
Тема 9. Компоненты математического обеспечения.....	57
Тема 10. Математические модели анализа на макроуровне.....	60
Тема 11. Методы и алгоритмы анализа на макроуровне .....	75
Тема 12. Математическое обеспечение анализа на микроуровне.....	90
Глава 4. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИНТЕЗА ПРОЕКТНЫХ РЕШЕНИЙ.....	97
Тема 13. Постановка задач параметрического синтеза.....	97
Тема 14. Методы оптимизации .....	102
Тема 15. Постановка задач структурного синтеза .....	119
Глава 5. ДРУГИЕ ВИДЫ ОБЕСПЕЧЕНИЙ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ.....	128
Тема 16. Программное обеспечение систем автоматизированного проектирования.....	128
Тема 17. Организационное и методическое обеспечение систем автоматизированного проектирования.....	135
Глава 6. АВТОМАТИЗАЦИЯ КОНСТРУИРОВАНИЯ.....	141
Тема 18. Автоматизация конструирования машин на основе геометрического моделирования .....	141
Тема 19. Современные программные САД/САМ продукты для проектирования изделий машиностроения.....	154
Тема 20. Интеллектуальные подсистемы систем автоматизированного проектирования.....	158

Глава 7. КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ПОДГОТОВКА ПРОИЗВОДСТВА В СИСТЕМАХ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ .....	168
Тема 21. Понятие о конструкторско-технологической подготовке .....	168
Тема 22. Конструкторские базы и банки данных.....	176

Учебное электронное издание комбинированного распространения

Учебное издание

**Попов Виктор Борисович**

**СИСТЕМЫ АВТОМАТИЗИРОВАННОГО  
ПРОЕКТИРОВАНИЯ УЗЛОВ  
И АГРЕГАТОВ МАШИН**

**КУРС ЛЕКЦИЙ**

**для студентов специальности 1-36 12 01**

**«Проектирование и производство  
сельскохозяйственной техники»**

**Электронный аналог печатного издания**

Редактор

*В. В. Вороник*

Компьютерная верстка

*Е. Н. Герасименко*

Подписано в печать 29.09.2010.

Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».

Ризография. Усл. печ. л. 10,93. Уч.-изд. л. 10,52.

Изд. № 270.

E-mail: [ic@gstu.by](mailto:ic@gstu.by)

<http://www.gstu.by>

Издатель и полиграфическое исполнение:

Издательский центр учреждения образования

«Гомельский государственный технический университет

имени П. О. Сухого».

ЛИ № 02330/0549424 от 08.04.2009 г.

246746, г. Гомель, пр. Октября, 48.