



Министерство образования Республики Беларусь

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

Кафедра «Технология машиностроения»

В. С. Мурашко

ИНФОРМАЦИОННЫЕ СИСТЕМЫ В САПР

ПОСОБИЕ

для студентов специальности

**1-53 01 01 «Автоматизация производственных
процессов и производств (по направлениям)»
дневной формы обучения**

Гомель 2021

УДК 681.512.011.56(075.8)
ББК 30.2-5-05я73
М91

*Рекомендовано научно-методическим советом
машиностроительного факультета ГГТУ им. П. О. Сухого
(протокол № 4 от 06.04.2020 г.)*

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

Мурашко, В. С.

М91 Информационные системы в САПР : пособие для студентов специальности 1-53 01 01 «Автоматизация производственных процессов и производств (по направлениям)» днев. формы обучения / В. С. Мурашко. – Гомель : ГГТУ им. П. О. Сухого, 2021. – 210 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

Представлена классификация информационных систем, рассмотрены системы автоматизированного проектирования (САПР) и их место среди других автоматизированных систем, вопросы организации информационного фонда САПР с использованием баз данных и баз знаний, а также методы и алгоритмы поддержки принятия решений САПР.

Для студентов специальности 1-53 01 01 «Автоматизация производственных процессов и производств (по направлениям)» дневной формы обучения.

УДК 681.512.011.56(075.8)
ББК 30.2-5-05я73

© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2021

Введение

Прогресс науки и техники, потребности общества в новых промышленных изделиях обуславливают необходимость выполнения проектных работ большого объема. Требования к качеству проектов и срокам их выполнения становятся все более жесткими по мере увеличения сложности проектируемых объектов и повышения ответственности выполняемых ими функций. Удовлетворить эти требования с помощью простого увеличения численности конструкторов и технологов нельзя, так как возможность параллельного проведения проектных работ ограничена и численность инженерно-технических работников в проектных организациях страны не может быть существенно увеличена.

Решение этой проблемы возможно только на основе широкого применения систем автоматизированного проектирования (САПР). Однако широкое внедрения САПР и их дальнейшее развитие сдерживается недостатком кадров конструкторов и технологов, имеющих соответствующие знания для запуска и эффективного использования систем на практике. Современные технологические САПР базируются на информационных системах, использующих в основе банки данных.

Настоящее пособие предназначено для студентов специальности 1-53 01 01 «Автоматизация технологических процессов и производств (по направлениям)» и соответствует программе курса «Информационные системы в САПР»

В настоящем пособии представлен теоретический и практический материал необходимый для формирования у студентов навыков создания и использования банков данных и оптимизационных моделей различных технических объектов на основе информационных технологий.

В пособии представлена классификация информационных систем, рассмотрены системы автоматизированного проектирования (САПР) и их место среди других автоматизированных систем, рассмотрены вопросы организации информационного фонда САПР с использованием баз данных и баз знаний, а также методы и алгоритмы поддержки принятия решений в САПР.

1 ИНФОРМАЦИОННЫЕ СИСТЕМЫ

1.1 Основные понятия

Первые информационные системы начали появляться с моментом появления первых ЭВМ.

Система (греч. Systema – целое, составленное из частей; соединение) – множество элементов, находящихся в отношениях и связях друг с другом, которое образует определённую целостность, единство в интересах

Информационная система (ИС) – это определённая совокупность взаимосвязанных средств и методов, персонала для обработки, хранения и выдачи информации, с целью эффективного управления.

С понятием ИС тесно связаны и такие понятия как Информация, Информационное обеспечение. Информационные технологии.

Информация – это данные, сведения в определённом изложении.

Информационное обеспечение – это совокупность всех процессов: сбор, обработка, хранение, анализ и выдача информации, которая необходима для обеспечения управленческой деятельности и всех бизнес-процессов организации.

Информационные технологии – это по определению, принятому ЮНЕСКО, *информационной технологией* является совокупность взаимосвязанных научных, технологических и инженерных дисциплин, изучающих методы эффективной организации труда людей, которые заняты обработкой и хранением информации, а также вычислительная техника и методы организации и взаимодействия с людьми и производственным оборудованием.

Процесс информационного обеспечения ИС представлен на рисунке 1.1.

На современном этапе все ИС обрабатывают информацию используя компьютеры, поэтому их часто еще называют системами обработки данных.

Внедрение любой ИС производится с целью повышения эффективности производственно-хозяйственной деятельности организации за счет:

- обработки и хранения рутинной информации;
- автоматизации конторских (офисных) работ;
- применения принципиально новых методов управления, основанных на моделировании действий специалистов фирмы при

- принятии решений (методов искусственного интеллекта, экспертных систем и т. д.);
- использования современных средств телекоммуникаций (электронной почты, телеконференций), глобальных и локальных вычислительных сетей и т. д.

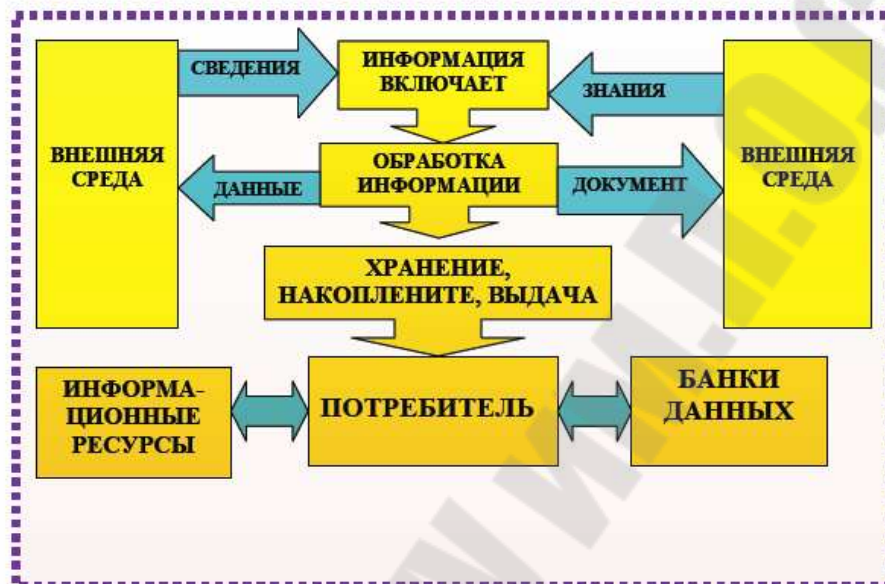


Рисунок 1.1 – Процесс информационного обеспечения ИС

Задачи обработки данных обеспечивают обычно рутинную обработку и хранение информации с целью выдачи (регулярной или по запросам) сводной информации, которая может потребоваться для управления объектом.

Автоматизация конторских (офисных) работ предполагает наличие в ИС системы ведения картотек, системы обработки текстовой информации, системы машинной графики, системы электронной почты и связи.

Методы (алгоритмы) искусственного интеллекта необходимы для решения задач принятия управленческих решений, основанных на моделировании действий специалистов предприятия при принятии решений. В современных информационных системах возможности искусственного интеллекта реализуются достаточно слабо.

Современное понимание информационной системы предполагает использование персонального компьютера в качестве основного технического средства переработки информации.

1.2 Этапы развития ИС

Примерно в 50–60-е года началось поэтапное осознание значения информации, как самого важного ресурса организации и с этого момента начинают разрабатывать первые ИС.

Первые ИС предназначались для обработки информации, связанной с заработной платой, и реализовывались в виде счетных машинок. Благодаря этому сокращалось время на обработку информации, благодаря чему сокращались затраты в организации.

На первоначальном этапе обработка информации происходила при помощи вычислительной техники, такие системы называли «системами обработки данных» (СОД). С течением времени этот термин стал широко использоваться в системах радиоуправления космическими объектами, ракетами, обработки статических данных и т. д.

С развитием ЭВМ стали и увеличивать объем памяти, что привело к организации баз данных (БД). На сегодня эта проблема обработки, создания, хранения БД также актуальна.

В 60–70-х годах появляется периодическая отчетность, а для развития естественно необходимо развивать ЭВМ и ИС широкого назначения, которые могли бы выполнять различные функции, а не просто рассчитывать зарплату и другие счета.

Основные черты ИС:

- техническое обеспечение систем составляли маломощные ЭВМ 2-3 поколения;
- информационное обеспечение представляло собой массивы (файлы) данных, структура которых определялась той программой, в которой они использовались;
- программное обеспечение было в виде специализированных прикладных программ, таких как программа начисления заработной платы;
- архитектура ИС – централизованная. Как правило, применялась пакетная обработка задач. Конечный пользователь не имел непосредственного контакта с ИС, вся предварительная обработка информации и ввод производились персоналом ИС.

Недостатки ИС:

- сильная взаимосвязь между программами и данными, то есть изменения в предметной области приводили к изменению

структуры данных, а это заставляло переделывать программы;

- трудоемкость разработки и модификации систем;
- сложность согласования частей системы, разработанных разными людьми в разное время.

В 80–90-х информация имеет важнейшие значения, появляются ИС способные решать задачи планирования организации.

Основные черты ИС:

- основу ИО составляет база данных;
- программное обеспечение состоит из прикладных программ и СУБД;
- технические средства: ЭВМ 3-4 поколения и ПЭВМ;
- средства разработки ИС: процедурные языки программирования 3-4 поколения, расширенные языком работы с БД (SQL, QBE);
- архитектура ИС: наиболее популярны две разновидности: персональная локальная ИС, централизованная БД с сетевым доступом.

В этот период появляются CASE-средства.

Недостатки ИС:

- большие капиталовложения в компьютеризацию предприятий не дали ожидаемого эффекта, соответствующего затратам (увеличились накладные расходы, но не произошло резкого повышения производительности);
- внедрение ИС столкнулось с инертностью людей, нежеланием конечных пользователей менять привычный стиль работы, осваивать новые технологии;
- к квалификации пользователей стали предъявляться более высокие требования (знание ПК, конкретных прикладных программ и СУБД, способность постоянно повышать свою квалификацию).

В 90–2000-е бурное развитие ИС для бизнеса. Появление интегрированных систем управления.

2000 – по настоящее время – развитие ИС, позволяющих вести бизнес в любой точке мира, не зависимо от местоположения офиса, широкое применение электронной цифровой подписи, безбумажный документооборот, интегрирование покупателя в ИС.

1.3 Структура информационной системы

Структуру информационной системы – составляет взаимосвязанная совокупность всех ее частей, называемых обеспечивающими подсистемами.

Подсистемы ИС – информационная, техническая, математическая, программная, организационная и правовая основа (правовое обеспечение).

Информационное обеспечение (ИО) – совокупность единой принятой классификации и кодирования информации, унифицированных систем документации, схем информационных потоков.

Техническое обеспечение – это комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Математическое и программное обеспечение – совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач ИС, а также нормального функционирования комплекса технических средств.

Организационное обеспечение – совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

Правовое обеспечение – совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.

На рисунке 1.2 представлена структура ИС, а на рисунке 1.3 – классификация задач, решаемых при помощи ИС.

Структурированная (формализуемая) задача – это такая задача, в которой известны не только все элементы, но и взаимосвязи между ними. В данной задаче выражается ее содержание в виде математической модели, которая точно описывается определенным алгоритмом решения. Данные задачи как правило решаются многократно и носят рутинный характер.

Использование ИС для решения структурированных задач обеспечивает полную автоматизацию их решения.



Рисунок 1.2 – Структура ИС с подсистемами



Рисунок 1.3 – Классификация задач, решаемых при помощи ИС

Примером структурированной задачи является расчет заработной платы, в которой известен алгоритм ее решения, все переменные известны. Рутинность проявляется в том, объем данных большой, а необходимость проведения расчетов ежемесячная.

Неструктурированная (неформализуемая) задача – в данной задаче не возможно выделить все элементы и взаимосвязи между ними. Создать алгоритм решения данной задачи практически не возможно, возможности ИС используется лишь для поддержки принятия решения. Само решение принимается человеком, на основе практического опыта и полученных знаний. Например, взаимоотношения в группе.

В деятельности организации мало полностью структурированных или совсем не структурированных задач. В основном встречаются задачи, где известна часть переменных и их взаимосвязь. *Такие задачи являются частично структурированными.*

В данном случае получается создать ИС, а получаемая информация анализируется и на основе этого анализа человек принимает решения. В этих системах человек играет определяющую роль. Данные системы относят к автоматизированным.

На рисунке 1.4 представлены ИС по способу решаемых задач.



Рисунок 1.4 – ИС по способу решаемых задач

ИС формирующие *управленческие отчеты*, тем самым позволяют обеспечить информационную поддержку пользователя, т.е. есть доступ к БД и ее частичная обработка. В процессе обработки данных в системе должны быть реализованы следующие возможности:

- комбинирование данных, формируемых различными источниками;
- быстрое удаление и добавление источников данных в обычном и автоматическом режиме, возможность переключения при поиске;
- управление данными БД;
- логическая независимость БД друг от друга в одной ИС;
- автоматическое отслеживание потока информации для наполнения баз данных.

Экспертные ИС обеспечивают выработку и оценку возможных альтернатив пользователем за счет создания экспертных систем, связанных с обработкой знаний.

Математические, статические, финансовые и другие модели, благодаря использованию которых облегчается выработка и оценка альтернатив решения. Пользователь получает информацию для принятия решения.

При решении задач применяются следующие функции:

- моделирование задач типа – «как сделать, чтобы?», «что будет, если?», анализ чувствительности и др.;
- быстрая и адекватная интерпретация результатов моделирования;
- оперативная подготовка и корректировка входных параметров и ограничений модели;
- графическое отображение динамики модели;
- возможность объяснения пользователю необходимых шагов формирования и работы модели.

1.4 Состав ИС

Любая информационная система, независимо от сферы применения (назначения) должна включать в себя необходимый набор элементов (основных компонентов), смотри на рисунке 1.5.

Функциональные подсистемы – это информационный обмен между определенными задачами в системе. Например, учет готовой продукции, оформление заказов, выдача заказов и т.д.

За счет информационного обеспечения происходит интеграция функциональных подсистем в единую систему. Состав системы определяется размером предприятия, отрасли в которой предприятие работает, формы собственности, размером, видом деятельности.

Функциональная система строится по разным принципам: *предметному, функциональному, проблемному, смешанному*.

Функциональные системы построенные по предметному принципу построены на управлении производственными и финансовыми ресурсами с возможностью решения задач на всех уровнях управления (т.е. управление сосредоточено на производстве, маркетинге, логистике, финансах).

Функциональная система построенная на проблемном принципе при помощи программ поддержки и принятия решения, таких как бизнес-планирование, информационные системы руководителей и т.д.

Однако в чистом виде не один принцип не реализуется, на практике применяются *смешанные типы*, которые соответствуют системе управления на предприятии, выполняемым функциям и задачам.



Рисунок 1.5 – Состав ИС

Информационное обеспечение системы – принципы организации и размещения информации, документации, ведения документооборота, форм документов, создания баз и запросов и т.д. Информационное обеспечение включает в себя техническое обеспечение, которое является комплексом средств, при помощи которого и функционирует система.

Программное обеспечение ИС – это совокупность средств и методов работы системы при ее эксплуатации.

Лингвистическое обеспечение системы – это применяемые языки программирования при создании системы, а также языки для общения пользователя с системой в процессе эксплуатации.

Математическое обеспечение – совокупность алгоритмов и программ, используемых в системе для решения задач и обработки информации.

Обеспечивающие подсистемы – общие для всей ИС, которые не зависят от конкретных функциональных подсистем, т.к. обеспечивают реализацию целей и функций системы.

Обеспечивающие объединяют в себя такие подсистемы как

- *информационные* – возможность ввода и вывода информации;
- *организационные* – возможность интеграции персонала при работе и управлении системы;

- *правовые* – возможность интеграции правовых актов при работе с заказчиками, поставщиками, клиентами, т.е. правовое регулирование отношений. Также возможность придание документам получаемых их ИС юридической силы.

1.5 Классификация ИС

1.5.1 Классификация информационных систем по функциональному признаку и уровням управления

Функциональный признак определяет назначение подсистемы, а также ее основные цели, задачи и функции. На практике ИС классифицируются по видам деятельности, таким: *производственная, маркетинговая, финансовая, кадровая.*

Производственная – выпуск продукции, создание и внедрение новшеств, организация МТС (материально-технического снабжения).

Финансовая деятельность – анализ и контроль финансов предприятия при помощи бухгалтерской, статистической и другого рода отчетности.

Кадровая деятельность организации – подбор необходимых специалистов, ведение документации.

На основании этих видов деятельности сформировались основные направления ИС, рисунок 1.6, рисунок 1.7



Рисунок 1.6 – Основные направления деятельности ИС

Система маркетинга	Производственные системы	Финансовые и учетные системы	Система кадров (человеческих ресурсов)	Прочие системы, например ИС руководства
Исследование и анализ рынка, прогнозирование продаж	Планирование объемов работ и разработка календарных планов	Управление портфелем заказов	Анализ и прогнозирование потребности в трудовых ресурсах	Контроль за деятельностью фирмы
Управление продажами	Оперативный контроль и управление производством	Управление кредитной политикой	Ведение архивов записей о персонале	Выявление оперативных проблем
Рекомендации по производству новой продукции	Анализ работы оборудования	Разработка финансового плана	Анализ и планирование подготовки кадров	Анализ управленческих и стратегических ситуаций
Анализ и установление цены	Участие в формировании заказов поставщикам	Финансовый анализ и прогнозирование		Обеспечение процесса выработки стратегических решений
Учет заказов	Управление запасами	Контроль бюджета		

Рисунок 1.7 – Функции ИС

1.5.2 Прочие классификации информационных систем

Информационные системы также классифицируют и по другим признакам, в зависимости от автоматизации, степени использования

информации и человека, характеру использования информации, алгоритмам, как представлено на рисунке 1.8.

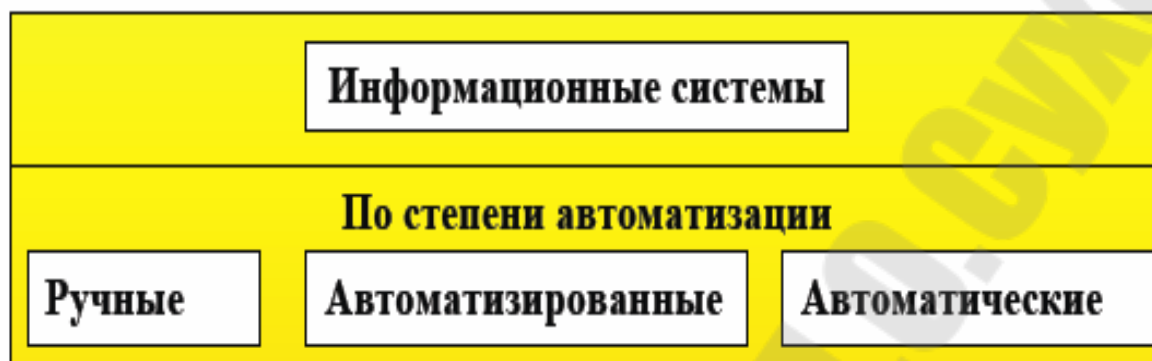


Рисунок 1.8 – Классификация ИС по степени автоматизации

Ручные ИС – это такие системы, в которых полностью отсутствуют технические средства переработки информации и все операции выполняются непосредственно человеком (персоналом).

Автоматические ИС, наоборот, – это системы, в которых все процессы выполняются в автоматическом режиме при минимальном участии человека (т.е. задается алгоритм действий и запускается процесс, где дальнейшее участие персонала не требуется).

Автоматизированные ИС – это системы, которые в процессе обработки информации используют персонал и технические средства, главная роль в этом процессе отведена ПК.

Интегрированные ИС автоматизируют все процессы организации в организации (предприятии) начиная с проектирования и до поставки потребителю (клиенту). Данные системы требуют при создании системного подхода с четко определенной главной целью предприятия (например, завоевания новых рынков сбыта, расширение внутренне сети, роста прибыли и т.д.).

Реализуя системный подход в проектировании данной системы, может полностью поменяться структура предприятия (организации), на что не всегда могут решиться руководители предприятий.

Информационные системы организационного управления позволяют автоматизировать управление персоналом в организации (например, оперативное управление сбытом и т.д.).

Информационные системы управления технологическими процессами используются для поддержания технологических процессов в машиностроении и промышленности.

САПР предназначены для проектировщиков, инженеров, конструкторов, архитекторов, дизайнеров и т.п. при создании документации, моделирования объектов и т.д.

