

Содержание и задачи курса

Дисциплина ПРОГРАММНЫЕ СРЕДСТВА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

” ставит целью изложение современного подхода к высокопроизводительному и качественному проектированию на основе широкого использования средств вычислительной техники.

При изучении курса студенты знакомятся с идеологией машинного проектирования, включающей совокупность всех видов проектной, конструкторской и технологической деятельности инженера.

В результате изучения дисциплины студенты должны знать:

- характерные особенности автоматизации проектирования на базе средств вычислительной техники;
- требования, которым должна удовлетворять система автоматизированного проектирования (САПР);
- структуру программного обеспечения САПР;
- результаты машинного проектирования .

После изучения курса студенты должны уметь:

- объяснять схему функционирования САПР;
- составлять этапы автоматизированного проектирования;
- выбирать аппаратные средства и практически целесообразные варианты функциональных схем САПР .

Объем аудиторных занятий по курсу - 214 часов, из них лекций - 36 часов, лабораторных 72 часа, курсовая работа - 90 часов.

Учебно-методические материалы

1. Мясников В.А., Игнатъев М.Б., Покровский А.М. Программное управление оборудованием Л.:Машиностроение, 1984.- 427с.
2. Четвериков В.Н., Ревунков Г.И. Базы и банки данных. - М.: Высш. шк., 1987.- 248с.
3. Жук К.Д и др. Построение современных систем автоматизированного проектирования Киев: Наук. думка, 1983.- 248с.
4. Кречко Ю.А., Полищук В.В. Автокад. Курс практической работы.- М.: Диалог-МИФИ, 1994.- 256с.
5. Бемер С. FoxPro 2.6 для Windows.-К.: BNV, 1995.-464с.
6. Поликарпов Ю.В., Акмаев И.И. AutoCad:Осваиваем графический редактор. - Уфа: УГАТУ, 1995-87с.
7. Николь Н., Альбрехт Р. Exel 5.0. Электронные таблицы.- М.: Эком, 1994.- 352с.
8. Зайцев А. Н. Курс лекций. САПР в сварочном производстве.
9. И. П. Норенков Автоматизированное проектирование. М:- 2000г.

Лекция 1

1.1. Введение в САПР

Автоматизация проектирования занимает особое место среди информационных технологий. Во первых, автоматизация проектирования — синтетическая дисциплина, ее составными частями являются многие другие современные информационные технологии. Так, техническое обеспечение систем автоматизированного проектирования (САПР) основано на использовании вычислительных сетей и телекоммуникационных технологий, в САПР используются персональные компьютеры и рабочие станции.

Математическое обеспечение САПР отличается богатством и разнообразием используемых методов вычислительной математики, статистики, математического программирования, дискретной математики, искусственного интеллекта. Программные комплексы САПР относятся к числу наиболее сложных современных программных систем, основанных на операционных системах Unix, Windows 95/NT, языках программирования C, C++, Java и других, современных CASE технологиях, реляционных и объектно-ориентированных системах управления базами данных (СУБД), стандартах открытых систем и обмена данными в компьютерных средах.

Во вторых, знание основ автоматизации проектирования и умение работать со средствами САПР требуется практически любому инженеру разработчику. Компьютерами насыщены проектные подразделения, конструкторские бюро и офисы. Работа конструктора за обычным кульманом, расчеты с помощью логарифмической линейки или оформление отчета на пишущей машинке стали анахронизмом. Предприятия, ведущие разработки без САПР или лишь с малой степенью их использования, оказываются неконкурентоспособными как из-за больших материальных и временных затрат на проектирование, так и из-за невысокого качества проектов. Появление первых программ для автоматизации проектирования за рубежом и в СССР относится к началу 60-х гг. Тогда были созданы программы для решения задач строительной механики, анализа электронных схем, проектирования печатных плат.

Дальнейшее развитие САПР шло по пути создания аппаратных и программных средств машинной графики, повышения вычислительной эффективности программ моделирования и анализа, расширения областей применения САПР, упрощения пользовательского интерфейса, внедрения в САПР элементов искусственного интеллекта.

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

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

1.2. Понятие инженерного проектирования

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

Обычно ТЗ представляют в виде некоторых документов. Результатом проектирования, как правило, служит полный комплект документации, содержащий достаточные сведения для изготовления объекта в заданных условиях. Эта документация и есть проект или точнее описание объекта. Более коротко, проектирование — процесс, заключающийся в получении и

преобразовании исходного описания объекта в окончательное описание на основе выполнения комплекса работ исследовательского, расчетного и конструкторского характера.

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

Проектирование, при котором все проектные решения или их часть получают путем взаимодействия человека и ЭВМ, называют **автоматизированным** в отличие от *ручного* (без использования ЭВМ) или **автоматического** (без участия человека на промежуточных этапах). Система, реализующая автоматизированное проектирование, представляет собой **систему автоматизированного проектирования (в англоязычном написании CAD System — Computer Aided Design System)**.

Автоматическое проектирование возможно лишь в отдельных частных случаях для сравнительно несложных объектов. Превалирующим в настоящее время является автоматизированное проектирование.

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

1.3. Системный подход к проектированию

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

Основной общий принцип системного подхода заключается в рассмотрении частей явления или сложной системы с учетом их взаимодействия. **Системный подход выявляет структуру системы ее внутренние и внешние связи.**

Системы автоматизированного проектирования и управления относятся к числу наиболее сложных современных искусственных систем. Их проектирование и сопровождение невозможны без системного подхода. Поэтому идеи и положения системотехники входят составной частью в дисциплины, посвященные изучению современных автоматизированных систем и технологий их применения.

1.4. Системы автоматизированного проектирования и их место среди других автоматизированных систем

Структура САПР. Как и любая сложная система, САПР состоит из подсистем»: Различают подсистемы проектирующие и обслуживающие.

Проектирующие подсистемы непосредственно выполняют проектные процедуры. Примерами проектирующих подсистем могут служить подсистемы геометрического трехмерного моделирования механических объектов, изготовления конструкторской документации, схемотехнического анализа, трассировки соединений в печатных платах.

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

1.5. Виды обеспечения: САПР

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

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

Математическое (МО), объединяющее математические методы, модели и алгоритмы для выполнения проектирования;

Программное (ПО), представляемое компьютерными программами САПР;

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

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

Методическое (МетО), включающее различные методики проектирования, иногда к МетО относят также математическое обеспечение;

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

1.6. Разновидности САПР

Классификацию САПР делают по приложению, целевому назначению, масштабам (комплексности решаемых задач), характеру базовой подсистемы и т. д.

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

- САПР для применения в отраслях общего машиностроения. Их часто называют машиностроительными САПР или MCAD (Mechanical CAD) системами;
- САПР для радиоэлектроники. Их названия — ECAD (Electronic CAD) или EDA (Electronic Design Automation) системы.
- САПР в области архитектуры и строительства;

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

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

- САПР функционального проектирования, иначе САПР-Ф или CAE (Computer Aided Engineering) системы.
- Конструкторские* САПР общего машиностроения — САПР-К, часто называемые просто CAD системами;
- Технологические* САПР общего машиностроения — САПР-Т, иначе называемые автоматизированными системами технологической подготовки производства АСТПП или системами САМ (Computer Aided Manufacturing).

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

По **характеру базовой подсистемы** различают следующие разновидности САПР:

- САПР на базе машинной графики и математического моделирования. Эти САПР ориентированы на приложения, где основной процедурой проектирования является конструирование, т.е. определение пространственных форм и взаимного расположения объектов. Поэтому к этой группе систем относится большинство графических ядер САПР в области машиностроения.
- САПР на базе СУБД. Они ориентированы на приложения, в которых при сравнительно несложных математических расчетах перерабатывается большой объем данных. Такие САПР преимущественно встречаются в технико-экономических приложениях, например, при проектировании бизнес-планов, но имеют место также при проектировании объектов, подобных щитам управления в системах автоматики.

- САПР на базе конкретного прикладного пакета. Фактически это автономно используемые программно-методические комплексы, например, имитационного моделирования производственных процессов, расчета прочности по методу конечных элементов, синтеза и анализа систем автоматического управления и т.п. Часто такие САПР относятся к системам САЕ. Примерами могут служить программы логического проектирования на базе языка VHDL, математические пакеты типа MathCAD.
- Комплексные (интегрированные) САПР, состоящие из совокупности подсистем предыдущих видов. Характерными примерами комплексных САПР являются САЕ/CAD/CAM-системы в машиностроении или САПР БИС. Так, САПР БИС включает в себя СУБД и подсистемы проектирования компонентов, принципиальных, логических и функциональных схем, топологии кристаллов, тестов для проверки годности изделий. Для управления столь сложными системами применяют специализированные *системные среды*.

Лекция 2

2.1. Техническое обеспечение САПР

С точки зрения системной модели САПР, техническое обеспечение представляет собой самый нижний уровень, в который “погружается” и реализуется операционно-программное и другие виды обеспечений САПР.

Задача проектирования технического обеспечения, таким образом, может быть сформулирована как задача оптимального выбора состава технических средств САПР. Исходной информацией при этом являются результаты анализа задач внутреннего проектирования и ресурсные требования к техническим средствам в виде критериев и ограничений.

Основные требования к техническим средствам САПР состоят в следующем:

- эффективность;
- универсальность;
- совместимость;
- надежность.

Технические средства (ТС) в САПР решают задачи:

- ввода исходных данных описания объекта проектирования;
- отображения введенной информации с целью ее контроля и редактирования;
- преобразования информации (изменения формы и структуры представления данных, перекодировки и др.);
- хранения информации;
- отображения итоговых и промежуточных результатов решения;
- оперативного общения проектировщика с системой в процессе решения задач.

Для решения этих задач ТС должны содержать:

- процессоры,
- оперативную память,
- внешние запоминающие устройства,
- устройства ввода-вывода информации,
- технические средства машинной графики,
- устройства оперативного общения человека с ЭВМ,
- устройства, обеспечивающие связь ЭВМ с удаленными терминалами и другими машинами.

При необходимости создания непосредственной связи САПР с производственным оборудованием в состав ТС должны быть включены устройства, преобразующие результаты проектирования в сигналы управления станками.

ТС САПР могут одно- и многоуровневыми.

ТС, в состав которых входит одна ЭВМ, оснащенная широким набором периферийного оборудования, носят название одноуровневых. Они широко применяются при проектировании изделий общепромышленного применения с установившейся конструкцией, имеющих узкоспециализированные математические модели и фиксированную последовательность этапов проектно-технологических работ.

Развитие САПР предполагает расширение набора терминальных устройств, представление каждому проектировщику возможности взаимодействия с ЭВМ, обработку технической информации непосредственно на рабочих местах. С этой целью терминальные устройства снабжаются мини- и микроЭВМ, имеющими специальное математическое обеспечение интеллектуальные терминалы. Они соединяются с ЭВМ высокой производительностью с помощью специальных или обычных телефонных каналов.

Для использования информации отдельных ЭВМ распределенных на относительно большой территории особый эффект дает применение вычислительных сетей.

