

Содержание

Введение

Цели создания САПР

Компоненты САПР.

Классификация САПР

Разработка САПР

Три уровня САПР

Модульный принцип построения САПР

Системы автоматического проектирования для машиностроительного комплекса.

Понятие CAD, CAM, CAE

Проектирование. Методы проектирования.

методы проектирования в САПР

Представление графической информации в ЭВМ

Задачи синтеза и анализа. Оптимальное проектирование конструкций

Методы решения задач оптимизации

Математические модели объектов проектирования

Классификация математических моделей

Стандарты обмена данными между системами

Общие сведения

Банки данных

Введение

САПР — это автоматизированные системы, которые призваны реализовывать ту или иную информационную технологию осуществления проектирования. На практике они представляют собой технические системы, позволяющие, таким образом, автоматизировать, обеспечить независимое от человека функционирование процессов, составляющих разработку проектов. В зависимости от контекста под САПР может пониматься:

- программное обеспечение, используемое в качестве основного элемента соответствующей инфраструктуры;
- совокупность кадровых и технических систем (включая те, что предполагают задействование САПР в виде ПО), применяемых предприятием в целях автоматизации разработки проектов.

Таким образом, можно выделить более широкую и узкую трактовку термина, о котором идет речь. К примеру, в контексте отдельно взятого производственного цеха под САПР, вероятно, будет пониматься конкретная программа для автоматизированного проектирования. Если речь идет о стратегическом планировании развития предприятия, данное понятие будет, вероятно, соответствовать более масштабной инфраструктуре, задействуемой в целях повышения эффективности разработки различных проектов.

САПР — это аббревиатура, которая и расшифровываться может по-разному. В общем случае она соответствует словосочетанию «система автоматизированного проектирования». Вместе с тем есть и другие варианты расшифровки соответствующей аббревиатуры. Например, она может звучать как «система автоматизации проектных работ».

Цели создания САПР

Главная цель разработки САПР — повышение эффективности труда специалистов предприятия, решающих различные производственные задачи. В частности, связанные с инженерным проектированием. Повышение эффективности в данном случае может осуществляться за счет:

- снижения трудоемкости процесса проектирования на производстве;
- сокращения сроков реализации проектов;
- снижения себестоимости проектных работ, а также издержек, связанных с эксплуатацией;
- обеспечения повышения качества инфраструктуры проектирования;
- снижения издержек на моделирование, а также проведение испытаний.

САПР — это инструмент, позволяющий добиться отмеченных преимуществ за счет:

- автоматизации документации;
- эффективной информационной поддержки специалистов, участвующих в разработке проектов;
- применения концепций параллельного проектирования;
- унификации различных решений;
- осуществления стратегического проектирования;
- применения математического моделирования как альтернативы дорогостоящим испытаниям;
- повышения качества процессов управления бизнесом;
- оптимизации методов проектирования.

Компоненты САПР.

Рассмотрим структуру САПР на примере системы автоматизированного проектирования технологических процессов, эта система, может включать следующие компоненты:

- комплекс элементов автоматизации;
- программно-техническую инфраструктуру;
- методические инструменты;
- элементы поддержки функциональности САПР.

Распространен подход, в соответствии с которым в структуре САПР следует выделять различные подсистемы. Ключевыми принято считать:

- обслуживающие подсистемы, которые поддерживают функционирование основных проектирующих компонентов САПР, инфраструктуры, отвечающей за обработку данных, поддержание ПО;
- проектирующие подсистемы, которые в зависимости от соотношения с объектом разработки могут быть представлены с объектными задачами или же инвариантными, то есть связанные с реализацией конкретных проектов или же с совокупностью нескольких.

САПР — это системы, которые включают в себя определенные функциональные компоненты. Рассмотрим их особенности.

Автоматизированное проектирование систем управления и промышленной инфраструктуры, как мы уже знаем, состоит из различных подсистем. В свою очередь, их составляющими являются компоненты, которые обеспечивают функционирование соответствующих элементов САПР. Например, это может быть та или иная программа, файл, аппаратное обеспечение. Компоненты, обладающие общими признаками, формируют средства обеспечения систем проектирования. Таковые могут быть представлены следующими основными разновидностями:

- техническим обеспечением, которое представляет собой совокупность различных технических средств, таких как компьютеры, сетевые компоненты, измерительные приборы;
- математическими моделями, которые объединяют те или иные алгоритмы, что задействуются в целях решения различных задач;
- программным обеспечением — системным, прикладным;
- информационным обеспечением, представляющим собой совокупность различных данных, что необходимы в целях внедрения проектирования;
- лингвистическими моделями, представляющими собой совокупность различных языков, которые применяются в САПР в целях отражения сведений о проектировании;
- методическим обеспечением, представляющим собой совокупность подходов к обеспечению функционирования САПР, различных методов подбора технологических концепций для достижения оптимальных результатов при реализации тех или иных проектов;
- организационным обеспечением, которое представлено главным образом источниками, которые определяют структуру проектной документации, а также характеристики системы автоматизации и то, каким образом должны отражаться результаты реализации проектов.

Автоматизированные системы проектирования, обработки информации могут быть классифицированы по различным критериям. Рассмотрим их специфику.

Классификация САПР

В числе распространенных критериев классификации САПР — отраслевое назначение соответствующих систем. Так, выделяют:

- автоматизированное проектирование инфраструктуры машиностроения;
- САПР для электронного оборудования;
- автоматизированное проектирование в сфере строительства.

Первый тип систем САПР используется в широком спектре отраслей — в автомобилестроении, авиастроении, судостроении, в сегментах выпуска различных товаров народного потребления. Соответствующая инфраструктура применяется в целях разработки, как отдельных деталей, так и различных механизмов с использованием всевозможных подходов в рамках параметрического проектирования, моделирования.

Второй тип САПР применяется для проектирования готового электронного оборудования, а также отдельных его элементов, например

процессоров, интегральных микросхем и прочих видов аппаратного обеспечения.

Третий тип САПР задействуется в целях проектирования различных зданий, сооружений, элементов инфраструктуры.

Следующий критерий, по которому могут быть классифицированы системы автоматизированного проектирования, программирования, — целевое назначение САПР. Так, выделяют:

- средства проектирования, задействуемые в целях автоматизации двумерных либо трехмерных геометрических моделей, формирования конструкторской документации;
- системы, применяемые в целях разработки различных чертежей;
- САПР, созданные для геометрического моделирования;
- системы, предназначенные для автоматизации расчетов в рамках инженерных проектов, а также динамического моделирования;
- САПР, предназначенные для осуществления компьютерного анализа различных параметров по проектам;
- средства автоматизации, используемые в целях технологической оптимизации проектов;
- САПР, используемые в целях автоматизации планирования.

Стоит отметить, что данную классификацию следует считать условной.

Автоматизированная система технологического проектирования может включать в себя самый широкий спектр функций. Конкретный перечень возможностей САПР определяет, прежде всего, разработчик соответствующей системы. Рассмотрим, какие в принципе задачи он может решать.

Разработка САПР

Проектирование автоматизированных систем обработки информации, управления, программирования и реализации иных функций, направленных на повышение эффективности разработки проектов в тех или иных отраслях, — процесс, который характеризуется высоким уровнем сложности и требует от его участников осуществления вложения значительных ресурсов — трудовых, финансовых. Выделяют несколько основных принципов, в соответствии с которыми может вестись разработка САПР. В числе таковых:

- унификация;
- комплексность;
- открытость;
- интерактивность.

Рассмотрим их подробнее.

Унификация как принцип разработки САПР

Работа с системами автоматизированного проектирования как на стадии их разработки, так и в период пользования соответствующей инфраструктурой предполагает следование принципу унификации, в соответствии с которым, те или иные решения могут одинаково эффективно и по схожим алгоритмам внедряться в различных отраслях производства. Данный принцип предполагает, что человек, использующий знакомый ему модуль САПР или, к примеру, методику автоматизированного проектирования в одной среде, без труда сможет адаптировать их к специфике применения в иных условиях.

Унификация САПР имеет значение и с точки зрения развития предприятия - разработчика соответствующей системы: чем более универсальными будут модули и подходы, которые данный хозяйствующий субъект предлагает рынку, тем более интенсивным может быть его рост, тем выше конкурентоспособность и готовность новых потребителей к сотрудничеству.

Следующий принцип, который характеризует процесс проектирования автоматизированных систем, — комплексность. Он предполагает, что производитель САПР сможет наделить свой продукт компонентами, которые позволят его пользователю решать поставленные задачи на самых разных уровнях реализации проекта. Данный аспект, возможно, является ключевым с точки зрения обеспечения конкурентоспособности продукта и освоения им новых рынков. Но при этом следует иметь в виду, что даже самые комплексные решения должны удовлетворять иным ключевым принципам разработки САПР.

Открытость в данном контексте может пониматься по-разному, но во всех случаях ее интерпретация будет уместной. Разработка системы автоматизированного проектирования — процесс, который должен прежде всего характеризоваться открытостью с точки зрения формирования обратной связи между производителем САПР и ее пользователями. Человек, задействующий соответствующую систему, должен иметь возможность информировать ее разработчика о выявленных проблемах, особенностях функционирования САПР в различных условиях, передавать бренду-производителю свои пожелания касательно улучшения продукта.

Открытость в разработке САПР также может выражаться в готовности производителя осуществлять активный мониторинг технологических разработок, в том числе от конкурирующих производителей, отслеживать различные тренды. В данном случае ведущую роль в бизнесе могут играть не

только технологические подразделения, но, к примеру, маркетологи компании, специалисты по PR, менеджеры, отвечающие за переговоры фирмы с партнерами.

Открытость при разработке САПР — это также готовность разработчика соответствующей системы к прямому диалогу с другими поставщиками, которые опять же могут быть его прямыми конкурентами. Обмен технологиями, позволяющими создавать продукты, посредством которых может быть осуществлено эффективное автоматизированное проектирование систем управления, промышленной инфраструктуры, инженерных разработок, также является значимым фактором повышения конкурентоспособности бренда, поставляющего САПР в тех или иных сегментах рынка.

Следующий важнейший принцип создания САПР — интерактивность. Он предполагает, прежде всего создание разработчиком соответствующих систем интерфейсов, максимально облегчающих процедуру их задействования человеком, а также осуществления им необходимых коммуникаций с другими пользователями САПР.

Еще один аспект интерактивности — обеспечение в необходимых случаях взаимодействия между различными модулями систем автоматизированного проектирования в рамках формирования производственной инфраструктуры.

Можно отметить, что принцип интерактивности тесно связан с первым — унификацией. Дело в том, что обмен данными в рамках тех или иных интерактивных процедур наиболее эффективным будет при условии необходимой стандартизации взаимодействия между теми или иными субъектами. Это может выражаться в унификации файловых форматов, документов, процедур, языка, инженерных подходов при разработке тех или иных проектов.

Особенно большое значение рассматриваемый принцип играет в САПР, посредством которых осуществляется автоматизированное проектирование информационных систем. Данная сфера применения САПР характеризуется, в частности, высокой степенью потребности пользователей соответствующей инфраструктуры:

- в регулярном, динамичном взаимодействии между собой;
- обеспечении связей между большим количеством модулей САПР;
- осуществлении оптимизации различных интерактивных процедур;
- оперативном формировании отчетности.

Только при условии достаточной интерактивности систем автоматизированного проектирования пользователи вправе рассчитывать на эффективное решение подобных производственных задач.

Три уровня САПР

Одна из важнейших задач современных САПР — избавить инженера от рутинной работы, предоставить ему возможность заниматься творческими процессами. Сейчас заводы используют большую номенклатуру САПР: от небольших графических программ до мощных специализированных пакетов. Их стоимость колеблется от ста до нескольких десятков тысяч долларов за одно рабочее место. В зависимости от возможностей, а соответственно и стоимости, современные САПР можно условно разбить на три уровня: нижний, средний и высший. Деление на уровни в специальной литературе производят либо по их возможностям, либо по стоимости, например: до 1000 долл., от 1000 до 10 000 долл., свыше 10 000 долл.

При разделении по возможностям предполагается, что системы нижнего уровня (например, AutoCAD, VersaCAD, CADKEY) обеспечивают выполнение чертежных работ.

Системы среднего уровня (например, T-FLEX CAD, SolidEdge) сокращают сроки выпуска документации и время разработки проектов за счет автоматизации выпуска конструкторской и технологической документации, программирования, обработки заготовок на станках с ЧПУ. Эти системы позволяют создать объемную модель изделия, по которой определяются инерционно-массовые, прочностные и иные характеристики, контролируется взаимное расположение деталей, моделируются все виды ЧПУ-обработки, отрабатывается внешний вид по фотореалистичным изображениям, выпускается документация. Кроме того, обеспечивается управление проектами на базе электронного документооборота. Экономический эффект состоит в многократном повышении производительности труда при резком сокращении ошибок и соответственно в улучшении качества изделий.

Системы высшего уровня (EDS Unigraphics, Pro/Engineer, CATIA или CADDs) обеспечивают интеграцию всего цикла создания изделия от проектирования, подготовки к производству до изготовления. Они позволяют конструировать детали с учетом особенностей материала (пластмасса, металлический лист), проводить динамический анализ сборки с имитацией сборочных приспособлений и инструмента, проектировать оснастку с

моделированием процессов изготовления (штамповки, литья, гибки), что исключает брак в оснастке и делает ненужным изготовление натурных макетов, то есть значительно уменьшаются затраты и время на подготовку к производству изделия. Программы математического анализа таких САПР могут включать прочностной, кинематический и динамический анализ. Моделирование механообработки позволяет оценить качество деталей с точки зрения их деформации. Экономический эффект подобных систем зависит от размера зарплаты конструктора или технолога и навыков использования САПР.

Модульный принцип построения САПР

Значительная часть современных САПР состоит из нескольких модулей: сборки, механообработки, управления инженерными данными и т.п. Их объединяет общая методология и инструментальные средства. Высокая степень интеграции модулей конструирования и технологической подготовки производства обеспечивает преобразование графической информации об объекте в табличную, используемую при технологическом проектировании. Геометрические объекты, имеющие математическое описание, аппроксимируются с заданной точностью кривыми и поверхностями.

Объединение модулей конструкторских и технологических разработок в единую САПР снижает стоимость и уменьшает время выхода новой продукции на рынок, позволяет конструировать детали с учетом их технологичности и используемого материала (пластмасса, металлический лист).

Модульная архитектура САПР облегчает расширение системы и адаптирование ее в соответствии с требованиями пользователя, позволяет приобретать только необходимые компоненты. Многие САПР снабжены контекстно-зависимыми справочниками и собственными базами данных или предлагают интерфейс с существующими базами данных.

Системы автоматического проектирования для машиностроительного комплекса.

Понятие CAD, CAM, CAE

Для современного производства характерно использование компьютерной техники на всех уровнях. Это осуществляется на базе так называемых CAD/CAM/CAE систем.

CAD (Computer Aided Design) - компьютерная помощь проектированию. Самая основная функция CAD - определение геометрии конструкции,

поскольку геометрия определяет все последующие этапы жизненного цикла продукта и является основой для последующего использования в системах САМ и САЕ.

САМ (Computer Aided Manufacturing) - компьютерная помощь изготовлению. Системы САМ используется для планирования, управления и контроля операций производства. Наиболее важным качеством этих системы является способность автоматизировать расчеты траекторий перемещения инструмента для обработки на станках с ЧПУ.