ИПС (*информационно-поисковые системы*) позволяют хранить, вводить, перерабатывать информацию по запросу пользователя, без каких либо дополнительных манипуляций.

ИР системы (*информационно-решающие системы*) обрабатывают информацию по определенному алгоритму.

Различают два вида выбранной информации: *по принятию решений и предоставлению информации для последующего анализа с целью принятия решения.*

ИР по принятию решений сразу выдают информацию по результатам которой принимается решение. Например расчет объема выпуска продукции, учет и т.д.

ИР для последующего анализа предоставляют информацию, которая используется для сведений, которые в дальнейшем способствуют принятию решений. Данная система обрабатывает не данные, а знания.

Например, система рассчитает максимально возможный объем производства в определенном периоде, а руководство предприятия принимает план выпуска с учетом этих сведений.

На рисунке 1.9 представлена классификация ИС *по масштабу и интеграции компонент:*

- *локальный АРМ* (автоматизированное рабочее место) – программно-технический комплекс, предназначен для реализации управленческих функций на отдельном рабочем месте; информационно и функционально не связан с другими информационными системами;
- *комплекс информационно и функционально связанных АРМ*, реализующих в полном объеме функции управления;
- *компьютерная сеть АРМ на единой информационной базе*, обеспечивающая интеграцию функций управления в масштабе предприятия или группы бизнес-единиц;
- *корпоративная информационная система (КИС)*, обеспечивающая полнофункциональное распределенное управление крупномасштабным предприятием.



Рисунок 1.9 – Классификация ИС по масштабу и интеграции компонент

По характеру обработки информации на различных уровнях управления предприятием на рисунке 1.10 представлена классификация ИС.



Рисунок 1.10 – Классификация ИС по характеру обработки информации на различных уровнях управления предприятием

EDP – Electronic data processing – системы обработки данных, которые предназначены для ведения учета и оперативного управления хозяйственными операциями, создания и подготовки стандартных документов (в соответствии с законодательством) таких как платежные поручения, отчеты и т.д. Оперативное управление включает в себя регистрацию и обработку определенных событий, т.е. выполнение заказов, отгрузка потребителям и другое. Оперативное управление включает в себя постоянный мониторинг информации, который носит

регулярный характер, а также непосредственно связан с оформлением и передачей (пересылкой) информации – т.е. четкий алгоритм действий. Данные выводятся в виде определенных схем, форм, таблиц, т.е. форма документооборота регламентирована.

К данным системам можно отнести ИС обработки первичной документации, такие как учет запасов готовой продукции, складской учет, бухгалтерский и т.д., т.е. системы, которые собирают и регистрируют данные. В таких системах обычно используются простые алгоритмы расчетов и запросов, но применяется хорошая защита баз данных от несанкционированного доступа, сбоев при работе. Для эффективной работы используют компьютерные сети – КЛИЕНТ – СЕРВЕР.

MIS – Management Information System – информационные системы управления, которые ориентированы на среднесрочное планирование. Данные ИС в основном используются в отделах снабжения, сбыта и т.п. для прогнозирования объемов, составления производственных программ и т.п.

DSS – Decision Support System – системы поддержки принятия решений, предназначены для руководителей верхнего уровня с целью формирования стратегических планов, привлечения финансирования и т.п.

Данные ИС ориентированы на реализацию сложных процессов организации, которые требуют проведения аналитической обработки данных, и имеет целевую ориентацию. Например, анализ финансового состояния предприятия.

На рисунке 1.11 представлена классификация ИС по уровням управления.



Рисунок 1.11 – Классификация ИС по уровням управления

Информационные системы оперативного уровня обрабатывают данные, в которой задачи, источники и цели структурированы и определены. Такими системами являются – банковские системы, бухгалтерские, обработки заказов и др. В таких системах фиксируются сделки и события (кредиты, счета, поток материалов, зарплата и т.п.). Такая система – основной источник информации для других систем.

Информационные системы специалистов позволяют обрабатывать бумажные документы, тем самым интегрируя новые данные в систему.

Информационные системы менеджмента предназначены для мониторинга, принятия решений. Назначение данных систем анализ и сравнение показателей, составление отчетов, хранение информации, доступ к информации и пр. Среди данных систем выделяют:

- система для контроля и принятия решения, на основе полученной информации;
- система поддержки принятия решений, которая решает частично структурированные задачи, а также имеет возможность моделировать и составлять прогнозы.

Стратегические информационные системы предназначены для реализации долгосрочного планирования, за счет анализа изменений во внешней среде и организации.

На рисунке 1.12 представлена укрупненная классификация систем, предназначенных для автоматизации различных видов хозяйственного учета.



Рисунок 1.12 – Укрупненная классификация систем, предназначенных для автоматизации различных видов хозяйственного учета

Локальные системы – не дорогие системы для отдельных задач учета на предприятии, автоматизируют учет по отдельным модулям, не предоставляя целостной информации. Основное преимущество низ-

кая цена и легкость при внедрении. Это такие ИС, как «1С:Бухгалтерия» фирмы «1С», «ИнфоБухгалтер» фирмы «Информатик» и др. Данные программы хорошо адаптируются на предприятиях.

Средние интегрированные системы – системы, имеющие ограниченные функциональные возможности. Например КИС «Галактика» фирмы «Галактика», «Парус» фирмы «Парус», «БЭСТ ПРО» фирмы «Интеллект-сервис», «1С: Предприятие» фирмы «1С» и др.

Крупные интегрированные системы, которые развитые, функциональны, сложны. В данных системах реализуются стандарты MRP, ERP, SCRP. Такими программами являются «SAP» фирмы «R3 (Accelerated Solutions)», «PeopleSoft» фирмы «PeopleSoft Select», «Галактика ERP» фирмы Галактика и др. В данных программах автоматизируются все процессы предприятия, все модули взаимосвязаны. Данные системы предназначены для управления производственными и торговыми предприятиями, а также бюджетными и страховыми организациями, данные системы позволяют осуществить переход к международным стандартам отчетности.

1.6 Типы ИС

В зависимости от уровня управления и выполняемых задач можно выделить ИС:

- по уровням управления;
- по уровню квалификации работника.

ИС можно представить в виде пирамиды, в основании которой лежат кадры, финансы, производство, маркетинг, а сотрудники данного уровня являются исполнителями, оперативное управление.

На следующем уровне располагаются менеджеры среднего звена (специалисты), занимающиеся тактическим управлением.

Далее менеджеры высшего звена – стратегическое управление.

1.7 Жизненный цикл ИС

Необходимость проектирования ИС может обуславливаться разработкой и внедрением информационных технологий в организации (построение новой информационной системы) либо при модернизации существующих информационных процессов, либо при реорга-

низации деятельности предприятия (проведении бизнес-реинжиниринга). Потребности проектирования ИС указывают:

- для достижения каких целей необходимо разработать систему;
- к какому моменту времени целесообразно осуществить разработку;
- какие затраты необходимо осуществить для проектирования системы.

Проектирование ИС является трудоемким, длительным и динамическим процессом. Технологии проектирования, применяемые в современных условиях, предполагают поэтапную разработку системы. Этапы по общности целей могут объединяться в стадии. Совокупность стадий и этапов, которые проходит ИС в своем развитии от момента принятия решения о создании системы до момента прекращения функционирования системы, называется *жизненным циклом ИС*.

Планирование и анализ требований (предпроектная стадия) – системный анализ. Проводится исследование и анализ существующей информационной системы, определяются требования к создаваемой ИС, формируются технико-экономическое обоснование (ТЭО) и техническое задание (ТЗ) на разработку ИС.

Системный анализ. Основными целями этапа являются: 1) формулировка потребностей в новой ИС (определение всех недостатков существующей ИС); 2) выбор направления и определение экономической обоснованности проектирования ИС.

Системный анализ ИС начинается с описания и анализа функционирования рассматриваемого объекта в соответствии с требованиями (целями), которые предъявляются к нему. В результате этого этапа выявляются недостатки существующей ИС, на основе которых формулируется потребность в совершенствовании системы управления этим объектом, и ставится задача определения экономически обоснованной необходимости автоматизации определенных функций управления (создается технико-экономическое обоснование проекта ИС). После определения этой потребности возникает проблема выбора направлений совершенствования объекта на основе выбора программно-технических средств. Результаты оформляются в виде технического задания на проект, в котором отражаются технические условия и требования к ИС, а также ограничения на ресурсы проекти-

рования. Требования к ИС определяются в терминах функций, реализуемых системой.

Проектирование (техническое и логическое проектирование).

В соответствии с требованиями формируются состав автоматизируемых функций (функциональная архитектура) и состав обеспечивающих подсистем (системная архитектура), проводится оформление технической проектной ИС.

Этап проектирования предполагает:

- проектирование функциональной архитектуры ИС, которая отражает структуру функциональных подсистем и связей между ними; является наиболее ответственным и важным этапом с точки зрения качества всей последующей разработки ИС;
- проектирование системной архитектуры ИС (состав обеспечивающих подсистем). Построение системной архитектуры на основе функциональной предполагает определение элементов и модулей информационного, технического, программного обеспечения и других обеспечивающих подсистем, связей по информации и управлению между выделенными элементами и разработку технологии обработки информации.

Реализация (рабочее и физическое проектирование, кодирование). Разработка и настройка программ, формулировка рабочих инструкций для персонала, создание информационного обеспечения, включая формирование и наполнение баз данных, оформление рабочего проекта.

Важной особенностью жизненного цикла ИС является его повторяемость (цикличность) «системный анализ» – «разработка» – «сопровождение» – системный анализ». Это соответствует представлению об ИС, как о развивающейся, динамической системе.

Существуют различные модели жизненного цикла ИС. Среди известных можно выделить следующие.

Каскадная модель (до 70-х годов) – последовательный переход на следующий этап только после полного завершения предыдущего.

Достоинство – планирование времени осуществления всех этапов проекта, упорядочении хода конструирования.

Недостатки каскадной модели:

- модель недостаточно гибкая – реальные проекты часто требуют отклонения от стандартной последовательности шагов;
- цикл основан на точной формулировке исходных требований к программному обеспечению (ПО) (реально в начале проекта требования заказчика определены лишь частично);
- результаты проекта доступны заказчику только в конце работы.

Итерационная модель (70–80-е годы) – с итерационными возвратами на предыдущие этапы после выполнения очередного этапа.

Построение комплексных ИС подразумевает согласование проектных решений, получаемых при реализации отдельных задач. Подход к проектированию «снизу вверх» предполагает необходимость таких итерационных возвратов, когда проектные решения по отдельным задачам объединяются и общие системные решения, и при этом возникает потребность в пересмотре ранее сформулированных требований.

Недостаток. Вследствие большого числа итераций возникают рассогласования и несоответствия в выполненных проектных решениях и документации.

Спиральная модель (80–90-е годы) прототипная модель, предполагающая постепенное расширение ПО.

Спиральная модель определяет четыре действия, представляемые четырьмя квадрантами спирали:

- планирование – определение целей, вариантов и ограничений;
- анализ риска – анализ вариантов и распознавание (выбор) риска;
- конструирование – разработка продукта следующего уровня;
- оценивание – оценка заказчиком текущих результатов конструирования.

С каждой итерацией по спирали (продвижением от центра к периферии) строятся все более полные версии ПО.

Спиральная модель жизненного цикла ИС реально отображает разработку программного обеспечения; позволяет явно учитывать риск на каждом витке эволюции разработки; включает шаг системного подхода в итерационную структуру разработки; использует моделирование для уменьшения риска и совершенствования программного изделия.

Недостатками спиральной модели являются:

- новизна (отсутствует достаточная статистика эффективности модели);
- повышенные требования к заказчику;
- трудности контроля и управления временем разработки.

В основе спиральной модели жизненного цикла лежит применение *прототипной технологии или RAD-технологии (rapid application development – технологии быстрой разработки приложений)*.

Основная идея этой технологии заключается в том, что ИС разрабатывается путем расширения программных прототипов, повторяя путь от детализации требований к детализации программного кода.

При прототипной технологии сокращается число итераций, возникает меньше ошибок и несоответствий, которые необходимо исправлять на последующих итерациях, а само проектирование ИС осуществляется более быстрыми темпами, упрощается создание проектной документации. Для более точного соответствия проектной документации разработанной ИС все большее значение придается использованию CASE-технологий.

RAD-технология обеспечивает экстремально короткий цикл разработки ИС. При полностью определенных требованиях и ограниченной проектной области RAD-технология позволяет создать полностью функциональную систему за очень короткое время (60-90 дней).

Выделяют следующие этапы разработки ИС с использованием RAD-технологии:

- бизнес-моделирование. Моделируется информационный поток между бизнес-функциями. Определяются ответы на вопросы: Какая информация руководит бизнес-процессом? Какая информация генерируется? Кто генерирует ее? Где информация применяется? Кто обрабатывает информацию?
- моделирование данных. Информационный поток отображается в набор объектов данных, которые требуются для поддержки деятельности организации. Определяются характеристики (свойства, атрибуты) каждого объекта, отношения между объектами;
- моделирование обработки. Определяются преобразования объектов данных, обеспечивающие реализацию бизнес-функций. Создаются описания обработки для добавления, модификации, удаления или нахождения (исправления) объектов данных;

- генерация приложения. Предполагается использование методов, ориентированных на языки программирования 4-го поколения. Вместо создания ПО с помощью языков программирования 3-го поколения, RAD-процесс работает с повторно используемыми программными компонентами или создает повторно используемые компоненты. Для обеспечения конструирования используются утилиты автоматизации (CASE-средства);
- тестирование и объединение. Поскольку применяются повторно используемые компоненты, многие программные элементы уже протестированы, что сокращает время тестирования (хотя все новые элементы должны быть протестированы).

Применение RAD имеет и свои недостатки, и ограничения:

- большие проекты в RAD требуют существенных людских ресурсов (необходимо создать достаточное количество групп);
- RAD применима только для приложений, которые можно разделять на отдельные модули и в которых производительность не является критической величиной;
- RAD неприменима в условиях ВЫСОКИХ технических рисков.

1.8 Основы применения инструментальных средств ИТ

Переход на промышленную технологию производства программ, стремление к сокращению сроков, трудовых и материальных затрат на производство и эксплуатацию программ, обеспечение гарантированного уровня качества ИС обусловили бурно развивающееся направление – программотехнику, связанное с технологией создания программных продуктов.

Инструментарий технологии программирования – программные продукты поддержки технологии программирования. В рамках этих направлений сформировались следующие группы:

- средства для создания приложений;
- CASE-технологии (Computer-Aided Software Engineering), предназначенные для автоматизации процессом разработки и реализации информационных систем.

Средства для создания приложений включают языки и системы программирования, а также инструментальную среду разработчика.

Язык программирования – формализованный язык для описания алгоритма решения задачи на компьютере.

Средства для создания приложений – совокупность языков и систем программирования, а также различные программные комплексы для отладки и поддержки разрабатываемых программных продуктов.

Языки программирования разделяют на следующие классы (по синтаксису конструкций языка):

- машинные языки – языки программирования, воспринимаемые аппаратной частью компьютера (машинные коды);
- машинно-ориентированные языки – языки программирования, которые отражают структуру конкретного типа компьютера (ассемблеры);
- алгоритмические языки, не зависящие от архитектуры компьютера языки программирования для отражения структуры алгоритма (Паскаль, Фортран, Бейсик и др.);
- процедурно-ориентированные языки – языки программирования, где имеется возможность описания п-граммы как совокупности процедур, подпрограмм;
- проблемно-ориентированные языки – языки программирования, предназначенные для решения задач определенного класса (ЛИСП, РПГ, Симула и др.);
- интегрированные системы программирования.

Другой классификацией языков программирования является их деление на языки, предназначенные для реализации основ структурного программирования, и объектно-ориентированные языки, поддерживающие понятие объектов, их свойств и методов обработки.

Программа, написанная на языке программирования, проходит этап трансляции, когда происходит преобразование исходного кода программы и объектный код, который далее пригоден к обработке редактором связей.

Редактор связей – специальная подпрограмма, обеспечивающая построение загрузочного модуля, пригодного к выполнению.

Трансляция может выполняться с использованием средств компиляторов или интерпретаторов.

Компиляторы транслируют всю программу, но без ее выполнения.

Интерпретаторы, в отличие от компиляторов, выполняют по-операторную обработку и выполнение программы.

Необходимым средством для профессионального разработчика являются специальные программы, предназначенные для трассировки и анализа выполнения других программ, отладчики.

Современная система программирования состоит из следующих компонентов:

- компилятор;
- интегрированная среда разработчика программ;
- отладчик;
- средства оптимизации кода программ;
- набор библиотек (возможно, с исходными текстами программ); редактор связей;
- сервисные средства (утилиты) для работы с библиотеками, текстовыми и двоичными файлами;
- справочные системы;
- документатор исходного кода программы;
- систему поддержки и управления проектом программного комплекса.

CASE-технология – программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем. Основное преимущество CASE-технологии – возможность коллективной работы над проектом за счет поддержки работы разработчиков в локальной сети, экспорта-импорта любых фрагментов проекта организационного управления проектом.

1.9 Создание автоматизированных информационных систем

Создание автоматизированных информационных систем (АИС) регламентируется комплексом стандартов и руководящих документов. Можно выделить следующие стадии и этапы создания АИС (рисунки 1.13).

Стадии	Этапы работ
1. Формирование требований к ИС	1.1. Обследование объекта и обоснование необходимости создания ИС. 1.2. Формирование требований пользователя к ИС. 1.3. Оформление отчета о выполненной работе и заявки на разработку ИС (тактико-технического задания)
2. Разработка концепции ИС	2.1. Изучение объекта. 2.2. Проведение необходимых научно исследовательских работ. 2.3. Разработка вариантов концепции ИС и выбор варианта концепции ИС, удовлетворяющего требованиям пользователя. 2.4. Оформление отчета о выполненной работе
3. Техническое задание	3.1. Разработка и утверждение технического задания на создание ИС (см. ниже)
4. Эскизный проект	4.1. Разработка предварительных проектных решений по системе и ее частям. (см. ниже) 4.2. Разработка документации на ИС и ее части
5. Технический проект	5.1. Разработка проектных решений по системе и ее частям. (см. ниже) 5.2. Разработка документации на ИС и ее части.
	5.3. Разработка и оформление документации на поставку изделий для комплектования ИС и (или) технических требований (технических задания) на их разработку. 5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации
6. Рабочая документация	6.1. Разработка рабочей документации на систему и ее части. 6.2. Разработка или адаптация программ (см. ниже)
7. Ввод в действие	7.1. Подготовка объекта автоматизации к вводу ИС в действие. 7.2. Подготовка персонала. 7.3. Комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями). 7.4. Строительно-монтажные работы. 7.5. Пусконаладочные работы. 7.6. Проведение предварительных испытаний. 7.7. Проведение опытной эксплуатации. 7.8. Проведение приемочных испытаний

Рисунок 1.13 – Стадии и этапы создания АИС

Одним из центральных элементов всего процесса создания АИС является разработка *технического задания*, структура которого, согласно ГОСТ 34.602–89, содержит следующие разделы:

1. общие сведения;
2. назначение и цели создания (развития) системы;
3. характеристика объектов автоматизации;
4. требования к системе:
 - 4.1. требования к системе в целом – отражают концептуальные параметры и характеристики создаваемой системы, среди которых указываются требования к структуре и функционированию системы,

к надежности и безопасности, к численности и квалификации персонала и т. д.

4.2. требования к функциям (задачам), выполняемым системой – содержат перечень функций, задач или их комплексов; временной регламент каждой функции, задачи или комплекса задач; требования к качеству реализации каждой функции; к форме представления выходной информации; характеристики необходимой точности и времени выполнения, требования одновременности выполнения группы функций; достоверности выдачи результатов.

4.3. требования к видам обеспечения:

- к составу, структуре и способам организации данных в системе (информационно-логическая схема);
- к информационному обмену между компонентами системы;
- к информационной совместимости со смежными системами;
- по использованию общероссийских и других классификаторов, унифицированных документов;
- по применению систем управления базами данных;
- к структуре процесса сбора, обработки, передачи данных в системе И представлению данных;
- к защите данных от разрушений при авариях и сбоях в электропитании системы;
- к контролю, хранению, обновлению и восстановлению данных;
- к процедуре придания юридической силы документам, продуцируемым техническими средствами ПС;

5. состав и содержание работ по созданию системы;

6. порядок контроля и приемки системы;

7. требования к составу и содержанию работ ПО подготовке объекта автоматизации к вводу системы в действие;

8. требования к документированию;

9. источники разработки.