Отличительные признаки вычислительной сети состоят в следующем:

- большое число взаимодействующих друг с другом вычислительных машин, выполняющих функции сбора, хранения, передачи, обработки и выдачи информации;
- чрезвычайно большие вычислительные мощности;
- распределенная обработка информации;

- надежная и гибкая связь пользователя с вычислительными мощностями;
- возможность взаимного обмена информацией между вычислительными машинами;
- расширение до любой мощности и протяженности.

Лекция 3

Математическое обеспечение САПР

3.1. Общие положения

Математическое обеспечение (МО) объединяет в себе математические модели проектируемых объектов, методы и алгоритмы выполнения проектных процедур, используемые при автоматизированном проектировании.

Элементы МО чрезвычайно многообразны, среди них имеются инвариантные элементы, широко применяемые в различных САПР.

К ним относятся принципы построения функциональных моделей, методы численного решения алгебраических и дифференциальных уравнений, постановки экстремальных задач, поиска экстремума.

Специфика предметных областей проявляется, прежде всего, в математических моделях (ММ) проектируемых объектов, она заметна также в способах решения задач структурного синтеза. Формы представления МО также разнообразны, но его практическое использование происходит после реализации ПО.

3.2. Математические модели

3.2.1. Требования к математическим моделям

Требования к математическим моделям:

- универсальность;
- адекватность;
- точность;
- экономичность.

Степень универсальности ММ характеризует полноту отображения в модели свойств реального объекта.

Точность ММ оценивается степенью совпадения значений параметров реального объекта и значений тех же параметров, рассчитанных с помощью оцениваемой ММ.

Например, пусть $Y = (y_1, y_2, \dots, y_m)$ - вектор входных параметров, тогда относительная погрешность расчета j -го параметра может быть оценена по формуле

$$\varepsilon_j = (y_{jm} - y_{j\text{н\text{м}}}) / y_{j\text{н\text{м}}}, \quad (3.1)$$

где y_{jm} , $y_{\text{н\text{м}}}$ - значения выходного параметра истинное и рассчитанное по математической модели.

Адекватность ММ - способность отражать заданные свойства объекта с погрешностью не выше заданной. Адекватность ММ, как правило, имеет место лишь в ограниченной области изменения внешних параметров - в области адекватности (ОА):

$$OA = \{Q | \varepsilon_m \leq \delta\}, \quad (3.2)$$

где $\delta > 0$ - заданная константа, равная предельно допустимой погрешности ММ; Q - вектор внешних параметров.

Экономичность модели характеризуется затратами вычислительных ресурсов (времени и памяти) на ее реализацию.

3.2.2. Классификация математических моделей

ММ классифицируются по следующим признакам:

- характер отображаемых свойств объекта;
- принадлежность к иерархическому уровню;
- степень детализации описания внутри одного уровня;
- способ получения модели.

По характеру отображаемых свойств объекта ММ делятся на структурные и функциональные.

Различают структурные топологические и геометрические ММ.

В топологических ММ отображают состав и взаимосвязи элементов объекта. Эти ММ чаще применяют для описания объектов, состоящих из большого числа элементов, например, при решении задач привязки конструктивных элементов к определенным пространственным позициям или относительным моментам времени при разработке технологических процессов.

В геометрических ММ отображаются геометрические свойства объектов, в них дополнительно к сведениям о взаимном расположении объектов содержатся сведения о форме деталей. Геометрические модели могут выражаться, например, совокупностью уравнений линий и поверхностей.

Функциональные математические модели предназначены для отображения физических и информационных процессов, протекающих в объекте при его функционировании или изготовлении.

Использование блочно-иерархического подхода к проектированию приводит к появлению иерархии математических моделей проектируемых объектов.

В зависимости от места в иерархии описаний математические модели делятся на ММ микро-, макро- и метауровня.

Особенностью ММ на микроуровне является отражение физических процессов, протекающих в непрерывном пространстве и времени. Типичными ММ этого уровня являются дифференциальные уравнения в частных производных. В них независимыми переменными являются пространственные координаты и время.

ММ на макроуровне используют укрупненную дискретизацию пространства по функциональному признаку, что приводит к представлению ММ на этом уровне в виде систем обыкновенных дифференциальных уравнений.

На метауровне в качестве элементов принимают достаточно сложные совокупности деталей. Метауровень характеризуется большим разнообразием типов используемых ММ. Здесь ММ также представляются в виде систем обыкновенных дифференциальных уравнений. В этих моделях не описываются внутренние для элементов фазовые переменные, а фигурируют только фазовые переменные, относящиеся к взаимным связям элементов.

По способу представления свойств объектов функциональные модели делятся на аналитические и алгоритмические.

Аналитические ММ представляют собой явные выражения выходных параметров как функций входных и внутренних, т.е. имеют вид:

$$Y=F(X, Q), \quad (3.3)$$

где $Y=(y_1, y_2, \dots, y_m)$ - вектор выходных параметров;

$X=(x_1, x_2, \dots, x_n)$ - вектор внутренних параметров;

$Q=(q_1, q_2, \dots, q_l)$ - вектор внешних параметров.

Аналитические модели характеризуются высокой экономичностью, однако их получение возможно лишь в частных случаях и, как правило, при принятии существенных допущений и ограничений, снижающих точность и сужающих адекватность модели.

Алгоритмические модели выражают связи выходных параметров с параметрами внутренними и внешними в форме алгоритма.

Для получения моделей используют неформальные и формальные методы.

Неформальные методы используют на различных иерархических уровнях для получения ММ элементов. Формальные методы применяют для получения ММ систем при известных математических моделях элементов.

3.2.3. Методика получения математических моделей

В общем случае методика получения ММ включает в себя следующие операции:

1. Выбор свойств объекта, которые подлежат отражению в модели;
2. Сбор исходной информации о выбранных свойствах объекта;
3. Синтез структуры ММ;
4. Расчет числовых значений параметров ММ. Эта задача ставится как задача минимизации погрешности модели заданной структуры, т.е.

$$\min \square_M(X),$$

$X \in X_d$

где X - вектор параметров ММ; X_d - область варьирования параметров; σ_M - погрешность ММ(см.3.1);

5. Оценка точности и адекватности ММ.

Лекция 4

Математическое обеспечение САПР

4.1. Алгоритмы выполнения проектных процедур

Как отмечалось ранее схема процесса проектирования в САПР может быть представлена следующей схемой рис.4.1.

4.2. Постановка и решение задач анализа

Рассмотрим математическую постановку типовых проектных задач анализа.

Анализ динамических процессов выполняется путем решения обыкновенных дифференциальных уравнений (с известными начальными условиями) вида:

$$F(dU/dt, V, t)=0 \quad (4.1)$$

где $V=(U,W)$ - вектор фазовых переменных; U - вектор, характеризующий запасы энергии в элементах объекта; t - время.

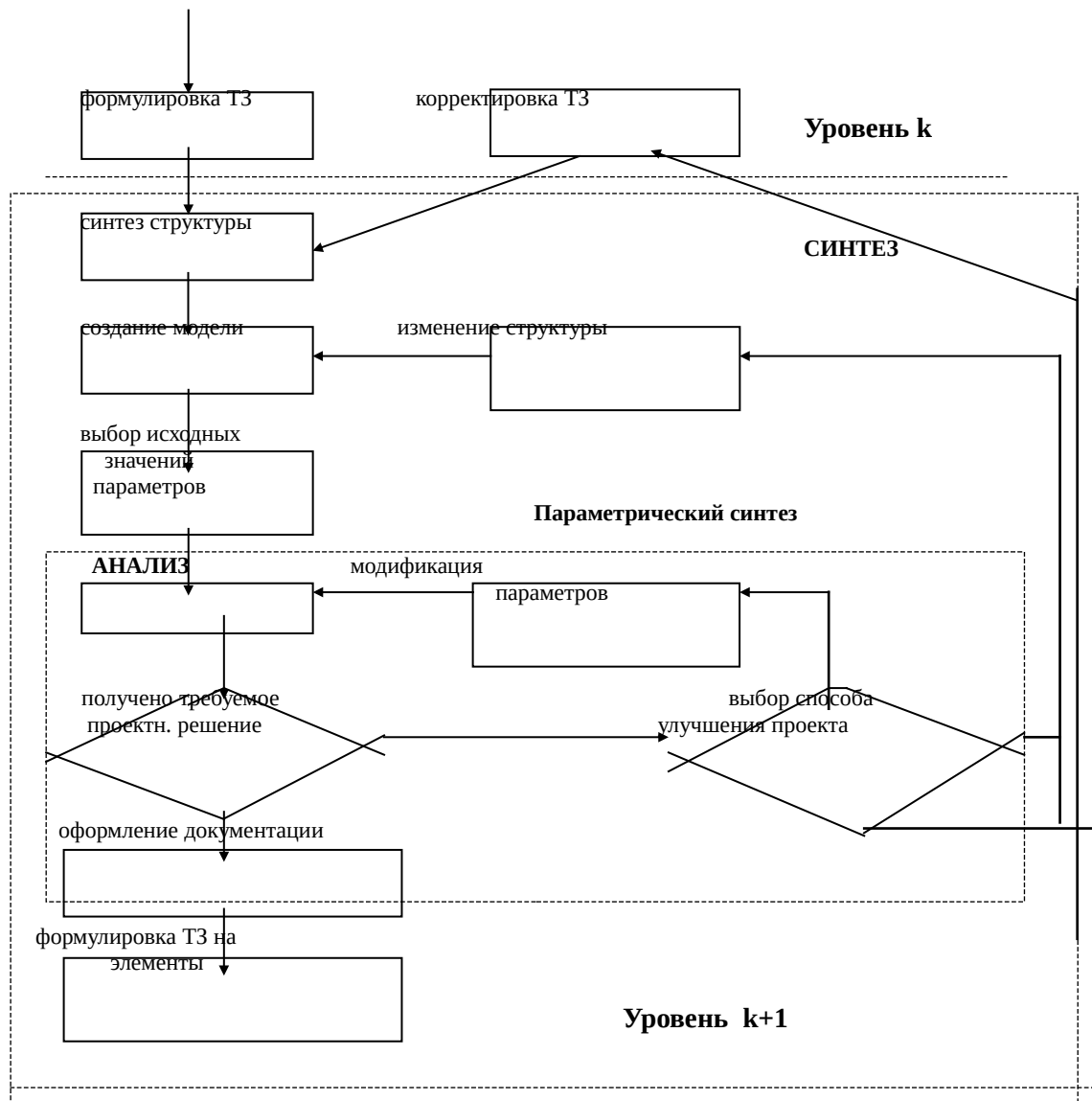


Рис.4.1. Схема процесса проектирования

Решение ОДУ позволяет получить зависимость вектора фазовых переменных $V=(U,W)$ от t в табличной форме.

Большинство выходных параметров Y проектируемых объектов являются функционалами зависимостей $V(t)$, например определенных интегралами, экстремальными значениями и др. Решение системы(4.1)

и расчет выходных параметров- функционалов составляют содержание процедуры анализа переходных процессов.

Анализ статических состояний объектов также может быть выполнено путем интегрирования уравнений типа(4.1), но, поскольку в статике $dU/dt=0$, такой анализ может быть сведен к решению систем алгебраических уравнений

$$F(V)=0 . \quad (4.2)$$

При проектировании САУ важное значение имеет задача анализа устойчивости.