САЕ (Computer Aided Engineering) - компьютерная помощь инженерии. САЕ - это технология для анализа геометрии САД, моделирования и изучения поведения продукта, решения различных инженерных задач (расчет конструктивной прочности, нагрузки, напряжения, деформации, анализ тепловых процессов, расчет гидравлических систем и механизмов и др.).

САД/САМ/САЕ системы условно делятся на три категории:

Системы нижнего уровня (легкие системы) предназначены в основном для выполнения чертежей, автоматизации выпуска конструкторской и технологической документации, двухмерного моделирования и трёхмерных построений. Системы просты в использовании, содержат множество библиотек стандартных элементов, поддерживают различные стандарты оформления графической документации.

Использование этих систем позволяет сократить время разработки проекта, но не гарантируют проектировщиков от ошибок даже при полном соответствии документации ЕСКД и ЕСТД.

Системы среднего уровня предназначены для создания объемной модели изделия, проектирования сборочных единиц среднего уровня сложности, подготовки управляющих программ для оборудования с ЧПУ, выпуска конструкторско-технологической документации, определения инерционно-массовых, прочностных и прочих характеристик.

Системы позволяют сократить время на разработку проектов, на доводку опытных образцов изделий в результате исключения ошибок при проектировании. С их помощью можно решать большинство типичных машиностроительных задач, не привлекая мощные и дорогие системы тяжёлого класса.

Системы высшего уровня (тяжелый класс), кроме перечисленных выше систем низшего и среднего уровней, дают возможность: моделировать работу механизмов и все виды ЧПУ-обработки, конструировать детали с учетом особенностей материала, проводить динамический анализ сборки,

проектировать оснастку с моделированием процессов изготовления (штамповки, литья, гибки).

Системы высшего уровня позволяют значительно сократить время разработки и изготовления изделия.

Проектирование. Методы проектирования.

Основные понятия.

Проектирование - это комплекс работ с целью получения описаний нового или модернизируемого технического объекта, достаточных для реализации или изготовления объекта в заданных условиях. В процессе проектирования возникает необходимость создания описания, необходимого для построения еще не существующего объекта. Получаемые при проектировании описания бывают окончательными или промежуточными. Окончательные описания представляют собой комплект конструкторско-технологической документации в виде чертежей, спецификаций, программ для ЭВМ и автоматизированных комплексов и т.д.

Процесс проектирования, осуществляемый полностью человеком, называют неавтоматизированным.

Автоматизированное проектирование – проектирование, при котором происходит взаимодействие человека и ЭВМ.

Представления о сложных технических объектах в процессе их проектирования разделяются на *аспекты* и *иерархические уровни*. *Аспекты* характеризуют ту или иную группу родственных свойств объекта. Типичными аспектами в описаниях технических объектов являются: функциональный, конструкторский и технологический.

Функциональный аспект отражает физические и информационные процессы, протекающие в объекте при его функционировании.

Технологический аспект определяет технологичность, возможности и способы изготовления объекта в заданных условиях.

Конструкторский аспект характеризует структуру, расположение в пространстве и форму составных частей объекта.

Разделение описаний проектируемых объектов на иерархические уровни по степени подробности отражения свойств объектов составляет сущность блочно-иерархического подхода к проектированию.

Типичными иерархическими уровнями функционального проектирования являются: функционально-логический (функциональные и логические схемы); схемотехнический (электрические схемы узлов и отдельных блоков); компонентный (проектирование элементов и их размещение).

Проектирование делится на стадии, этапы и процедуры. Выделяют стадии научно-исследовательских работ (НИР), опытно-конструкторских работ (ОКР),

эскизного проекта, технического проекта, рабочего проекта, испытаний опытного образца.

Проектное решение - описание объекта или его части, достаточное для принятия заключения об окончании проектирования или путях его продолжения.

Маршрутом проектирования называется последовательность проектных процедур, ведущая к получению требуемых проектных решений.

Проектная процедура - часть проектирования, заканчивающаяся получением проектного решения.

Проектные процедуры делятся на процедуры синтеза и анализа. Процедура синтеза заключается в создании описаний проектируемого объекта. В описаниях отображаются структура и параметры объекта (т.е. осуществляется структурный и параметрический синтез). Процедура анализа - исследование объекта. Собственно, задача анализа формулируется как задача установления соответствия двух различных описаний одного и того же объекта.

Процесс проектирования на предприятиях начинается с получения ТЗ (технического задания). В ТЗ содержатся основные сведения об объекте проектирования, условиях его эксплуатации, а также требования, предъявляемые заказчиком к проектируемому изделию. Важнейшее требование к ТЗ - это его полнота. Выполнение этого требования определяет сроки и качество проектирования.

Следующий этап - предварительное проектирование - связан с поиском принципиальных возможностей построения системы, исследованием новых принципов, структур, обоснованием наиболее общих решений. Результатом этого этапа является техническое предложение.

На этапе эскизного проектирования производится детальная проработка возможности построения системы, его результатом является эскизный проект.

На этапе технического проектирования выполняется укрупненное представление всех конструкторских и технологических решений; результатом этого этапа является технический проект.

На этапе рабочего проектирования производится детальная проработка всех блоков, узлов и деталей проектируемой системы, а также технологических процессов производства деталей и их сборки в узлы и блоки.

Заключительный этап - изготовление опытного образца, по результатам испытаний которого вносят необходимые изменения в проектную документацию.

При неавтоматизированном проектировании наиболее трудоемкими являются этапы технического и рабочего проектирования.

В процессе проектирования сложной системы формируются определенные представления о системе, отражающие ее существенные свойства с той или иной

степенью подробности. В этих представлениях можно выделить составные части - уровни проектирования. В один уровень, как правило, включаются представления, имеющие общую физическую основу и допускающие для своего описания использование одного и того же математического аппарата. Уровни проектирования можно выделять по степени подробности, с какой отражаются свойства проектируемого объекта. Тогда их называют **горизонтальными (иерархическими) уровнями проектирования**.

Выделение горизонтальных уровней лежит в основе **блочно-иерархического подхода** к проектированию. Горизонтальным уровням свойственно следующее:

- при переходе с некоторого уровня K_1 , на котором рассматривается система S , на соседний, более низкий уровень K_2 происходит разделение системы S на блоки и рассмотрение вместо системы S ее отдельных блоков;
- рассмотрение каждого из блоков на уровне K_2 с большей степенью детализации, чем на уровне K_1 , **приводит к получению** задач приблизительно одинаковой сложности с точки зрения возможностей восприятия человеком и возможностей решения с помощью имеющихся средств проектирования;
- использование своих понятий системы и элемента на каждом иерархическом уровне, т.е. если элементами проектируемой системы S считались блоки S_k , то на соседнем, низшем уровне K_2 те же блоки S_k рассматриваются уже как системы.

Уровни проектирования можно выделять также по характеру учитываемых свойств объекта. В этом случае их называют **вертикальными уровнями проектирования**. При проектировании устройств автоматизации основными вертикальными уровнями являются функциональное (схемное), конструкторское и технологическое проектирования. При проектировании автоматизированных комплексов к этим уровням добавляется алгоритмическое (программное) проектирование.

Функциональное проектирование связано с разработкой структурных, функциональных и принципиальных схем. При функциональном проектировании определяются основные особенности структуры, принципы функционирования, важнейшие параметры и характеристики создаваемых объектов.

Алгоритмическое проектирование связано с разработкой алгоритмов функционирования ЭВМ и вычислительных систем (ВС), с созданием их общего системного и прикладного программного обеспечения.

Конструкторское проектирование включает в себя вопросы конструкторской реализации результатов функционального проектирования, т.е. вопросы выбора форм и материалов оригинальных деталей, выбора типоразмеров

унифицированных деталей, пространственного расположения составных частей, обеспечивающего заданные взаимодействия между элементами конструкции.

Технологическое проектирование охватывает вопросы реализации результатов конструкторского проектирования, т.е. рассматриваются вопросы создания технологических процессов изготовления изделий.

Для этапа НИР целесообразно использование специальных **систем автоматизации научных исследований и экспериментов**. В этих системах используются многие элементы математического и программного обеспечения САПР, обслуживающие другие этапы проектирования.

В зависимости от порядка, в каком выполняются этапы проектирования, различают восходящее и нисходящее проектирование. **Восходящее проектирование** (проектирование снизу-вверх) характеризуется решением задач более низких иерархических уровней перед решением задач более высоких уровней. Противоположная последовательность приводит к **нисходящему проектированию** (проектированию сверху вниз).

В настоящее время проектирование сложного оборудования и его элементов и узлов осуществляется на разных предприятиях с помощью различных САПР, в том числе типовых, например, САПР проектирования электронной и вычислительной аппаратуры, САПР проектирования электрических машин и т.д.

Функциональное проектирование в САПР включает в себя два больших горизонтальных уровня - системный и функционально-логический. Для выполнения задач этих уровней обычно используется нисходящее проектирование.

На **системном уровне** проектируются структурные схемы устройств, в связи с чем данный уровень называют также **структурным уровнем**. На этом уровне ведется укрупненное рассмотрение всей системы в целом, а элементами системы являются такие устройства, как процессоры, каналы связи, различные датчики, исполнительные устройства и др.

На **функционально-логическом уровне** проектируются функциональные и принципиальные схемы устройств. Здесь выделяют подуровни - регистровый и логический. На регистровом подуровне проектируются устройства из блоков (блоки типа регистров, счетчиков, дешифраторов и логических преобразователей, составляющих цепи межрегистровых пересылок). На логическом подуровне проектируются устройства или составляющие их блоки из отдельных логических элементов, (например, вентилях и триггеров).

Задачи функционально-логического уровня в САПР устройств автоматизации аналогичны задачам такого же уровня в других САПР, связанных с проектированием технических объектов.

На **схемотехническом уровне** проектируются принципиальные электрические схемы устройств. Элементами здесь являются компоненты электронных схем (резисторы, конденсаторы, транзисторы, диоды).

На **компонентном уровне** разрабатываются отдельные компоненты устройств, рассматриваемые как системы, состоящие из элементов.

Функциональное проектирование в САПР может быть, как восходящим, так и нисходящим. Восходящее проектирование характеризуется использованием типовых конфигураций компонентов.

Нисходящее проектирование характеризуется стремлением использовать схемотехнические решения, являющиеся наилучшими для конкретного устройства или элемента автоматизации, и связано с разработкой оригинальных принципиальных схем и структур компонентов.

Высшие иерархические уровни алгоритмического проектирования служат для создания программного обеспечения ЭВМ. Для сложных программных систем обычно выделяют два иерархических уровня. На высшем из них производится планирование программной системы и разрабатываются схемы алгоритмов; элементами схем являются программные модули. На следующем уровне эти модули программируются на каком-либо алгоритмическом языке. Здесь используется нисходящее проектирование.

Основная задача **архитектурного уровня** проектирования - выбор архитектуры системы, т.е. определение таких структурно-алгоритмических особенностей, как форматы данных и команд, система команд, принципы выполнения операций, условия возникновения и дисциплина обслуживания прерываний и т.п.

Микропрограммный уровень предназначен для проектирования микропрограмм операций и процедур, выполняемых в ЭВМ аппаратным способом. Этот уровень тесно связан с функционально-логическим уровнем проектирования.

Конструкторское проектирование включает в себя иерархические уровни проектирования стоек, панелей, типовых элементов замены (ТЭЗов). Для решения конструкторских задач характерно восходящее проектирование.

Основные задачи **системного** и **архитектурного** уровней проектирования, следующие:

- определение принципов организации системы;
- выбор архитектуры, уточнение функций системы и их разделение на функции, реализуемые аппаратным и программным путем;
- разработка структурной схемы, т.е. определение состава устройств и способов их взаимодействия;

- определение требований к выходным параметрам устройств и формирование технических заданий (ТЗ) на разработку отдельных элементов системы.

В ТЗ на разработку отдельных устройств САПР входят: перечисление функций, выполняемых устройством; условия работоспособности устройства, требования к его выходным параметрам, данные о содержании и форме информации, которой данное устройство обменивается с другими устройствами системы. Кроме того, на этапе функционального проектирования устройств уже известно принятое на этапе предварительного проектирования решение относительно характера элементной базы.

Поэтому в задачи **микропрограммного уровня** алгоритмического проектирования и регистрового подуровня **функционально-логического уровня** проектирования входят:

- детализация выполняемых устройством функций, их алгоритмическая реализация и представление алгоритмов в одной из принятых форм;
- выбор принципов организации устройства, включающий, например, декомпозицию устройства на ряд блоков с выбором их структуры и т.п.;
- разработка микропрограмм, т.е. определение для каждой команды совокупности микрокоманд и последовательности их выполнения;
- синтез конечных автоматов (блоков), реализующих заданные функции, с определением типа и емкости памяти автоматов, функций выхода и возбуждения элементов памяти.

На **логическом подуровне функционально-логического уровня** проектирования решаются следующие задачи:

- синтез функциональных и принципиальных схем выделенных блоков;
- проверка работоспособности синтезируемых блоков с учетом задержек сигналов и ограничений выбранной элементной базы или выработка требований к элементам в составе САПР;
- синтез контролирующих и диагностических тестов;
- формулировка ТЗ для схемотехнического уровня проектирования.

Основную часть ТЗ на схемотехническом уровне проектирования составляют требования к выходным параметрам электронных схем: задержкам распространения сигналов, мощностям рассеяния, уровням выходного напряжения, запасам помехоустойчивости и т.п. Кроме того, в ТЗ оговариваются условия функционирования в виде указания допустимых диапазонов изменения внешних параметров (температуры, напряжений питания и др.).

На **схемотехническом уровне** основные задачи проектирования, следующие:

- синтез структуры принципиальной схемы;

- расчет параметров пассивных компонентов и определение требований к параметрам активных компонентов;

- расчет вероятности выполнения требований ТЗ к выходным параметрам;
- формулировка ТЗ на проектирование компонентов.

На **компонентном уровне** задачи функционального, конструкторского и технологического проектирования тесно связаны друг с другом. Это:

- выбор физической структуры и расчет параметров полупроводниковых компонентов;
- выбор топологии компонентов и расчет геометрических размеров;
- расчет электрических параметров и характеристик компонентов;
- расчет параметров технологических процессов, обеспечивающих получение желаемого конечного результата;
- расчет вероятности выполнения требований к выходным параметрам элементов и устройств.

При нисходящем проектировании связь иерархических уровней проявляется через формирование ТЗ на разработку элементов с учетом требований, предъявляемых к системе.

При восходящем проектировании разработка элементов предшествует разработке системы, поэтому обычно ТЗ на элементы формируются на основе мнений экспертов на том же уровне, на каком эти элементы проектируются. Связь между уровнями проявляется, прежде всего, в том, что при проектировании системы учитываются свойства уже спроектированных элементов через использование макромоделей элементов.

Подходы и методы проектирования в САПР

Три типа подходов проектирования:

1. Структурный подход требуется синтезировать варианты системы из компонентов (блоков) и оценивать варианты при их частичном переборе с предварительным прогнозированием характеристик компонентов.

2. 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.

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.

3. Объектно-ориентированный подход к проектированию.

Следующие 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.

Представление графической информации в ЭВМ

Создавать и хранить графические объекты в компьютере можно двумя способами: как **растровое** или как **векторное** изображение. Для каждого типа изображения используется свой способ кодирования.

Растровое изображение представляет собой совокупность точек, используемых для его отображения на экране монитора.

Векторное изображение представляет собой совокупность графических примитивов. Каждый примитив состоит из элементарных отрезков кривых, параметры которых (координаты узловых точек, радиус кривизны и пр.) описываются математическими формулами.