На основе установленных в техническом задании основных требований и технических решений на последующих этапах конкретизируются и непосредственно разрабатываются компоненты и элементы системы.

В частности, на этапе 4.1 «Разработка предварительных проектных решений по системе и ее частям» определяются:

- функции ИС;
- функции подсистем;

- концепция информационной базы и ее укрупненная структура;
- функции системы управления базой данных;
- состав вычислительной системы;
- функции и параметры основных программных средств.

На этапе 5.1 «Разработка проектных решений по системе и ее частям» осуществляется разработка общих решений по системе и ее частям:

- по функционально-алгоритмической структуре системы;
- по функциям персонала и организационной структуре;
- по структуре технических средств;
- по алгоритмам решения задач и применяемым языкам;
- по организации и ведению информационной базы (структура базы данных);
- по системе классификации и кодирования информации (словарно-классификационная база);
- по программному обеспечению.

Разработка и документация программного обеспечения в процессе создания или комплектования автоматизированных систем (п. 6.2) регламентируются комплексом стандартов, объединенных в группу «Единая система программной документации (ЕСПД)».

2 СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ (САПР) И ИХ МЕСТО СРЕДИ ДРУГИХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

2.1 Системы управления жизненным циклом в современном машиностроении

2.1.1 Жизненный цикл изделия (ЖЦИ). Этапы жизненного цикла изделия

Жизненный цикл промышленных изделий включает ряд этапов, начиная от зарождения идеи нового продукта до утилизации по окончании срока его использования. Основные этапы жизненного цикла изделия представлены на рисунке 2.1.

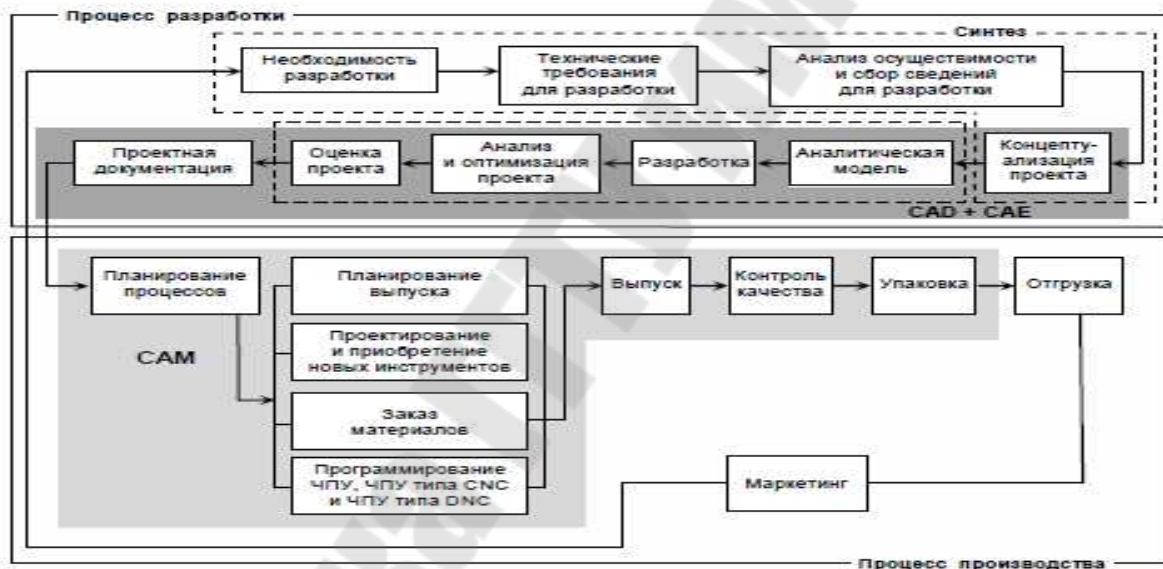


Рисунок 2.1 – Этапы жизненного цикла изделия

Совокупность видов деятельности и функций, необходимых для осуществления проекта и изготовления изделий, называется *производственным циклом*. Поведение этого цикла определится заказчиком изделий и потребностями рынков сбыта.

Производственный цикл может активизироваться различными путями в зависимости от особенностей той или иной группы заказчиков-потребителей. В одних случаях функции проектирования выполняет сам заказчик, а изготовление изделия – сторонние фирмы, в других и проектирование и производство выполняет одна и та же фирма. Однако, как бы это не происходило, производственный цикл всегда

начинается с выработки концепций нового изделия (т.е. возникновение определенной идеи).

Первоначально концепция нового изделия прорабатывается, затем в деталях уточняется, анализируется, совершенствуется и после конструктивной разработки воплощается в план выполнения нового изделия. Этот план подкреплен документацией в виде набора конструктивных чертежей, показывающих, как должно изготавливаться изделие и совокупность технических описаний, спецификаций, отражающих принципы функционирования изделия. Этим завершается деятельность по проектированию, если не считать конструкцию изделия на протяжении всего ЖЦ.

Следующим видом деятельности становится *изготовление изделия*, которое начинается составлением плана, определяющего последовательность операций, необходимых для выполнения изделия. Иногда могут потребоваться новое оборудование, инструмент и оснастка.

На этапе составления графика производства выбранного плана, в соответствии с которым фирма фактически принимает на себя обязательство выпустить определенное количество изделий в конкретные сроки. После того, как сформированы планы-графики по всем изделиям, они запускаются в производство, и затем проходят через операции контроля качества и отгрузки заказчику.

Достижение поставленных целей на современных предприятиях, выпускающих сложные промышленные изделия, оказывается невозможным без широкого использования автоматизированных систем (АС), основанных на применении компьютеров и предназначенных для создания, переработки и использования всей необходимой информации о свойствах изделий и сопровождающих процессов. Специфика задач, решаемых на различных этапах жизненного цикла изделий, обуславливает разнообразие применяемых АС.

2.1.2 Информация об изделии

Информация об изделии – это набор данных, которые порождаются и используются на всем протяжении его жизненного цикла (ЖЦ) и включают:

- информацию о конфигурации и структуре изделия;
- характеристики и свойства;

- организационную информацию (описание процессов, связанных с изменением данных об изделии, необходимые ресурсы – люди, материалы, т.д.;
- информацию о проведенных контрольных испытаниях;
- документы, которыми обрастает изделие с момента его проектирования до его продажи и дальнейшего обслуживания, и т.д.

Классификация информации об изделии по этапам ЖЦ.

1. Конструкторские данные об изделии – совокупность информационных объектов, порождаемых в процессе проектирования и разработки изделия, содержащая сведения:

- о составе изделия;
- о геометрических моделях изделия;
- об его компонентах и их технических характеристиках;
- об отношениях компонентов в структуре изделия;
- о результатах расчетов и моделирования;
- о допусках на изготовление деталей и т.д.

2. Технологические данные об изделии – совокупность информационных объектов, порождаемых на стадии технологической подготовки производства и ассоциированных с информационными объектами, описывающими изделие и его компоненты. Содержит сведения:

- о способах изготовления и контроля изделия;
- описание маршрутных и операционных технологий;
- нормы времени и расхода материалов;
- управляющие программы для станков с ЧПУ;
- данные для проектирования оснащения.

3. Производственные данные об изделии – содержат сведения о статусе конкретных экземпляров изделия и его компонентов в производственном цикле.

4. Данные о качестве изделия – совокупность информационных объектов, порождаемых при выполнении всех видов контроля. Содержат сведения о степени соответствия изделия заданным техническим требованиям, стандартам и др.

5. Логистические данные об изделии – порождаются в процессе проектирования и разработки, содержат сведения о пространственно-временной увязки планирования взаимодействий всех субъектов и для поддержки изделия на постпроизводственных стадиях ЖЦ.

6. Эксплуатационные данные об изделии – порождаются в процессе проектирования и разработки, содержат руководства по эксплуатации, сведения для организации обслуживания, ремонта и других действий, обеспечивающих работоспособность изделия.

Единая интегрированная модель изделия:

- содержит всю информацию об изделии (его свойства, знания о нем и его производстве), требуемую на любом из этапов ЖЦИ;
- сопровождает изделие на всем протяжении его ЖЦ от замысла до утилизации;
- при построении каждого модуля модели должны использоваться единые средства и методы построения моделей и обеспечение целостности всей модели, описывающей изделие.

На рисунке 2.2 представлена интегрированная модель изделия.

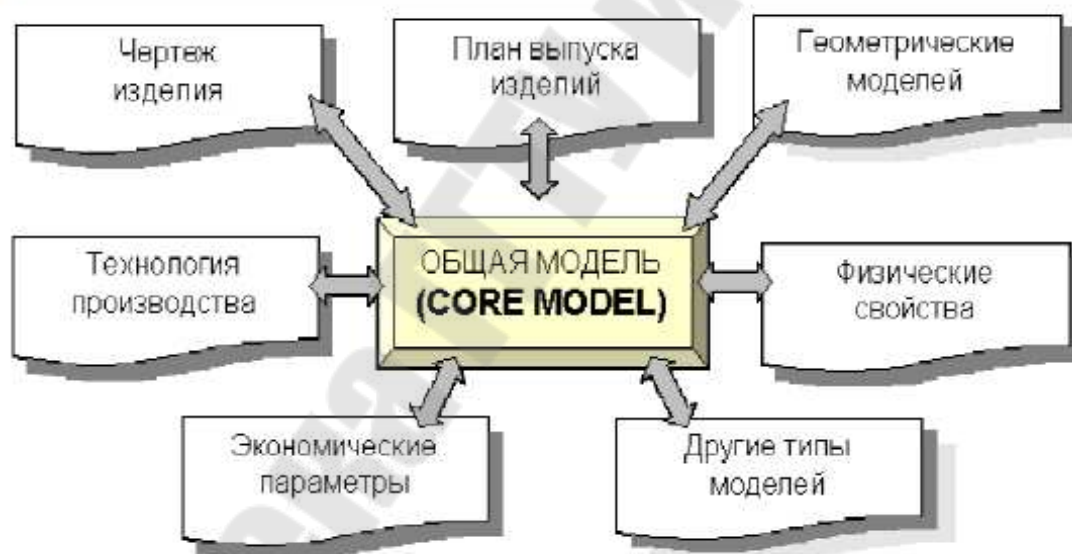


Рисунок 2.2 – Интегрированная модель изделия

2.1.3 Понятие PLM-технологии

Управление производством. Для решения проблем совместного функционирования компонентов САПР различного назначения, координации работы систем CAE/CAD/CAM, управления проектными данными и проектированием разрабатываются системы, получившие название систем управления проектными данными PDM (Product Data Management).

Системы PDM либо входят в состав модулей конкретной САПР, либо имеют самостоятельное значение и могут работать совместно с разными САПР.

Главная цель PDM – поддержка электронного описания продукта (изделия) на всех стадиях его жизненного цикла

Под PLM (Product Lifecycle Management) понимают систему управления жизненным циклом продукции реализующую технологии управления данными в информационном пространстве, едином для различных автоматизированных систем.

Технологии PLM объединяют методики и средства информационной поддержки изделий на протяжении всех этапов жизненного цикла изделий.

Характерная особенность PLM – обеспечение взаимодействия как средств автоматизации разных производителей, так и различных автоматизированных систем многих предприятий, то есть технологии PLM (включая технологии CPC(*Collaborative Product Commerce*)) являются основой, объединяющей информационное пространство, в котором функционируют САПР, ERP, PDM, SCM, CRM и другие автоматизированные системы многих предприятий.

В целом же PLM – это совокупность программных систем CAE/CAD/CAM/PDM и ERP/CRM/SCM, методики их применения, а главное – людей, обладающих должной компетентностью. Можно сказать, это целая философия жизни производства, опирающаяся на те, самые, программные средства. Перевод мыслей, идей, опыта сотрудников предприятия в явные знания предприятия в виде документов, схем, чертежей и т.д., является важной задачей в управлении ЖЦИ. Процесс наполнения банка знаний предприятия постоянен и непрерывен.

PLM – это стратегический подход к бизнесу, при котором применяется набор интеллектуальных инструментальных средств, поддерживающих совместное создание, управление, изменение и использование данных о продукции в рамках расширенной модели предприятия (то есть модели предприятия, включающей, помимо собственно производства, привлечение к работе с данными о продукции как контрагентов, так и конечных заказчиков продукции), и обмен данными осуществляется в рамках единого информационного пространства (включая единые бизнес-процессы, бизнес-приложения и данные).

При этом PLM управляет данными, находящимися преимущественно в цифровом виде. При этом особо подчеркивается, что PLM не является какой-либо конкретной технологией или каким-либо отдельным программным продуктом. Таким образом, речь идет о стратегическом подходе, для реализации которого требуется использовать не одну, а несколько систем в рамках единого интегрированного решения (причем не исключено, что эти системы могут быть от разных производителей).

Концепция PLM возникла в отраслях, связанных с разработкой и производством сложных технических изделий (авиационно-космическая отрасль, оборонно-промышленный комплекс, точное машиностроение и др.). Поэтому довольно долго PLM-решения ассоциировались именно с предприятиями данного профиля, и предполагалось, что они оперируют в основном техническими (в первую очередь конструкторскими) данными. Но в последние годы ситуация качественно изменилась.

Теперь уже не имеет значения, что именно понимается под словом «продукция» – станок, корабль, самолет, нефтепровод или сложная информационная система. Задачи, решаемые PLM-системой, представлены на рисунке 2.3.



Рисунок 2.3 – Задачи, решаемые PLM-системой

Концепция PLM предполагает, что создается единая информационная база, описывающая три краеугольные компоненты: *Продукт - Процессы - Ресурсы* и взаимосвязи между ними.

Наличие такой объединенной модели обеспечивает возможность быстро, эффективно увязывать все эти три компонента, оптимизируя решение под требования бизнеса. Работа всех проектантов, конструкторов, технологов с единой моделью обеспечивает снижение издержек на многочисленные согласования, неизбежные при традиционной

технологии работы, и исключает наличие дублирующих или взаимоисключающих документов. На практике это позволяет значительно сократить материальные и временные затраты на создание продукта и запуск его в производство, минуя многочисленные отладочные варианты, воплощаемые в реальности, то есть получить проект продукта, готового буквально с первых экземпляров к отправке потребителю.

Мировая практика уже имеет примеры в даже таких сложных отраслях, как, например, авиастроение, когда самый первый собранный самолет нового проекта после проверочных испытаний был передан в реальную эксплуатацию. Конечно, такие идеальные случаи все-таки редки, но количество испытательно-доводочных вариантов продукции в современной автомобильной, авиационной, станкостроительной промышленности сократилось кардинально, а сроки на создание новых продуктов буквально в разы. Существуют целые классы технических объектов, в которых опытные образцы просто невозможны (например, целый завод) и «натурные эксперименты» на доводку их функционирования до оптимального уровня баснословно дороги.

2.1.4 Понятие CALS-технологии

CALS-технологии (англ. Continuous Acquisition and Life cycle Support – непрерывная информационная поддержка поставок и ЖЦИ) – современный подход к проектированию и производству высокотехнологичной и наукоёмкой продукции, заключающийся в использовании компьютерной техники и современных информационных технологий на всех стадиях ЖЦИ.

Первоначально *CALS* создавалась как совокупность методов и средств решения логистических задач и аббревиатура *CALS* расшифровывалась как Computer Aided Logistics Systems.

В дальнейшем сфера применения *CALS* расширилась и охватила все стороны информационной поддержки промышленных изделий, включая проектирование, управление предприятиями и технологическими процессами. Соответственно *CALS* получила новую интерпретацию и стала рассматриваться как Continuous Acquisition and Life cycle Support.

Что же такое *CALS* в современном понимании?

Существует и используется несколько толкований.

В широком смысле слова CALS – это методология создания единого информационного пространства промышленной продукции, обеспечивающего взаимодействие всех промышленных автоматизированных систем. В этом смысле предметом CALS являются методы и средства как взаимодействия разных АС и их подсистем, так и сами АС с учетом всех видов их обеспечения. Практически синонимом CALS в этом смысле становится термин PLM, широко используемый в последнее время ведущими производителями автоматизированных систем (АС).

В узком смысле слова CALS – это технология интеграции различных АС со своими лингвистическим, информационным, программным, математическим, методическим, техническим и организационным видами обеспечения.

За счет непрерывной информационной поддержки обеспечиваются единообразные способы управления процессами и взаимодействия всех участников этого цикла: заказчиков продукции, поставщиков/производителей продукции, эксплуатационного и ремонтного персонала. Информационная поддержка реализуется в соответствии с требованиями системы международных стандартов, регламентирующих правила указанного взаимодействия преимущественно посредством электронного обмена данными.

ИПИ (информационная поддержка процессов жизненного цикла изделий – русскоязычный аналог понятия **CALS**).

Стратегия CALS объединяет в себе следующие основные принципы:

- применение современных информационных технологий (электронный обмен данными, многопользовательские базы данных, распределённые системы поддержки принятия решений, виртуальное предприятие);
- единство представления и интерпретации данных в процессах информационного обмена между АС и их подсистемами, что обуславливает разработку онтологий приложений и соответствующих языков представления данных;
- доступность информации для всех участников ЖЦИ в любое время и в любом месте, что обуславливает применение современных телекоммуникационных технологий;
- инжиниринг и реинжиниринг бизнес-процессов;
- поддержка процедур совмещенного (параллельного) проектирования изделий;

- унификация и стандартизация средств совместного использования АС и электронного обмена данными;
- интеграцию образования и производства, подготовки и переподготовки специалистов, связь эффективности производства с качеством подготовки и переподготовки специалистов.

Цель реализации CALS-стратегии – качественное повышение эффективности деятельности за счет ускорения процессов исследования, разработки и модернизации продукции.

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

Суть концепции CALS – в организации единого информационного пространства (Интегрированной информационной среды), объединяющего автоматизированные системы, предназначенные как для эффективного решения задач инженерной деятельности, так и для планирования и управления производством и ресурсами предприятия. В единый процесс вовлекается множество проектирующих и машиностроительных предприятий с удаленным доступом к информации, прямой передачей информации от компьютера к машиностроительному оборудованию и т.д.

Интегрированная информационная среда представляет собой совокупность распределенных баз данных, в которой действуют единые, стандартные правила хранения, обновления, поиска и передачи информации, через которую осуществляется безбумажное информационное взаимодействие между всеми участниками жизненного цикла изделия. При этом однажды созданная информация хранится в интегрированной информационной среде, не дублируется, не требует каких-либо перекодировок в процессе обмена, сохраняет актуальность и целостность.

Целостность данных поддерживается в процессе управления конфигурацией проекта, а также тем, что нельзя одновременно изменять один и тот же объект разным разработчикам, каждый из них

должен работать со своей рабочей версией. Другими словами, необходимо обеспечение синхронизации изменения данных, разделяемых многими пользователями.

Для этого выполняется авторизация пользователей и разрабатываются средства ведения многих версий проекта.

Во-первых, пользователи подразделяются на классы (администрация системы, руководство проектом и частями проекта, группы исполнителей-проектировщиков) и для каждого класса вводят определенные ограничения, связанные с доступом к разделяемым данным;

во-вторых, доступ регламентируется по типам разделяемых данных. Данным могут присваиваться различные значения статуса, например, «правильно», «необходимо перевычисление», «утверждено в качестве окончательного решения» и т.п.

Концепция CALS реализуется в виде соответствующих CALS-технологий и определяет набор правил, регламентов, стандартов, взаимодействия участников процессов проектирования, производства, испытаний и т.д.

Назначение CALS-технологий – обеспечивать предоставление необходимой информации в нужное время, в нужном виде, в конкретном месте любому из участников жизненного цикла промышленных изделий.

Развитие CALS-технологий должно привести к появлению так называемых виртуальных производств, в которых процесс создания спецификаций с информацией для программно-управляемого технологического оборудования, достаточной для изготовления изделия, может быть распределён во времени и пространстве между многими организационно-автономными проектными студиями.

Для обеспечения информационной интеграции CALS использует стандарты IGES и STEP в качестве форматов данных. В CALS входят также стандарты электронного обмена данными, электронной технической документации и руководства для усовершенствования процессов.

2.2 Основные принципы создания САПР

2.2.1 Основные понятия теории САПР

Проектирование – это процесс, заключающийся в преобразовании исходного описания объекта в окончательное описание на основе выполнения комплекса работ исследовательского, расчетного и конструкторского характера.

В любом случае проектирование начинается при наличии задания на проектирование, которое отражает потребности в получении некоторого технического объекта. Это задание представляется в виде тех или иных документов и является *исходным (первичным) описанием объекта*. Результатом проектирования, как правило, служит полный комплект конструкторской документации, содержащий достаточные сведения для изготовления объекта в заданных условиях. Эта документация представляет собой *окончательное описание объекта*.