Анализ чувствительности заключается в определении внутренних и внешних параметров x_i на выходные y_j . Количественная оценка этого влияния представляется матрицей чувствительности A с элементами $a_{ij}=dy_j/dx_i$.

Статистический анализ выполняется с целью получения тех или иных о распределении параметров y_j при задании статистических сведений о параметрах x_i . Результаты статистического анализа могут быть представлены в виде гистограмм распределения y_j , оценок числовых характеристик распределений мат. ожидания, дисперсии и т.д.

4.3 Постановка и решение задач синтеза

4.3.1.Классификация задач параметрического синтеза

К задачам параметрического синтеза относится совокупность задач, связанных с определением требований к параметрам объекта, номинальных значений параметров и их допусков. Задачи параметрического синтеза могут быть классифицированы на 3 группы:

- назначение технических требований;
- расчет параметров элементов;
- идентификация математических моделей.

Группа 1 задач параметрического синтеза связана с назначением технических требований к выходным параметрам объекта.

Группа 2 задач параметрического синтеза связана с расчетом параметров элементов объекта при заданной структуре объекта.

Группа 3 задач параметрического синтеза связана с определением параметров используемых в САПР математических моделей и определением областей их адекватности.

Большинство задач параметрического синтеза элементов сводится к решению задач математического программирования.

Задача математического программирования формулируется следующим образом:

$$\text{extr } F(X), \quad X \in X_D \quad (4.2)$$

т.е. нужно найти экстремум целевой функции $F(X)$, которая называется функцией качества, в пределах допустимой области X_D изменения управляемых параметров X .

Область X_D может задаваться совокупностью ограничений типа неравенств $\varphi(x) \geq 0$

и типа равенств

$$\psi(x) = 0 .$$

4.3.2. Классификация задач структурного синтеза

Задачи структурного синтеза классифицируют по следующим признакам:

- в зависимости от стадии проектирования, на которой производится синтез;
- в зависимости от возможностей формализации;
- по типу синтезируемых структур.

Рассмотрим эти задачи более подробно.

В зависимости от стадии проектирования, на которой производится синтез, различают процедуры:

- выбор основных признаков функционирования будущего объекта;
- выбор технического решения в рамках заданных принципов функционирования;
- оформления технической документации.

В зависимости от возможностей формализации задачи синтеза делятся на несколько уровней сложности.

К 1-му уровню сложности относят задачи, в которых требуется выполнение лишь параметрического синтеза, а структура объекта определена либо спецификой ТЗ, либо результатами процедур, выполненных на предыдущих этапах проектирования.

К 2-му уровню сложности относят задачи, в которых возможен полный перебор известных решений, т.е. это комбинаторные задачи выбора элементов в конечных множествах малой мощности.

К 3-му уровню сложности относят комбинаторные задачи, которые при существующих технических и программных средствах не могут быть решены путем полного перебора за приемлемое время.

К 4-му уровню сложности относят задачи поиска вариантов структур в счетных множествах неизвестной или неограниченной мощности. Формализация этих задач создает наибольшие трудности, но содержит потенциальную возможность получения новых оригинальных патентоспособных решений.

К 5-му уровню относят задачи синтеза, решение которых является проблематичным. Главная проблема здесь заключается в нахождении принципиально новых основ построения целого класса технических объектов.

Лекция 5

Информационное обеспечение САПР

5.1. Характеристика входного и выходного информационного массива

Процесс проектирования может быть представлен организованной последовательностью преобразования информации.

Исходная (входная) информация включает большое количество данных: стандарты, нормативы, каталоги комплектующих изделий и материалов, методики проектирования, сведения, содержащиеся в ТЗ, результаты предпроектных исследований.

С целью систематизации и облегчения анализа исходная информация группируется по классам, например:

- информация справочного характера (стандарты, каталоги, справочники, книги, отчеты);
- данные прототипов объектов проектирования;
- методики проектирования;
- специфические условия и требования к конкретному объекту проектирования.

Первые три класса являются неизменными на сравнительно длительном отрезке времени, значительно превышающем длительность процесса проектирования. Информация четвертого класса меняется от объекта к объекту.

Выходная информация охватывает все данные проекта, полученные на промежуточных и конечных этапах проектирования. Она определяет конструкторско-технологическую и монтажную реализацию объекта проектирования; параметры, процессы и режимы его функционирования и т.п. С другой стороны, выходная информация должна содержать критериальные оценки проекта, необходимые для введения коррекций в принятые решения.

5.2. Информационное обеспечение и информационный фонд САПР

Информационный фонд (ИФ) - совокупность данных, используемых всеми компонентами САПР.

Назначение информационного обеспечения (ИО) САПР- реализация информационных потребностей всех составных компонентов САПР. Основная функция ИО САПР - ведение информационного фонда, т.е. обеспечение создания, поддержки и организации доступа к данным. Таким ИО САПР есть совокупность информационного фонда и средств его ведения.

5.3. Состав информационного фонда САПР

Программные модули хранятся в виде символических и объектных текстов их потребителями являются мониторы различных подсистем САПР.

Исходные и результирующие данные необходимы при выполнении программных модулей в процессе преобразования.

Нормативно- справочная проектная документация, как правило представляет собой хорошо структурированный фактографический материал.

Содержание экранов дисплеев - представляет собой связанную совокупность данных, задающих форму кадра и, следовательно, позволяющих отобразить на экран дисплея информацию с целью организации диалогового взаимодействия в ходе проектирования. Обычно эти данные имеют фиксированный размер, и занимают промежуточное место между программными модулями и исходными данными; используются диалоговыми системами САПР.

Текущая проектная документация отражает состояние и ход выполнения проекта.

5.4. Способы ведения информационного фонда САПР

Различают следующие способы ведения информационного фонда САПР:

1. использование файловой системы;
2. построение библиотек;
3. использование банков данных (БнД);
4. создание информационных программных адаптеров.

Способы 1 и 2 широко распространены в организации информационного фонда вычислительных систем, поскольку поддерживаются средствами операционных систем (ОС). Однако для обеспечения быстрого доступа к справочным данным, хранения меняющихся данных, организации взаимодействия между разноязыковыми модулями эти способы малопригодны.

Способ 3- использование БД (рис.5.1) позволяет:

- централизовать информационный фонд САПР;
- произвести структурирование данных в форме удобной для проектировщика;
- обеспечить поиск информационно- справочной и проектной документации;
- упростить организацию межмодульного интерфейса путем унификации промежуточных данных.

Способ 4- предполагает использование специальных систем и программных технологий для организации межмодульного интерфейса и построения крупных программных комплексов из готовых модулей.

Лекция 6

Информационное обеспечение САПР (продолжение)

6.1. Принципы построения банков данных (БНД)

Банк данных (БНД) - совокупность базы данных (БД) и системы управления базами данных (СУБД).

База данных - структурированная совокупность данных. Наименьшая единица описания данных называется *элементом описания*. Совокупность элементов описания, объединенных отношением принадлежности к одному объекту, называется *записью*.

СУБД - состоит из языковых и программных средств, предназначенных для создания и использования базы, данных прикладными программами, а также непосредственно пользователями- непрограммистами.

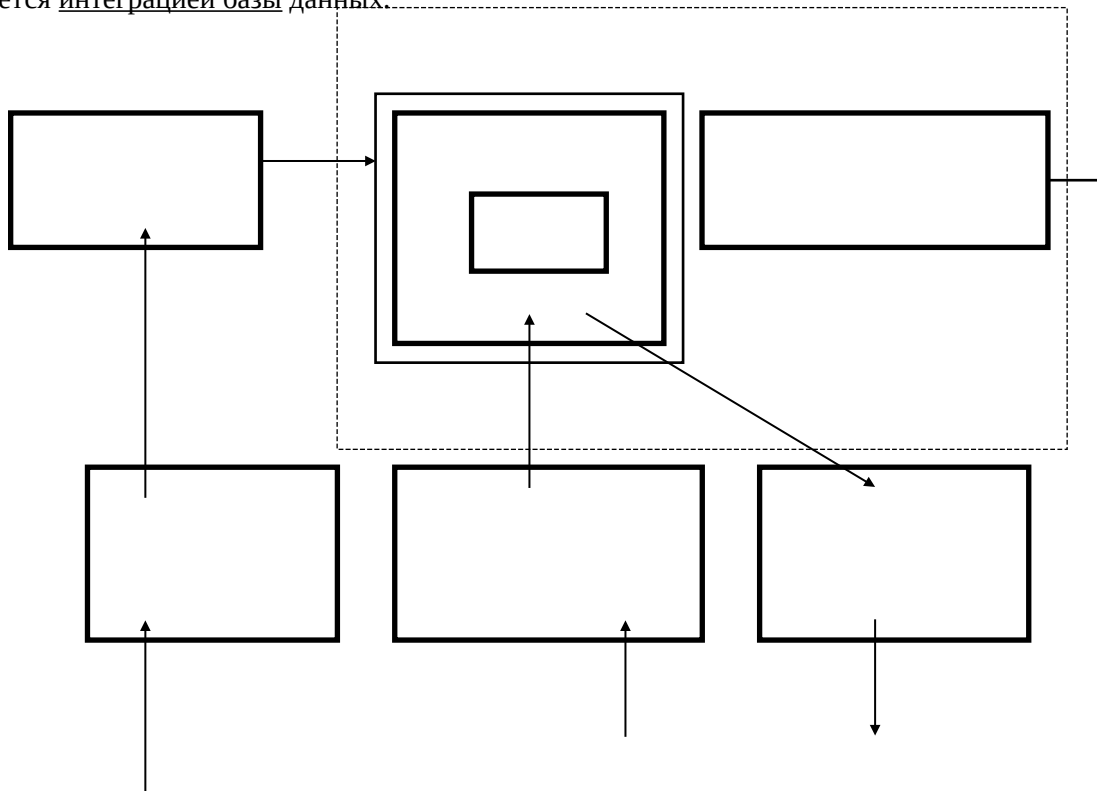
Применение БД позволяет решить следующие проблемы организации и ведения больших массивов информации:

1. сокращение избыточности;
2. обеспечение целостности;
3. разграничение доступа;
4. обеспечение независимости представления данных.

Избыточность вызывается наличием разных форм представления одних и тех же данных, размножением части данных для дальнейшего использования прикладными программами, повторными записями одинаковых данных на различных носителях информации.

Целостностью называется свойство БД в любой момент содержать лишь достоверные данные. Наличие избыточных данных, противоречивых и неверно составленных данных нарушает целостность БД.

Для сокращения избыточности производится объединение одинаковых по смыслу, но имеющих различный тип данных в единую БД с приведением к общему, стандартизованному виду. Процесс объединения данных, используется различными пользователями, в одну общую БД - называется интеграцией базы данных.