Для каждой линии указываются её тип (сплошная, пунктирная, штрихпунктирная), толщина и цвет, а замкнутые фигуры дополнительно характеризуются типом заливки.

Фрактальная графика

Фрактал - это рисунок, который состоит из подобных между собой элементов. Существует большое количество графических изображений, которые являются фракталами: треугольник Серпинского, снежинка Коха, "дракон" Хартера-Хейтуея, множество Мандельброта. Построение фрактального рисунка осуществляется по какому-то алгоритму или путём автоматической генерации изображений при помощи вычислений по конкретным формулам. Изменения значений в алгоритмах или коэффициентов в формулах приводит к модификации этих изображений. Главным преимуществом фрактальной графики есть то, что в файле фрактального изображения сохраняются только алгоритмы и формулы.

Трёхмерная графика

Трёхмерная графика (3D-графика) изучает приёмы и методы создания объёмных моделей объектов, которые максимально соответствуют реальным. Такие объёмные изображения можно вращать и рассматривать со всех сторон. Для создания объёмных изображений используют разные

графические фигуры и гладкие поверхности. При помощи их сначала создаётся каркас объекта, потом его поверхность покрывают материалами, визуально похожими на реальные. После этого делают освещение, гравитацию, свойства атмосферы и другие параметры пространства, в котором находится объект. Для двигающихся объектом указывают траекторию движения, скорость.

Задачи синтеза и анализа. Оптимальное проектирование конструкций

Проектирование — сложный и трудно формализуемый процесс, объединяющий такие важные процедуры, как синтез структуры, выбор параметров элементов, анализ и принятие решения. Особенно важна начальная стадия проектирования, когда выбираются эффективный физический принцип действия, рациональное техническое решение и определяются оптимальные значения параметров.

Задачи синтеза и анализа. Под *синтезом* понимаются проектные процедуры, направленные на получение новых описаний проектируемого объекта в соответствии с заданными показателями его функционирования. *Анализ* — это проектные процедуры, имеющие целью получение информации о свойствах проектируемого объекта по заданному его описанию.

Задачи синтеза связаны с созданием проектных документов и самого проекта, а задачи анализа связаны с оценкой проектных документов.

Процедуры синтеза делятся на процедуры структурного и параметрического синтеза.

Поиск рационального технического решения при выбранном физическом принципе действия осуществляется методами *структурного синтеза*. Определение оптимальных значений параметров элементов технической системы известной структуры представляет собой задачу *параметрического синтеза*, или *параметрической оптимизации*.

Целью структурного синтеза является определение структуры объекта — перечня типов элементов, составляющих объект, и способа связи элементов между собой в составе объекта.

Параметрический синтез заключается в определении числовых значений параметров элементов при заданных структуре и условиях работоспособности, влияющих на выходные параметры объекта, т. е. при параметрическом синтезе нужно найти точку или область в пространстве внутренних параметров, в которых выполняются те или иные условия (обычно условия работоспособности).

Классификация проектных процедур приведена на рис. 1.7.



Рис. 1.7. Классификация проектных процедур

Процедуры анализа делятся на процедуры одно- и многовариантного анализа.

При *одновариантном анализе* заданы значения внутренних и внешних параметров, требуется определить значения выходных параметров объекта. Удобно использовать геометрическую интерпретацию этой задачи, связанную с понятием пространства внутренних параметров; это « n -мерное пространство, в котором для каждого из n внутренних параметров x_i , выделена координатная ось. При одновариантном анализе задается также некоторая точка в пространстве внутренних параметров и требуется в этой точке определить значения выходных параметров. Подобная задача обычно сводится к однократному решению уравнений, составляющих математическую модель, что и обуславливает название данного вида анализа.

Многовариантный анализ заключается в исследовании свойств объекта в некоторой области пространства внутренних параметров. Такой анализ требует многократного решения систем уравнений (многократного выполнения одновариантного анализа).

Синтез называется *оптимизацией*, если определяются наилучшие в заданном смысле структура и значения параметров. При расчетах оптимальных значений параметров при заданной структуре говорят о параметрической оптимизации. Задачу выбора оптимальной структуры называют *структурной оптимизацией*.

Постановка задачи оптимизации имеет содержательный смысл только в том случае, когда появляется необходимость выбора одного из конкурирующих вариантов, полученных при ограниченности ресурсов. Техническое проектирование всегда ведется в условиях жестких ограничений на

материальные, энергетические, временные и прочие виды ресурсов. Вместе с тем средства САПР позволяют выполнить разработку нескольких альтернативных вариантов, поэтому окончательный выбор технического объекта (принятие решения) необходимо проводить с учетом выработанных *правил предпочтения* на основании установленных критериев. Выбор критерия является одним из важных этапов постановки задачи оптимизации, так как все последующие действия направлены на поиск объекта, наиболее близкого к оптимальному по выбранному критерию.

В основе построения правила предпочтения лежит *целевая функция*, количественно выражающая качество объекта и потому называемая также *функцией качества*, или *критерием оптимальности*. Формирование целевой функции всегда выполняется с учетом различных выходных параметров проектируемого устройства. В зависимости от содержательного смысла этих параметров и выбранного способа их сочетания в целевой функции качество объекта будет тем выше, чем больше ее значение (максимизация) или чем меньше ее значение (минимизация).

Выбор целевой функции носит субъективный характер, и поэтому объект может быть оптимален только в смысле данного критерия.

В большинстве подходов к оценке технического объекта принято ориентироваться на эталонные образцы, на мнение ведущих специалистов отрасли (экспертные оценки) или на технико-экономические показатели, определяемые техническим заданием (ТЗ) на проектирование. При подготовке ТЗ обычно учитываются достижения, полученные в мировой практике, а также в той или иной мере экспертные оценки, поэтому более объективной следует считать ориентацию на ТЗ. Наиболее полная оценка проектных решений может быть выполнена на основе анализа технико-экономических показателей с учетом требований, сформулированных в ТЗ.

Качество функционирования любой системы характеризуется вектором выходных параметров $V = (y_1, y_2, \dots, y_m)$. Некоторые из выходных параметров могут быть измерены количественно, другие представляют собой качественную сторону объекта, поэтому все выходные параметры обычно делят соответственно на количественные и качественные. В дальнейшем под вектором V будем подразумевать вектор количественных параметров.

К выходным параметрам, определяющим качество гидропривода, можно отнести тяговые и скоростные характеристики, показатели разгонной и тормозной динамики, управляемости и устойчивости, универсальности, расход среды, количественный и качественный состав загрязняющих веществ, выбрасываемых в атмосферу, и многие другие.

Значение целевой функции может возрастать или убывать с увеличением качества выходного параметра, поэтому в первом случае необходимо искать максимум, а во втором — минимум целевой функции.

Каждый из составляющих вектор V выходных параметров зависит от множества внутренних параметров проектируемого объекта. Следует отметить, что значения некоторых внутренних параметров назначаются и не подлежат

изменению. К таким параметрам, например, можно отнести параметры унифицированных элементов или те из них, значения которых оговорены в техническом задании на проектирование. Остальные параметры можно выбрать, ориентируясь на прототипы с учетом собственного опыта и оценки возможных перспектив развития конструкций подобных объектов.

Внутренние параметры, значения которых могут меняться в процессе оптимизации и которые являются аргументами целевой функции, называют *управляемыми параметрами*.

Пусть в проектируемом гидроприводе имеется n управляемых параметров, образующих вектор $X = (x_1, x_2, \dots, x_n)$. Обозначим целевую функцию через $F(X)$, а область ее определения — через XP . Вектор X определяет координаты точки в области определения XP . Если элементы вектора X принимают только дискретные значения, XP является *дискретным множеством точек* и задача оптимизации относится к области *дискретного* (в частном случае целочисленного) *программирования*.

Большинство задач параметрической оптимизации технических объектов формулируется в терминах непрерывных параметров. Если экстремум целевой функции ищется в неограниченной области XP , его называют *безусловным*, а методы поиска — *методами безусловной оптимизации*. Если экстремум целевой функции ищется в ограниченной области XP , его называют *условным*.

Для решения задач проектирования в машиностроении характерны методы условной оптимизации.

Таким образом, задачу поиска оптимального решения можно в общем случае сформулировать следующим образом:

$$\min(\max)F(X), X \in XP,$$

где X — вектор управляемых параметров; $F(X)$ — целевая функция; XP — область допустимых значений вектора управляемых параметров.

Постановка задачи оптимизации. Основная проблема постановки задачи оптимизации заключается в выборе целевой функции. Сложность выбора целевой функции состоит в том, что любой технический объект первоначально имеет векторный характер критериев оптимальности (многокритериальность), причем улучшение одного из выходных параметров, как правило, приводит к ухудшению другого, так как все выходные параметры являются функциями одних и тех же управляемых параметров и не могут изменяться независимо друг от друга. Такие выходные параметры называют *конфликтными*.

Целевая функция должна быть одна (принцип однозначности). Сведение многокритериальной задачи к однокритериальной называют *сверткой векторного критерия*. Задача поиска его экстремума сводится к задаче математического программирования.

В зависимости от того, каким образом выбираются и объединяются выходные параметры в скалярной функции качества, различают частные, аддитивные, мультипликативные, минимаксные, статистические критерии и т.д.

Частные критерии могут применяться в случаях, когда среди выходных параметров можно выделить один основной параметр $y_i(X)$, наиболее полно

отражающий эффективность проектируемого объекта; этот параметр принимают за целевую функцию. Различные группы и категории транспортных и тяговых средств очень сильно отличаются по назначению (например, тракторы, грузовые автомобили и спортивные автомобили), поэтому для них в качестве выходных параметров, применимых как целевая функция, могут быть и грузоподъемность, и расход топлива, и динамические качества, и многие другие. Условия работоспособности объекта по всем остальным выходным параметрам относят при этом к функциональным ограничениям. Оптимизация на основе такой постановки называется *оптимизацией по частному критерию*.

Достоинство такого подхода — его простота, а существенный недостаток — то, что возможно получить высокое качество объекта только по тому параметру, который принят в качестве целевой функции, в то время как остальные выходные параметры будут только приемлемыми.

Теоретически возможен следующий подход. Можно выбрать один, наиболее значимый выходной параметр и провести его оптимизацию. Затем выбрать следующий по значимости параметр и провести его оптимизацию при условии, что значения управляемых параметров, входящие в выражение как для первого, так и для второго выходных параметров, при оптимизации второго параметра должны быть зафиксированы на уровне, достигнутом при оптимизации первого параметра, а оптимизация должна проводиться только с использованием управляемых параметров, не вошедших в описание выходного параметра, оптимизируемого первым. Далее по такому же принципу можно оптимизировать и все остальные выходные параметры. Однако это возможно только в тех случаях, когда выходные параметры зависят хотя бы частично от различных наборов управляемых параметров, что на практике бывает далеко не всегда.

Взвешенный аддитивный критерий применяют тогда, когда условия работоспособности позволяют выделить две группы выходных параметров. В первую группу входят выходные параметры. $F(X)$, значения которых в процессе оптимизации нужно увеличивать (производительность, вероятность безотказной работы, для гидравлических устройств — грузоподъемность, максимальная скорость и т.п.), во вторую — выходные параметры. (X) , значения которых следует уменьшать (расход, масса гидропривода, время разгона и пр.).

В этом случае целевая функция свертки будет иметь вид:

$$F(X) = \sum_{i=1}^k \lambda_i y_i^-(X) - \lambda_i y_i^+(X), \quad (3.1)$$

где $\lambda_i > 0$ — весовой коэффициент, определяющий степень важности i -го выходного параметра (обычно значения λ_i выбираются проектировщиком и в процессе оптимизации остаются постоянными).

Объединение нескольких выходных параметров, имеющих в общем случае различную физическую размерность и величину, в одной скалярной целевой функции требует предварительного нормирования этих параметров. Способы нормирования параметров могут быть различны. Наиболее простым является

отнесение этого параметра к некоторой заранее заданной средней величине, определяемой из опыта предыдущего проектирования, или к допускаемой величине (например, допускаемые напряжения). В этом случае все составляющие выражения (3.1) будут безразмерными величинами одного порядка, что делает удобным манипулирование коэффициентами λ_i в процессе проектирования.

Наиболее типичным случаем параметрической оптимизации технических объектов является поиск значений вектора управляемых параметров X , определяющих экстремум целевой функции при наличии ограничений. Для дальнейшего изложения будем считать, что в процессе оптимизации ищется минимум целевой функции.

Таким образом, задачу поиска оптимального решения можно в общем случае сформулировать следующим образом:

$$\min F(X). \quad (3.2)$$

Здесь $X \in X_P$ при прямых ограничениях $x_{ni} \leq x_i \leq x_{vi}$ для любого $i \in [1, n]$, где x_{ni}, x_{vi} — максимально и минимально допустимые значения i -го управляемого параметра; n — размерность пространства управляемых параметров и при функциональных ограничениях.

Функциональные ограничения, как правило, представляют собой условия работоспособности выходных параметров, не вошедших в целевую функцию, и могут быть:

типа равенств

$$\psi(X) = 0,$$

типа неравенств

$$\phi(X) > 0,$$

где $\psi(X)$ и $\phi(X)$ — вектор функции.

Выбор метода решения зависит от способа постановки задачи оптимизации.

Любая из точек $X \in X_P$ является *допустимым решением задачи*. Часто параметрический синтез ставится как задача определения любого из допустимых решений. Однако гораздо важнее решить задачу оптимизации — найти оптимальное решение среди допустимых.

Методы решения задач оптимизации

Расчет выходных параметров объекта в САПР, как правило, выполняется через численное решение систем алгебраических и дифференциальных уравнений. Целевая функция и функциональные ограничения определяются неявным образом, что не позволяет оценить их свойства (выпуклость, вогнутость и т.д.) с приемлемыми затратами вычислительных ресурсов. Поэтому решение задач оптимизации в САПР ведется с помощью *поисковых методов* математического программирования, использующих предшествующую информацию для построения улучшенного решения задачи, т.е. эти методы являются итерационными.

По ряду признаков задачи поиска экстремума могут быть отнесены к тому или иному классу. Большинство постановок задач параметрической оптимизации технических систем сводятся к за-Дачам нелинейного программирования, так как целевая функция и ограничения описываются нелинейными зависимостями от вектора управляемых параметров. В отдельных случаях при проектировании удастся так сформулировать задачу, что целевая функция и ограничения являются линейными функциями своих аргументов. Тогда имеет место задача линейного программирования. В зависимости от типа искомого экстремума различают методы локальной и глобальной, условной и безусловной оптимизации. Практически используемые методы в основном являются методами локального поиска.

Одновременно надежные и экономичные методы поиска глобального экстремума в настоящее время неизвестны. Надежным, но крайне неэкономичным методом глобального поиска является *метод сканирования*. При его применении область определения $F(X)$ в пространстве управляемых параметров разбивается на k подобластей, в центре каждой из которых вычисляется значение целевой функции. Если функция зависит от n параметров, необходимо выполнить k^n вариантов расчетов. Чтобы получить достоверную картину поведения гиперповерхности отклика целевой функции, необходимо сканировать допустимую область с достаточно малым шагом, поэтому даже для сравнительно несложных задач затраты машинного времени на поиск становятся недопустимо большими. Этот недостаток характерен и для *методов случайного поиска глобального экстремума*. Однако затраты ресурсов на случайный поиск можно сделать приемлемыми, если не предъявлять высоких требований к надежности определения экстремума.