Основным первичным описанием объекта проектирования является техническое задание (ТЗ).

С точки зрения математики ТЗ задает область допустимых значений внутренних и выходных параметров.

Система автоматизированного проектирования – комплекс средств автоматизации проектирования, взаимосвязанных с необходимыми подразделениями проектной организации или коллективом специалистов (пользователей системы), выполняющих автоматизированное проектирование.

Различают автоматизированное и автоматическое проектирование.

Автоматизированным называют проектирование, при котором все требования описаний объекта и алгоритма его функционирования, а также представление описаний на различных языках осуществляются взаимодействием человека и ЭВМ.

Автоматическим является проектирование, при котором все преобразования описаний объекта и алгоритма его функционирования, а также представление описаний на различных языках осуществляются на ЭВМ без участия человека.

2.2.2 Подсистемы САПР

Составными структурными частями САПР являются *подсистемы*, обладающие всеми свойствами систем и создаваемые как само-

стоятельные системы. По назначению подсистемы САПР подразделяются на два вида: *проектирующие и обслуживающие*.

К *проектирующим* относятся подсистемы, выполняющие проектные процедуры и операции, например:

- подсистема оптимизации характеристик изделия;
- подсистема проектирования узлов деталей и сборочных единиц;
- подсистема технологического проектирования;
- подсистема проектирования приспособлений.

Обслуживающими называются подсистемы, предназначенные для расширения возможностей проектирующих подсистем, например,

- подсистема графического отображения объектов проектирования;
- подсистема документирования;
- подсистема обслуживания базы данных.

По отношению к объекту проектирования различают два вида проектирующих подсистем: *объектно-ориентированные (объектные)*; *объектно-независимые (инвариантные)*.

К *объектным* относятся подсистемы, выполняющие одну или несколько проектных процедур или операций, непосредственно зависящих от конкретного объекта проектирования. Инвариантными называются подсистемы, выполняющие унифицированные проектные процедуры и операции.

Проектная процедура – это формализованная совокупность действий, выполнение которых оканчивается проектным решением.

Проектной операцией называют действие или формализованную совокупность действий, составляющих часть проектной процедуры, алгоритм которых остается неизменным для ряда проектных процедур.

Унифицированная проектная процедура – процедура, алгоритм которой остается неизменным для различных объектов проектирования или различных стадий проектирования одного и того же объекта.

2.2.3 Типовые проектные процедуры

Проектная процедура называется *типовой*, если она предназначена для многократного применения при проектировании многих ти-

пов объектов. Классификация типовых проектных процедур представлена на рисунке 2.4.

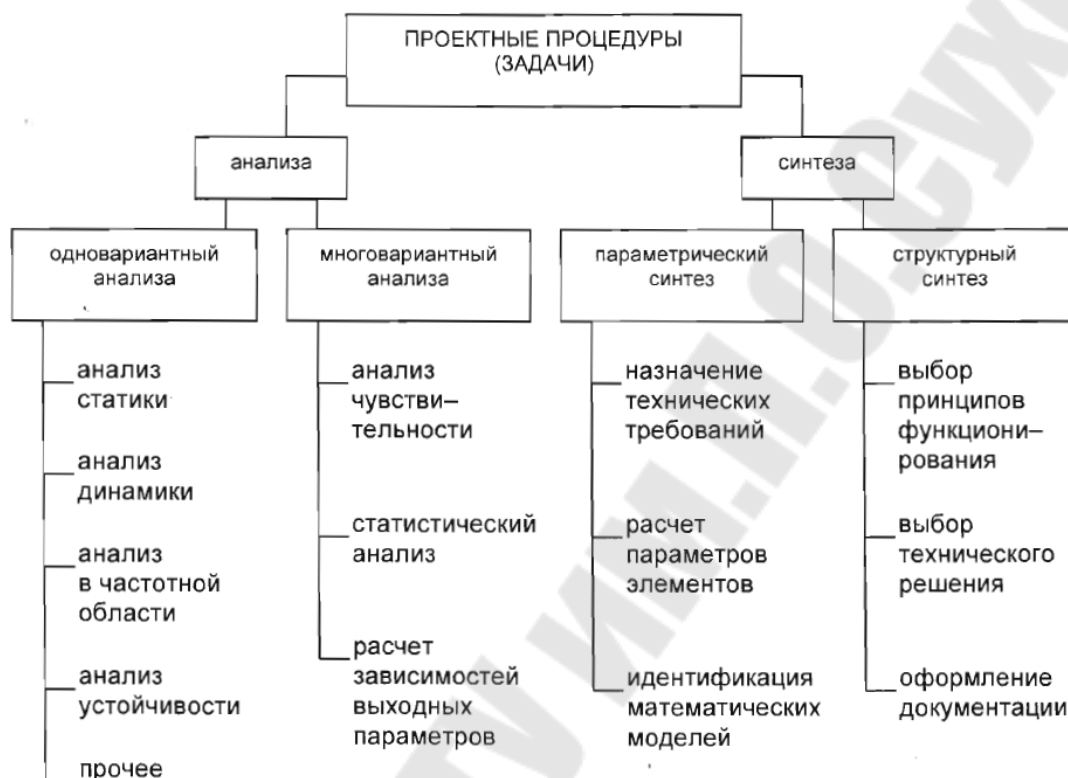


Рисунок 2.4 – Классификация основных проектных процедур

Различают проектные процедуры *анализа* и *синтеза*. Синтез заключается в создании описания объекта, а анализ – в определении свойств и исследовании работоспособности объекта по его описанию, т.е. при синтезе создаются, а при анализе оцениваются проекты объектов.

Накопленный в технологии машиностроения теоретический и практический материал касается в основном задач *анализа технологических процессов*, с помощью которого устанавливается влияние различных технологических факторов на производительность, геометрическую точность и качество поверхности при обработке деталей.

Процедуры анализа делятся на процедуры одно- и многовариантного анализа.

При *одновариантном анализе* заданы значения внутренних и внешних параметров, требуется определить значения выходных па-

раметров объекта. Полезно 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 этого 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).

Процедуры синтеза делятся на процедуры структурного и параметрического синтеза.

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 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 (обычные условия работоспособности).

2.2.4 Иерархические уровни проектирования

Разделение описаний проектируемых объектов по степени детализации отображаемых свойств и характеристик объекта лежит в основе *блочного-иерархического подхода к проектированию* и приводит к появлению иерархических уровней (уровней абстрагирования) в представлениях об объекте.

На верхнем уровне используется наименее детализированное представление, которое отображает только самые общие черты и особенности проектируемого объекта. На следующем уровне степень подробности описания возрастает. При этом рассматриваются отдельные блоки системы с учетом взаимодействий между собой этих блоков. Такой подход позволяет на каждом иерархическом уровне формулировать задачи приемлемой сложности, поддающиеся реше-

нию с помощью имеющихся средств проектирования. Описание на соответствующем уровне проектирования (документация на проектируемый объект) должно быть таким, чтобы оно было обозримо и воспринимательно человеком.

Следовательно, блочно-иерархический подход основан на разбиении сложной задачи большой размерности на последовательно или параллельно решаемые задачи малой размерности, что позволяет уменьшить время процесса проектирования и сократить требования к вычислительным ресурсам.

Для большинства задач проектирования характерны следующие иерархические уровни проектирования.

Системный уровень, на котором решаются наиболее общие задачи проектирования. В этом случае результаты проектирования представляются в виде структурных схем, укрупненных алгоритмов работы, диаграмм потоков данных и т.п.

Макроуровень, на котором проектируются отдельные устройства и узлы машины. Результаты проектирования представляются в виде функциональных, схем, сборочных чертежей и т.п. Макроуровень также называют схмотехническим уровнем.

Микроуровень, на котором проектируются отдельные детали и элементы машины.

Если решение задач высоких иерархических уровней предшествует решению задач более низких иерархических уровней, то проектирование называют *нисходящим*. Если раньше выполняются этапы, связанные с низшими иерархическими уровнями, проектирование называют *восходящим*.

У каждого из этих двух видов проектирования имеются преимущества и недостатки. На практике при проектировании сложных систем обычно используют нисходящее проектирование.

Типичная последовательность проектных процедур на одном из этапов нисходящего проектирования представлена на рисунке 2.5.

На предыдущем этапе решались задачи k -го иерархического уровня, одним из результатов решения этих задач при нисходящем проектировании является формулировка ТЗ на проектирование систем $(k+1)$ -го рассматриваемого уровня.

Проектирование системы начинается с синтеза исходного варианта ее структуры. Для оценки этого варианта создается модель: математическая – при автоматизированном проектировании, экспериментальная или стенд – при неавтоматизированном проектировании.

После выбора исходных значений параметров элементов выполняется анализ варианта, по результатам которого становится возможной его оценка. Обычно оценка заключается в проверке выполнения условий работоспособности, сформулированных в ТЗ. Если условия работоспособности выполняются в должной мере, то полученное проектное решение принимается, система ($k+1$)-го уровня описывается в принятой форме и формулируется ТЗ на проектирование элементов данного уровня (т.е. систем следующего уровня). Если же полученное проектное решение неудовлетворительно, выбирается один из возможных путей улучшения проекта.

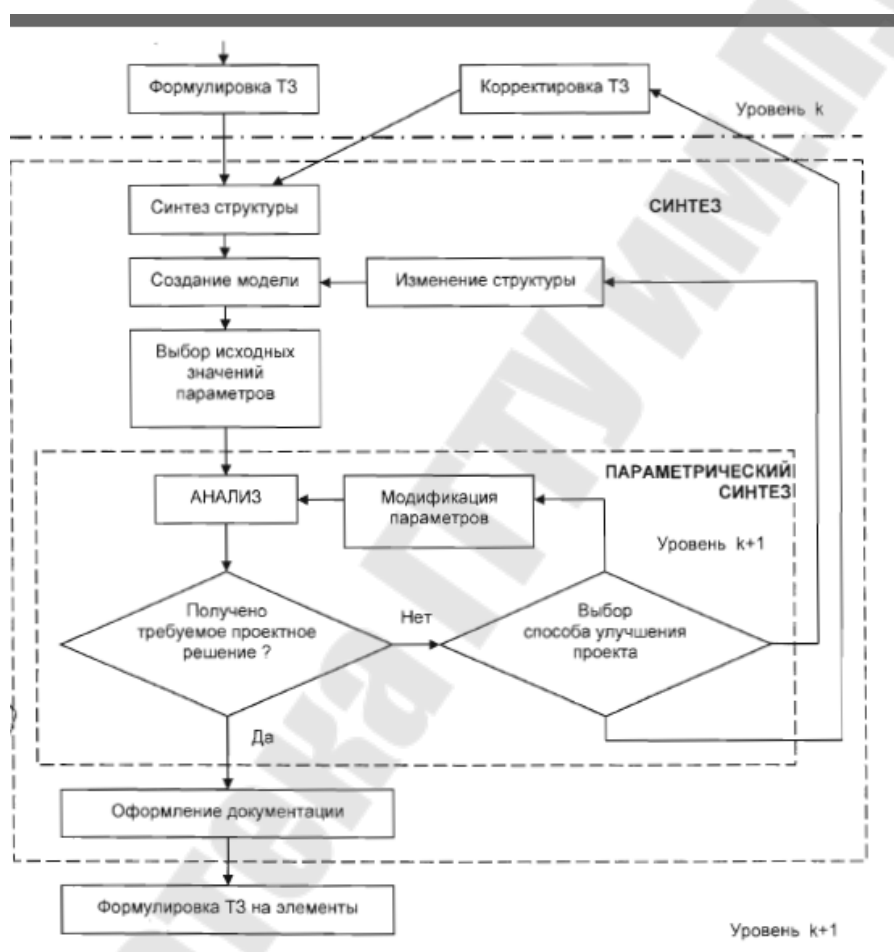


Рисунок 2.5 – Типичная последовательность проектных процедур

Рисунок 2.5 позволяет установить характерную особенность взаимосвязи проектных процедур анализа и синтеза. Эта взаимосвязь имеет характер вложенности процедуры анализа в процедуру оптимизации (параметрического синтеза) и процедуры оптимизации в проце-

дуру синтеза, объединяющую синтез структурный и параметрический.

Вложенность означает, во-первых, что анализ входит как составная часть в оптимизацию, а оптимизация – в синтез, во-вторых, что однократное выполнение процедуры оптимизации требует многократного выполнения процедуры анализа, а однократное решение задачи синтеза – многократного решения задачи оптимизации. Очевидно, что такой же характер взаимодействия имеют процедуры анализа – однократный многовариантный анализ основан на многократном одновариантном анализе. Нетрудно подсчитать, что синтез проектного решения на очередном этапе проектирования может потребовать выполнения чрезмерно большого количества вариантов анализа и разработка способов сокращения затрат вычислительных ресурсов на выполнение проектных процедур – актуальная проблема автоматизированного проектирования.

Маршрут проектирования объекта - последовательность этапов и (или) проектных процедур, используемая для проектирования этого объекта.

Маршрут называют типовым, если он применяется при проектировании многих объектов определенного класса.

2.2.5 Классификация САПР. Структура

В целях создания условий для типизации, унификации и стандартизации различных типов САПР, а также повышения эффективности их функционирования разработана классификация САПР. Важными признаками которой являются:

- тип объекта проектирования;
- комплексность автоматизации проектирования;
- количество уровней в структуре технического обеспечения

Тип объекта проектирования определяет следующие виды САПР:

- САПР изделий машиностроения и приборостроения (проектирует изделия машиностроения и приборостроения).
- САПР технологических процессов в машиностроении и приборостроении (проектирует технологические процессы в машиностроении и приборостроении).
- САПР объектов строительства.
- САПР организационных систем.

Комплексность автоматизации проектирования обуславливает следующие модификации САПР:

- одноэтапная САПР (выполняет один этап проектирования из всех, установленных для объекта, проектируемого системой);
- многоэтапная САПР (выполняет несколько этапов проектирования из всех установленных для объекта, проектируемого системой);
- комплексная САПР (выполняет все этапы проектирования, из всех установленных для объекта, проектируемого системой).

Под *этапом проектирования* понимается условно выделенная часть проектирования, в результате которой находят проектное решение, необходимое и достаточное для рассмотрения и принятия решения по продолжению проектирования.

Количество уровней в структуре технического обеспечения определяет следующие виды САПР:

- одноуровневая САПР – система, построенная на основе ЭВМ среднего или высокого класса со штатным набором периферийных устройств, который в необходимых случаях может быть дополнен некоторыми средствами обработки графической информации;
- двухуровневая САПР – система, построенная на основе ЭВМ среднего и высокого класса и одного или несколько автоматизированных рабочих мест (АРМ);
- трехуровневая САПР – система, построенная на основе ЭВМ высокого класса и одного или несколько автоматизированных рабочих мест (АРМ) и периферийного программно-управляемого оборудования (станки с ЧПУ, промышленные роботы и др.).

2.2.6 Виды обеспечения САПР

Основой САПР является совокупность различных видов обеспечения автоматизированного проектирования, необходимых для решения проектных задач.

Комплекс средств автоматизации проектирования современных САПР включает семь видов обеспечения:

- техническое;
- математическое;
- программное;

- информационное;
- лингвистическое;
- методическое;
- организационное.

Техническое обеспечение – совокупность взаимосвязанных и взаимодействующих технических средств, предназначенных для выполнения автоматизированного проектирования (hardware).

Используемые в САПР технические средства должны обеспечить.

1. Выполнение всех необходимых проектных процедур, что обеспечивается наличием вычислительных систем с достаточными производительностью и емкостью памяти.

2. Взаимодействие между проектировщиком и ЭВМ. Обеспечивается за счет использования средств ввода-вывода данных, в первую очередь устройств обмена графической информацией.

3. Взаимодействие между членами коллектива, выполняющего работу над общим проектом. Обеспечивается за счет вычислительной сети.

4. Достаточную производительность и объем оперативной памяти ЭВМ для решения задач всех этапов проектирования.

5. Приемлемую стоимость и др.

6. Высокую надежность.

К техническим средствам (ТС) относятся:

- устройства вычислительной и организационной техники;
- средства передачи данных;
- измерительные и другие устройства и их сочетания, обеспечивающие определенную техническую функцию соответствующих подсистем САПР.

Реализацию однородных функций различных подсистем САПР выполняют следующие группы ТС:

- подготовки и ввода данных;
- передачи данных;
- программной обработки данных;
- отображения и документирования данных;
- архива проектных решений.

Математическое обеспечение (МО) САПР состоит из математических моделей объектов проектирования, методов и алгоритмов выполнения проектных операций и процедур. Основу математического обеспечения САПР составляет математический аппарат для моде-

лирования, синтеза структуры, одновариантного и многовариантного анализа, структурной и параметрической оптимизации. Математическое обеспечение состоит из двух частей: специальное МО и инвариантное МО.

Специальное МО отражает специфику объекта проектирования, физические и информационные особенности его функционирования и тесно привязано к конкретным задачам проектирования. Эта часть математического обеспечения охватывает математические модели, методы и алгоритмы их получения, алгоритмы одновариантного анализа, а также большую часть используемых алгоритмов синтеза.

Инвариантное МО включает методы и алгоритмы, слабо связанные с особенностями математических моделей и используемые при решении различных задач проектирования. Это – методы и алгоритмы многовариантного анализа и параметрической оптимизации.

При создании математического обеспечения САПР должны учитываться следующие показатели: *универсальность, алгоритмическая надежность, точность, затраты машинного времени, объем используемой памяти.*

Универсальность МО определяет его применимость к широкому классу проектируемых объектов. Особенно это важно при создании комплексных САПР, включающих различные виды задач: от конструирования изделия и проектирования технологических процессов до выбора режущего инструмента и проектирования конструкций специальной оснастки на основе анализа типовых технологических решений. При этом может использоваться единая группа стандартных программ.

Универсальность МО упрощает методику автоматизированного проектирования. В то же время следует отметить, что универсальность не имеет количественной оценки. Реализуя ту или иную модель и метод, разработчик МО должен указать четкие границы их применимости.

Алгоритмическая надежность – свойство компонента МО давать при его применении и заранее определенных ограничениях правильные результаты. Количественной оценкой алгоритмической надежности служит вероятность получения правильных результатов при соблюдении оговоренных ограничений на применение метода. Если эта вероятность равна единице или близка к ней, то говорят, что метод алгоритмически надежен.

С алгоритмической надежностью тесно связана проблема *обусловленности* математических моделей и задач. О плохой обусловленности говорят в случаях, когда малые погрешности исходных данных приводят к большим погрешностям результатов. В результате не только снижается точность результатов проектирования, но и растут затраты машинного времени. Для анализа и оптимизации объектов с плохо обусловленными моделями необходимо применять специальные методы с повышенной алгоритмической надежностью.

Точность является наиболее важным свойством всех компонентов МО и определяет степень совпадения расчетных и истинных результатов. Алгоритмически надежные методы могут давать различную точность. И лишь в случаях, когда точность оказывается хуже предельно допустимых значений или решение вообще невозможно получить, говорят не о точности, а об алгоритмической надежности.

В большинстве случаев решение проектных задач характеризуется совместным использованием многих компонентов МО, что затрудняет оценку влияния погрешности отдельных компонентов. При необходимости оценки их точности проводят вычислительные эксперименты с использованием тестовых задач.

Затраты машинного времени во многом определяются сложностью проектируемых объектов и размерностью решаемых задач. Машинное время вычислительного процесса является главным ограничивающим фактором при попытках повысить сложность проектируемых на ЭВМ объектов.

Одним из путей сокращения сроков проектирования является применение в САПР многопроцессорных вычислительных систем, обеспечивающих распараллеливание процесса вычисления. В связи с этим важнейшим показателем экономичности МО является его приспособленность к распараллеливанию процесса проектирования.

Используемая память является вторым после затрат машинного времени показателем экономичности МО. Затраты памяти определяются длиной программы и объемом используемых массивов данных. Несмотря на значительное увеличение емкости оперативной памяти в современных ЭВМ, требования к снижению затрат памяти остаются актуальными. Это связано с тем, что в мультипрограммном режиме функционирования ЭВМ задача с запросом большего объема памяти получает более низкий приоритет, и в результате время ее пребывания в системе увеличивается и производительность процесса проектирования снижается.

В целях экономии затрат оперативной памяти используют внешнюю память. Однако частые обращения к внешней памяти приводят к увеличению затрат машинного времени, поэтому при разработке методов проектирования, алгоритмов и программ приходится решать вопрос рационального использования двух видов памяти ЭВМ – внутренней (оперативной) и внешней.