источники информации для САПР

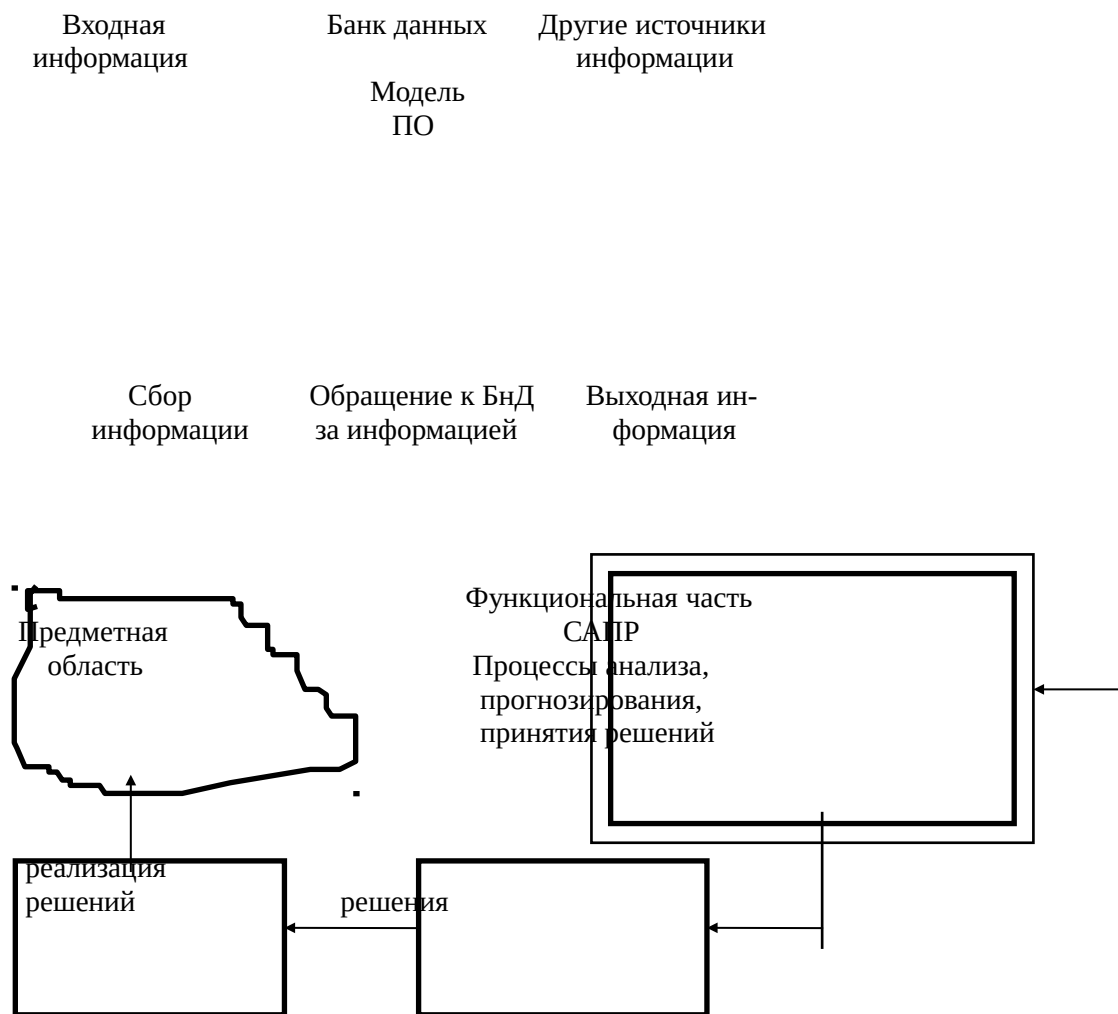


Рис.5.1. БНД в составе САПР

Каждый конкретный пользователь получает доступ к некоторому подмножеству данных из БД, необходимых для выполнения своих прикладных программ. Одновременно с этим обеспечивается режим секретности и повышается степень защищенности данных от несанкционированного доступа.

Одним из важнейших преимуществ применения БД является возможность обеспечения независимости представления данных в прикладных программах от типов запоминающих устройств и способов их физической организации. В основном это достигается построением двух уровней представления данных:

- логического;
- физического.

На логическом уровне данные представляются в виде, удобном для использования в прикладных программах или непосредственно проектировщиками.

Физический уровень представления данных отражает способ хранения и структуру данных с учетом их расположения на носителях информации в запоминающих устройствах ЭВМ.

Важнейшим понятием в БНД является модель данных - формализованное описание, отражающее состав и типы данных, а также взаимосвязи между ними. Модели данных классифицируются по ряду признаков.

В зависимости от объема описываемой информации на логическом уровне различают внешнюю и внутреннюю модели данных.

Внешняя модель (или логическая подхема) - описывает структуру информации, относящейся к конкретной процедуре или группе родственных процедур.

Внутренняя логическая модель данных объединяет все внешние модели (логические подсхемы) БД.

По способам отражения связей между данными на логическом уровне различают модели - иерархическую, сетевую и реляционную. Модель называют сетевой, если данные и связи между ними имеют структуру графа. Если структура отражаемых связей представляется в виде

дерева, то модель называют иерархической. Представление данных в виде таблиц соответствует реляционной модели данных.

Задание модели данных в БД осуществляется на специальном языке описания данных (ЯОД). Прикладные программы, использующие БД, записываются на некотором алгоритмическом языке (например, Паскаль, Си), называемом включающим языком. Для обеспечения взаимодействия с БД в эти программы должны быть введены операторы обращения к СУБД. Совокупность операторов обращения к СУБД- язык манипулирования данными (ЯМД).

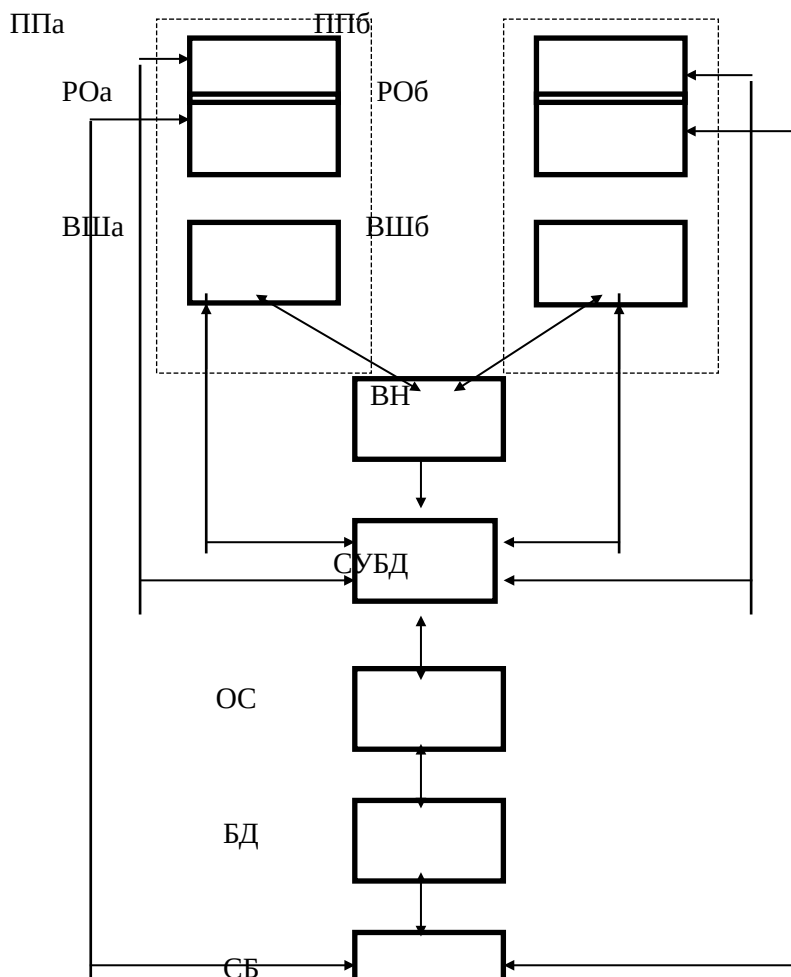


Рис.6.1.Схема взаимосвязи СУБД с прикладными программами: ППа, ППб - прикл. progr. пользователей; РОа, РОб - рабочие области пользователей; ВШа, ВШб-внешние модели данных пользователей; ВН-внутренняя логическая модель данных; СБ-системный буфер

Основные операции с данными, выполняемыми на ЯМД следующие:

- поиск информации по заданным поисковым признакам в БД;
- включение в БД новых записей;
- удаление из БД лишних или ненужных в дальнейшем записей;
- изменение значений элементов данных в записях.

Банк данных - сложная информационно- программная система, функционирование которой невозможно выполнить полностью в автоматическом режиме. Контроль за ее состоянием и управление режимами осуществляется человеком- администратором банка данных.

Взаимосвязь БД с прикладными программами представлена на рис.6.1. Прикладные программы пользователей а и б обращаются с запросами в СУБД, которая, пользуясь информацией

о конкретной внешней модели и основываясь на описании логической схемы БД, формирует обращение к программным средствам метода доступа ОС.

Полученные данные поступают вначале в системный буфер, а затем поступают в доступную пользователю рабочую область.

Совокупность модели данных и операций, определенных над данными, называется подходом. В соответствии с моделями данных различают реляционный, сетевой и иерархические подходы. Так как подход лежит в основе СУБД, различают реляционные, сетевые и иерархические СУБД.

6.2. Иерархический и сетевой подходы

Для организации поиска нужных записей используются понятия ключа и связи.

Ключ - уникальное имя записи. С помощью ключа производится идентификация каждой конкретной записи, а также упорядочение записей в файле. Упорядочение по ключу может быть либо прямым, либо выполнено с помощью хеш-функции.

Прямое упорядочение предполагает лексикографическое расположение записей: записи могут быть записаны в порядке увеличения ключа. При этом простой ключ можно рассматривать как некоторое число.

Хэш-функция производит пересчет ключа в адрес записи на файле. Эта операция осуществляется СУБД всякий раз при поиске новой записи по ключу.

Лекция 6

Информационное обеспечение САПР (продолжение)

6.3. Реляционный подход

Реляционные модели данных в последнее время получили широкое распространение вследствие простой формы представления данных, а также благодаря развитому теоретическому аппарату, позволяющему описывать различные преобразования реляционных данных. Основу реляционной модели данных составляет понятие отношения, представляющего собой подмножество декартова произведения доменов. *Домен* - это некоторое множество элементов, например множество целых чисел или множество допустимых значений, которые может принимать объект по некоторому свойству.

Декартовым произведением доменов D_1, D_2, \dots, D_k

$$D_1 * D_2 * \dots * D_k,$$

где

$$D_1 = \{d_{1.1}, d_{1.2}, \dots, d_{1.i1}, \dots, d_{1.n1}\},$$

$$D_2 = \{d_{2.1}, d_{2.2}, \dots, d_{2.i1}, \dots, d_{2.n1}\},$$

.....

$$D_k = \{d_{k.1}, d_{k.2}, \dots, d_{k.i1}, \dots, d_{k.n1}\},$$

называется множеством всех **кортежей** длины k , т.е. состоящих из k элементов - по одному из каждого домена $\langle D_i \rangle = (d_{1.i1}, d_{2.i2}, \dots, d_{k.ik1})$.