Наиболее многочисленную группу составляют методы безусловной оптимизации. Некоторое представление о широко применяемых методах этой группы дает рис. 3.6. В зависимости от порядка используемых производных целевой функции по управляемым параметрам методы безусловной оптимизации делят на методы нулевого, первого и второго порядков.

В методах нулевого порядка (прямых методах) информация о производных не используется. Для методов первого порядка необходимо вычислять как значение функции качества, так и ее первые частные производные (градиентные методы). В методах второго порядка организация поиска экстремума ведется с учетом значений целевой функции, ее первых и вторых производных.

В зависимости от количества управляемых параметров целевой функции различают методы одномерного и многомерного поиска. Одномерный поиск может рассматриваться как самостоятельная задача, если аргументом целевой функции является один параметр. Такой же поиск используется в качестве части процедуры многомерной оптимизации в тех случаях, когда необходимо найти оптимальный шаг в выбранном направлении.

Задача условной оптимизации может быть сформулирована как задача безусловной оптимизации с помощью методов Лагранжа или штрафных

функций. Тогда для ее решения применяются методы безусловной оптимизации.

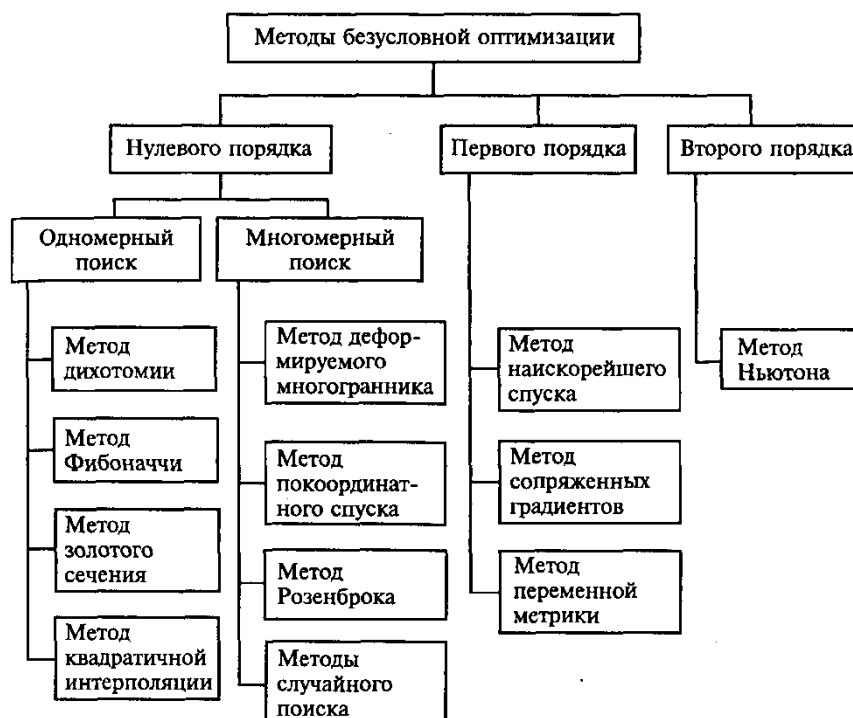


Рисунок 3.6. Классификация методов оптимизации

Задача условной оптимизации может быть решена и методами условной оптимизации (методы проекции градиента, допустимых направлений и др.).

Практически во всех методах оптимизации стремятся построить такую последовательность значений X_0, X_1, X_2, \dots , при которой $F(X_0) > F(X_1) > F(X_2) > \dots$. В этом случае обеспечивается сходимость результатов и можно надеяться, что минимум функции будет найден.

Важной характеристикой методов является их скорость сходимости. Однако оценка сходимости того или иного метода обычно базируется на некоторых теоретических предпосылках относительно особенностей целевой функции (например, функция дважды непрерывно дифференцируема или сильно выпукла) и зависит от выбора начальной точки поиска. Теоретические предпосылки относительно реальных целевых функций могут не удовлетворяться, поэтому скорость сходимости в этих случаях можно рассматривать как сравнительную оценку метода.

Численные методы поиска оптимума позволяют построить последовательность шагов от начальной точки X_0 через некоторые промежуточные точки X_k к локальному экстремуму X^* .

Схема алгоритма поиска оптимального решения для общего случая показана на рис. 3.7. Как отмечалось выше, выбор исходной точки поиска X_0 во многом определяет успех решения всей задачи. Очевидно, что X_0 должна принадлежать области определения целевой функции и, чем ближе к экстремуму выбрана X_0 , тем быстрее и с большей вероятностью экстремум

будет найден. Сущность метода оптимизации определяется этапами 2 и 3 алгоритма, на которых выбирается направление дальнейшего поиска и вычисляются координаты очередной точки X_{k+1} на траектории поиска. Далее в точке X_{k+1} вычисляются значения целевой функции $F(X_{k+1})$ и функций ограничений, т.е. определяется информация, позволяющая судить о достигнутом успехе. Инженер может назначить различные условия прекращения поиска, и, в зависимости от степени их выполнения, поиск будет продолжен или прекратится.

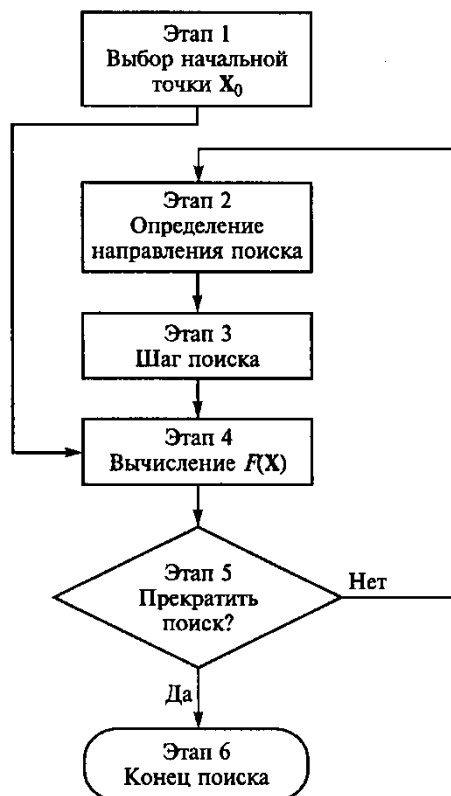


Рисунок 3.7. Алгоритм поиска оптимального решения

Методы одномерного поиска строятся в предположении унимодальности (одноэкстремальности) функции $F(x)$ на заданном интервале $[a, b]$. К функции не предъявляются требования дифференцируемости или непрерывности. Предполагается, что для любого $x \in [a, b]$ значение $F(x)$ может быть вычислено, т.е. найдено путем вычислительного эксперимента.

Методы одномерного поиска можно разделить на методы последовательного поиска (методы дихотомии или половинного деления, Фибоначчи и золотого сечения) и методы, использующие аппроксимацию функции (методы квадратичной и кубической интерполяции и др.).

Для большинства задач, связанных с поиском оптимальных решений при проектировании таких сложных технических систем, как гидропневмосистемы, методы одномерного поиска практически не применимы, поскольку выходные параметры этих систем, как правило, зависят от множества управляемых параметров. Обычно в этом случае применяются методы многомерного поиска.

Математические модели объектов проектирования

Под математической моделью обычно понимается отображение реального объекта с помощью системы математических соотношений.

При построении математических моделей будем исходить из уровней описания САУ, т.е. система — устройство-элемент. Наибольшие трудности возникают при построении математических моделей объектов управления, так как учет физических процессов и явлений в этих объектах требует длительных и весьма трудоемких математических операций. Большое количество однообразных и утомительных операций при построении математических моделей, в том числе дифференцирование, умножение полинома на полином, раскрытие скобок, приведение подобных членов, неизбежно связаны с много-численными ошибками и «ненадежностью» математических моделей. Автоматизация этих операций является обязательной в САПР САУ.

Будем различать графические, аналитические и численные методы построения математических моделей в САПР.

Графические методы применяются при построении математических моделей на верхнем уровне их описания в форме структурных схем и их графовых эквивалентов. В этом случае проектировщик исходит из представления об объекте как о сложной системе, состоящей из устройств и элементов. «Внутреннее содержание» устройств обычно отражается типовыми линейными звеньями САУ или условными обозначениями.

Важным условием автоматизации графических способов является разработка алгоритмов ввода структурных и графовых моделей в ЭВМ. Графические методы построения геометрических моделей устройств и элементов САУ широко применяются в конструкторском и технологическом проектировании.

При аналитическом построении математических моделей исходными являются общие законы физики, результаты обобщения тысячелетнего опыта, накопленного ранее усилиями ученых и инженеров. Это прежде всего законы сохранения энергии, массы и вытекающие из них принципы наименьшего действия, непрерывности материальных потоков, кинетико-химических реакций, теплового баланса и т. п. Естественно, что приложении этих законов и принципов применительно к тому или иному физическому устройству САУ приводит к различным выражениям.

Уравнения движения объектов управления, устройств САУ также выводятся на основе соответствующих законов и принципов. В частности, уравнения движения механических и электро-механических устройств строятся с помощью формализма Лагранжа (математический маятник):

Уравнения движения гидравлических, пневматических и других устройств САУ выводятся на основе аналогичных законов.

Удобен для вывода дифференциальных уравнений движения динамической системы формализм Лагранжа. Замечательное свойство формализма Лагранжа — его независимость от выбора обобщенных координат q_i , поэтому появляется

возможность выбора удобной для исследований системы координат. Обобщенные силы Q_i являются производными от функций энергии различного происхождения: потенциальной $U(q_i)$, «энергии рассеяния» D , энергии регуляторов $E(q_i, \dot{q}_i)$, т. е.

Численные методы формирования математических моделей, включают в себя последовательность операций по обработке и анализу априорной и апостериорной числовой информации об объекте. В результате выполнения анализа получают структура и параметры математических моделей объекта.

Последовательность операций при этом обычно такая:

Задаются математические зависимости между входными и выходными переменными в статике. Проводится факторный анализ априорных и апостериорных данных с целью отсеивания несущественных переменных в этих зависимостях. Под факторным анализом понимается приближенная оценка параметров и коэффициентов уравнений в зависимости от выбранной меры точности совпадения экспериментальных и теоретических данных.

Проводятся эксперименты по определению реакции системы на воздействия по времени. Строятся оценки динамических характеристик объекта во времени. Строятся математические зависимости между выходными и входными переменными в динамике.

Перечисленные операции методики построения математических моделей входят как составляющие в большой раздел теории САУ, известный как идентификация систем и устройств.

Под идентификацией в теории САУ обычно понимают построение математических моделей объектов по априорной и апостериорной информации и, в частности, по известным их входным и выходным сигналам. Наиболее распространены методы параметрической идентификации, когда структура математических моделей уже задана, а требуется найти только ее параметры в соответствии с заданными критериями адекватности математических моделей и объекта.

Наиболее целесообразным способом построения математических моделей САУ, который можно положить в основу построения соответствующей подсистемы САПР САУ, является сочетание аналитических методов с численными. При этом аналитическими методами строятся возможно более полные математических моделей, а с помощью численных методов идентификации осуществляется количественная оценка параметров модели и обеспечивается адекватность ее реальному объекту. Такой подход, реализуемый средствами САПР, оказывается эффективным не только для экономии высококвалифицированного труда, но и для значительного повышения адекватности ММ САУ реальному объекту.

При этом процедура построения математических моделей состоит из трех этапов:

- вывод полной математической модели в аналитической форме на основе классических принципов и формализмов динамики;
- упрощение и преобразование математических моделей в соответствии с назначением и особенностью модели;

— параметрическая идентификация упрощенных математических моделей по ре-зультатам экспериментальных исследований и испытаний.

Классификация математических моделей

В зависимости от характера отображаемых свойств объекта модели делятся на структурные и функциональные.

В процедурах, относящихся к процессу проектирования, преобладает использование математических моделей, отражающих только структурные свойства объекта, например, его геометрическую форму, размеры, взаимное расположение элементов в пространстве. Такие модели называют *структурными*.

Различают структурные ММ топологические и геометрические. В топологических моделях отображаются состав и взаимосвязь элементов объекта. Их часто используют для описания объектов, состоящих из большого числа элементов, при решении задач привязки элементов к определенным пространственным позициям (например, задача компоновки машины) или к относительным моментам времени (например, при разработке технологического процесса). Топологические модели часто имеют форму графов, таблиц, списков и т. п. В геометрических моделях помимо сведений о взаимном расположении элементов объекта содержатся сведения о форме элементов.

Такие модели могут выражаться совокупностью уравнений линий и поверхностей, алгебраических соотношений, описывающих области, составляющие тело объекта. При описании конструкции, состоящей из типовых элементов, используются графы и списки.

В проектных процедурах, связанных с функциональным аспектом проектирования, как правило, используются ММ, отражающие закономерности процессов функционирования объектов. Такие модели называют *функциональными*. Типичная функциональная модель представляет собой систему уравнений, описывающих механические, гидравлические, пневматические, электрические, тепловые процессы. Поскольку характер функционирования объекта в большинстве случаев невозможно описать без учета его структуры, в функциональных ММ отражаются также и структурные свойства объекта. Функциональные модели более сложные по сравнению со структурными и считаются основным типом моделей в САПР.

Существует классификация математических моделей в зависимости от степени детализации описываемых свойств и процессов, протекающих в объекте. Рассмотрим ее применительно к функциональным ММ.

Блочнo-иерархический подход к проектированию технических объектов включает в качестве своей основы и иерархию математических моделей. На каждом иерархическом уровне используются свои ММ, сложность которых согласована с возможностями анализа. Деление моделей по иерархическим уровням происходит по степени детализации описываемых свойств и процессов, протекающих в объекте. При этом на каждом иерархическом уровне используют свои понятия «система» и «элементы». Так, система k -го уровня рассматривается как элемент на соседнем более высоком $(k-1)$ -м уровне абстрагирования.

Структуру объекта на любом иерархическом уровне можно представить в виде совокупности элементов и связей между ними. Свойства каждого элемента описываются математической *моделью элемента*. Она представляет собой соотношения, связывающие внешние по отношению к элементу фазовые переменные. Некоторые подмножества элементов с их связями можно по каким-либо общим признакам объединить в группы или *блоки*. Математическая модель, полученная непосредственным объединением моделей элементов блока в общую систему уравнений, называется *полной моделью* блока. Характерной ее особенностью является присутствие в ней вектора внутренних фазовых переменных, т. е. она описывает и состояние каждого из элементов блока.

При большом количестве элементов порядок системы уравнений становится чрезмерно большим и требуются упрощения. При переходе к более высокому иерархическому уровню упрощения основаны на исключении из модели вектора внутренних переменных. Такая модель называется *макромоделью*. Она уже не описывает процессы внутри блока, а характеризует только процессы взаимодействия данного блока с другими в составе системы блоков. Понятие макромодели имеет существенное значение в блоч-но-иерархическом подходе к проектированию. Замена полных моделей блоков их макромоделями позволяет перейти на более высокий иерархический уровень, где блок низшего уровня выступает уже в качестве элемента нового, укрупненного блока (или системы полностью).

Существует также понятие многоуровневых моделей, когда одни блоки системы описываются полными моделями, а другие — макромоделями.

В зависимости от сложности объекта при его проектировании используют большее или меньшее число уровней абстракции. Число используемых иерархических уровней при проектировании конкретных объектов зависит от традиций предприятия, принятой организации САПР, возможностей используемого математического и программного обеспечения. Увеличение числа уровней позволяет использовать более простые ММ на каждом из них, однако усложняет согласование результатов, полученных на различных уровнях.