Математическое моделирование при автоматизированном проектировании технологических процессов. Выполнение проектных процедур при автоматизированном проектировании основано на оперировании с математическими моделями.

Математическая модель технологического процесса – это система математических объектов (чисел, переменных, множеств, графов, матриц и т.д.) и отношений между ними, отражающая некоторые свойства технологического процесса.

В САПР технологических процессов находят применение *структурно–логические* и *функциональные математические модели*.

Структурно–логические математические модели подразделяются на *табличные, сетевые и перестановочные*.

Табличная модель описывает одну конкретную структуру технологического процесса. В табличной модели каждому набору условий соответствует единственный вариант проектируемого технологического процесса. Поэтому табличные модели используют для поиска типовых проектных решений.

Пример. При обработке группы деталей d_1, d_2, d_3 на прутковом токарном автомате последовательность обработки их поверхностей устанавливается с помощью табличных моделей. Каждая деталь (рисунок 2.6) имеет поверхности с определенными свойствами K_1, K_2, \dots, K_8 :

$$F(d_1) = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8\} = F_1';$$

$$F(d_2) = \{F_1, F_2, F_3, F_4, F_5, F_8\} = F_2';$$

$$F(d_3) = \{F_1, F_2, F_5, F_8\} = F_3'.$$

На рисунке 2.7 представлены табличные модели в виде графов взаимосвязей переходов при обработке деталей d_1, d_2, d_3 на данной операции.

На рисунке 2.7 приняты следующие обозначения: $\tau_1, \tau_2, \dots, \tau_8$ – операторы (технологические переходы): τ_1 – подрезка торца; τ_2, τ_3, τ_4 – точение наружной цилиндрической поверхности; τ_5 – сверление; τ_6 – зенкерование; τ_7 – зенкование; τ_8 – отрезка.

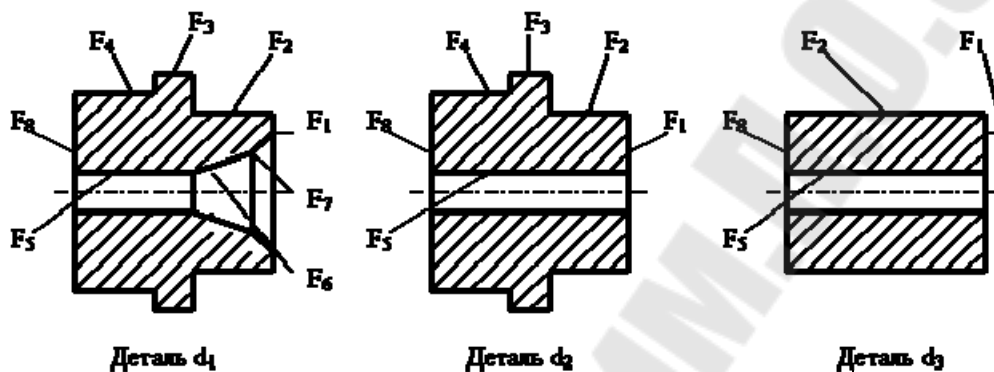


Рисунок 2.6 – Эскизы деталей для обработки на прутковом токарном автомате

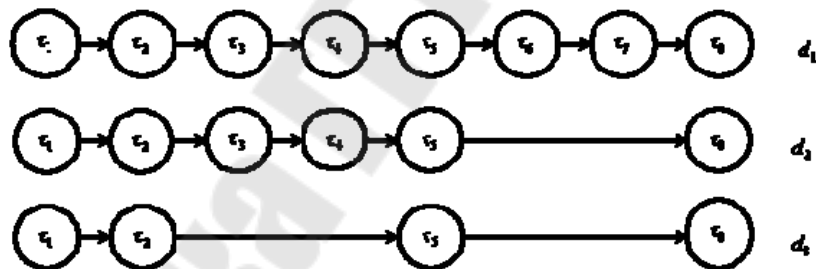


Рисунок 2.7 – Графы взаимосвязей переходов при обработке деталей d_1, d_2, d_3

Для представления данных об обработке деталей на данной операции на языке, понятном компьютеру, удобном для программирования, представленная выше информация может быть описана в виде двух таблиц (2.1 и 2.2), которые легко превращаются в массивы.

Таблица 2.1 – Связи между свойствами поверхностей деталей и операторами (технологическими переходами)

ε_i	F_j							
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
ε_1	1	0	0	0	0	0	0	0
ε_2	0	1	0	0	0	0	0	0
ε_3	0	0	1	0	0	0	0	0
ε_4	0	0	0	1	0	0	0	0
ε_5	0	0	0	0	1	0	0	0
ε_6	0	0	0	0	0	1	0	0
ε_7	0	0	0	0	0	0	1	0
ε_8	0	0	0	0	0	0	0	1

В этой, а также последующих таблицах логическая единица обозначает наличие связи, а ноль – отсутствие таковой.

Таблица 2.2 – Связи между совокупностями свойств деталей и операторами (технологическими переходами)

ε_i	F_j'		
	F_1'	F_2'	F_3'
ε_1	1	:	1
ε_2	1	:	1
ε_3	1	:	0
ε_4	1	:	0
ε_5	1	:	1
ε_6	1	0	0
ε_7	1	0	0
ε_8	1	:	1

Сетевая модель описывает множество структур технологического процесса, отличающихся количеством и (или) составом элементов структуры при неизменном отношении порядка.

Структура элементов сетевой модели описывается ориентированным графом, не имеющим ориентированных циклов. В модели может содержаться несколько вариантов проектируемого технологического процесса, однако во всех вариантах порядок элементов одинаков.

Пример. Сетевая модель технологического проектирования маршрута обработки детали «Зубчатое колесо», эскиз которой представлен на рисунке 2.8.

На рисунке 2.9 показан граф взаимосвязи операторов (технологических операций) по возможной последовательности их выполнения.

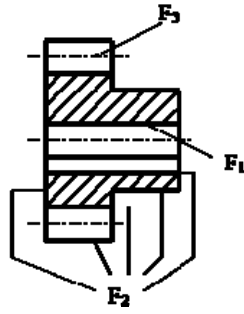
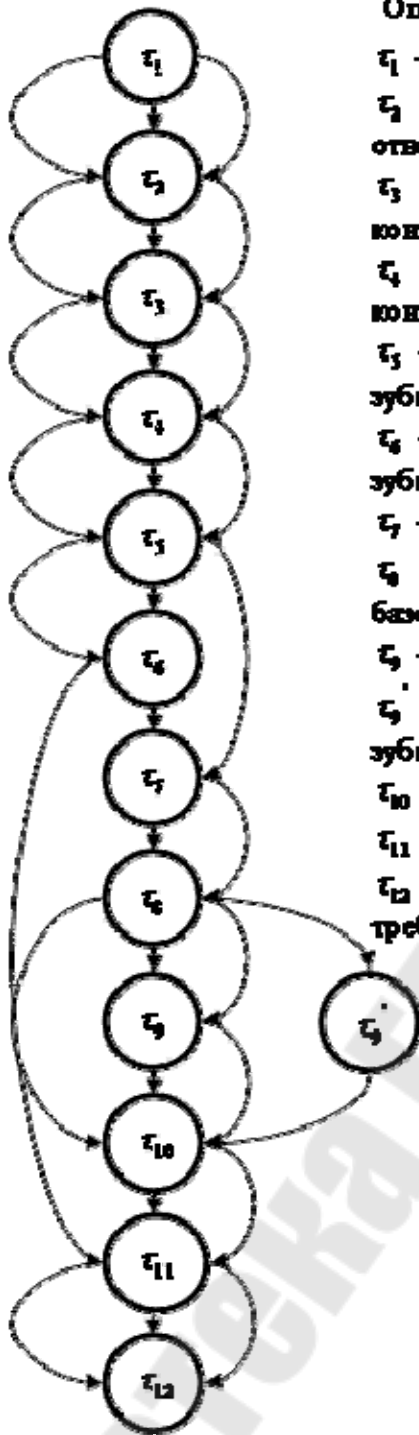


Рисунок 2.8 – Эскиз детали «Зубчатое колесо»

Кроме данного графа сетевая модель включает в себя таблицу связей свойств поверхностей детали и операторов технологического процесса (в этом примере – технологических операций) – таблица 2.3.

Таблица 2.3 – Связи между свойствами поверхностей детали и операторами технологического процесса

τ_i	R_j		
	R_1	R_2	R_3
τ_1	1	1	0
τ_2	1	0	0
τ_3	0	1	0
τ_4	0	1	0
τ_5	0	0	1
τ_6	0	0	1
τ_7	1	1	1
τ_8	1	0	0
τ_9	0	0	1
τ_{10}	0	0	1
τ_{11}	1	1	1
τ_{12}	1	1	1



- Операторы τ_i (технологические операции):
- τ_1 - заготовительная;
 - τ_2 - протяжная (протягивание базового отверстия);
 - τ_3 - черновая токарная (черновое точение контура зубчатого колеса);
 - τ_4 - чистовая токарная (чистовое точение контура зубчатого колеса);
 - τ_5 - зубофрезерная черновая (черновая нарезка зубьев);
 - τ_6 - зубофрезерная чистовая (чистовая нарезка зубьев);
 - τ_7 - термическая (объемная закалка);
 - τ_8 - внутренняя шлифовальная (шлифование базового отверстия);
 - τ_9 - зубошлифовальная (шлифование зубьев);
 - τ_{10} - зубошвинговальная (швингование зубьев);
 - τ_{11} - притирочная (притирка зубьев);
 - τ_{12} - моечная (мойка детали);
 - τ_{13} - контрольная (контроль технических требований детали)

Рисунок 2.9 – Граф взаимосвязи операторов (технологических операций) по возможной последовательности их выполнения

Перестановочная модель описывает множество структур технологического процесса, отличающихся количеством и (или) составом элементов структуры при изменении отношения порядка.

Отношения порядка в этих моделях задаются с помощью графа, содержащего ориентированные циклы.

Пример. Расцеховка при изготовлении изделия (рисунок 2.10).

На рисунке 2.10 через P_1, P_2, \dots, P_8 обозначены цеха: P_1 – литейный; P_2 – кузнечный; P_3 – механический; P_4 – термический; P_5 – механосборочный; P_6 – общей сборки; P_7 – испытательный; P_8 – упаковочный.

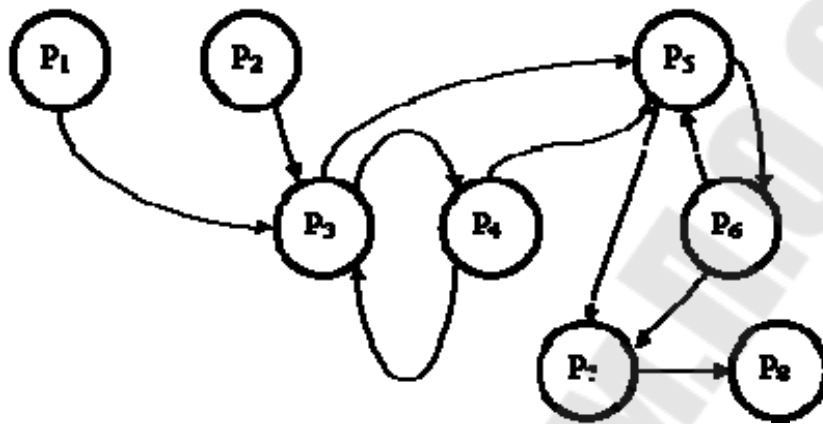


Рисунок 2.10 – Граф, отображающий расцеховку при изготовлении изделия

Сетевые и перестановочные модели используют для получения типовых, групповых и индивидуальных технологических процессов. Наличие в них вариантов позволяет производить оптимизацию технологических процессов.

Характерным примерами функциональных моделей являются математические модели, используемые при расчете и оптимизации режимов резания.

Лингвистическое обеспечение представлено совокупностью языков, применяемых для описания процедур автоматизированного проектирования и проектных решений.

В соответствии с классификацией языков САПР различают языки программирования и проектирования.

Языки программирования – языки, предназначенные для написания программного обеспечения. Эти языки – средство разработчика САПР. Программа, написанная на языке высокого уровня, должна быть преобразована в машинное представление. Это преобразование выполняется языковыми процессорами. Языковые процессоры делятся на компиляторы и интерпретаторы. При компиляции исходная программа переводится в машинные коды и создается исполнимый файл, который затем загружается и исполняется на компьютере. При интер-

претации анализируется очередной оператор исходной программы, а затем он выполняется.

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

Среди языков проектирования выделяют языки *входные, выходные, сопровождения, управления, промежуточные и внутренние*.

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*. Эти языки используются для описания исходной информации представленной в виде соответственно некоторой схемы конструкторского чертежа, алгоритма функционирования. *Схемные языки* широко применяют при описании принципиальных электрических схем, в подсистемах проектирования электронных устройств, функциональных схем, подсистемах функционально логического проектирования ЭВМ.

Графические языки – основа лингвистического обеспечения в подсистемах геометрического моделирования и машинной графики.

Выходные языки используются для вывода результатов выполнения проектных процедур на ЭВМ

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

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

Пример современного языка проектирования – VHDL (very high speed IC hardware description language).

Информационное обеспечение – совокупность сведений, необходимых для выполнения автоматизированного проектирования и представленных в заданной форме.

Информацию, используемую при автоматизации режимов резания, условно можно разделить на исходную и производную. *Исходной* называется информация, существующая до начала машинного проек-

тирования. Она подразделяется на переменную и условно-постоянную.

В качестве *переменной* информации можно отнести, например требования технологического процесса, принятую схему обработки (наладки), паспорт оборудования и т.д. Эта информация вводится в оперативное запоминающее устройство каждый раз при новых расчетах, например, режимов резания.

Условно-постоянная информация, например, для расчетов режимов резания состоит из карт и нормативных таблиц. Эта информация является достаточно стабильной и постоянно хранится во внешней памяти ЭВМ.

Производная информация формируется в процессе расчетов режимов резания и содержит, например, расчет машинного времени при протягивании отверстий или наружных поверхностей.

Условно-постоянная информация может быть представлена в виде базы данных.

В широком смысле слова *база данных* – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Под *предметной областью* принято понимать часть реального мира, подлежащую изучению для организации управления и в конечном счете автоматизации.

Программное обеспечение – совокупность машинных программ, необходимых для выполнения автоматизированного проектирования и представленных в заданной форме. Программное обеспечение (ПО) делится на общесистемное, базовое и прикладное (специальное).

Общесистемное ПО предназначено для организации функционирования технических средств, т.е. для планирования и управления вычислительным процессом, распределением имеющихся ресурсов. Оно представлено операционными системами ЭВМ и ВС. Общесистемное ПО обычно создается для многих приложений и *специфику САПР не отражает*.

Базовое и *прикладное ПО* создаются для нужд САПР. В *прикладном ПО* реализуются математическое обеспечение для непосредственного выполнения проектных процедур. Прикладное ПО обычно имеет форму пакетов прикладных программ (ППП), каждый из которых обслуживает определенный этап процесса проектирования или группу однотипных задач внутри различных задач. В *базовое ПО*

входят программы, обеспечивающие правильное функционирование прикладных программ.

В САПР, как правило, применяется много различных пакетов прикладных программ (ППП), каждый из которых имеет ориентацию на определенную подсистему САПР. Так, известны пакеты программ геометрического моделирования; оформления конструкторской документации, используемые в подсистеме машинной графики; синтеза маршрутной технологии; проектирования штампов; выбора установочных баз в подсистемах технологического проектирования; расчетов на прочность в подсистемах проектирования корпусных деталей летательных аппаратов и т. п.

Пакеты прикладных программ в свою очередь могут иметь достаточно сложную структуру с выделением управляющей и обрабатывающей частей.

Процесс автоматизированного проектирования заключается в выполнении маршрутов проектирования с помощью большого количества взаимодействующих программных модулей.

Проблемы создания гибкого и универсального программного обеспечения для систем автоматизированного проектирования сложных объектов в значительной мере являются проблемами информационного согласования различных частей программного обеспечения при реализации множества возможных маршрутов проектирования.

Существуют три основных способа реализации связей по информации:

- через передачу параметров из вызывающей программы в вызываемую подпрограмму;
- через общие области (обменные зоны) взаимодействующих модулей;
- через банк данных.

Реализация информационных связей через передачу параметров означает, что передаются либо значения параметров, либо их адреса. Такая передача удобна только при сравнительно небольшом объеме передаваемых данных и их простой структуре.

Если же одни и те же модули могут входить в различные маршруты, взаимодействовать со многими другими модулями, то целесообразно унифицировать средства информационного обмена. Такая унификация осуществляется с помощью концепции банков и баз данных,

Основные преимущества информационного взаимодействия через банк данных заключаются в следующем:

1. Снимаются ограничения на число обслуживаемых маршрутов проектирования и на число вхождений модулей в различные маршруты.

2. Возможно развитие и модификации программной системы путем добавления новых модулей и маршрутов без изменений ранее созданных модулей.

3. Возможна модернизация технических средств для хранения БД без изменений в ППП.

4. Обеспечивается целостность данных.

Целостность данных подразумевает их непротиворечивость и достоверность.

Однако реализация информационных связей через банк данных имеет и свои недостатки, связанные со значительными затратами времени на поиск необходимых данных в БД.

Методическое обеспечение – совокупность документов, устанавливающих состав, а также правила отбора и эксплуатации средств обеспечения автоматизированного проектирования, необходимых для решения проектных задач.

Организационное обеспечение – совокупность документов, устанавливающих состав проектной организации и ее подразделений, связи между ними, их функции, а также форму представления результата проектирования и порядок рассмотрения проектных документов.

2.2.7 Основные принципы создания САПР

При создании САПР на различных стадиях, а также ее подсистем и компонентов необходимо учитывать следующие положения и принципы.

САПР – это *человеко-машинная система*. Коллектив разработчиков и пользователей системы является ее основной частью и, взаимодействуя с техническими средствами САПР, выполняет проектирование. При этом часть проектных процедур не может быть автоматизирована и решается при участии человека. Об автоматическом проектировании можно говорить лишь в отношении отдельных задач.

САПР – *развивающаяся система*. Суть положения состоит в том, что САПР должна создаваться и функционировать с учетом пополнения, совершенствования и обновления ее подсистем и компо-

нентов. С этой целью на предприятии, где используется САПР, должна быть создана группа специалистов, способная совершенствовать и развивать имеющуюся САПР.

Принцип системного единства САПР состоит в том, что при создании, функционировании и развитии САПР связи между подсистемами должны обеспечивать целостность всей системы. Наибольший эффект от САПР достигается при комплексной (сквозной) автоматизации проектирования на всех уровнях. Последнее позволяет исключить многократное описание информации об объектах проектирования, обеспечив ее преемственность для различных подсистем.

Принцип совместимости компонентов САПР состоит в том, что языки, символы, коды, информационные и технические характеристики структурных связей между подсистемами, средствами обеспечения САПР должны обеспечивать совместное функционирование подсистем. Особенно важным является информационная и программная согласованность отдельных подсистем. Так, например, информационная совместимость обеспечивает работу различных подсистем с одной и той же базой данных и единым входным языком.

Принцип стандартизации САПР заключается в проведении унификации, типизации и стандартизации подсистем и компонентов, инвариантных к проектируемым объектам, а также в установлении правил с целью упорядочения деятельности в области создания и развития САПР. Унификация программного, лингвистического, технического и других видов обеспечения открывает широкую возможность внедрения САПР и ее адаптации на различных предприятиях.

Принцип независимости отдельных подсистем (программ) САПР определяет возможность для подсистем (программ) введения в действие и функционирования их независимо от других подсистем. Этот принцип называется также модульным принципом построения САПР.

Принцип открытости САПР определяет возможность внесения изменений в систему во время ее разработки и эксплуатации. Изменения могут заключаться в добавлении новых или замене старых элементов программного, информационного, технического и лингвистического обеспечения.

Принцип согласованности автоматизированного (традиционного) проектирования и САПР должен учитываться при внедрении САПР на уже действующем предприятии со сложившейся структурой, взаимоотношениями, формами и способами использования про-

ектной документации. При этом внедрение САПР не должно нарушать на длительный срок нормального функционирования предприятия.