Таким образом, декартово произведение

$$D = \begin{pmatrix} (d_{1.1}, d_{2.1}, \dots, d_{(k-1).1}, d_{k.1}), \\ (d_{1.1}, d_{2.1}, \dots, d_{(k-1).1}, d_{k.2}), \\ \dots \\ (d_{1.1}, d_{2.1}, \dots, d_{(k-1).3}, d_{k.2}), \\ (d_{1.n1}, d_{2.n2}, \dots, d_{(k-1).n(k-1)}, d_{k.nk}) \end{pmatrix}$$

Пример (рис.6.1).

Если $D_1 = \{A, 2\}$, $D_2 = \{B, C\}$, $D_3 = \{4, 5, D\}$, то $k=3$ и соответственно декартово произведение $D = D_1 * D_2 * D_3 = \{(A, B, 4), (A, B, 5), (A, B, D), (A, C, 4), (A, C, 5), (A, C, D), (2, B, 4), (2, B, 5), (2, B, D), (2, C, 4), (2, C, 5), (2, C, D)\}$.

Отношением R на множествах D_1, D_2, \dots, D_k называется подмножество декартова произведения $D_1 * D_2 * \dots * D_k$, то есть R , определенное на множествах $D_1 * D_2 * \dots * D_k$ есть некоторое множество кортежей: $(d_{1.i1}, d_{2.i2}, \dots, d_{i.ik})$.

Атрибутом A называется некоторое подмножество домена D . Например, для домена D_3 атрибутом может быть $A_3 = \{4, 5\}$.

Для рассмотренного примера отношения имеют следующий вид

$$R_1 = \{(A, B, 4), (A, C, 4)\} \subseteq D_1 \times D_2 \times D_3,$$

$$R_2 = \{(2, C, 5), (A, C, 4)\} \subseteq D_1 \times D_2 \times D_3,$$

$$R_3 = \{(2, C, D), (A, B, 5), (A, B, 4), (2, B, D)\} \subseteq D_1 \times D_2 \times D_3,$$

$$R_4 = \{(A, B, 4), (A, B, 5), \dots, (2, C, D)\} \subseteq D_1 \times D_2 \times D_3,$$

$$R_5 = \emptyset.$$

В ряде случаев отношения удобно представлять в виде таблицы, где каждая строка соответствует кортежу, а столбец атрибуту, например,

R1

| | | |
|---|---|---|
| A | B | 4 |
| A | C | 4 |

R2

| | | |
|---|---|---|
| 2 | C | 5 |
|---|---|---|

R3

| | | |
|---|---|---|
| 2 | C | D |
| A | B | 5 |
| A | B | 4 |
| 2 | B | D |

6.4. Реляционное исчисление

Реляционное исчисление базируется на теоретических основах исчисления предикатов. Предикат $P(x_1, x_2, \dots, x_n)$ это функция, принимающая значения «Истина» или «ложь», от аргументов, определенных в конкретных областях D_1, D_2, \dots, D_n . При построении высказываний используются логические связки, например

\wedge конъюнкция, дизъюнкция, отрицание,

\neg

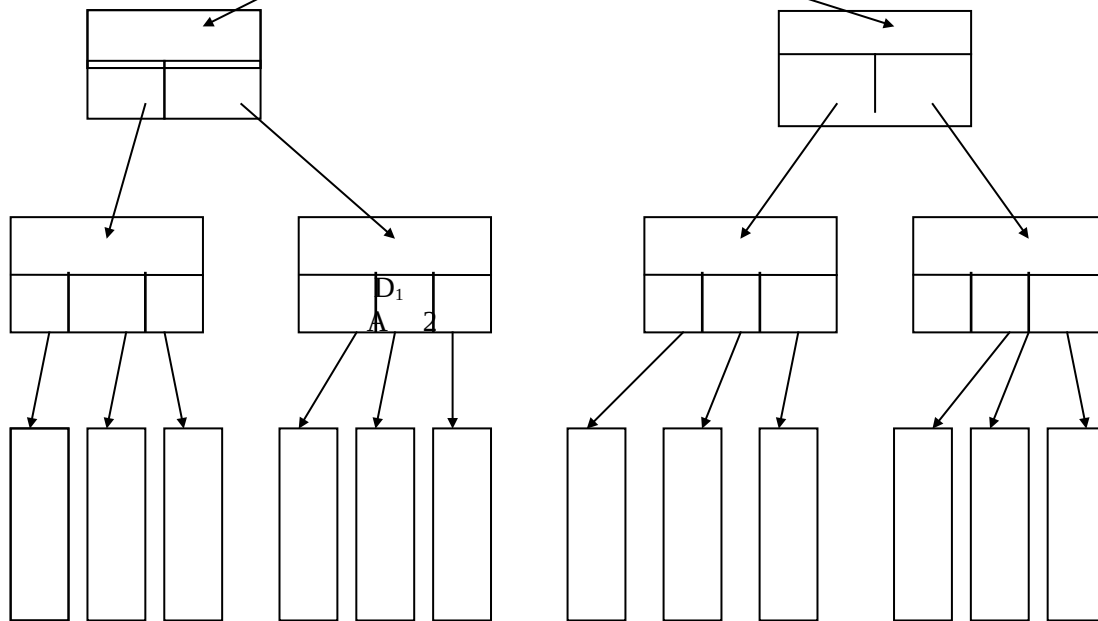
термы сравнения с операциями

\geq, \neq, \leq и др., а также кванторы существования \exists и общности \forall .

Рассмотрим пример применения кванторов. Высказывание $\exists x \in X (f(x) > a)$ означает, что среди элементов множества X найдется, по крайней мере, один, при котором оказывается истинным неравенство, заключенное в скобках. Высказывание $\forall x \in X (f(x) > a)$ означает, что для всех элементов множества X некоторая функция $f(x)$ больше заданного a .

В реляционном исчислении принято связывать отношением $R(A_1, \dots, A_n)$ некоторый предикат $P(x_1, \dots, x_n)$, аргументы, которых имеют одинаковые области определения, таким образом, что если $P(a_1, a_2, \dots, a_n) = 1$, то кортеж $\langle a_1, a_2, \dots, a_n \rangle$ принадлежит отношению R , $a_i \in A_i$, для $i=1, n$, в противном случае кортеж не входит в состав указанного отношения. Отсюда следует, что посредством задания некоторого предиката может быть задано и соответствующее ему отношение.

Из сказанного видно, что реляционное исчисление позволяет описать самые разнообразные виды искомых отношений. Однако отсутствие процедурности существенно затрудняет реализацию языков, основанных на реляционном исчислении. Решению этой проблемы возможно при использовании методов реляционной алгебры.

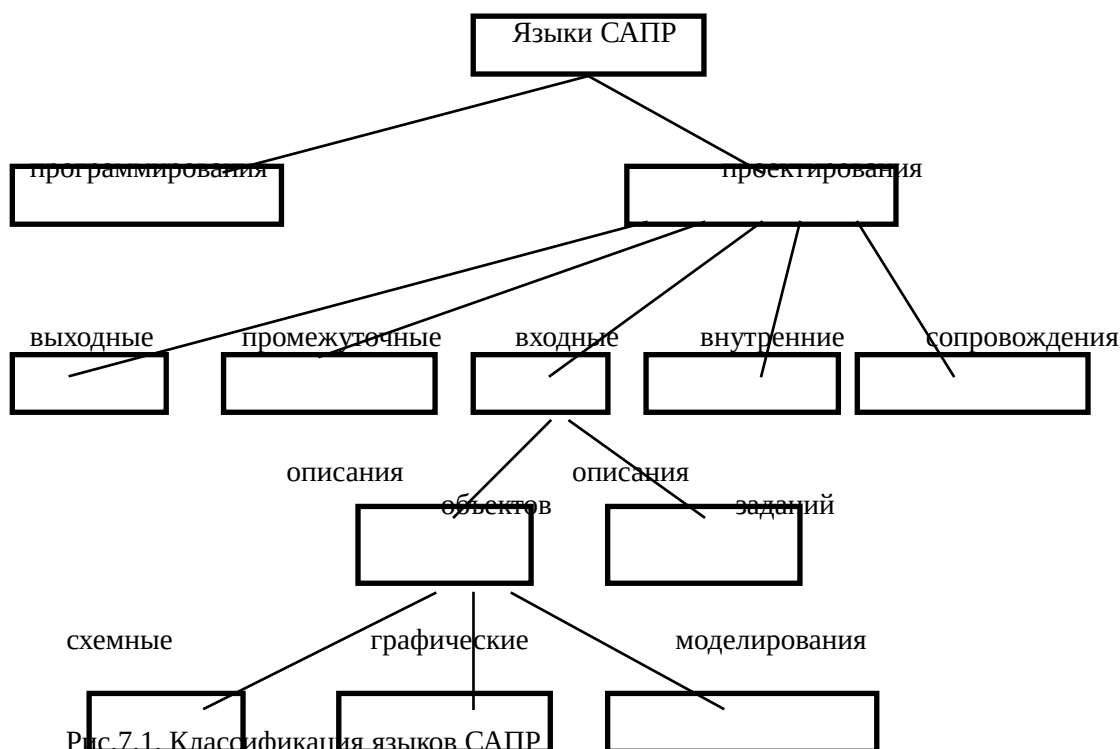


Лекция 7

Лингвистическое обеспечение САПР

Лингвистическое обеспечение САПР представлено совокупностью языков, применяемых для описания процедур автоматизированного проектирования и проектных решений. Основная часть лингвистического обеспечения САПР- языки общения человека с ЭВМ.

В соответствии с принятой классификацией языков САПР(рис.7.1) различают языки программирования и проектирования.



7.1. Языки программирования

Языки программирования- языки, предназначенные для написания программного обеспечения. Эти языки - средство разработчика САПР.

К языкам программирования предъявляют требования удобства использования, универсальности и эффективности объектных программ (т.е. программ полученных после трансляции на машинный язык).

Удобство использования выражается в затратах времени программиста на освоение языка и главным образом на написание программ на нем.

Универсальность определяется возможностями языка для описания разнообразных алгоритмов, характерных для программного обеспечения САПР.

Эффективность объектных программ определяется свойствами используемого транслятора, которые в свою очередь зависят от свойств языка. Эффективность оценивается затратами машинных времени и памяти на исполнение программ.

С позиций универсальности и эффективности объектных программ наилучшими свойствами обладают машинно-ориентированные языки. Близость к машинным кодам (языки машинных команд) обуславливают простоту и эффективность трансляторов на машинный язык называемых ассемблерами. Однако языки ассемблера неудобны для человека, их использование снижает производительность труда программистов. Поэтому их применяют для разработки лишь тех модулей программного обеспечения САПР, которые работают в реальном

масштабе времени или требуют для своего исполнения чрезмерно больших вычислительных ресурсов.

Среди алгоритмических языков высокого уровня, созданных на ранних этапах развития вычислительной техники, наибольшее распространение получил Фортран. Он легок в освоении и весьма эффективен при решении задач численного анализа. Программное обеспечение таких систем как PARIS, ModAPT-Caspa, APT-СМ написаны на языке Фортран. Однако Фортран имеет ограниченные возможности для описания сложных алгоритмов логического характера. Поэтому при создании таких программ, как мониторы или языковые процессоры, используют либо языки ассемблера, либо языки высокого уровня с более развитыми возможностями невычислительных процедур- C, Pascal, Pl-1, Ada.