В иерархии функциональных моделей для большинства проектируемых сложных объектов объединение уровней, родственных по характеру используемого математического аппарата, приводит к образованию трех укрупненных уровней: микро-, макро- и метаяуровень.

На микроуровне используют математические модели, описывающие физическое состояние и процессы в сплошных средах. Фазовые переменные являются в данном случае функциями нескольких независимых переменных, таких, как пространственные координаты и время, при этом и пространство, и время непрерывны.

Примерами таких моделей служат дифференциальные уравнения в частных производных — уравнения упругости, электродинамики, теплопроводности, гидродинамики, газовой динамики, которые описывают напряженно-деформированное состояние деталей механических конструкций, поля электрического потенциала и температуры и т. п.

К типичным фазовым переменным на микроуровне относятся механические напряжения и деформации, давления, температуры, электрические потенциалы, концентрации частиц, плотности токов. В связи с учетом характера воздействий и фазовых переменных, распределенных в пространстве, эти модели называют *распределенными*.

Анализ распределенных моделей сводится к решению краевых задач математической физики и представляет значительные трудности вычислительного характера. Использование их ограничивается случаями объектов с малым числом участков. Усложнение задачи при увеличении протяженности пространственных и временных областей приводит к необходимости перехода к следующему иерархическому уровню — макроуровню.

На макроуровне производится дискретизация пространства с выделением в качестве элементов отдельных деталей. Такая дискретизация означает переход от распределенных моделей к *сосредоточенным*, при этом из числа независимых переменных исключают пространственные координаты. Элементами этого уровня являются объекты, которые на микроуровне рассматривались как системы (например, валы, пружины, элементы сопротивления). Параметры этих элементов, будучи на микроуровне выходными, становятся внутренними.

Математические модели на макроуровне представляют собой системы обыкновенных дифференциальных уравнений, которые в частных случаях решения статических задач превращаются в системы алгебраических или трансцендентных уравнений. Для их получения и решения используют соответствующие численные методы. В качестве фазовых переменных фигурируют силы, скорости, температуры, расходы, электрические напряжения, токи и т.д. Они характеризуют проявления внешних свойств элементов при их взаимодействии между собой и внешней средой.

С увеличением числа элементов системы возможности решения задач с использованием ММ макроуровня резко сужаются. В этом случае целесообразен переход к следующему, более высокому иерархическому уровню.

На метауровне с помощью дальнейшего абстрагирования от характера физических процессов удается получить приемлемое по сложности описание процессов, протекающих в проектируемых объектах. Математические модели на метауровне — системы обыкновенных дифференциальных уравнений, системы алгебраических уравнений, системы логических уравнений, имитационные модели систем массового обслуживания. Здесь роль элементов и внутренних параметров выполняют системы и выходные параметры предыдущего иерархического уровня. Моделирование на метауровне позволяет выполнить динамический расчет гидропривода, решить вопросы компоновки машины, выполнить основные расчеты на прочность и сопротивление усталости деталей.

Математические модели различают также в зависимости от формы их представления.

Инвариантная форма — запись соотношений модели с помощью традиционного математического языка безотносительно к методу решения уравнений модели.

Алгоритмическая форма — запись соотношений модели и выбранного численного метода решения в форме алгоритма. Вычисление значений искомых величин производится путем решения; систем уравнений.

Аналитическая форма — запись модели в виде результата аналитического решения исходных уравнений модели. При этом модели в аналитической форме обычно представляют собой явные выражения выходных параметров как функций внутренних и внешних параметров. С точки зрения удобства реализации на ЭВМ они выгодно отличаются от других, однако их сложно и не всегда возможно получить.

Схемная форма (называемая также *графической*) — представление модели на некотором графическом языке, например, на языке графов, эквивалентных схем, диаграмм и т. п. Графические формы удобны для восприятия человеком. Использование таких форм возможно при наличии правил однозначного истолкования элементов чертежей и их перевода на язык инвариантных или алгоритмических форм.

Математические модели в алгоритмической и аналитической формах называют соответственно *алгоритмическими* и *аналитическими*. Среди алгоритмических моделей важный класс составляют *имитационные модели*, предназначенные для имитации происходящих в объекте процессов при задании различных зависимостей входных воздействий от времени. Собственно, имитацию названных процессов называют *имитационным моделированием*. Результат имитационного моделирования — зависимости фазовых переменных в избранных элементах системы от времени. Примерами имитационных моделей могут служить модели разгона и торможения выходного звена гидропривода, модель рабочего цикла гидропривода и т. д.

Требования к математическим моделям. Основными требованиями, предъявляемыми к ММ, являются требования адекватности, универсальности и экономичности.

Адекватность. Модель считается адекватной, если она отражает заданные свойства объекта с приемлемой точностью. Точность определяется как степень совпадения предсказанных с помощью модели значений выходных параметров объекта с истинными значениями этих параметров.

Точность модели оценивается относительной погрешностью

$$\varepsilon = \frac{y_m - y_{ист}}{y_m},$$

где y_m — выходной параметр, рассчитанный с помощью модели; $y_{ист}$ — тот же выходной параметр, имеющий место в моделируемом объекте.

Количественная оценка точности модели в большинстве случаев вызывает затруднения по ряду причин.

Во-первых, реальные объекты и их модели характеризуются не одним, а несколькими выходными параметрами, поэтому для возможности сопоставления моделей друг с другом используют сведение векторной оценки точности к скалярной.

Пусть объект характеризуется m выходными параметрами. Относительные погрешности ε_j , по каждому из них будут различны. В связи с этим для общей оценки погрешности модели ε_m по совокупности учитываемых выходных параметров обычно используют одну из норм вектора $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m)$:

$$\varepsilon = \max_{j=1, m} |\varepsilon_j| \quad \text{или} \quad \varepsilon_m = \sqrt{\sum_{j=1}^m \varepsilon_j^2}$$

Во-вторых, так как характер проявления свойств объекта зависит от особенностей его взаимодействия с внешней средой и другими объектами системы, то и показатели точности отображения этих свойств в модели будут зависеть от условий функционирования объекта. В результате оценка точности становится неоднозначной.

Кроме того, значения $y_{ист}$, обычно отождествляют с экспериментальными полученными. Однако погрешности эксперимента во многих случаях оказываются соизмеримыми с погрешностью ММ или даже превышают их. Чтобы уменьшить влияние этих факторов на результаты оценки погрешности модели, сравнение моделей проводят по результатам их использования в некоторых стандартных ситуациях, соответствующих наиболее характерным условиям работы реального объекта, причем число этих ситуаций стараются максимально ограничить.

Конечно, ММ должна иметь высокую точность отображения свойств объекта в конкретных условиях его эксплуатации. Однако целью ее создания является возможность использования модели для исследования поведения объекта в некоторых областях изменения его внутренних и внешних параметров. Поскольку выходные параметры системы являются функциями внешних $Q = (q_1, q_2, \dots, q_k)$ и внутренних $X = (x_1, x_2, \dots, x_n)$ ее параметров, погрешность ε_m зависит от Q и X . Обычно внутренние параметры ММ выбираются из условия минимизации погрешности ε_m в некоторой точке $Q_{ном}$ пространства внешних параметров, и величина погрешности модели становится функцией Q .

Если задаться предельно допустимой погрешностью δ , то в пространстве внешних параметров можно выделить область, в которой выполняется условие $\varepsilon_m < \delta$.

Такую область называют областью адекватности (ОА) модели.

Универсальность. Степень универсальности ММ характеризует полноту отображения в них свойств реального объекта и определяется возможностью использования модели для анализа более или менее многочисленной группы однотипных объектов, а также числом доступных для анализа режимов функционирования. Использование машинных методов проектирования станет неудобным, если в процессе анализа объекта при каждом изменении режима функционирования пользователю потребуется смена ММ.

Универсальность модели в первую очередь зависит от числа и состава учитываемых в модели внешних и выходных параметров. Увеличение их расширяет применимость модели, но существенно усложняет ее разработку. Тем не менее степень использования универсальных ММ в САПР является одним из основных критериев ее выбора.

Экономичность. Экономичность ММ характеризуется затратами вычислительных ресурсов для ее реализации, а именно затратами машинного времени и памяти. Общие затраты на выполнение в САПР какой-либо проектной процедуры зависят как от особенностей выбранных моделей, так и от методов решения.

В большинстве случаев при реализации численного метода происходят многократные обращения к модели элемента, входящего в состав моделируемого объекта. Тогда удобно экономичность модели элемента характеризовать затратами машинного времени, получающимися при обращении к модели, а число обращений к модели должно учитываться при оценке экономичности метода решения.

Экономичность модели по затратам памяти оценивается объемом оперативной памяти, необходимой для реализации модели.

Требования широких областей адекватности, высокой степени универсальности, с одной стороны, и высокой экономичности, с другой, являются противоречивыми. Наилучшее компромиссное удовлетворение этих требований оказывается неодинаковым в различных применениях. Данное обстоятельство обуславливает использование в САПР многих моделей для объектов одного и того же типа.

Стандарты обмена данными между системами

В последние годы широкое распространение нашли программные системы, обеспечивающие поддержку жизненного цикла наукоемкой продукции, которые, как правило, имеют собственную конфигурацию и особенности применения, связанные с кругом решаемых задач. В этой связи актуальной является проблема стандартизации обмена информацией между этими системами [1]. Так, например, в области машиностроения необходимость стандартного представления информации определяется следующими причинами:

- в разработке и создании изделий могут принимать участие несколько организаций и предприятий;
- в большинстве случаев, не только на разных предприятиях, но и в рамках одного из них, используются разные САПР, системы управления данными об изделии – PDM-системы и ERP-системы.

В настоящей работе приводится критический анализ существующих методов стандартизации обмена данными между различными системами поддержки жизненного цикла изделий машиностроения и предлагается новая идеология стандартного представления этих данных, основанная на языке XML. Приводится пример описания структуры данных об изделии на этом языке.

Остановимся сначала на существующей нормативной базе, регламентирующей обмен информацией о составе и структуре машиностроительных изделий.

Для этих целей существует ряд нормативных документов серии ГОСТ-Р-ИСО-10303, определяющих формат обмена данными об изделии на основе стандарта STEP [2]. Не останавливаясь на особенностях стандартизации графических данных, также определяемых стандартами этой серии, ограничимся здесь только

обсуждением вопросов, связанных с обменом данными о структуре изделия между PDM/PLM – системами.

Стандарт STEP позволяет описывать структуру изделия с помощью языка EXPRESS. Важно отметить, что тексты самого стандарта основаны на зарубежных аналогах.

- Так, например, ГОСТ-Р-ИСО-10303-11-2000 основан на языке EXPRESS, в основу которого положены устаревшие языки программирования, такие как Ada, Algol, Euler и PL/1. Широкое распространение стандарта STEP и языка EXPRESS за рубежом, по-видимому объясняется возможностью использования и простого переноса уже накопленного опыта в новые условия.

- Другим примером может служить серия стандартов ГОСТ-Р-ИСО-10303, которые являются образцом классического подхода к стандартизации обмена информацией между разнородными системами. Так, согласно требованиям стандарта ГОСТ-Р-ИСО-10303-11-2000 в процессе работы с информацией о структуре изделия необходим синтаксический анализатор языка EXPRESS, который практически игнорирует возможности современных средств, предоставляемых операционными системами и языками программирования.

- Кроме отмеченных особенностей стандартов серии ГОСТ-Р-ИСО-10303, можно также указать и на отсутствие в них регламентации некоторых новых возможностей современных PDM/PLM в части описания структуры изделий с целью обмена информацией с другими программными системами.

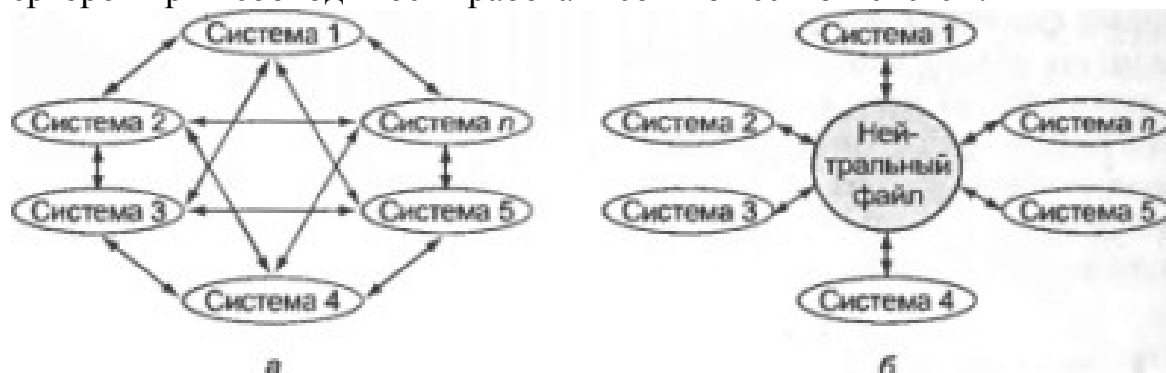
В связи с тем, что отечественные предприятия переходят на новые системы управления жизненным циклом изделий позже, чем аналогичные зарубежные организации, было бы целесообразно отказаться от простого воспроизведения их опыта, а с ним и от использования зарубежных стандартов, а создавать собственные стандарты, основанные на современных языках программирования, с учетом отечественной специфики и уже имеющихся и широко используемых в промышленности стандартов.

Когда две или более CAD/CAM/CAE-системы объединяются и связываются в единое приложение для совместного использования данных, часто возникает проблема обмена данными. Фактически всегда имеется потребность связать воедино несколько систем либо внутри одной организации, либо внешне, как в случае со смежниками или поставщиками компонентов.

Для решения этой коммуникационной проблемы необходима возможность преобразовывать данные технических требований одной системы в форму, понятную для других систем, и наоборот.

Различные CAD/CAM/CAE-системы хранят данные технических требований в структурах разного вида, поэтому для переноса данных необходимо преобразовать данные технических требований одной системы в формат другой системы, еще один конвертор необходим для переноса данных между двумя системами в противоположном направлении. Следовательно, для каждой пары систем необходимо иметь два конвертора. Если у нас есть p различных систем, нам необходимо разработать $p(p - 1)$ конверторов, поскольку количество пар систем равно $p(p - 1) / 2$. Например, для обмена данными между 10 системами придется

разработать 90 конверторов. Таким образом, метод прямого конвертирования непрактичен, так как требует разработки слишком большого количества конверторов при необходимости работать со множеством систем.



Однако обмен данными можно обеспечить, введя нейтральную структуру базы данных, называемую нейтральным файлом, которая была бы независима от существующих САПР. Эта структура будет действовать как промежуточная точка коммуникации между различными структурами баз данных САПР. Таким образом, в каждой системе будет своя пара конверторов для экспорта и импорта данных в этот нейтральный формат.

Рассмотрим три типичных формата нейтрального файла: **IGES** (Initial Graphics Exchange Specification — первоначальная спецификация обмена графическими данными), **DXF** (Drawing interchange Format — формат обмена чертежами) и **STEP** (STandard for Exchange of Product model data — стандарт обмена данными о модели продукта). В настоящее время IGES является самым популярным форматом нейтрального файла, а формат DXF используется главным образом для обмена данными чертежей. STEP — это стандартный формат данных, используемый для хранения полной информации обо всем жизненном цикле продукта, включая проектирование, анализ, производство, контроль качества, испытания и обслуживание помимо обычных данных технических требований. В настоящее время CAD-системы, перешли к формату STEP.