2.2.8 Системное проектирование технологических процессов

Системное проектирование технологических процессов особенно с использованием ЭВМ включает в себя использование двух основных принципов:

Принцип 1. Применение при проектировании технологических процессов *системного подхода*, который основывается на следующем:

а) технологический процесс нужно рассматривать, с одной стороны, как просто перечень отдельных его элементов (операций, переходов и т.д.), а с другой стороны, как совокупность взаимосвязанных и взаимообусловленных элементов. Т.е. необходимо говорить о структуре технологического процесса;

Структура технологического процесса – это множество его элементов и множество связей между ними.

Если $V = \{v_1, v_2, \dots, v_n\}$ – множество элементов технологического процесса, $S = \{s_1, s_2, \dots, s_n\}$ – множество связей между элементами, то $Str = \{V, S\}$ – структура технологического процесса (рисунки 2.11 и 2.12).

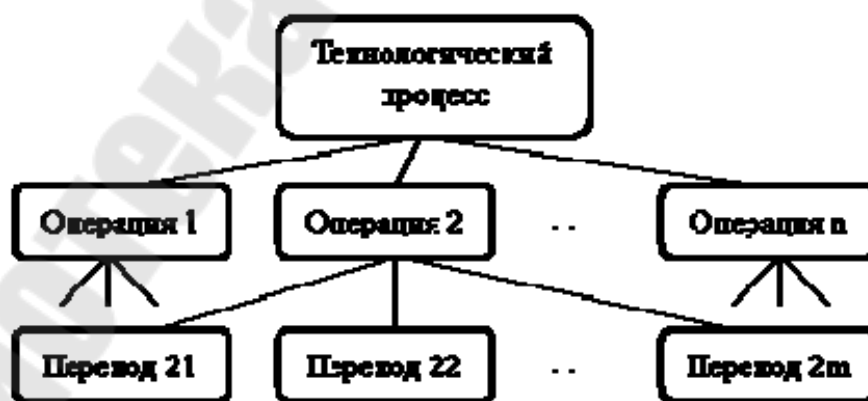


Рисунок 2.11 – Представление структуры технологического процесса в виде дерева



Рисунок 2.12 – Представление структуры технологического процесса в виде графа

б) процесс проектирования технологического процесса – это, с одной стороны, просто перечень отдельных его этапов (выбор заготовки, определение маршрута обработки детали и т.д.), а с другой стороны, совокупность взаимосвязанных и взаимообусловленных этапов;

в) рациональное разбиение процесса проектирования на части. Проектирование технологического процесса – сложная задача. Общепринятый подход к решению сложных задач – разбиение их на простые задачи и их решение во взаимосвязи друг с другом. «Простые» задачи при проектировании технологического процесса: выбор типа заготовки, расчет режимов резания и т.д.;

г) принятие оптимальных решений.

Принцип 2. Использование при проектировании технологических процессов рационального сочетания традиционных (иногда «ручных») методов проектирования и достижений теории множеств, теории графов, теории оптимизации и других современных системных наук, ориентированных на использование ЭВМ.

Применение принципов системного проектирования позволяет систематизировать знания в любой области, «навести в ней порядок». Рисунок 2.13 (а, б) показывает, чем отличается представление знаний без использования принципов системного проектирования и с использованием этих принципов.

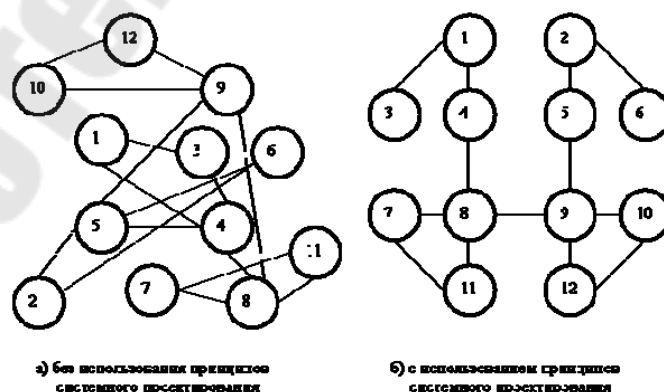


Рисунок 2.13 – Представление знаний в определенной области

2.2.9 Стратегии проектирования технологических процессов

При «ручном» проектировании технологических процессов, а особенно при создании (использовании) САПР технологических процессов важно иметь четкое представление, с использованием какой (каких) стратегий они проектируются. Стратегия проектирования технологического процесса определяет методику его проектирования. Правильный выбор стратегии проектирования чрезвычайно важен (особенно в САПР). Это определяет эффективность САПР. Ниже приведены некоторые стратегии проектирования технологических процессов (рисунки 2.14 –2.17).

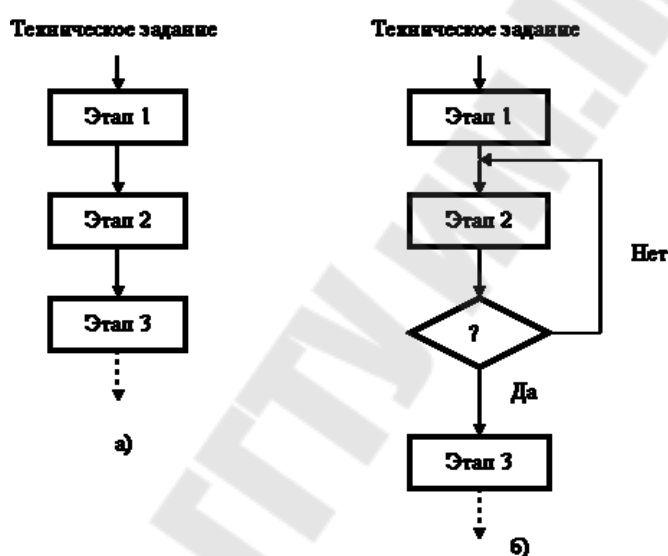


Рисунок 2.14 – Линейная (а) и циклическая (б) стратегия проектирования

В идеале необходимо стремиться к выбору или разработке линейной стратегии проектирования. Она является идеальной особенно при проектировании с использованием ЭВМ. Эта стратегия имеет минимальную трудоемкость, максимальную надежность.

Циклическая стратегия (схема с петлями) характерна для многих программ ЭВМ и носит название итерационного процесса. Другими словами это процесс последовательного приближения к цели путем улучшения разрабатываемых вариантов.

Наличие параллельных этапов в разветвленной стратегии очень выгодно. Это позволяет сократить сроки проектирования.

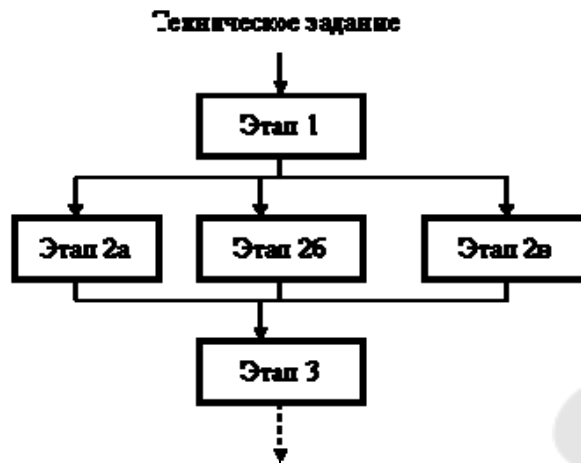


Рисунок 2.15 – Разветвленная стратегия проектирования

В адаптивных стратегиях проектирования с самого начала определяется только первое действие. В дальнейшем выбор каждого последующего действия зависит от результатов предыдущего. В принципе, это самая разумная стратегия, т.к. схема поиска определяется на основе наиболее полной информации. Эта стратегия используется при создании систем искусственного интеллекта.

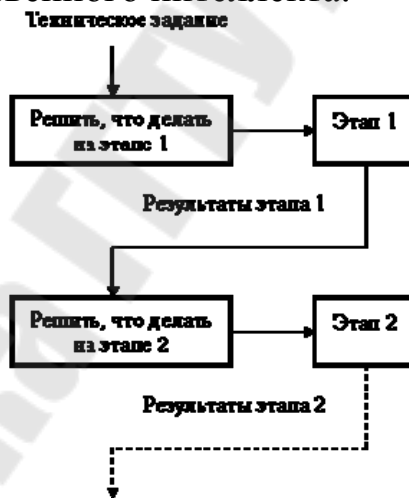


Рисунок 2.16 – Адаптивная стратегия проектирования

Стратегия случайного поиска отличается абсолютным отсутствием плана. Она используется в новаторском проектировании, например, при разработке новых технологических процессов.



Рисунок 2.17 – Стратегия случайного поиска

Необходимо добиваться максимальной линейзации процесса проектирования с включением параллельных этапов, а цикличность стараться исключать, особенно на верхних уровнях проектирования. К сожалению, из-за недостаточной информации часто не удастся задать линейную стратегию, которая особенно целесообразна в САПР.

Стратегия проектирования может детализироваться от одного уровня проектирования к другому. На определенных этапах проектирования приходится вводить методы управления стратегией (рисунок 2.18).



Рисунок 2.18 – Управление стратегией проектирования

Целесообразно процесс проектирования разбивать на частные задачи. Результат выполнения каждой задачи оформляется в виде технического задания, которое дает информацию о последующем плане (стратегии) ее детализации (дальнейшего решения).

2.3 Состав и назначение интегрированных САПР

2.3.1 Интеграция систем проектирования и изготовления

Одним из направлений развития современного машиностроения является создание гибких (быстроперенастраиваемых) автоматизированных производств. В основу этих промышленных систем положены принципы централизованной переработки конструкторской, технологической информации и обеспечения управления станками с ЧПУ, промышленными роботами, системами транспортирования заготовок и инструмента с помощью ЭВМ. Централизация переработки всех видов информации приводит к необходимости интеграции (объединения) систем проектирования, использующих и порождающих эту информацию, т. е. к созданию интегрированных САПР.

Под интеграцией САПР понимают объединение некоторых их основных частей и элементов. Основой эффективной интеграции систем является объединение иерархически сгруппированных, в значительной степени самостоятельных, целесообразно связанных и взаимодополняющих друг друга систем проектирования.

На рисунке 2.19 показана схема интегрированной системы проектирования, изготовления деталей, сборки машин и управления производством от ЭВМ. Использование таких систем открывает возможность создания «безлюдной» технологии, главной особенностью которой является переработка и передача информации с помощью вычислительных систем (одной или нескольких ЭВМ) от проектировщика-конструктора или технолога непосредственно исполнительному элементу производственной системы – станку или промышленному роботу без бумажной документации и участия рабочего-станочника.

Интеграция систем проектирования включает:

- интеграцию информации (единая классификация, единая система документации);
- организационную интеграцию (единая система сбора, поиска и передачи информации);
- технико-математическую интеграцию (унифицированные технико-математические методы анализа решаемых задач);
- программную интеграцию (унификация программного обеспечения);

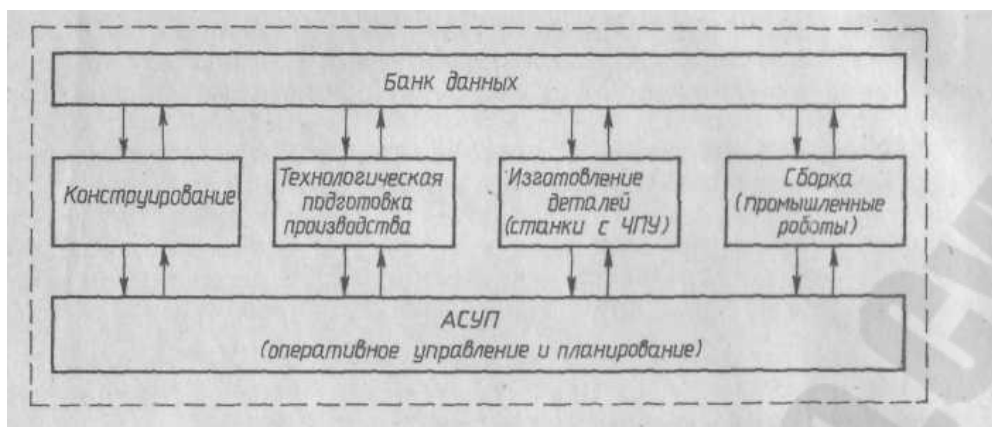


Рисунок 2.19 – Схема интегрированной системы проектирования и изготовления изделий

- техническую интеграцию (унификация используемой вычислительной техники, копировально-множительного оборудования и средств связи).

2.3.2 Требования к интегрированным САПР

Разрабатываемые интегрированные САПР должны отвечать следующим требованиям:

- 1) охватывать все этапы проектирования – от ввода описания проектируемого объекта до получения проектной документации в виде технологических карт, эскизов и управляющих программ для станков с ЧПУ;
- 2) отличаться модульным принципом построения, позволяющим изменять и наращивать системы;
- 3) обладать развитой иерархией языков проектирования, развитой операционной системой управления, а также развитыми банками данных;
- 4) иметь управляющий программный комплекс по выбору и генерированию прикладных программ, оптимальных применительно к конкретному проектируемому объекту;
- 5) характеризоваться возможностью моделирования и контроля на различных этапах процесса проектирования;
- 6) отличаться хорошей приспособляемостью к тиражированию и использованию на различных типах ЭВМ.

2.3.3 Состав интегрированных САПР

Интегрированные САПР, построенные на принципах стандартизации и модульного построения, позволяют обеспечить гибкость, компактность и экономичность системы. Основными компонентами интегрированной САПР являются: система автоматизации проектно-конструкторских работ, система автоматизации технологической подготовки производства, банк данных.

Управление процессом проектирования в рассматриваемых системах осуществляется программной частью операционной системы – монитором. Основные функции монитора следующие:

- 1) формирование задания на выбор пакета программ и установление приоритета для решения отдельных задач;
- 2) выполнение директив диалогового режима;
- 3) управление процессом проектирования в автоматическом режиме (анализ качества каждого этапа проектирования и выработка условий для перехода к следующим этапам).

На рисунке 2.20 показана схема подготовки основных производственных документов в условиях интегрированной САПР.

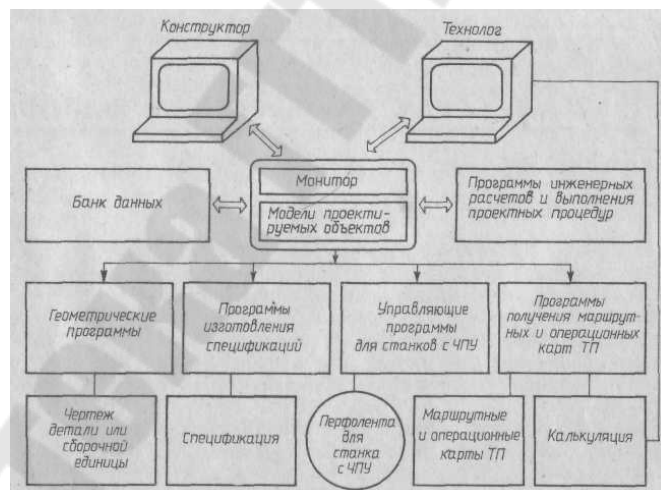


Рисунок 2.20 – Схема подготовки производственных документов в условиях интегрированных САПР

2.3.4 CAD /CAM /CAE системы

По мнению ведущих мировых аналитиков, основными факторами успеха в современном промышленном производстве являются: сокращение срока выхода продукции на рынок, снижение ее себестоимости и повышение качества. К числу наиболее эффективных технологий, позволяющих выполнить эти требования, принадлежат так называемые CAD/CAM/CAE-системы.

CAD-системы (computer-aided design) – компьютерная поддержка проектирования) предназначены для решения конструкторских задач и оформления конструкторской документации (у нас они именуются системами автоматизированного проектирования – САПР). Как правило, в современные CAD-системы входят модули моделирования трехмерной объемной конструкции (детали) и оформления чертежей и текстовой конструкторской документации (спецификаций, ведомостей и т. д.). Ведущие трехмерные CAD-системы позволяют реализовать идею сквозного цикла подготовки и производства сложных промышленных изделий.

CAM-системы (computer-aided manufacturing) – компьютерная поддержка производства) предназначены для проектирования обработки изделий на станках с числовым программным управлением (ЧПУ) и выдачи программ для этих станков (фрезерных, сверлильных, эрозионных, пробивных, токарных, шлифовальных и др.). CAM-системы еще называют системами технологической подготовки производства. В настоящее время они являются практически единственным способом для изготовления сложно-профильных деталей и сокращения цикла их производства. В CAM-системах используется трехмерная модель детали, созданная в CAD-системе.

CAE-системы – (computer-aided engineering) – поддержка инженерных расчетов) представляют собой обширный класс систем, каждая из которых позволяет решать определенную расчетную задачу (группу задач), например расчеты на прочность, анализ и моделирования тепловых процессов, анализ электромагнитной совместимости. В CAE-системах также используется трехмерная модель изделия, созданная в CAD-системе. CAE-системы еще называют системами инженерного анализа.

В настоящее время общепризнанным фактом является невозможность изготовления сложной наукоемкой продукции (кораблей, самолетов, различных видов промышленного оборудования и др.) без применения CAD/CAM/CAE-систем. За последние годы

CAD/CAM/CAE-системы прошли путь от сравнительно простых чертежных приложений до интегрированных программных комплексов, обеспечивающих единую поддержку всего цикла разработки, начиная от эскизного проектирования и заканчивая технологической подготовкой производства, испытаниями и сопровождением. В настоящее время наметилась тенденция к интеграции CAD/CAM/CAE систем.

2.3.5 Функции, характеристики и примеры CAE/CAD/CAM-систем

САПР Легкий вес. Эти САПР служат для выполнения почти всех работ с двумерными чертежами и имеют ограниченный набор функций по трехмерному моделированию. С помощью этих систем выполняются порядка 80 % всех работ по проектированию (эта цифра сильно колеблется в зависимости от отрасли), хотя имеющиеся ограничения делают их не всегда довольно удобными.

Область их работы – создание чертежей отдельных деталей. Характерные представители таких САПР – AutoCAD, T-FlexCAD 2D. Можно выделить среди Легкого класса САПР, подгруппу CAD систем на платформе DWG. К этой подгруппе можно отнести множество программ, которые являются аналогами AutoCAD и используют тот же формат хранения данных dwg, dxf.

САПР Средний вес. По своим возможностям они полностью охватывают САПР «легкого веса» плюс позволяют работать со сборками, и по некоторым параметрам они уже не уступают тяжелым САПР. А в удобстве работы даже превосходят. Обязательным условием является наличие интеграции с САМ-программами. Это не просто программы, а программные комплексы, в частности Autodesk Mechanical Desktop, Intergraph, Solid Edge, T-Flex, Solid Works.

САПР Тяжелый вес. Эти системы применяются для решения наиболее трудоемких задач – моделирования поведения сложных механических систем в реальном масштабе времени, оптимизирующих расчетов с визуализацией результатов, расчетов температурных полей и теплообмена и т.д. Обычно в состав системы входят графические модули, модули для проведения расчетов, постпроцессоры для станков с ЧПУ. Примерами «тяжелых» САПР могут служить такие продукты, как CATIA, CADD5, EMS, Pro/ENGINEER.

3 ОРГАНИЗАЦИЯ ИНФОРМАЦИОННОГО ФОНДА САПР С ИСПОЛЬЗОВАНИЕМ БАЗ ДАННЫХ

3.1 Банки данных в САПР

3.1.1 Основные понятия

В основе решения многих задач лежит обработка информации, для которой создаются *информационные системы* (ИС).

Банк данных – разновидность ИС, в которой реализованы функции централизованного хранения и накопления обрабатываемой информации, организованной в одну или несколько *баз данных*.

В широком смысле слова *база данных* – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Под *предметной областью* принято понимать часть реального мира, подлежащую изучению для организации управления и в конечном счете автоматизации. Предметная область – это, например, «Круглошлифовальные станки. Шлифование шеек».

Создавая базы данных, пользователь стремится упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков. Сделать это возможно, только если данные структурированы, т.е. введены соглашения о способах представления данных.

База данных (БД) – это поименованная совокупность структурированных данных, относящихся к определенной предметной области.

Пользователями базы данных могут быть различные прикладные программы, программные комплексы, а также специалисты предметной области, выступающие в роли потребителей или источников данных, называемые конечными пользователями.

Создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляется централизованно с помощью специального программного инструмента – *системы управления базами данных* (СУБД).

База данных и СУБД образует *банк данных*, который часто называют *автоматизированным банком данных* (АБД).

Понятие базы данных тесно связано с такими понятиями структурных элементов, как поле, запись, файл (таблица).