7.2. Языки проектирования

Языки проектирования- языки, предназначенные для описания информации об объектах и задачах проектирования. Большинство этих языков относится к средствам пользователя САПР.

Среди языков проектирования выделяют

- входные;
- выходные;
- сопроводени;
- управления;
- промежуточные;
- внутренние.

Входные языки

Входные языки служат для задания исходной информации об объектах и задачах проектирования и включают в себя языки описания объектов (ЯОО) и языки описания заданий (ЯОЗ). Первые служат для описания свойств проектируемых объектов, а вторые - для описания заданий на выполнение проектных операций и процедур. ЯОО в свою очередь делятся на языки схемные, графические и моделирования.

Схемные языки широко применяют для описания принципиальных электрических и функциональных схем. Графические языки - основа лингвистического обеспечения в подсистемах машинной графики и геометрического моделирования. Языки моделирования развиты в подсистемах имитационного моделирования.

Выходные языки используются для выражения результатов проектных процедур на ЭВМ.

Языки сопровождения применяют для корректировки и редактирования данных при выполнении проектных процедур.

Языки управления служат для представления управляющей информации для программно-управляемого исполнительного оборудования, например для устройств документирования.

Промежуточные и внутренние языки предназначены для представления информации на определенных стадиях ее переработки в ЭВМ.

Недостаток промежуточных узкоспециализированных языков - в необходимости существенной перестройки связанной с ними программной системы при изменении условий проектирования. Недостаток универсальных языков связан с их громоздкостью и, следовательно, с неудобствами применения конечным пользователем.

Устранение указанных выше недостатков осуществляется с помощью транслирующих программ- конверторов. Здесь пользователь составляет описание на входном языке, это описание переводится конвертором на промежуточный (специализированный) язык и далее работает основной транслятор, который переводит описание задачи с промежуточного языка в объектную программу (пример, ГЕММА-> АРТ->УП).

7.3. Языки процедурные и непроцедурные

Языки проектирования, предназначенные для описания развивающихся во времени процессов называют процедурными, а языки, предназначенные для описания статических структур проектируемых объектов называют непроцедурными.

7.4. Диалоговые языки

Различают пассивный и активный диалоговый режим работы оператора с ЭВМ и соответственно этому активные и пассивные диалоговые языки.

В пассивном диалоговом режиме инициатива диалога принадлежит ЭВМ. Прерывание вычислительного процесса и обращение к пользователю в нужных местах осуществляется с помощью диалоговых программных средств включенных в мониторинговую систему САПР или монитор ППП. Обращение ЭВМ к пользователю в этом случае может быть в виде запроса (*исходных данных по шаблону или варианта дальнейшего проектирования по меню*), информационного сообщения (*для вывода промежуточных или окончательных результатов решения*) или подсказки (*сообщения об ошибках*).

В активном диалоговом режиме инициатива начала диалога может быть двусторонней - возможности прерывания процесса имеются и к ЭВМ и у пользователя. Активные диалоговые языки, как правило, близки к естественному человеческому, но с ограниченным набором слов. Очевидно, что для реализации активного диалогового режима требуется более сложное ПО.

Лекция 8

Структура программного обеспечения САПР

8.1. Основные компоненты программного обеспечения САПР

Структура ПО САПР зависит от следующих факторов:

- предметная область, аспекты и уровни создаваемых с помощью ПО описаний проектируемых объектов;
- степень автоматизации отдельных проектных операций и процедур;
- архитектура и состав технических средств, режим функционирования;
- ресурсы, отпущенные на разработку ПО.

Рассмотрим вариант организации одноуровневой САПР (рис.8.1).

Программное обеспечение САПР может быть разделено на проектирующие и обслуживающие подсистемы.

К обслуживающим подсистемам ПО относятся: диалоговая ДП; СУБД; инструментальная подсистема; монитор - обеспечивающий взаимодействие всех подсистем и управление их выполнением.

Диалоговая подсистема ПО организует интерактивное взаимодействие пользователя САПР с управляющей и проектирующими подсистемами ПО, подготовку и редактирование исходных данных, просмотр результатов проектирующих подсистем, функционирующих в пакетном режиме.

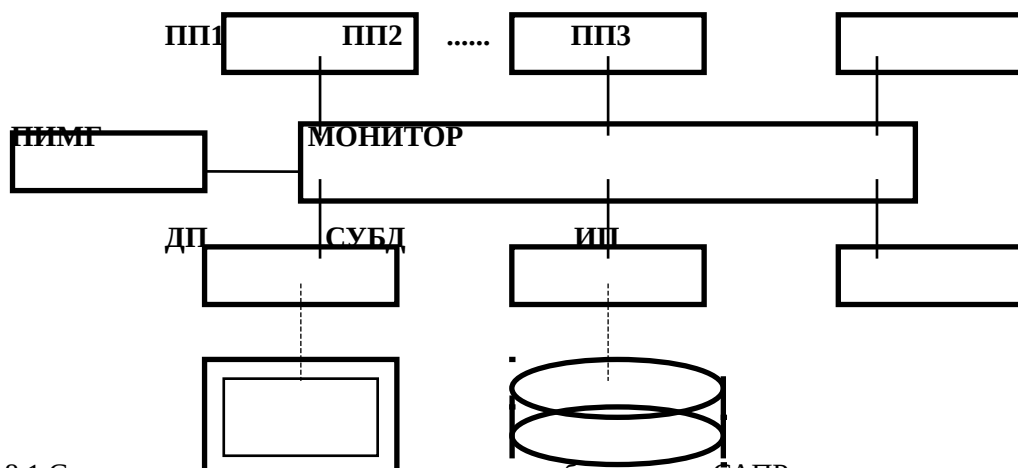


Рис.8.1. Структура специального программного обеспечения САПР:

ПП₁-проектирующая подсистема; ДП-диалоговая подсистема; ИП-инструментальная подсистема; СУБД- система управления базой данных; ПИМГ - система интерактивной машинной графики

Подсистема управления базами данных (СУБД) реализует единообразный доступ к общей базе данных САПР. Назначение БД:

- хранение сведений нормативно- справочного характера;
- хранение результатов выполнения этапов текущего проекта;
- обеспечение информационной согласованности различных подсистем САПР.

Инструментальная подсистема программирования, основу которой составляет генератор прикладных программ, синтезирующий новые программы из унифицированных модулей и программ разработанных пользователем, необходима для обеспечения открытости ПО САПР. Генератор прикладных программ включает в себя также средства автоматической разработки трансляторов для входных языков проектирующих подсистем САПР.

Проектирующие подсистемы ПО могут быть объектно-зависимыми(проблемно-ориентированными) или объектно-независимыми(методо-ориентированными).

Объектно-независимые подсистемы ПО ориентированы на решение задач проектирования при наличии предварительно выполненной математической постановки(например, подсистема

структурно параметрической идентификации - ПАРИС, подсистема решения уравнений в частных производных или обыкновенных диф.уравнений и др.). Объектно-независимые подсистемы составляют основу для генерации проблемно-ориентированных подсистем САПР.

Проектируемыми подсистемами ПО САПР могут быть простые программы, ориентированные на узкий класс объектов и использующих аналитические модели. Но чаще проектирующие подсистемы ПО представляют собой универсальные пакеты прикладных программ сложной структуры, обладающие своими мониторами, локальными БД и СУБД.

Подсистема интерактивной машинной графики (ПИМГ) занимает промежуточное место между проектируемыми и обслуживаемыми подсистемами ПО САПР. С одной стороны, средства машинной графики обслуживают ряд проектирующих подсистем(например, для наглядного представления информации в виде графиков, гистограмм), а с другой стороны они входят во многие подсистемы конструкторского проектирования как основная часть.

8.2. Монитор САПР

Управление ходом вычислительного процесса и координация взаимодействия подсистем САПР осуществляется монитором. В рамках пакетов ПП аналогичные функции решаются локальными мониторами.

В функции мониторов входят:

- прием и интерпретация обращенных к ним команд пользователя;
- загрузка и активизация компонентов ПО, организация маршрутов их выполнения;
- установления взаимодействия между подсистемами;
- динамическое распределение памяти;
- обработка прерываний от дисплея пользователя;
- сервисные функции(регистрация пользователей, сбор статистики, обработка сбоя системы и др.);

Язык управления монитором САПР достаточно прост, в его основе лежат команды вызова необходимых подсистем ПО САПР и задания им управляющих параметров, а также команды, описывающие способ информационного обмена между подсистемами(через ОЗУ, через внешнюю память или БД). Средства этого языка должны составлять макрокоманды, определяющие маршруты выполнения проектирующих подсистем ПО САПР.

В общем случае загруженные проектирующие подсистемы ПО могут функционировать либо как обычные подпрограммы, подчиненные управляющей подсистеме ПО, либо как параллельные выполняемые подзадачи, способные соревноваться между собой и монитором за управление.

8.3.Взаимодействие подсистем

Взаимодействие управляющей подсистемы ПО и мониторов проектирующих пакетов осуществляется через стандартный интерфейс, представляющий собой формальные правила передачи фактических параметров.

В проектирующие подсистемы ПО передаются:

- параметры, задающие режим функционирования;
- адреса точек входа в обслуживаемые подсистемы ПО;
- адреса динамически распределенных областей памяти, предназначенных для информационного обмена между различными подсистемами ПО.

Каждый проектирующий пакет, входящий в САПР, имеет паспорт, хранящийся в базе данных САПР. Паспорт содержит следующие сведения о проектирующем пакете:

- размер занимаемой области ОЗУ;
- имена требуемых обслуживаемых подсистем;
- имена режимных параметров и их значения по умолчанию;
- имя языка программирования, в стандарте которого пакет использует представление структур данных;
- местонахождение в пакете обработчика прерываний от дисплея пользователя;
- указатели на возможные способы обмена информацией с другими проектируемыми подсистемами ПО(ОЗУ, СУБД, файловая система).

Монитор САПР, получив команду на активизацию какой-либо проектирующей подсистемы ПО, считывает из базы данных ее паспорт, проверяет корректность команды и возможность загрузки подсистемы.

Далее он помещает в ОЗУ необходимые обслуживающие подсистемы, проектирующую подсистему, а затем в строгом соответствии с данными из паспорта строится обращение к этой подсистеме.

Важной функцией управляющей подсистемы САПР и мониторов проектирующих пакетов является динамическое распределение памяти, необходимое всегда, когда пакет работает с данными переменного объема. Средства динамического распределения памяти- обязательные компоненты всех современных языков программирования.

для эффективной работы коллектива пользователей необходим множественный доступ к САПР. Эту проблему решает режим разделения времени, реализуемый с помощью ПО ОС ЭВМ.

Лекция 9

Уровни, аспекты и этапы автоматизированного проектирования

9.1. Иерархические уровни описаний проектируемых объектов

В основе блочно-иерархического подхода к проектированию лежит принцип разделения описания объекта по степени детализации его отображаемых свойств и характеристик. Это приводит к появлению уровней абстрагирования.