Форматы IGES, DXF, STEP

Описание формата IGES версии 1.0, опубликованное в январе 1980 г. В 1981 г. оно было принято Американским Национальным институтом стандартов (ANSI) в качестве стандарта.

IGES был первым стандартным форматом обмена данными, разработанным для нужд передачи данных технических требований между различными САПР.

Ранние версии IGES были неявным образом ориентированы на CAD/CAM-системы 1970 -х и начала 1980-х гг., то есть главным образом на обмен чертежами. В более поздних версиях спектр типов данных, подлежащих обмену, был расширен.

Формат DXF (Drawing interchange Format — формат обмена чертежами) изначально разрабатывался для того, чтобы предоставить пользователям гибкость в управлении данными и преобразовании чертежей программы AutoCAD в форматы файлов, которые могли читаться и использоваться другими САПР. Из-за популярности AutoCAD формат DXF стал фактическим стандартом обмена

файлами CAD-чертежей почти для всех САПР. На самом деле почти в каждой из появляющихся новых САПР имеется транслятор в формат DXF и обратно.

DXF-файл — это текстовый ASCII-файл, состоящий из пяти разделов: Header (Заголовок), Table (Таблица), Block (Блок), Entity (Элемент) и Terminate (Конец). В разделе Header описывается среда AutoCAD, в которой был создан DXF-файл. В разделе Table содержится информация о типах линий, слоях, стилях текста и видах, которые могут быть определены на чертеже. В разделе Block содержится список графических элементов, определенных как группа. Конкретные данные по каждому элементу хранятся в соответствующем разделе Entity, который следует сразу за разделом Block. Раздел Entity — это главный раздел DXF- файла, в котором описываются все элементы, присутствующие на чертеже.

Форматы IGES и DXF были разработаны для обмена данными технических требований, а не данными о продукте. Под данными о продукте мы понимаем данные, относящиеся ко всему жизненному циклу продукт.

PDES (Product Data Exchange Specification — спецификация для обмена данными о продуктах).

Целью PDES было устранить потребность в инженерных чертежах и других бумажных документах при обмене информацией о различных фазах жизненного цикла продукта между сходными или различающимися САПР.

В основе разработки STEP лежат следующие принципы.

- Стандарт STEP должен ориентироваться на данные о продукте, которые включают информацию обо всем жизненном цикле продукта: проектировании, производстве, контроле качества, испытании и поддержке.
- В структурах данных STEP информация, относящаяся к приложению, должна храниться в модуле уровня приложения, отдельно от общей информации о форме. Благодаря такому подходу структура данных сможет поддерживать широкий спектр приложений, избегая при этом избыточности в общей структуре данных.
- Для определения структуры данных должен использоваться формальный язык. Спецификации IGES и DXF описывают формат физического файла, в котором хранятся все геометрические и прочие данные. В STEP данные описываются на языке EXPRESS, а затем результат преобразовывается в физический файл. Таким образом можно избежать неоднозначностей при интерпретации данных о продукте, извлеченных из файла.

Использование механизмов OLE и COM в САПР

- COM (Component Object Model)
- OLE (Object Linking and Embedding)

Говоря об обмене данными между различными CAD/CAM/CAE-системами, мы обычно подразумеваем широко используемый механизм экспорта/импорта из формата хранения данных одной системы в формат хранения данных другой. Все это выполняется на файловом уровне и нередко приводит к частичной потере данных из-за их некорректной интерпретации. Связано это с тем, что разработчики программ преобразования данных из одного формата в другой зачастую лишены

возможности взаимодействия с создателями данного формата и не имеют полной информации по его структуре.

В качестве альтернативы ряд компаний предлагает так называемый прямой интерфейс между приложениями, при котором данные опять же передаются из одного формата в другой. По сути, это то же самое, только реализовано совместными усилиями разработчиков соответствующих форматов и позволяет практически полностью устранить потерю конвертируемых данных. В то же время системы постоянно развиваются, их форматы хранения данных претерпевают различные модификации, требуя внесения соответствующих изменений и в интерфейсы обмена данными. Это приводит к необходимости постоянного привлечения специалистов для поддержки обмена данными между приложениями.

Пользователи программного обеспечения Microsoft Office с технологией Object Embedding and Linking (OLE) — внедрение и связывание объектов — работают уже давно. Первоначально она обозначала возможность внедрения документа, созданного в одном приложении, в документ, созданный в другом приложении. Хорошим примером этому могут послужить таблицы Excel в документах Word. Изначальная концепция OLE изменялась со временем и в итоге была заменена идеей модели объектных компонентов (Component Object Model — COM). COM представляет собой глобальный интерфейс для создания программных компонентов, которые можно совместить с другими компонентами в любом сочетании.

Особенностью COM-объектов является то, что они существуют в пределах одного компьютера. Следующим шагом стало появление Distributed Common Object Model (DCOM), которые, по сути своей, практически ничем не отличаются от COM-объектов, за исключением того, что существуют и взаимодействуют друг с другом они не только в пределах одного компьютера, но и в компьютерной сети. Здесь мы уже имеем распределенную объектную модель, когда множество взаимодействующих друг с другом объектов находятся на разных рабочих местах, объединенных сетью.

OLE for D&M (в литературе можно встретить обозначение OLE4DM) — это один из аспектов распределенной модели, когда предлагается набор стандартных интерфейсов для обмена и управления данными между трехмерными CAD-, CAM- и CAE-систем. Эта технология нацелена на предоставление прямого доступа одной системы к данным математической модели другой системы, минуя файловый обмен. OLE for D&M поддерживает клиент-серверную технологию и выступает одним из эффективных средств интеграции отдельных Windows-приложений в единый комплекс. Если приложение поддерживает этот интерфейс, оно легко интегрируется с другой системой.

Общие сведения

Одним из условий успешного функционирования САПР является наличие необходимой информации.

Под информационным обеспечением САПР понимаются документы, содержащие описания стандартных проектных процедур, типовых проектных решений, типовых элементов, комплектующих изделий, материалов и другие

данные, а также файлы и блоки данных на машинных носителях с записью указанных документов.

Основной задачей информационного обеспечения процессов проектирования является своевременная выдача источнику запроса полной и достоверной информации, приводящей к выполнению определенной части процесса проектирования.

Исходя из этого к информационному обеспечению САПР предъявляются следующие требования:

- обеспечение информацией как автоматизированных, так и ручных процессов проектирования;
- хранение и поиск информации, представляющей результат ручных и автоматизированных процессов проектирования;
- достаточный объем хранилищ информации. Структура системы должна допускать возможность наращивания емкости памяти вместе с ростом объема информации, подлежащей хранению, одновременно должны обеспечиваться компактность хранимой информации и минимальный износ носителей информации;
- быстродействие системы информационного обеспечения, благодаря которой время получения информации проектировщиками гораздо меньше времени, необходимого для получения этой же информации традиционными средствами;
- возможность быстрого внесения изменений и корректировки информации, доведения этих изменений до потребителя и получения твердой копии документа.

При создании информационного обеспечения основная проблема заключается в преобразовании информации, необходимой для выполнения проектно-конструкторских работ над данным классом объектов, в приемлемую и наиболее рациональную для машинной обработки форму.

Основу информационного обеспечения САПР составляет совокупность всевозможных данных, которые используются в процессе проектирования. При проектировании сложного объекта, в котором участвует более одного проектировщика, данные, необходимые каждому из них, должны быть легко доступны одновременно.

Данные — это сведения о некоторых фактах, позволяющие делать определенные выводы. Взаимосвязанные данные часто называют системой данных, а хранимые данные — информационным фондом. Основное назначение информационного обеспечения — предоставлять пользователям САПР достоверную информацию в необходимом им виде.

На заре развития вычислительной техники данные хранились на перфоленте или перфокартах, которые могли считываться электронным оборудованием. Для обработки данных существовали специальные прикладные программы, представлявшие собой независимые программные единицы и не являвшиеся частью более сложной системы, которую в современных условиях представляет САПР.

Со временем оборудование и используемые для эксплуатации системы электронной обработки данных процедуры существенно усложнились. В качестве

информационных носителей в настоящее время преимущественно используются магнитные и оптические носители, отличающиеся большей на несколько порядков плотностью записи. Стало возможным использовать как независимые информационные системы, так и являющиеся подсистемами более сложных систем.

Главная цель информационного обеспечения осталась прежней — предоставление по запросу заинтересованного пользователя достоверной информации в определенное время.

Чтобы понять процесс электронной обработки данных, необходимо знать ряд терминов, которые применяются при описании и представлении данных.

Предметная область может относиться к любому типу организации (банк, университет, больница или завод) или автоматизированной системы. Для предметной области САПР может потребоваться информация, характерная для данной области проектирования.

Объектом может быть любой предмет, событие, понятие и т.п., о котором приводятся данные.

Каждый объект характеризуется рядом основных атрибутов. Например, конструкционный материал характеризуется плотностью, прочностными параметрами, тепло- и электропроводностью и т. п., автомобиль можно характеризовать такими атрибутами, как максимальная скорость, динамика разгона, расход топлива, масса, габариты и др.

Атрибут называют также элементом данных, полем данных или просто полем.

Сведения, содержащиеся в каждом атрибуте, называют значениями данных. Значения данных представляют действительные данные, содержащиеся в каждом элементе данных; ими могут быть конкретные величины упомянутых выше характеристик конструкционного материала.

Среди атрибутов имеются такие, по значениям которых возможна идентификация объекта. Атрибуты, по значениям которых определяют значения других атрибутов, называют идентификаторами объекта, или ключевыми элементами данных. Отметим, что один и тот же объект могут идентифицировать несколько элементов данных. Их тогда считают кандидатами в идентификаторы. Проблему выбора идентификатора из нескольких кандидатов решает пользователь САПР. Например, зная какую-либо из характеристик конструкционного материала, можно определить вид этого материала, его свойства.

Совокупность значений связанных элементов данных образует запись данных. В приведенном выше примере с конструкционным материалом такими элементами данных являются конкретное его обозначение и численные значения его свойств.

Упорядоченную совокупность записей данных называют файлом данных, или набором данных.

Наиболее высокой формой организации информационного обеспечения для больших систем являются банки данных, представляющие собой совокупность средств для централизованного накопления и коллективного использования данных в САПР. Банк данных является проблемно-ориентированной информационно-справочной системой, которая обеспечивает ввод необходимой

информации, автономное от конкретных задач ведение и сохранение информационных массивов и выдачу необходимой информации по запросу пользователя или программы. Банк данных может быть определен как система программных, языковых, организационных и технических средств, предназначенных для хранения и многоцелевого использования информации.

Банки данных должны обеспечивать:

- сокращение времени поиска данных;
- многократность использования данных;
- простоту и удобство обращения к данным пользователей;
- достоверность хранения данных.

Основными частями банка данных является база данных (БД), представляющая собой систематизированные взаимосвязанные совокупности данных, и система управления базами данных (СУБД), обеспечивающая необходимые манипуляции с информационными массивами.

Множество данных, которые потенциально могут использоваться при функционировании САПР или являются запоминаемым результатом ее работы, образует БД системы.

База данных — сами данные, находящиеся в запоминающих устройствах ЭВМ и структурированные в соответствии с принятыми в данном банке данных правилами.

Система управления базами данных — совокупность программных средств, обеспечивающих функционирование банка данных. С помощью СУБД производятся запись данных в БД, их выборка по запросам пользователей и прикладных программ, обеспечивается защита данных от искажений и несанкционированного доступа и т. п.

Файлы данных чаще всего специально создаются для использования конкретными программами (подпрограммами), которые реализуют ввод данных из файла в строго определенной последовательности.

К недостаткам размещения данных, необходимых предметной САПР, в файлах данных можно отнести следующие.

Избыточность данных. Некоторые элементы данных, необходимых САПР, неизбежно используются во многих прикладных программах. Поскольку данные требуются нескольким прикладным программам, они часто записываются в несколько файлов, при этом одни и те же данные хранятся в разных местах. Такое положение называют избыточностью данных. Оно делает проблематичным обеспечение непротиворечивости данных и обуславливает еще один недостаток — сложность в управлении.

Проблемы непротиворечивости данных. Одной из причин нарушения непротиворечивости данных является их избыточность, что связано с хранением одной и той же информации в нескольких местах. При необходимости обновления информации ее нужно изменить во всех файлах, что бывает затруднительно. В результате об одном и том же объекте предметной области в разных местах хранится различная информация.

Ограничения по доступности данных. В современных условиях лицо

с соответствующими правами доступа должно иметь возможность получить данные за приемлемый отрезок времени. Если же данные содержатся в разных файлах, доступность данных, комбинируемых из этих файлов, ограничена. —

Для решения вышеуказанных проблем были разработаны системы с базами данных, представляющими собой совокупность специально организованных данных, рассчитанных на применение в большом количестве прикладных программ конкретной предметной области, работа с которыми обеспечивается специальными пакетами

прикладных программ — системой управления базой данных с целью создания массивов данных, их обновления и получения справок. Основное отличие БД от файла данных состоит в том, что файл данных может иметь несколько назначений, но соответствует одному представлению о хранимых данных; БД также имеет несколько назначений, но соответствует различным представлениям о хранимых данных.

Программный модуль, входящий в состав САПР, при своей работе обращается за необходимой информацией не к какому-то массиву информации, как это имело место в автономных программах, а к СУБД. Последняя организует поиск необходимой информации в сложной информационной структуре — БД, упорядочивание и представление этой информации в необходимом объеме соответствующему модулю.

Различие в методах использования файлов данных и БД проиллюстрировано на рис. 4.1 и 4.2.





Основные требования к БД следующие:

- целостность данных — их непротиворечивость и достоверность;
- универсальность, т. е. наличие в БД всех необходимых данных и возможность доступа к ним в процессе решения проектной задачи;
- открытость БД для внесения в нее новой информации;
- наличие языков высокого уровня взаимодействия пользователей с БД;
- секретность, т. е. невозможность несанкционированного доступа к информации и ее изменений;
- оптимизация организации БД — минимизация избыточности данных.

Одним из принципов построения САПР является информационная согласованность частей ее программного обеспечения, т. е. пригодность результатов выполнения одной проектной процедуры для использования другой проектной процедурой без их трудоемкого ручного преобразования пользователем. Отсюда вытекают следующие условия информационной согласованности:

- использование программами одной и той же подсистемы САПР единой БД;
- использование единого внутреннего языка для представления данных.

Комплексная автоматизация процесса проектирования объекта предполагает информационную согласованность не только отдельных программ подсистем САПР, но и самих подсистем между собой. Способом достижения этой согласованности является единство информационного обеспечения.

Основные способы информационного согласования подсистем САПР достигаются либо созданием единой БД, либо сопряжением нескольких БД с помощью специальных программ, которые перекодируют информацию, приводя ее к требуемому виду.

Сложность разработки БД определяется тем, что формирование структуры БД возможно только после разработки алгоритмов, реализуемых при проектировании. При этом необходимо, чтобы степень разработки алгоритмов была доведена до машинной реализации, так как структура БД должна учитывать специфику автоматизированного преобразования информации при решении проектных задач с целью эффективного использования вычислительной техники. Однако для разработки программ необходимы сведения о структуре БД. Следовательно,

информационное обеспечение САПР и специальное программное обеспечение должны создаваться параллельно.



Информация, имеющая место при проектировании, может быть разделена на статическую и динамическую (рис. 4.3).