Поле – элементарная единица логической организации данных, которая соответствует неделимой единице информации – реквизиту. Для описания поля используются следующие характеристики:

- *имя*, например, «характер обработки», «класс чистоты поверхности», «точность обработки»;
- *тип*, например, символьный, числовой, календарный;
- *длина*, например, 15 байт, причем будет определяться максимально возможным количеством символов;
- *точность* для числовых данных, например, два десятичных знака для отображения дробной части числа.

Запись – совокупность логически связанных полей. *Экземпляр записи* – отдельная реализация записи, содержащая конкретные значения её полей.

Файл (таблица) – совокупность экземпляров записей одной структуры.

В структуре записи файла указываются поля, значения которых являются *ключами*: *первичными (ПК)*, которые идентифицируют экземпляр записи, и *вторичными (ВК)*, которые выполняют роль поисковых или группировочных признаков (по значению вторичного ключа можно найти несколько записей).

3.1.2 Виды архитектуры информационных систем на основе баз данных

Эффективность ИС во многом зависит от ее архитектуры. В настоящее время перспективной является архитектура *клиент-сервер* – предполагает наличие компьютерной сети и *распределенной базы данных*, включающей *корпоративную базу данных (КБД)* и *персональные базы данных (ПБД)*.

Размещаются: КБД – на компьютере-сервере, ПБД – на компьютерах сотрудников подразделений, являющихся клиентами КБД.

Сервер определенного ресурса в компьютерной сети – компьютер (или программа), управляющий этим ресурсом, *клиент* – компьютер (программа), использующий этот ресурс. В качестве ресурса компьютерной сети могут выступать, например, БД, файловые системы, службы печати, почтовые службы. Тип сервера определяется видом ресурса, которым он управляет. Например, если управляемым ресурсом является БД, то соответствующий сервер называется *сервером БД*.

Основное преимущество организации ИС по архитектуре клиент-сервер – удачное сочетание:

- *централизованного хранения* и обслуживания данных;
- *коллективного доступа* к общей корпоративной информации;
- *индивидуальной работы* пользователей над персональной информацией.

Архитектура клиент-сервер допускает различные варианты реализации. Исторически первыми появились *распределенные ИС* с применением *файл-сервера* (рисунок 3.1).

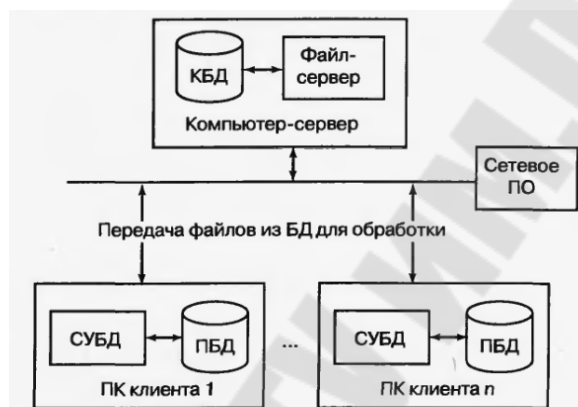


Рисунок 3.1 – Структура ИС с файл-сервером

В ИС архитектуры файл-сервер по запросам пользователей *файлы* БД передаются на ПЭВМ, где и производится их обработка.

Недостатки:

- *Высокая интенсивность передачи обрабатываемых данных.* Причем часто передаются *избыточные данные*: вне зависимости от того, сколько записей (строк) из БД требуется пользователю, файлы БД передаются целиком.
- *Обработка основных объемов данных на «слабых» (относительно сервера) ЭВМ.*

При *клиент-серверной архитектуре* (рисунок 3.2) сервер БД выполняет основную обработку данных. Формируемые пользователем или приложением запросы поступают к серверу БД в виде инструкций специального языка запросов (*SQL – Structured Query Language*). Сервер БД выполняет поиск и извлечение нужных данных (*не файлов!*), которые затем передаются на компьютер пользователя.



Рисунок 3.2 – Структура ИС с сервером БД

Достоинствами такого подхода в сравнении с предыдущим являются:

- меньший объем передаваемых по сети данных;
- более быстрая обработка данных, т. к. основная работа выполняется на, как правило, более мощном компьютере-сервере (по сравнению с ПК пользователя).

Для создания и управления ПБД и приложений, работающих с ними, используются различные СУБД: *Access, Visual FoxPro* (Microsoft); *Paradox* (Borland), ... КБД создается, поддерживается и работает под управлением сервера БД: *Microsoft SQL Server, Oracle Server, ...*

В зависимости от размеров и решаемых задач ИС может иметь одну из конфигураций:

- компьютер-сервер, содержащий корпоративную и персональные БД;
- компьютер-сервер и ПК с ПБД;
- несколько компьютеров-серверов и ПК с ПБД.

Другие преимущества клиент-серверной архитектуры:

- возможность постепенного наращивания ИС предприятия при развитии предприятия и/или самой ИС;
- разделение общей БД на КБД и ПБД позволяет уменьшить сложность проектирования БД по сравнению с централизованным вариантом, а следовательно, снизить вероятность ошибок и стоимость проектирования.

Примеры СУБД, представленных на современном рынке программных продуктов: *Lotus Approach, Data Ease, Paradox, MS Access, FoxPro, Clarion.*

Примеры приложений для создания приложений клиент-сервер: MS SQL Server, Sybase SQL Server, Informix, Oracle, PowerBuilder, Delphi, Visual Basic.

При работе приложения клиент-сервер база данных располагается на мощном компьютере - сервере, который принимает от программ, выполняемых на других компьютерах, - клиентов –запросы на получение той или иной информации из базы данных или осуществление тех или иных манипуляций с данными, обрабатывает полученные запросы и передает клиентам результаты.

3.1.3 Классификация систем управления базами данных: признаки классификации и виды СУБД. Аппаратное и программное обеспечение СУБД. Функции различных видов СУБД

Основные классификационные признаки СУБД:

- вид программы;
- характер использования;
- модель данных.

Эти признаки влияют на выбор СУБД и эффективность работы ИС. В общем случае под СУБД можно понимать любой программный продукт, поддерживающий процессы создания, ведения и использования БД. Рассмотрим, какие из программ имеют отношение к БД, и в какой мере они связаны с БД.

Классификация СУБД по виду программы:

- полнофункциональные СУБД;
- серверы БД;
- клиенты БД;
- средства разработки программ работы с БД.

Полнофункциональные СУБД (ПФСУБД) – «традиционные» СУБД, сначала – для больших машин, затем – для мини-ЭВМ и ПЭВМ. Среди всех СУБД современные ПФСУБД наиболее многочисленные и мощные по функциям: Clarion Database Developer, DataEase, dBASE IV, Access, FoxPro, Paradox. Обычно ПФСУБД имеют *графический интерфейс* для основных действий с БД (создания/изменения структуры таблиц, ввода/обработки данных, разработки форм/отчетов). Создание запросов – не только программирование на *SQL*, но и язык *QBE*. ПФСУБД могут включать: средства программирования и *CASE-подсистемы*; *факультативные модули* для доступа к другим БД или к данным SQL-серверов.

Серверы БД предназначены для организации центров обработки данных в сетях ЭВМ. Эта группа БД в настоящее время менее многочисленна, но их количество постепенно растет. Серверы БД реализуют функции управления БД, запрашиваемые другими (клиентскими) программами обычно с помощью операторов SQL. Примеры серверов БД: NetWare SQL (Novell), MS SQL Server (Microsoft), InterBase (Borland), SQLBase Server (Gupta), Intelligent Database (Ingress).

Клиентские программы для серверов БД – различные программы: ПФСУБД, электронные таблицы, текстовые процессоры, программы электронной почты и т. д.

Элементы пары «клиент – сервер» могут быть от одного или разных производителей ПО. Если клиентская и серверная части выполнены одной фирмой, то больше вероятность, что распределение функций между ними выполнено рационально.

В остальных случаях доступ к данным обеспечивается «любой ценой». Пример: для MS SQL Server в роли клиентских программ могут выступать Paradox, DataEase, Focus, 1-2-3.

Средства разработки программ работы с БД используются для создания:

- 1) клиентских программ;
- 2) серверов БД и их отдельных компонентов;
- 3) пользовательских приложений.

1), 2) – относительно мало программ (в основном, для системного программирования).

3) – гораздо больше (но меньше, чем ПФСУБД):

- системы программирования;
- библиотеки программ для языков программирования;
- CASE-пакеты.

Примеры: Delphi, Power Builder, Visual Basic, SILVERRUN, S-Designor, AllFusion ERwin Data Modeler.

Классификация СУБД по характеру использования:

- персональные;
- многопользовательские

Персональные СУБД – обеспечивают возможность создания ПБД и недорогих приложений, работающих с ними. Персональные СУБД или разработанные с их помощью приложения часто выступают в роли клиентской части многопользовательской СУБД.

Visual FoxPro, Paradox, Clipper, Access, ...

Многопользовательские СУБД включают сервер БД и клиентскую часть и, как правило, могут работать в *неоднородной вычислительной среде* (с разными типами ЭВМ и операционными системами).

Классификация СУБД (и БД) по модели данных:

- *иерархические;*
- *сетевые;*
- *реляционные;*
- *объектно-ориентированные;*
- *постреляционные;*
- *другие модели.*

Некоторые СУБД могут одновременно поддерживать несколько моделей данных.

Функции, выполняемые различными видами СУБД

С точки зрения пользователя, СУБД реализует следующие основные функции:

- хранение данных;
- изменение (добавление/ редактирование/ удаление) данных;
- обработка информации по заданным правилам;
- отображение данных в требуемом виде;
- разработка *и получение различных выходных документов.*

Для работы с данными СУБД предоставляет программам/пользователям 2 типа языков:

- *язык описания данных* – высокоуровневый непроцедурный язык декларативного типа для описания логической структуры данных;
- *язык манипулирования данными* – набор конструкций, обеспечивающих выполнение основных операций по работе с данными: ввод, модификацию и выборку данных по запросам.

ЯОД (*DDL*) и ЯМД (*DML*) в разных СУБД могут отличаться.

Наиболее распространены – два языка:

- *QBE* – язык запросов по образцу;
- *SQL* – структурированный язык запросов.

QBE в основном обладает свойствами *языка манипулирования данными*, *SQL* сочетает в себе свойства языков обоих типов – *описания и манипулирования данными.*

Перечисленные выше функции СУБД, в свою очередь, используют следующие основные функции более низкого уровня – низкоуровневые функции СУБД:

- управление данными во внешней памяти;
- управление буферами оперативной памяти;
- управление транзакциями;
- ведение журнала изменений в БД;
- обеспечение целостности и безопасности БД.

Управление данными во внешней памяти в разных системах может различаться:

- на уровне управления ресурсами (используя файловые системы ОС или непосредственное управление устройствами ПЭВМ);
- по логике самих алгоритмов управления данными.

Методы и алгоритмы управления данными являются «внутренним делом» СУБД и прямого отношения к пользователю не имеют. Качество реализации этой функции наиболее сильно влияет на эффективность работы специфических ИС, например, с большими БД и объемами обработки данных, со сложными запросами.

Управление буферами оперативной памяти.

Необходимость *буферизации данных* и как следствие реализации *функции управления буферами оперативной памяти* обусловлена тем, что объем оперативной памяти меньше объема внешней памяти. *Буферы* – области оперативной памяти, предназначенные для ускорения обмена данными между внешней и оперативной памятью. В буферах временно хранятся фрагменты БД, данные из которых предполагается использовать при обращении к СУБД или планируется записать в БД после обработки.

Управление транзакциями.

Транзакция – неделимая последовательность операций над данными БД, которая отслеживается СУБД от начала и до завершения. Механизм транзакций используется в СУБД для поддержания целостности данных. Если по каким-либо причинам (сбои и отказы оборудования, ошибки в программном обеспечении) транзакция остается незавершенной, то она отменяется.

Транзакции имеют три основных свойства:

- атомарность (выполняются все входящие в транзакцию операции или ни одна из операций);

- сериализуемость (отсутствует взаимное влияние выполняемых в одно и то же время транзакций);
- долговечность (результаты зафиксированной транзакции должны сохраняться при любых условиях).

Пример транзакции. Перевод денег с одного счета на другой в банковской системе.

Необходим, по крайней мере, двухшаговый процесс:

1) сначала снимают деньги с одного счета (обнулить значение поля в одной таблице);

2) затем добавляют их к другому счету (занести числовое значение в другую таблицу).

Если хотя бы одно из действий не выполнится успешно, результат операции окажется неверным и будет нарушен баланс между счетами. Контроль транзакций важен в однопользовательских и многопользовательских СУБД, где транзакции могут быть запущены параллельно. В этом случае говорят о *сериализуемости транзакций*.

Сериализация параллельно выполняемых транзакций – составление такого плана их выполнения (*сериального плана*), при котором суммарный эффект реализации транзакций = эффекту их последовательного выполнения. При параллельном выполнении смеси транзакций возможно возникновение конфликтов (*блокировок*), разрешение которых является функцией СУБД. При обнаружении таких случаев обычно производится «откат» путем отмены изменений, произведенных одной или несколькими транзакциями.

Ведение журнала изменений в БД (*журнализация изменений*) выполняется СУБД для обеспечения надежности хранения данных в БД при аппаратных сбоях, ошибках в ПО. *Журнал СУБД* – особая БД или часть основной БД, напрямую недоступная пользователю и используемая для записи информации обо всех изменениях БД. В различных СУБД в журнал могут заноситься записи, об изменениях в СУБД на разных уровнях: от минимальной внутренней операции модификации страницы внешней памяти до логической операции модификации БД (например, вставки записи, удаления столбца, изменения значения в поле) и транзакции.

Обеспечение целостности и безопасности БД.

Целостность БД – свойство БД, означающее, что в ней содержится полная, непротиворечивая и адекватно отражающая предметную область информация. Поддержание целостности БД включает проверку целостности и ее восстановление в случае обнаружения

противоречий в БД. Целостное состояние БД описывается с помощью *ограничений целостности* – условий, которым должны удовлетворять хранимые в базе данные. *Обеспечение безопасности* достигается в СУБД шифрованием прикладных программ, данных, защитой паролем, поддержкой уровней доступа к БД и к отдельным ее элементам (таблицам, формам, и т. д.).

3.1.4 Принципы разработки и выполнения приложений для информационных систем на основе СУБД

Современные СУБД позволяют решать широкий круг задач по работе с БД без разработки приложения. Также бывают случаи, когда целесообразно *разработать приложение*. Для разработки приложений СУБД должна иметь программный интерфейс, основу которого составляют функции и/или процедуры соответствующего языка программирования. Существующие СУБД поддерживают следующие технологии (и их комбинации) разработки приложений:

- ручное кодирование программ (Clipper, FoxPro, Paradox, ...);
- создание текстов приложений с помощью генераторов (FoxApp в FoxPro, Personal Programmer в Paradox);
- автоматическая генерация готового приложения методами визуального программирования (Delphi, Access, ...).

Ручное кодирование – текст программ приложений набирается вручную, затем – отладка.

Использование генераторов – упрощает разработку приложений, так как при этом можно получать программный код без ручного набора. Генераторы приложений облегчают разработку основных элементов приложений (меню, экранных форм, запросов, ...), но часто не могут полностью исключить ручное кодирование.

Средства визуального программирования приложений – развитие идеи использования генераторов приложений – приложение «строится» из готовых «строительных блоков» с помощью удобной интегрированной среды.

При необходимости в приложение легко вставляется программный код. Интегрированная среда предоставляет мощные средства создания, отладки и модификации приложений. Визуальное программирование позволяет быстрее создавать более надежные и эффективные приложения по сравнению с приложениями, полученными первыми двумя способами.

Виды приложений.

Независимые приложения – выполняются автономно от среды СУБД (основной файл приложения – исполняемый, например, ехе-файл); создаются компиляцией исходных текстов программ, полученных: ручным набором, с помощью генератора или среды визуального программирования.

Разрабатываются средствами: *FoxPro, Delphi* (с привлечением процессора БД Borland DataBase Engine, играющего роль ядра СУБД). Одно из первых средств разработки таких приложений – система *Clipper* – «чистый компилятор».

Интерпретируемые приложения – без среды СУБД не выполняются. Выполнение приложения: СУБД анализирует содержимое файлов приложения (в частном случае – текст исходной программы) и автоматически строит необходимые исполняемые машинные команды; т.е. приложение выполняется *методом интерпретации*.

Режим интерпретации реализован во многих современных СУБД: Access, Visual FoxPro, Paradox, а также в СУБД недавнего прошлого: FoxBase, FoxPro.

Приложения с псевдокомпиляцией – используют промежуточный вариант между компиляцией и интерпретацией – так называемую *псевдокомпиляцию*.

Исходная программа путем компиляции преобразуется в промежуточный код (*псевдокод*) и записывается на диск. В этом виде ее в некоторых системах разрешается даже редактировать, но главная цель псевдокомпиляции – преобразовать программу к виду, ускоряющему процесс ее интерпретации. Такой прием широко применялся в СУБД, работающих под управлением DOS.

В современных СУБД псевдокод чаще используют для запрета модификации приложения (защиты от случайного или преднамеренного повреждения работающей программы). Пример – СУБД Paradox for Windows – разработанные экранные формы и отчеты можно преобразовывать в объекты, не поддающиеся редактированию. Некоторые СУБД дают пользователю возможность выбора варианта разработки приложения:

- *интерпретируемый СУБД программный код;*
- *независимая программа.*

Преимущества независимых приложений:

- время выполнения машинной программы обычно меньше, чем при интерпретации;

- более эффективны на слабых машинах;
- удобны при установке систем «под ключ», когда необходимо «закрыть» приложение от доработок со стороны пользователей.

Преимущества интерпретируемых приложений:

- легкость модификации. Если готовая программа подвергается частым изменениям, то для их внесения нужна инструментальная система (СУБД или аналогичная среда). Для интерпретируемых приложений такой инструмент всегда «рядом».
- наличие мощных встроенных средств контроля целостности данных и защиты от несанкционированного доступа (в системах компилирующего типа эти функции приходится программировать вручную).

3.1.5 Система управления базой данных (СУБД), информационные потоки в САПР

Ядром любой базы данных является модель данных. *Модель данных* представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

Модель данных – совокупность структур данных и операций их обработки.

СУБД основывается на использовании иерархической, сетевой или реляционной модели, на комбинации этих моделей или на некотором их подмножестве.

Рассмотрим три основных типа моделей: иерархическую, сетевую и реляционную.

Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево).

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь.

Узел – это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящемся на более высоком уровне. Иерархиче-

ское дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне.

Зависимые (подчинённые) узлы находятся на втором, третьем и т.д. уровнях (рисунок 3.3). Количество деревьев в базе данных определяется числом корневых записей.

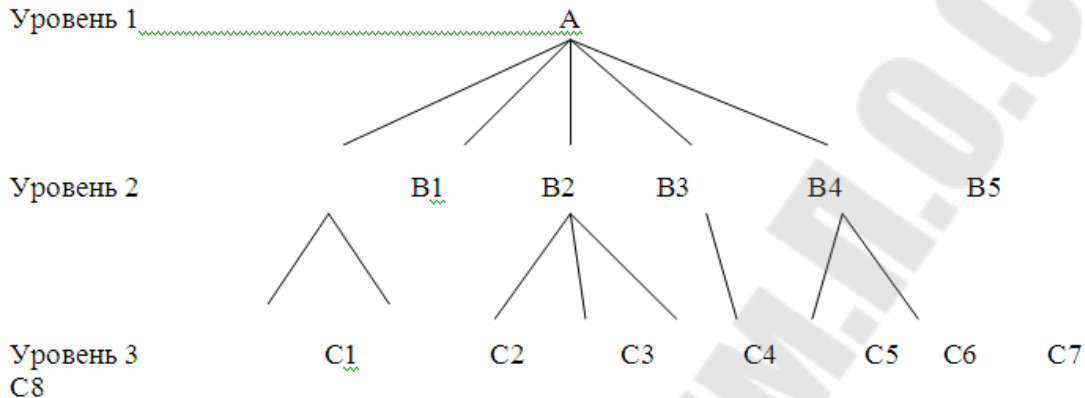


Рисунок 3.3 – Графическое изображение иерархической структуры БД

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом (рисунок 3.4).

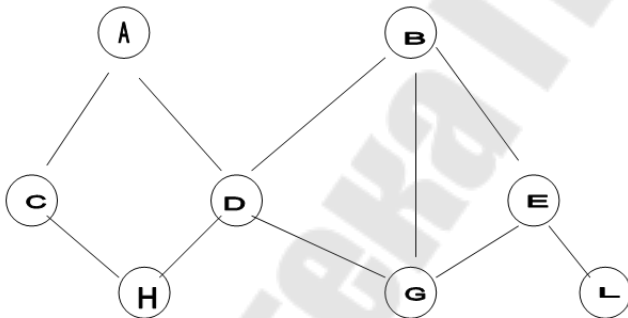


Рисунок 3.4 – Графическое изображение сетевой структуры

Реляционная модель данных.