На каждом иерархическом уровне используются свои понятия системы и элементов. На уровне 1 (верхнем уровне) объект S рассматривается как система S из n взаимосвязанных и взаимодействующих элементов S_i (рис.9.1).

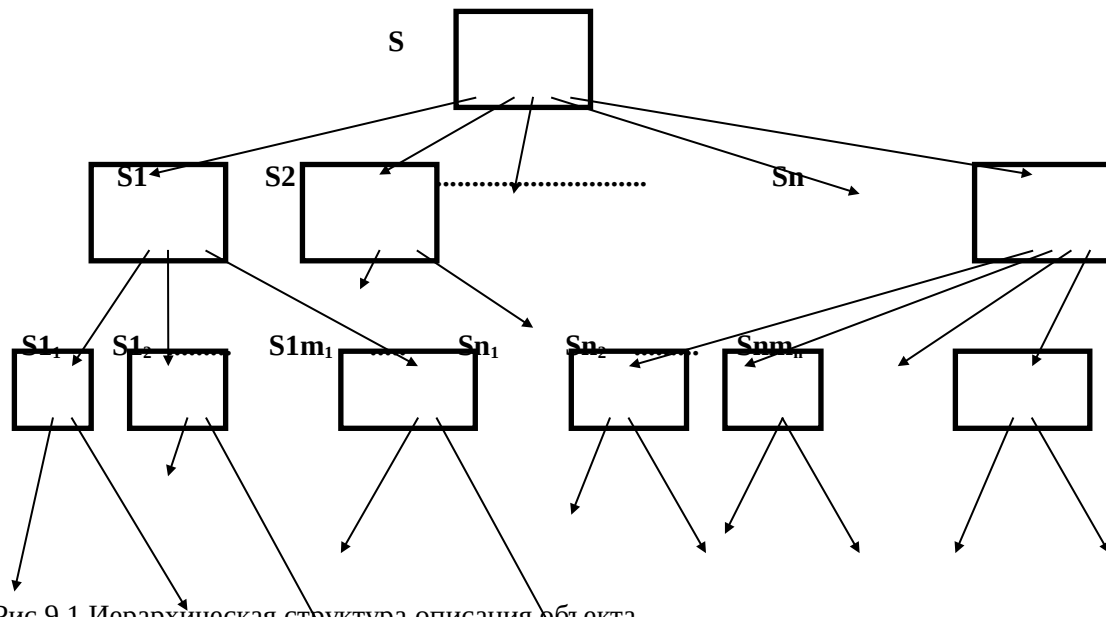


Рис.9.1. Иерархическая структура описания объекта

Каждый из элементов в описании уровня 1 представляет собой также сложный объект, который в свою очередь рассматривается как система S_i на уровне 2. Элементами систем S_i являются объекты S_{ij} , $j=1,2,\dots,m_i$ (где m_i - количество элементов в описании системы S_i). Как правило, выделение элементов S_{ij} происходит по функциональному признаку. Деление элементов на уровни происходит вплоть до появления элементов, которые дальнейшему делению не подлежат. Такие элементы по отношению к S называются **базовыми**.

Примером иерархического описания объектов является структура технологического процесса изготовления детали с уровнями: маршрут -> операция -> переход -> проход; другим примером является декомпозиция конструкции изделия принятая в спецификации: изделие - узел - деталь.

Таким образом, принцип иерархичности означает структурирование представлений об объектах проектирования по степени детальности описаний, а принцип декомпозиции (блочности) - разбиение представлений каждого уровня на ряд составных блоков с возможностью отдельного блочного проектирования.

9.2. Аспекты описаний проектируемых объектов

Кроме иерархического описания объекта используется описание по характеру отображаемых свойств, т.е. аспектов описаний. Наиболее важными аспектами для технических объектов являются конструкторский и технологический и функциональный аспекты.

Функциональный аспект связан с отображением основных принципов функционирования, характера физических и информационных процессов, протекающих в объекте, и находит выражение в принципиальных, функциональных, структурных, кинематических схемах и сопровождающих их документах.

Конструкторский аспект связан с реализацией результатов функционального проектирования, т.е. с определением геометрических форм объектов и их взаимным расположением в пространстве.

Технологический аспект относится к реализации результатов конструкторского проектирования, т.е. связан с описанием методов и средств изготовления объектов.

Возможно более дифференцированное описание свойств объекта, а внутри каждого аспекта возможно специфическое выделение иерархических уровней.

9.3. Составные части процесса проектирования

Процесс проектирования расчленяется на стадии, этапы, проектные процедуры и операции.

Стадии:

- предпроектные исследования;
- техническое задание;
- эскизный проект;
- технический проект;
- рабочий проект;
- испытание;
- внедрение в производство.

Этап проектирования - часть процесса проектирования, включающая в себя формирование всех требующихся описаний объекта, относящихся к одному или нескольким иерархическим уровням и аспектам.

Составные части этапа проектирования называют проектными процедурами. Проектная процедура - часть этапа, которая заканчивается получением проектного решения. Более мелкие составные части процесса проектирования, входящие в состав проектных процедур, называются проектными операциями. Например, для проектной процедуры оформления чертежа изделия проектной операцией может быть вычерчивание типового графического изображения зубчатого венца и т.п.

9.4. Нисходящее и восходящее проектирование

Если решение задач более высоких иерархических уровней предшествует решению задач более низких иерархических уровней, то проектирование называют нисходящим, в противном случае восходящим.

При нисходящем проектировании система разрабатывается в условиях, когда ее элементы еще не определены и, следовательно, сведения об их возможностях и свойствах носят предположительный характер. При восходящем проектировании элементы проектируются раньше и, следовательно, требования к системе имеют предположительный характер.

На практике, как правило, сочетают оба указанных вида проектирования.

Например, восходящее проектирование имеет место на всех уровнях, на которых используются унифицированные элементы, на остальных используется нисходящее.

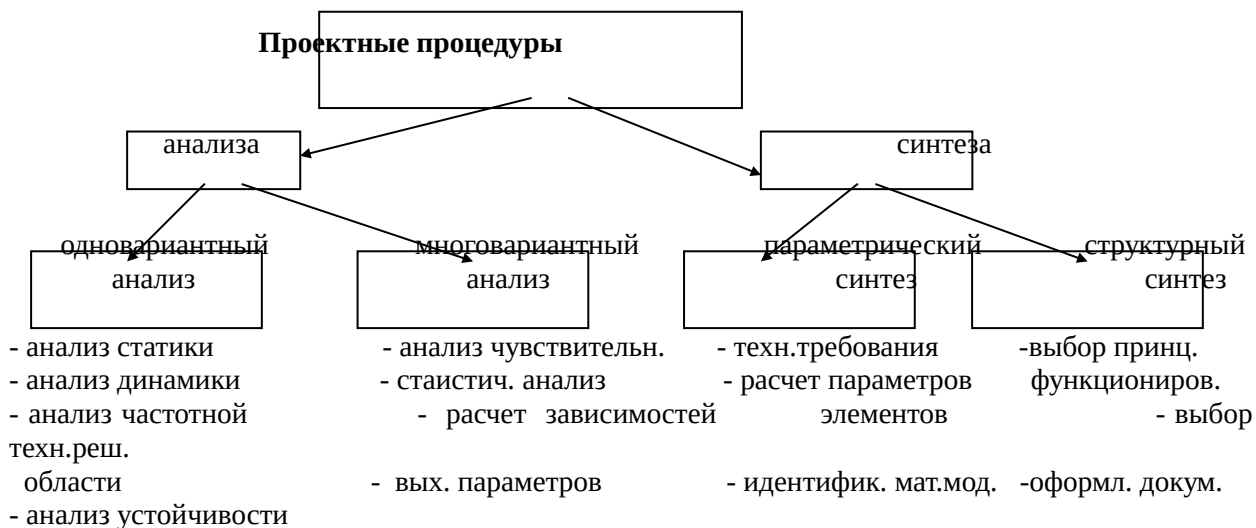
9.5. Внешнее и внутреннее проектирование

При нисходящем проектировании формулировка ТЗ на разработку элементов к-го иерархического уровня относится к проектным процедурам этого же уровня. Однако разработка ТЗ на систему более высокого иерархического уровня или на систему унифицированных элементов универсального назначения является самостоятельным этапом проектирования, которое называют внешним. Соответственно, проектирование объекта по сформулированным ТЗ называют внутренним. На практике сочетание процессов внутреннего и внешнего проектирования образуют итерационный процесс корректировки ТЗ.

9.6. Унификация проектных решений и процедур

Использование унифицированных и типовых проектных решений упрощает и ускоряет процесс проектирования. Однако унификация целесообразна только в тех случаях, когда из сравнительно небольшого числа элементов проектируется большое разнообразие систем. В некоторых случаях (например, при многократном повторении одинаковых проектных процедур для различных объектов проектирования) возникает потребность в унификации не только самих объектов, но и собственно проектных процедур в рамках САПР.

9.7. Классификация типовых задач проектирования



Лекция 10

Автоматизация проектирования технологических Процессов

10.1. Проектирование на основе методов типизации

Типовые процессы разрабатывают на основе анализа, систематизации и обобщения технологических решений, учитывают передовые достижения технологии машиностроения.

Для данного класса деталей устанавливают так называемый обобщенный маршрут обработки. Он включает перечень операций характерный для определенного класса деталей, имеет типовую их последовательность и содержание.

Если известно множество индивидуальных маршрутов M_1, M_2, \dots, M_n (1...n - номера индивидуальных маршрутов для группы деталей) для данной группы деталей, то обобщенный маршрут M^* может быть формально описан следующим выражением (рис.10.1)

$$M^* = \bigcup_{i=1}^n M_i, \text{ где } n \rightarrow \max$$

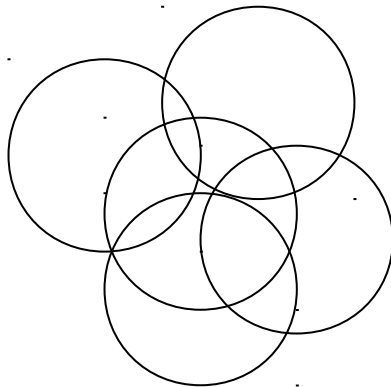


Рис.10.1 Схема объединения индивидуальных технологических маршрутов в обобщенный технологический маршрут

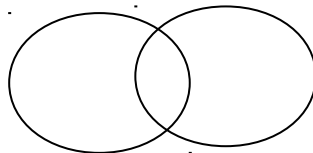


Рис.10.2 Схема пересечения технологических маршрутов с образованием области пересечения операций

Необходимым условием включения индивидуального маршрута в обобщенный является наличие области пересечения операций как непустого множества

$$M_i \cap M_j \neq \emptyset$$

Важной характеристикой (критерием эффективности) формирования обобщенного маршрута является мощность пересечения множеств операций индивидуальных маршрутов $M_{\text{пер}}$, т.е. количество одинаковых операций, входящих в это множество без учета отношения порядка элементов (операций) множества:

$$M_{\text{пер}} = \bigcap_{i=1}^n M_i \quad \text{а} \quad \max$$

Мощность пересечения в обобщенном маршруте должна стремиться к максимуму (рис.10.2). Это будет основным условием объединения нескольких индивидуальных маршрутов в обобщенный. Тогда мощность обобщенного маршрута

$$/M^*/ = \bigcup_{i=1}^n Mi(i = 1, 2, \dots, n) \rightarrow \min$$

должна стремиться к минимуму.