Статическая информация характеризуется сравнительно редкими изменениями и используется только в режиме чтения. К этой информации следует отнести данные технического задания на проектирование и справочные данные, которые характеризуются большим объемом. Формирование, загрузка и корректировка справочных данных осуществляются исключительно администратором базы данных, т. е. программистом системного профиля, формирующим БД. Объем данных технического задания на проектируемый объект значительно меньше объема справочных данных. Круг лиц, имеющих право вносить изменения в данные технического задания, более ограничен, чем круг лиц, имеющих право производить корректировку справочных данных.

Динамическая информация состоит из данных, накапливаемых для выполнения определенных процессов проектирования (промежуточные данные), и данных, представляющих собой результат проектирования при выполнении этих процессов. Промежуточные данные меняются при функционировании САПР. Вносить же коррективы в проектные результаты имеет право только главный конструктор проекта при соответствующем обосновании.

Широко используется в САПР фактографическая информация, составляющая основу БД. Эта информация представляет собой числовые и буквенные справочные данные о материалах, ценах, комплектующих изделиях, спроектированных в САПР объектах, их зарубежных и отечественных аналогах и т. п. Сюда же относятся данные, необходимые для выполнения расчетов: коэффициенты, таблицы, аппроксимированные графические зависимости и т.д.

При формировании БД в первую очередь надлежит исследовать информацию, необходимую для решения проектно-конструкторских задач. К этому исследованию следует подходить с двух позиций: с точки зрения полезности информации и с позиции эффективности обработки информации и пропускной способности вычислительной техники и человека.

Как уже отмечалось выше, части программного обеспечения и методы, осуществляющие управление базой данных, составляют СУБД, которая допускает множество различных представлений о хранимых данных, а также позволяет прикладным программам работать с БД без знания конкретного способа размещения данных в памяти ЭВМ. СУБД выступает как совокупность программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями и должна обеспечивать:

эффективное выполнение различных функций предметной области;
простоту физической реализации БД;
возможность централизованного и децентрализованного управления БД;
минимизацию избыточности хранимых данных;
предоставление пользователю по запросам непротиворечивой информации;
простоту разработки, ведение и совершенствование прикладных программ.
СУБД реализует два интерфейса:

- между логическими структурами данных в программах и БД;
- между логической и физической структурами БД.



Структурная схема СУБД приведена на рис. 4.4.

Манипулятор является ядром СУБД. В его функции входит управление компонентами системы, организация их взаимодействия и осуществление связи с операционной системой ЭВМ и администратором банка данных, выполнение основных операций над БД, контроль и защита целостности и секретности данных,

редактирование вывода, кодирование (декодирование) и сжатие (расширение) данных, сбор статистики и ведение протокола.

Администратор банка данных осуществляет внешнюю координацию всей работы банка данных и выполняет действия, пока не поддающиеся формализации. Прежде всего он отвечает за разработку концептуальной модели предметной области, описывающей все представляющие интерес объекты и взаимосвязи. Концептуальная модель должна быть трансформирована в модель данных, поддерживаемую конкретной СУБД. Следовательно, администратору необходимо спроектировать логическую модель. Наконец, исходя из логической модели, он должен спроектировать физическую (внутреннюю) модель, которая будет обеспечивать требуемые операционные характеристики. Таким образом, в его функции входит создание БД, согласование требований пользователей, управление загрузкой баз, распределение паролей; общее управление работой банка данных (наблюдение за ходом работы, подключение новых пользователей, управление восстановлением при сбоях, принятие решений в случае нарушений секретности, анализ статистики, оценка и обеспечение эффективности работы БД); реорганизация БД (изменение схем, реорганизация баз); генерация и развитие СУБД.

Администратор и манипулятор выполняют свои функции, взаимодействуя с сервисными программами. Ряд сервисных программ допускает двойной запуск: автоматически со стороны манипулятора БД и вручную со стороны администратора, другие запускаются только администратором БД (например, программы моделирования БД). Администратор БД может пополнять набор новыми сервисными программами (контроля данных, измерения производительности и т.д.).

Сервисные программы осуществляют основные операции над БД, в частности сортировку, выборку данных, слияние, дополнение и изменение БД, редактирование ответов.

При функционировании банка данных различают следующие основные виды запросов к банку со стороны пользователей: на выдачу справок в виде показателей и документов для пользователей системы;

- формирование и выдачу рабочих массивов для прикладных программ САПР;
- выдачу рабочих массивов со стороны других банков данных;
- пополнение или изменение БД;
- ввод входных массивов.

Поступивший в банк данных запрос проходит этап предварительной обработки, на котором осуществляется синтаксический и логический контроль, включающий проверку пароля абонентов, недопустимых сочетаний признаков и т.д. При обнаружении ошибок запрос к дальнейшей обработке не принимается, а на печать выдается информация об ошибках.

Следующий этап — интерпретация запроса — состоит в распознавании вида запроса: на выдачу показателя и документов, на формирование рабочих массивов, на изменение или пополнение БД. На этом этапе запрос с языка запросов переводится на язык манипулирования данными.

В соответствии с поисковым предписанием происходят обращение к рабочей области в памяти и выборка искомых данных или корректировка данных в базе. Найденные в БД данные тщательно контролируются и анализируются, затем редактируются, а на печать выдается информация об ошибках.

Разработка банков данных и его основного инвариантного компонента, т. е. СУБД, представляет собой сложную задачу, требующую значительных затрат времени и предполагающую высокую квалификацию разработчиков в области системного программирования. В связи с этим при построении информационного обеспечения САПР конкретных технических объектов на основе банков данных необходимо сделать анализ существующих банков данных и принять для САПР один из уже разработанных. Однако и в этом случае необходимость использования такого сложного компонента, каким является банк данных, должна быть всесторонне обоснована. Следует иметь в виду, что информационное обеспечение САПР может быть организовано и на основе более простой, но специально разработанной информационной структуры, учитывающей специфику проектируемых технических объектов. Подобные ориентированные информационные структуры при относительной простоте и невысоких требованиях к конфигурации технических средств могут реализовать набор необходимых функций по обработке данных, для чего должно быть разработано необходимое программное обеспечение.

Такая альтернатива к использованию банков данных объясняется тем, что универсальные информационные системы в виде банков данных рассчитаны на решение задач с большей номенклатурой различных параметров и характеристик объектов, поэтому применение их при создании САПР с относительно небольшой номенклатурой данных иногда оказывается просто нецелесообразным, а порой и невозможным.

Одной из важнейших характеристик СУБД является модель данных, которая поддерживается СУБД. Как отмечалось, существуют три модели данных: иерархическая, сетевая и реляционная. Выбор той или иной модели данных для построения информационного обеспечения САПР зависит от требований, которые предъявляются к информационной базе, создаваемой САПР.

Большое значение при выборе СУБД имеют средства взаимодействия пользователей с БД. Непрограммисты, а ими являются большинство пользователей САПР, взаимодействуют с БД с помощью языков манипулирования данными с использованием средств телеобработки. Тогда СУБД САПР должна включать в себя язык манипулирования данными, доступный непрограммисту, и средства доступа к данным через терминал.

Кроме того, при выборе для САПР той или иной СУБД следует учитывать следующие требования:

- возможность обеспечения мультипрограммного режима;
- независимость данных от конкретных пакетов программ;
- наличие средств, позволяющих сократить дублирование данных в информационной базе; возможность восстановления БД;
- совместимость СУБД с операционной системой ЭВМ и т.д.

Рациональная организация и функционирование банка данных возможны лишь при максимальном учете специфики информации, заносимой в БД. Если специфические особенности такой информации не учитываются, это приводит к значительному перерасходу машинной памяти и увеличивает затраты времени на обработку данных.

База данных организуется на основе принципов системного подхода. При этом предполагаются:

- неизбыточность, данных;
- независимость данных от программ решения задач;
- выбор структур данных, ориентированных на все задачи, решаемые системой;
- возможность дополнения, развития и обновления данных;
- типизация алгоритмов обработки данных.

Организация, структура и состав БД зависят от информационных моделей проектируемых объектов, от методов получения проектных решений и от используемых в САПР технических и программных средств. С другой стороны, все перечисленные факторы зависят от структуры БД. Основной особенностью БД по сравнению с массивами записей является наличие связей между структурными единицами данных.

Базы данных иногда определяют, как избыточную совокупность элементов данных. Однако в действительности для уменьшения времени доступа к данным во многих БД избыточность в незначительной степени присутствует. Некоторые записи повторяются для того, чтобы обеспечить возможность восстановления данных при их случайной потере. В этом случае говорят о минимальной избыточности БД.

Таким образом, БД можно определить, как совокупность взаимосвязанных, совместно хранящихся данных при наличии такой минимальной избыточности, при которой допускается их использование оптимальным образом для одного или нескольких приложений. Для этого данные запоминаются так, чтобы они были независимы от прикладных программ, использующих эти данные. Для добавления новых или модификации существующих данных и для поиска данных в БД применяется стандартный набор управляющих алгоритмов.

База данных представляет собой постоянно развивающийся объект (к ней добавляются новые записи, а в существующие включаются новые элементы данных). С целью повышения эффективности функционирования БД изменяется и ее структура. Используемые на практике способы построения БД реализуются в виде иерархических (древовидных), сетевых или реляционных моделей (моделей отношений) данных.

Банки данных

Модель- совокупность структур данных и операций их обработки.

Выбор той или другой модели происходит после получения всей информации о предметной области, ее описаний и детализации. Перед началом моделирования в собранной информации выделяются объекты, их характеристики и связи между

ними. Для выбора конкретной модели первоначально предметную область желательно оценить с т.з. прямого моделирования понятий, т.е. оценивается возможность определения выделенных объектов в терминах и структурах выбранной модели данных.

Кроме возможности прямого моделирования оцениваются следующие свойства модели данных:

1. сложность модели для изучения пользователем;
2. наглядность;
3. сложность и трудоемкость написания программ для манипулирования структурами данных;
4. соблюдение правил композиции;
5. оптимальное число базисных структур и т.д.

Сетевая модель

Основные термины СМ:

1. Элемент данных – наименьшая поименованная единица данных, в которой содержится минимальный набор информации. С помощью элемента данных выполняется построение всех остальных структур (аналог поля).

2. Агрегат данных – совокупность элементов данных внутри записи, имеющая имя, причем эта совокупность рассматривается как единое целое. Агрегаты бывают простые и сложные.

3. Запись – поименованная совокупность элементов и агрегатов.

4. Набор – поименованная совокупность записей, образующих двухуровневую иерархическую структуру. Каждый тип набора представляет собой отношение (связь) между двумя или несколькими типами записи. Для каждого типа набора один тип записи м.б. объявлен владельцем набора, тогда остальные записи м.б. объявлены членами этого набора.

5. База данных – поименованная совокупность записей различного типа, содержащая ссылки между записями, представленная экземплярами наборов.

СМ строятся на основании следующих композиционных правил:

1. БД может содержать любое количество типов записей и типов наборов.
2. между двумя типами записей может быть определено любое количество типов наборов.

3. тип записи может быть одновременно и владельцем и членом различных типов наборов. СМ позволяет реализовать связи такие как «1:1», «1:M», «M:1».

СМ м.б. представлена в форме графов (структурная диаграмма данных) или с помощью таблиц (расширенная структурная диаграмма).

СМ относится к типу симметричных моделей, однако, в ней самые сложные алгоритмы по поиску информации. В сетевых моделях предусмотрены 4 основные оператора: добавить (insert), удалить (delete), найти (find), обновить (update).

При выполнении данных операций следует учитывать важность связей между объектами, особенно это касается операции «удаление», при которой может произойти удаление связующей записи. Операция «обновление» выполняется легко, т.к. каждая запись в единственном экземпляре.

Иерархическая модель

Примерами иерархических структур могут служить: справочники, классификаторы, организационные структуры.

Структура ИМ описывается в терминах:

1. Элемент данных – наименьшая поименованная единица данных, в которой содержится минимальный набор информации. С помощью элемента данных выполняется построение всех остальных структур (аналог поля).

2. Агрегат данных – совокупность данных внутри записи, имеющая имя, причем эта совокупность рассматривается как единое целое. Агрегаты бывают простые и сложные.

3. Запись – поименованная совокупность элементов и агрегатов.

4. Групповое отношение.

5. База данных – поименованная совокупность записей различного типа, содержащая ссылки между записями, представленные экземплярами наборов.

Основные ограничения ИМ:

1. возможны связи 1:M и 1:1;

2. создаваемая структура д.б. древовидной.

Исходя из этих ограничений, ИМ должна удовлетворять условия:

1. на первом уровне иерархии м.б. только один узел называемый корневым;

2. вершины (узлы) на нижних уровнях называются порожденными или зависимыми;

3. каждый порожденный узел, находящийся на i -ом уровне связан только с одним исходным узлом, находящимся на $(i - 1)$ -ом уровне;

4. каждый исходный узел может иметь 1 или несколько порожденных узлов называемых подобными;

5. существует единственный путь доступа к узлу начиная от корня дерева.

6. иерархия всегда начинается от корня.

Применение ИМ не вызовет трудностей только в том случае если данные имеют естественную древовидную структуру. В принципе каждая сетевая модель м.б. представлена средствами ИМ, при этом сеть разбивается на несколько деревьев, а каждая сетевая запись разбивается на несколько иерархических записей. ИМ наряду со своими преимуществами может вызывать затруднения при выполнении основных операций, например,:

1. невозможно добавить информацию на самый нижний уровень без введения фиктивных верхних уровней;

2. удаление средних уровней или их редактирование может привести к потере связей;

3. при поиске информации сложность вызывает асимметрия модели.

+: простота, независимость данных, min расходы памяти; -: не универсальность, избыточность.

Реляционная модель

В основе РМ использовано понятие «отношение».

Отношение – дана совокупность множеств $D_1 \dots D_n$, отношение R определенное на n -множествах, есть множество упорядоченных строк (картежей) $d_1 \dots d_n$, таких что:

D_1, D_2, \dots, D_n – домены отношений, n – степень отношений.

Домен – множество значений, из которых извлекаются фактические значения, используемые в столбце (поле это не домен в чистом виде, это маленькая его часть).

В РМ сущность представлена отношением, *атрибут* – элементом домена, связь – самостоятельным отношением либо доменом внутри отношения.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами: каждый элемент таблицы – один элемент данных; все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип и длину; каждый столбец имеет уникальное имя; одинаковые строки в таблице отсутствуют; порядок следования строк и столбцов м.б. произвольным.

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, столбцы – атрибутам отношений, доменам, полям.

Ограничения модели:

1. модель позволяет реализовать связи «1:1» и «1: M»;
2. значение каждого атрибута в каждой строке является неделимым, т.е. оно должно состоять из одного значения, а не из множеств. Если это выполняется, то отношение называется нормализованным.

РМ БД – есть представление пользователя этой БД в виде совокупности изменяемых во времени нормализованных отношений различных степеней.

Реляционная БД задается реляционной схемой, состоящей из одной или нескольких схем отношений. Схема отношения задается именем отношения и именами соответствующих доменов. При создании реляционной БД каждая схема отношений представляется в виде таблицы, каждый столбец таблицы называется атрибутом. Отношения в БД обладают свойствами множеств. В табличном представлении отношений есть ограничение: дублирование строк не допускается. Это ограничение приводит к введению понятия ключа. Ключ подмножество атрибутов, совокупность значений которых уникально идентифицирует кортеж. Отношение может иметь несколько ключей, называемых возможными ключами. Для выбора ключа отношения используют два свойства:

- 1) значения ключа уникально идентифицируют кортеж отношения, т.е. не \exists двух строк, которые имели бы равные значения атрибутов, входящих в ключ и рассматриваемых как единое целое;
- 2) никакое подмножество атрибутов ключа, которое образуется при удалении из ключа любого атрибута, не обладает свойством 1).