10.2. Логические условия назначения операции в маршруте

При проектировании маршрута обработки необходимо решать две задачи: определить перечень операций и последовательность их выполнения. Логическое условие выбора операций по своему назначению можно разделить на ряд групп (рис.10.3). В общем случае логическая функция выбора k-й операции

$$f_k = \bigvee_{j=1}^{n_2} (\bigwedge_{i=1}^{n_1} A_i)_j$$

где A_i - условие из справочника логических условий для группы деталей;
 i - количество условий, связанных конъюнкцией; j - количество сочетаний, связанных дизъюнкцией.

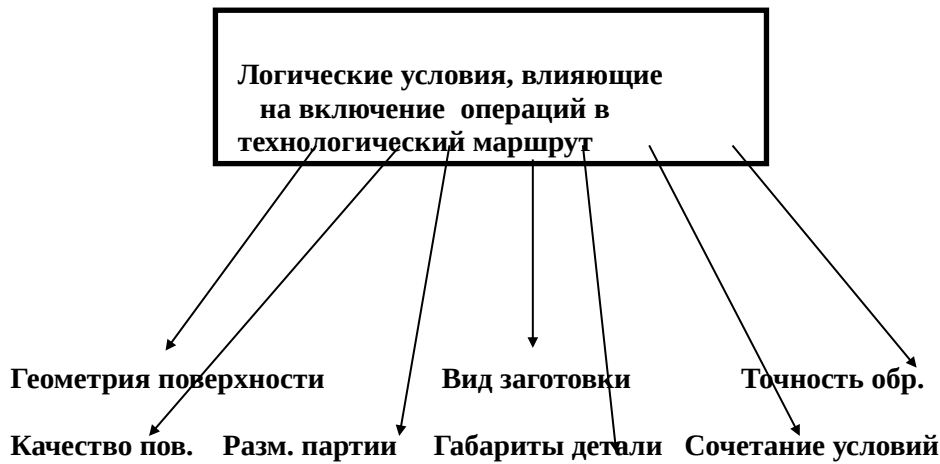


Рис.10.3. Логические условия, влияющие на включение операций в технологический маршрут

10.3. Формирование обобщенного маршрута

Формирование обобщенного маршрута начинают в принципе с любого имеющегося M_i , принимаемого за базовый. В него последовательно вставляются недостающие операции всех присоединяемых M_j маршрутов.

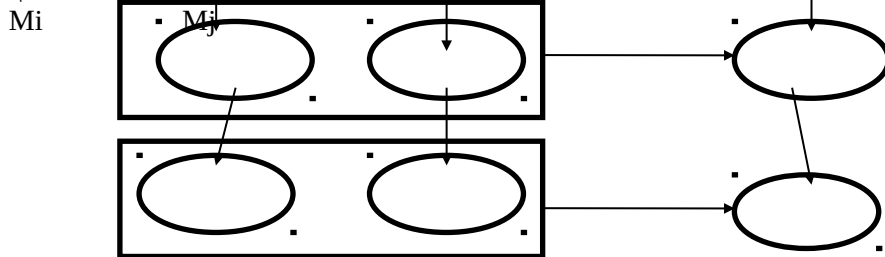
Для этого производится поиск в базовом маршруте для каждой присоединяемой операции эквивалентных операций. Вставляемые недостающие операции занимают определенные места в базовом маршруте. Полученный маршрут принимается как очередной базовый, к нему присоединяется следующий маршрут и т.д. для рассматриваемого класса деталей.

Полученный обобщенный маршрут представляет собой перечень операций, каждая из которых имеет логическую функцию, которая определяет условия включения данной операции в индивидуальный маршрут обработки.

На рис.10.4. показана схема построения обобщенного маршрута. К базовому маршруту M_i присоединяется маршрут M_j и в результате получается обобщенный маршрут M^* . Заштрихованные области показывают эквивалентные операции, которые определяют мощность пересечения маршрутов. Эквивалентность операций устанавливается по совпадению кодов.

10.4. Синтез технологических маршрутов

Синтез индивидуальных технологических маршрутов осуществляется путем выделения их из обобщенного маршрута (рис.10.5).



| | | |
|------|------|------|
| C1Mi | C1Mj | C1Mi |
| C2Mi | C2Mj | C1Mi |
| | C3Mj | C3Mj |
| C4Mi | | C4Mi |
| C5Mi | C5Mj | C5Mi |

| | | |
|------|------|------|
| C6Mi | C6Mj | C6Mi |
|------|------|------|

| | | |
|--|--|--|
| | | |
|--|--|--|

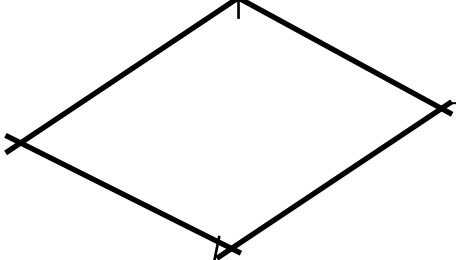
Рис.10.4.Схема построения обобщенного маршрута

| | | |
|--|--|--|
| | | |
|--|--|--|

| | | |
|--|--|--|
| | | |
|--|--|--|

| |
|--|
| |
|--|

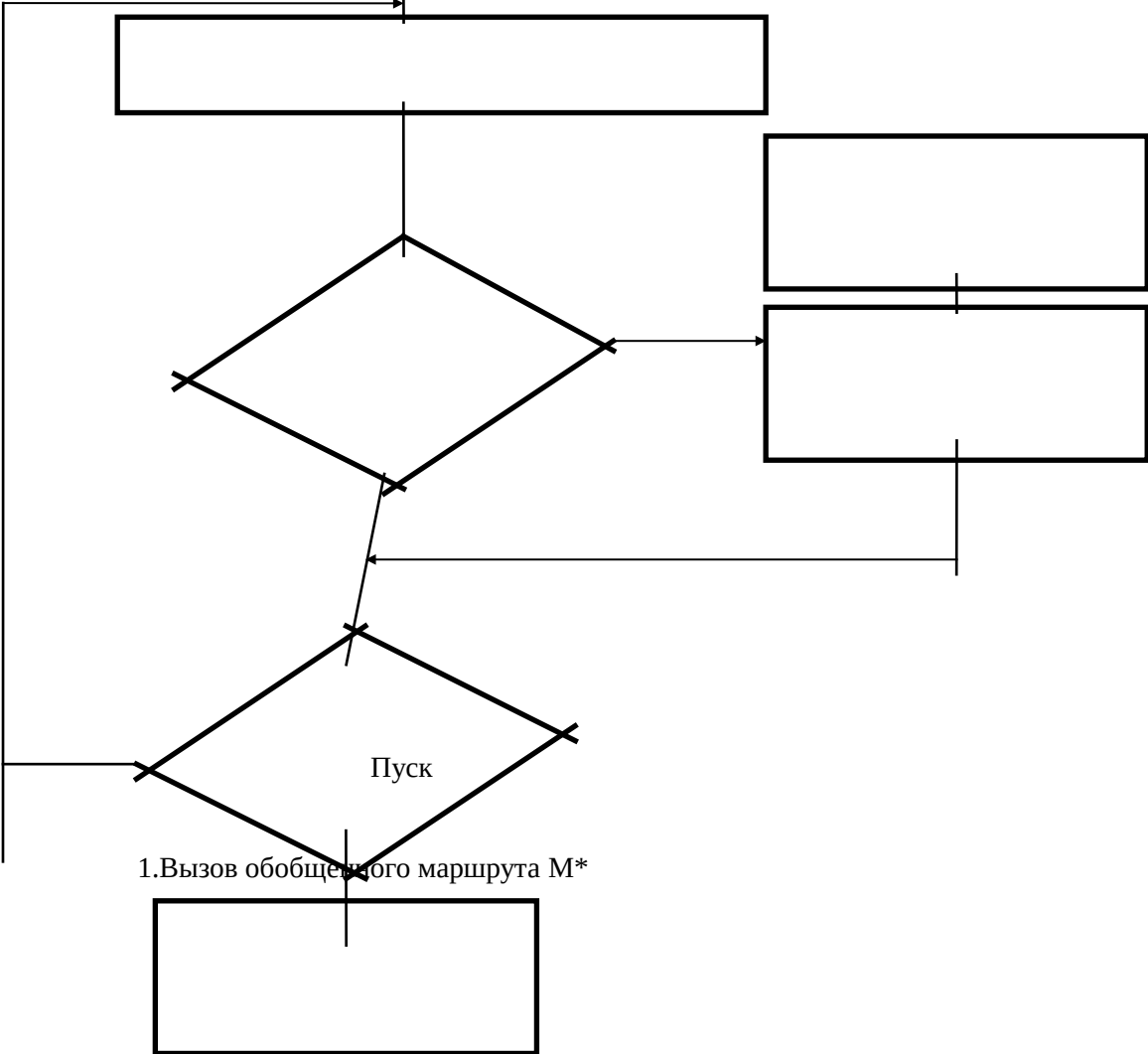
| |
|--|
| |
|--|



Пуск

1.Вызов обобщенного маршрута M*

| |
|--|
| |
|--|



с логическими функциями

2. Вызов логических условий, характеризующих деталь

3. Вызов k-й операции M^*

Печать маршрутной карты

$fk=1$

Формирование кодов инд. маршрута

последняя опер.?

Останов

Рис.10.5. Алгоритм синтеза маршрута обработки

Лекция 11

Системы автоматизированного программирования технологического оборудования с ЧПУ

11.1. Основные понятия и терминология

Совокупность команд на языке программирования, соответствующая заданному алгоритму функционирования станка, составляет управляющую программу(УП).

Программное обеспечение(ПО) системы ЧПУ представляет собой совокупность программ и документации на них для реализации целей и задач, возлагаемых на устройство ЧПУ, выдающее управляющие коды воздействия на исполнительные органы технологического оборудования.

УП содержит информацию об инструменте(электроде) и скоростях его перемещений относительно заготовки, частоте вращения шпинделя, смене инструмента, вводе коррекции, подаче охлаждения, других командах исполнительным механизмам станка, записанным в последовательности технологического процесса обработки детали на станке.

В процессе подготовки управляющей программы решаются следующие задачи:

- *0 диалог с пользователем;
- *1 синтаксический контроль исходной информации на входном языке;
- *2 проектирование элементов технологического процесса обработки;
- *3 расчет траекторий движения инструмента;
- *4 формирование и запись выходной информации на промежуточном языке;
- *5 выдача диагностических сообщений о разных этапах обработки исходной информации;
- *6 редактирование программ на уровнях входного, промежуточного и выходного языков;
- *7 формирование УП на выходном языке для конкретного станка и выдача ее на программноносителе;
- *8 распечатка УП и сопроводительной документации;
- *9 хранение и тиражирование УП.