После выбора ключевого атрибута необходимо помнить, что первичный ключ отличается от возможных ключей по операциям, которые над ним допустимы: первичный ключ нельзя обновлять; ни один из атрибутов первичного ключа не может принимать значение «не определено».

В реляционных моделях \exists внутренние ограничения, накладываемые на атрибуты и на кортежи:

1. специфицируется область значений атрибутов, определяющая допустимый тип значений;

2. \exists возможность задания домена сравнимости, применяемого для указания осмысленности сравниваемых значений, если два атрибута не имеют общего домена сравнимости, их нельзя сравнивать или использовать в операции сравнения;

3. общие ограничения, которые м.б. заданы посредством утверждений (условий, которым должна удовлетворять БД).

Атрибут отношения R является внешним ключом, если этот атрибут не является первичным ключом отношения R, но его значения являются значениями первичного ключа отношения R1 (другого отношения).

+ : простота, независимость данных от программы их реализации, теоретическое обоснование (нормализация отношений); - : низкое быстродействие, большой расход памяти.

Расширенная (посреляционная модель)

Важным аспектом традиционной реляционной модели данных является тот факт, что элементы данных, которые хранятся на пересечении строк и столбцов таблицы, должны быть неделимы и единственны. Это значит, что данные не могут быть развернуты в процессе дальнейшей обработки. Такое правило было заложено в основу реляционной алгебры при ее разработке как математической модели данных. Дальнейшие исследования показали, что существует ряд случаев, когда ограничения классической реляционной модели существенно мешают эффективной реализации приложений.

При том, что таблицы, строки и колонки так удобно отражают наше мышление, почему же они стали самоограничивающими для более крупных приложений? В основе проблемы лежат три вопроса:

1. работа с полями переменной длины и группами записей;
2. управление отношениями между таблицами и полями;
3. и отражение подлинно семантического содержания реальных структур, которые будут моделированы базой данных.

Основной принцип реляционной модели - устранять повторяющиеся поля и группы с помощью процесса, который называется нормализацией. Так как нормализация - это простой процесс, результат часто заключается в отображении единичных файлов в реляционных таблицах. Результат как непросто в понимании, так и неэффективен в обработке.

Опыт разработки прикладных информационных систем показал, что отказ от этой установки ведет к качественно полезному расширению модели данных. Если допустить, что значение данных может само состоять из подзначений, то в результате возникает понятие многозначного поля. Проще всего рассматривать набор многозначных полей в таблице как самостоятельную вложенную (nested) таблицу.

При условии, что вложенная таблица удовлетворяет общим критериям (например, имеет уникальный ключ), естественным образом происходит расширение операторов реляционной алгебры. Такая модель данных была названа

постреляционной. Многие методологии проектирования данных позволяют определять многозначные поля и затем удалять в процессе нормализации. При этом таблицы преобразуются в первую нормальную форму или 1NF. Однако удаление многозначных полей не всегда способствует улучшению прикладных программ. В случаях, когда обычная форма доступа к полю подразумевает обращение ко всем его значениям, базе данных 1NF придется проделывать операцию соединения (Join) каждый раз, когда нужно получить соответствующие значения, хранящиеся в другой таблице. Совершенно очевидно, что в подобной ситуации хранение значений физически в многозначных полях может обеспечить более эффективный доступ к информации.

На рис. 2 наглядно продемонстрированы преимущества хранения данных в базах uniVerse, относящихся к непервой нормальной форме, перед более громоздким хранением в базах данных первой нормальной формы. Пример представляет хранение части данных такого типичного документа, как накладная.

INVNO	CUSTNO	GOODS	QTY
0373	8723	Пиво	3
0373	8723	Вобла	2
8374	8232	Лимонад	1
8374	8232	Пиво	6
8374	8232	Вафли	2
7364	8723	Йогурт	1

Рис. 2а. Представление данных в 1-й нормальной форме (1NF)

ORDER

ORDER ITEMS

INVNO	CUSTNO		INVNO	GOODS	QTY
0373	8723		0373	Пиво	3
8374	8232		0373	Вобла	2
7364	8723		8374	Лимонад	1
			8374	Пиво	6
			8374	Вафли	2
			7364	Йогурт	1

Рис. 2в. Структура хранения данных в uniVerse NFNF (NF2)

ORDER

INVNO	CUSTNO	GOODS	QTY
0373	8723	Пиво	3
		Вобла	2
8374	8232	Лимонад	1
		Пиво	6
		Вафли	2
7364	8723	Йогурт	1

Рисунок 2. Структуры хранения данных с многозначными полями в uniVerse и в традиционных системах.

О таблицах, содержащих многозначные поля, говорят, что они находятся в непервой нормальной форме, или NF2 (Non First Normal Form). Как было замечено ранее, при условии, что используемые поля подчиняются определенным правилам, позволяющим обращаться с ними, как с таблицами, встроенными в другие таблицы, форма NF2 не нарушает принципы реляционной алгебры. Более того, такая информация полностью доступна, так как расширенные операторы, которые работают с таблицами NF2, позволяют извлекать встроенные таблицы и рассматривать данные как информацию, поступившую из таблиц 1NF. И все же во многих случаях форма 1NF будет скорее исключением, а не правилом. В большинстве случаев гораздо более эффективно осуществлять доступ к многозначным полям одновременно с остальными данными, зная, что их всегда можно извлечь и рассматривать как отдельную таблицу в тех случаях, когда это может понадобиться.

Многомерная модель

Достаточно очевидно, что даже при небольших объемах данных отчет, представленный в виде двухмерной таблицы (Модели автомобиля по оси Y и Время по оси X), нагляднее и информативнее отчета с реляционной построчной формой организации (рис. 1).

Реляционная модель

Модель	Месяц	Объем
"Жигули"	Июнь	12
"Жигули"	Август	5
"Москвич	Июнь	2
"Москвич"	Июль	18
"Волга"	Июль	19

Многомерная модель

	Июнь	Июль	Август
"Жигули"	12		5
"Москвич"	2	18	
"Волга"		19	

Рисунок 1. (Реляционная и многомерная модели представления данных).

А теперь представим, что у нас не три модели, а 30 и не три, а 12 различных месяцев. В случае построчного (реляционного) представления мы получим отчет в 360 строк (30x12), который займет не менее 5-6 страниц. В случае же многомерного (в нашем случае двухмерного) представления мы получим достаточно компактную таблицу 12 на 30, которая вполне уместится на одной странице и которую, даже при таком объеме данных, можно реально оценивать и анализировать

Основными понятиями, с которыми оперирует пользователь и проектировщик в многомерной модели данных, являются:

- измерение (Dimension);
- ячейка (Cell).

Иногда вместо термина "Ячейка" используется термин "Показатель" (Measure).

Измерение - это множество однотипных данных, образующих одну из граней гиперкуба. Например - Дни, Месяцы, Кварталы, Годы - это наиболее часто используемые в анализе временные Измерения. Примерами географических измерений являются: Города, Районы, Регионы, Страны и т.д.

В многомерной модели данных Измерения играют роль индексов, используемых для идентификации конкретных значений (Показателей), находящихся в Ячейках гиперкуба.

В свою очередь, *Показатель* - это поле (обычно цифровое), значения которого однозначно определяются фиксированным набором Измерений. В зависимости от того, как формируются его значения, Показатель может быть определен, как:

Переменная (Variable) - значения таких Показателей один раз вводятся из какого-либо внешнего источника или формируются программно и затем в явном виде хранятся в многомерной базе данных (МБД)

Формула (Formula)- значения таких Показателей вычисляются по некоторой заранее специфицированной формуле. То есть для Показателя, имеющего тип Формула, в БД хранится не его значения, а формула, по которой эти значения могут быть вычислены.

Заметим, что это различие существует только на этапе проектирования и полностью скрыто от конечных пользователей. В примере на рис. 1 каждое значение поля Объем продаж однозначно определяется комбинацией полей:

Модель автомобиля;

Месяц продаж.

Но в реальной ситуации для однозначной идентификации значения Показателя, скорее всего, потребуется большее число измерений, например:

Модель автомобиля;

Менеджер;

Время (например Год).

Измерения:

Время (Год) - 1994, 1995, 1995

Менеджер - Петров, Смирнов, Яковлев

Показатель:

Объем Продаж

И в терминах многомерной модели речь будет идти уже не о двухмерной таблице, а о трехмерном гиперкубе:

первое Измерение - Модель автомобиля;

второе Измерение - Менеджер, продавший автомобиль;

третье Измерение - Время (Год);

на пересечении граней которого находятся значения Показателя Объем продаж.

Заметим, что, в отличие от Измерений, не все значения Показателей должны иметь и имеют реальные значения. Например, Менеджер Петров в 1994 г. мог еще

не работать в фирме, и в этом случае все значения Показателя Объем продаж за этот год будут иметь неопределенные значения.

Гиперкубические и поликубические модели данных

В различных МСУБД используются два основных варианта организации данных:

- Гиперкубическая модель;
- Поликубическая модель.

В чем состоит разница? Системы, поддерживающие Поликубическую модель (примером является Oracle Express Server), предполагают, что в МБД может быть определено несколько гиперкубов с различной размерностью и с различными Измерениями в качестве их граней. Например, значение Показателя Рабочее Время Менеджера, скорее всего, не зависит от Измерения Модель Автомобиля и однозначно определяется двумя Измерениями: День и Менеджер. В Поликубической модели в этом случае может быть объявлено два различных гиперкуба:

Двухмерный - для Показателя Рабочее Время Менеджера;

Трехмерный - для Показателя Объем Продаж.

В случае же Гиперкубической модели предполагается, что все Показатели должны определяться одним и тем же набором Измерений. То есть только из-за того, что Объем Продаж определяется тремя Измерениями, при описании Показателя Рабочее Время Менеджера придется также использовать три Измерения и вводить избыточное для этого Показателя Измерение Модель Автомобиля.

Операции манипулирования Измерениями

Формирование "Среза". Пользователя редко интересуют все потенциально возможные комбинации значений Измерений. Более того, он практически никогда не работает одновременно сразу со всем гиперкубом данных. Подмножество гиперкуба, получившееся в результате фиксации значения одного или более Измерений, называется Срезом (Slice). Например, если мы ограничим значение Измерения Модель Автомобиля = "BA32108", то получим подмножество гиперкуба (в нашем случае - двухмерную таблицу), содержащее информацию об истории продаж этой модели различными менеджерами в различные годы.

Операция "Вращение". Изменение порядка представления (визуализации) Измерений (обычно применяется при двухмерном представлении данных) называется Вращением (Rotate). Эта операция обеспечивает возможность визуализации данных в форме, наиболее комфортной для их восприятия. Например, если менеджер первоначально вывел отчет, в котором Модели автомобилей были перечислены по оси X, а Менеджеры по оси Y, он может решить, что такое представление мало наглядно, и поменять местами координаты (выполнить Вращение на 90 градусов).

Отношения и Иерархические Отношения.

В нашем примере значения Показателей определяются только тремя измерениями. На самом деле их может быть гораздо больше и между их значениями обычно существуют множество различных Отношений (Relation) типа

"один ко многим". Например, каждый Менеджер может работать только в одном подразделении, а каждой модели автомобиля однозначно соответствует фирма, которая ее выпускает:

Менеджер -> Подразделение;

Модель Автомобиля -> Фирма-Производитель.

Заметим, что для Измерений, имеющих тип Время (таких как День, Месяц, Квартал, Год), все Отношения устанавливаются автоматически, и их не требуется описывать. В свою очередь, множество Отношений может иметь иерархическую структуру - Иерархические Отношения (Hierarchical Relationships). Вот только несколько примеров таких Иерархических Отношений:

День -> Месяц -> Квартал -> Год;

Менеджер -> Подразделение -> Регион -> Фирма -> Страна;

Модель Автомобиля -> Завод-Производитель -> Страна.

И часто более удобно не объявлять новые Измерения и затем устанавливать между ними множество Отношений, а использовать механизм Иерархических Отношений. В этом случае все потенциально возможные значения из различных Измерений объединяются в одно множество. Например, мы можем добавить к множеству значений Измерения Менеджер ("Петров", "Сидоров", "Иванов", "Смирнов"), значения Измерения Подразделение ("Филиал 1", "Филиал 2", "Филиал 3") и Измерения Регион ("Восток", "Запад") и затем определить между этими значениями Отношение Иерархии.

Операция Агрегации.

С точки зрения пользователя, Подразделение, Регион, Фирма, Страна являются точно такими же Измерениями, как и Менеджер. Но каждое из них соответствует новому, более высокому уровню агрегации значений Показателя Объем продаж. В процессе анализа пользователь не только работает с различными Срезами данных и выполняет их Вращение, но и переходит от детализированных данных к агрегированным, т.е. производит операцию Агрегации (Drill Up). Например, посмотрев, насколько успешно в 1995 г. Петров продавал модели "Жигули" и "Волга", управляющий может захотеть узнать, как выглядит соотношение продаж этих моделей на уровне Подразделения, где Петров работает. А затем получить аналогичную справку по Региону или Фирме. *Операция Детализации.*

Переход от более агрегированных к более детализированным данным называется операцией Детализации (Drill Down). Например, начав анализ на уровне Региона, пользователь может захотеть получить более точную информацию о работе конкретного Подразделения или Менеджера.

Объектная модель

Сразу же необходимо заметить, что общепринятого определения "объектно-ориентированной модели данных" не существует. Сейчас можно говорить лишь о некоем "объектном" подходе к логическому представлению данных и о различных объектно-ориентированных способах его реализации.

Мы знаем, что любая модель данных должна включать три аспекта: структурный, целостный и манипуляционный. Посмотрим, как они реализуются на основе объектно-ориентированной парадигмы программирования:

Структура: Структура объектной модели описывается с помощью трех ключевых понятий:

инкапсуляция- каждый объект обладает некоторым внутренним состоянием (хранит внутри себя запись данных), а также набором методов - процедур, с помощью которых (и только таким образом) можно получить доступ к данным, определяющим внутреннее состояние объекта, или изменить их. Таким образом, объекты можно рассматривать как самостоятельные сущности, отделенные от внешнего мира.

наследование - подразумевает возможность создавать из классов объектов новые классы объектов, которые наследуют структуру и методы своих предков, добавляя к ним черты, отражающие их собственную индивидуальность. Наследование может быть простым (один предок) и множественным (несколько предков).

полиморфизм - различные объекты могут по-разному реагировать на одинаковые внешние события в зависимости от того, как реализованы их методы.

Пример:

```
Begin
Point A(100,100);
CircleB(200,200,50);
A.Draw(); // рисует точку
B.Draw(); // рисует окружность
End.
```

Целостность данных:

Для поддержания целостности объектно-ориентированный подход предлагает использовать следующие средства:

автоматическое поддержание отношений наследования
возможность объявить некоторые поля данных и методы объекта как "скрытые", не видимые для других объектов; такие поля и методы используются только методами самого объекта
создание процедур контроля целостности внутри объекта

Средства манипулирования данными:

К сожалению, в объектно-ориентированном программировании отсутствуют общие средства манипулирования данными, такие как реляционная алгебра или реляционное счисление. Работа с данными ведется с помощью одного из объектно-ориентированных языков программирования общего назначения, обычно это SmallTalk, C++ или Java.