Понятие *реляционный* (англ. Relation – отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда. Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

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 (числовой, символьный и т.д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

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

Информационный объект – это описание некоторой сущности (реального объекта, явления, процесса, события) в виде совокупности логически связанных *реквизитов* (информационных элементов). Такими сущностями для информационных объектов могут служить: цех, склад, материал, скорость вращения детали, характеристика шлифовального круга и т.д.

Информационный объект определенного реквизитного состава и структуры образует класс (тип), которому присваивается уникальное имя (символьное обозначение), например, Характеристика шлифовального круга, Диаметр, Ширина.

Информационный объект имеет множество реализаций – *экземпляров*, каждый из которых представлен совокупностью конкретных значений реквизитов и идентифицируется значением ключа (простого – один реквизит или составного – несколько реквизитов). Остальные реквизиты информационного объекта являются описательными. При этом одни и те же реквизиты в одних информационных объектах могут быть ключевыми, а в других – описательными. Информационный объект может иметь несколько ключей.

Приведем пример информационного объекта «Характеристика шлифовального круга», представленного в виде графа (рисунок 3.5).

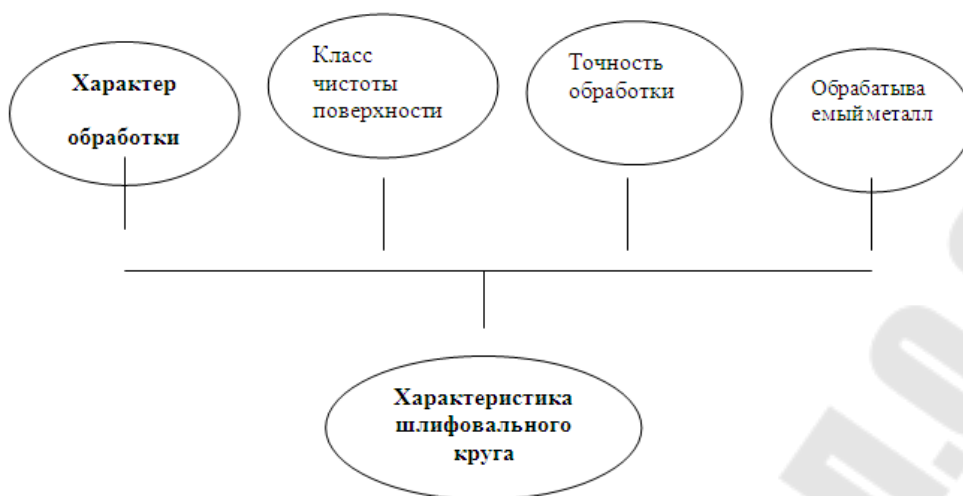


Рисунок 3.5 – Пример представления информационного объекта в виде графа

Типы связей.

Все информационные объекты предметной области связаны между собой. Различаются связи нескольких типов, для которых введены следующие обозначения:

- один к одному (1:1);
- один ко многим (1:M);
- многие ко многим (M:M).

Рассмотрим эти типы связей.

Связь *один к одному* (1:1) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует не более одного экземпляра информационного объекта В и наоборот (рисунок 3.6).

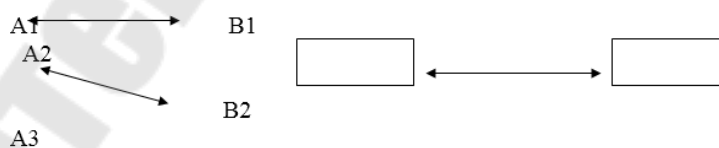


Рисунок 3.6 – Графическое изображение реального отношения 1:1

При связи *один ко многим* (1:M) одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В, но каждый экземпляр объекта В связан не более чем с 1 экземпляром объекта А (рисунок 3.7).

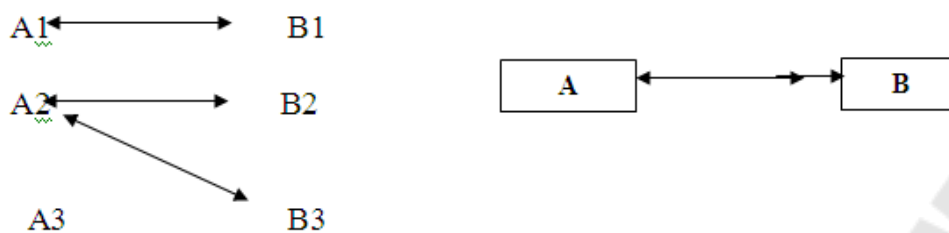


Рисунок 3.7 – Графическое изображение реального отношения 1:M

Связь *многие ко многим* (M:M) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В и наоборот (рисунок 3.8).



Рисунок 3.8 – Графическое изображение реального отношения M:M

3.1.6 Особенности и требования к базе данных САПР

К базам данных предъявляется ряд требований, среди которых можно выделить следующие основные требования.

1. *Минимальная избыточность*. Каждый элемент данных вводится в БД один раз и хранится в единственном экземпляре. При вводе данных СУБД выполняет проверку на дублирование. Этим достигается экономия внешней памяти и надежность информации.

2. *Независимость*. Модификация данных и изменения, вносимые в их структуру в связи с появлением новых пользователей и новых запросов, не должны отражаться на программах пользователей.

3. *Целостность данных*:

- *логическая* (СУБД должна защищать БД от некорректных действий пользователей путем восстановления состояния БД на момент, предшествующий ошибочной операции);
- *физическая* (защита носителей информации – дисков – от сбоев путем дублирования, например, двумя параллельно работающими зеркальными дисками).

4. *Секретность.* Пользователи должны работать только с теми данными (фрагментами данных), к которым им разрешен доступ.

3.1.7 Виды представления базы данных

Базы данных и программные средства их создания и ведения (СУБД) имеют многоуровневую архитектуру (рисунок 3.9).

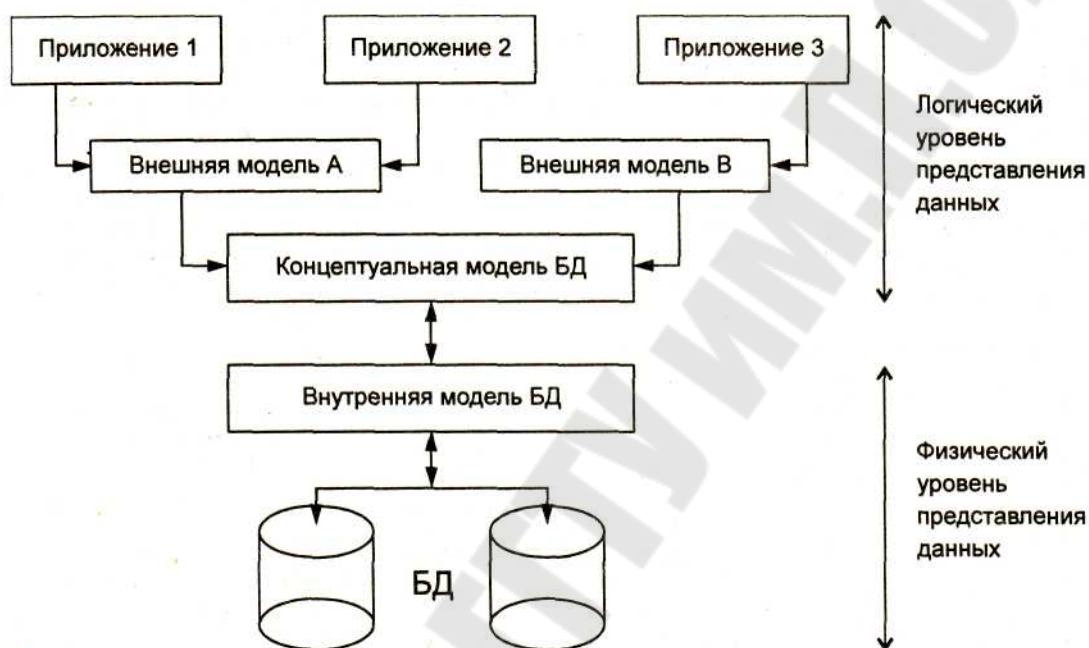


Рисунок 3.9 – Многоуровневое представление данных БД под управление СУБД

Различают концептуальный, внутренний и внешний уровни представления баз данных, которым соответствуют модели аналогичного назначения.

Концептуальный уровень соответствует логическому аспекту представления данных предметной области в интегрированном виде. *Концептуальная модель* состоит из множества экземпляров различных типов данных, сконструированных в соответствии с требованиями СУБД к логической структуре базы данных.

Внутренний уровень отображает требуемую организацию данных в среде хранения и соответствуют физическому аспекту пред-

ставления данных. *Внутренняя модель* состоит из отдельных экземпляров записей, физически хранимых во внешних носителях.

Внешний уровень поддерживает частные представления данных, требуемые конкретным пользователям. *Внешняя модель* является подмножеством концептуальной модели (рисунок 3.10). Возможно пересечение внешних моделей по данным. С помощью внешних моделей поддерживается санкционированный доступ к данным БД приложений (ограничен состав и структура данных концептуальной модели БД, доступных в приложении, а также заданы допустимые режимы обработки этих данных: ввод, редактирование, удаление, поиск).

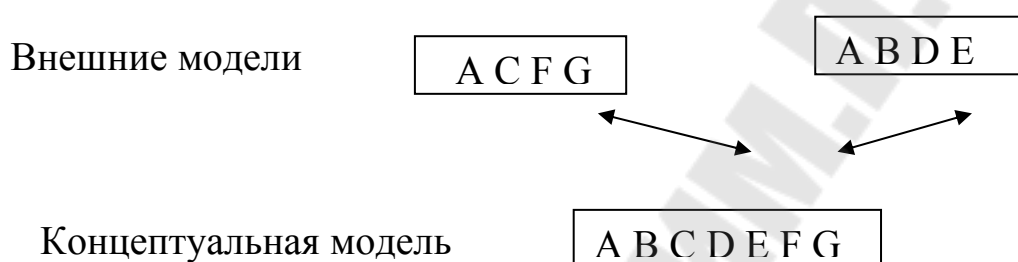


Рисунок 3.10 – Пример соотношения между концептуальной моделью и внешними моделями

Появление новых или изменение информационных потребностей существующих приложений требуют определения для них корректных внешних моделей, при этом на уровне концептуальной и внутренней модели данных изменений не происходит. Изменения в концептуальной модели, вызванные появлением новых видов данных или изменением их структур, могут затрагивать не все приложения, т.е. обеспечивается определенная независимость программ от данных. Изменения в концептуальной модели должны отражаться на внутренней модели, и при неизменной концептуальной модели возможна самостоятельная модификация внутренней модели БД с целью улучшения ее характеристик (время доступа к данным, расхода памяти внешних устройств и др.). Таким образом, БД реализует принцип относительной независимости логической и физической организации данных.

3.1.8 Структура построения базы данных

Проектирование базы данных состоит в построении комплекса взаимосвязанных моделей данных. На рисунке 3.11 условно отображены этапы процесса проектирования базы данных.

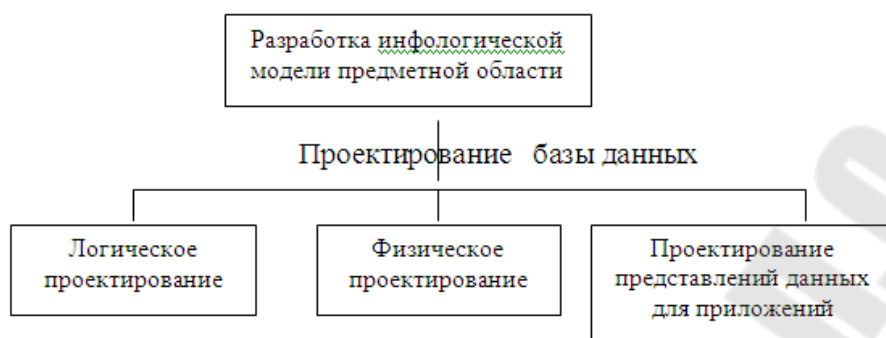


Рисунок 3.11 – Этапы процесса проектирования базы данных

Важнейшим этапом проектирования базы данных является разработка *информационно-логической (инфологической) модели* предметной области, не ориентированной на СУБД. В инфологической модели средствами структур данных в интегрированном виде отражают состав и структуру данных, а также информационные потребности приложений (задач и запросов).

Информационно-логическая (инфологическая) модель предметной области отражает предметную область в виде совокупности информационных объектов и их структурных связей.

Инфологическая модель предметной области строится первой. Затем на ее основе строятся концептуальная (логическая), внутренняя (физическая) и внешняя модели.

3.1.9 Обобщенная технология работы пользователей в среде СУБД

Каждая конкретная СУБД имеет свои особенности, которые необходимо учитывать. Однако, имея представление о функциональных возможностях любой СУБД, можно представить обобщенную технологию работы пользователя в этой среде.

В качестве основных этапов обобщенной технологии работы с СУБД можно выделить следующие (рисунок 3.12):

- создание структуры таблиц базы данных;

- ввод и редактирование данных в таблицах;
- обработка данных, содержащихся в таблицах;
- вывод информации из базы данных.

При формировании новой таблицы базы данных работа с СУБД начинается с создания структуры таблицы. Этот процесс включает определение перечня полей, из которых состоит запись таблицы, а также типов и размеров полей.

Заполнение таблиц данными возможно как непосредственным вводом данных, так и в результате выполнения программ и запросов.

Практически все СУБД позволяют вводить и корректировать данные в таблицах двумя способами:

- с помощью представляемой по умолчанию стандартной формы в виде таблицы;
- с помощью *экранных форм*, специально созданных для этого пользователем.

Обрабатывать информацию, содержащуюся в таблицах базы данных, можно путем использования запросов или в процессе выполнения специально разработанной программы.

Конечный пользователь получает при работе с СУБД такое удобное средство обработки информации, как запросы. *Запрос* представляет собой инструкцию на отбор записей.

Большинство СУБД разрешают использовать запросы следующих типов:

- запрос выборка, предназначенный для отбора данных, хранящихся в таблицах, и не изменяющий эти данные;
- запрос – изменение, предназначенный для изменения или перемещения данных; к этому типу запросов относятся: запрос на удаление записей, запрос на создание таблицы, запрос на обновление;
- запрос с параметром, позволяющий определить одно или несколько условий отбора во время выполнения запроса.

Практически любая СУБД позволяет вывести на экран и принтер информацию, содержащуюся в базе данных, из режимов таблицы или формы. Такой порядок вывода данных может использоваться, как черновой вариант, так как позволяет выводить данные только точно в таком же виде, в каком они содержатся в таблице или форме.

Каждый пользователь, работающий с СУБД, имеет возможность использования специальных средств построения *отчетов* для вывода

данных. Используя специальные средства отчетов, пользователь получает следующие дополнительные возможности вывода данных:

- включать в отчет выборочную информацию из таблиц базы данных;
- добавлять информацию, не содержащуюся в базе данных;
- при необходимости выводить итоговые данные на основе информации базы данных;
- располагать выводимую в отчете информацию в любом, удобном для пользователя виде;
- включать в отчет информацию из разных связанных таблиц базы данных.



Рисунок 3.12 – Схема обобщенной технологии работы в СУБД

3.2 Нормализация данных

3.2.1 Понятие нормализации данных

Одни и те же данные могут группироваться в реляционные таблицы различными способами. Группировка полей в таблицах должна быть рациональной, что означает *сведение к минимуму дублирования данных и упрощение процедур их обработки и обновления*.

Нормализация таблиц – это формальный аппарат ограничений (набор правил) на формирование таблиц, позволяющий:

- уменьшить дублирование данных;
- обеспечить целостность данных;
- уменьшить трудозатраты на ведение БД.

3.2.2 Основные нормальные формы реляционных отношений и их взаимосвязь

При практическом проектировании реляционных БД обычно используют *три нормальные формы таблиц*.

Таблица называется приведенной к *первой нормальной форме (1НФ)*, если все ее поля *простые* (далее неделимы).

Таблица в 1НФ не должна иметь повторяющихся записей и (или) групп полей. 1НФ предписывает, что все данные в столбцах реляционной таблицы должны быть атомарными или неделимыми.

Преобразование таблицы к 1НФ может привести к увеличению количества полей таблицы и изменению ключа.

Согласно 1НФ, в каждой позиции (клетке таблицы), определяемой строкой и столбцом, должна быть только одна величина, но не массив или список величин.

Рассмотрим пример таблицы 3.1, нарушающей требования 1НФ (звездочкой будем обозначать ключевое поле).

Таблица 3.1 – Заказы1

Заказы1		
КодЗаказа*	КодПокупателя	СодержаниеЗаказа
1	4	5 молотков, 3 дрели, 6 ключей
2	23	1 молоток, 4 ключа
3	15	2 ключа

Таблица Заказы1 служит для хранения данных о заказах на складе инструментов.

Эта таблица имеет простой ключ, состоящий из одного поля *КодЗаказа*. Данные в столбце *СодержаниеЗаказа* таблицы *Заказы1* содержат *списки величин и не являются атомарными*.

Так как в столбце *СодержаниеЗаказа* находится слишком много информации, то получить упорядоченную информацию из этой таблицы будет трудно. Например, трудоёмким окажется составление отчёта о суммарных закупках инструментов различных видов. Таблицу *Заказы1* можно преобразовать (*без потери данных*) в таблицу *Заказы2*, удовлетворяющую требованиям 1НФ.

Таблица 3.2 – *Заказы2*

Заказы2				
КодЗаказа*	КодСодержанияЗаказа*	КодПокупателя	Количество	СодержаниеЗаказа
1	1	4	5	молоток
1	2	4	3	дрель
1	3	4	6	ключ
2	1	23	1	молоток
2	3	23	4	ключ
3	3	15	2	ключ

Эта таблица имеет составной ключ из двух полей: *КодЗаказа* и *КодСодержанияЗаказа*. Данные всех столбцов таблицы *Заказы2* являются атомарными.

Таблица находится во *второй нормальной форме (2НФ)*, если она находится в 1НФ, и каждое неключевое (*описательное*) поле *функционально* зависит от всего ключа (простого или составного)

Функциональная зависимость полей – это зависимость, при которой определенному значению ключа соответствует *только одно значение описательного поля*.

Каждый неключевой (т.е. не входящий в ключ) столбец таблицы во 2НФ должен *полностью* (т.е. *функционально*) зависеть от *всего первичного ключа* (простого или составного).

Если таблица имеет простой первичный ключ, состоящий только из одного столбца, то она *автоматически находится во 2НФ* (при условии выполнения требований 1НФ). Если же первичный ключ *составной*, то таблица *необязательно находится во 2НФ*. В этом случае необходимо преобразовать таблицу (или разделить ее на несколько таблиц) так, чтобы *весь первичный ключ* однозначно определял значение в любом неключевом столбце. Одну и ту же таблицу можно привести к 2НФ разными способами, не теряя данных. Приведение таблицы к 2НФ позволяет избежать повторения одних и тех же данных, которое может появиться после приведения таблицы к 1НФ.

Например, таблица Заказы2 не находится во 2НФ. Эта таблица имеет составной ключ, включающий 2 поля: *КодЗаказа* и *КодСодержанияЗаказа*. В то же время, неключевое поле *СодержаниеЗаказа* однозначно определяется только полем *КодСодержанияЗаказа*, т.е. *только одним* из двух полей, входящих в составной ключ. В результате нет функциональной зависимости столбца *СодержаниеЗаказа* от *всего* ключа (рисунок 3.13).

составной первичный ключ

Есть 1 НФ		Заказы2		Нет 2 НФ
КодЗаказа*	КодСодержанияЗаказа*	КодПокупателя	Количество	СодержаниеЗаказа
1	1	4	5	молоток
1	2	4	3	дрель
1	3	4	6	ключ
2	1	23	1	молоток
2	3	23	4	ключ
3	3	15	2	ключ

нет функциональной зависимости от всего ключа

Рисунок 3.13 – Нет 2НФ

При заданной кодировке инструментов на складе для заполнения значения поля *СодержаниеЗаказа* достаточно знать только значение поля *КодСодержанияЗаказа* (1 – молоток, 2 – дрель и т. д.). Таким образом, для поля *СодержаниеЗаказа* отсутствует функциональная зависимость от *всего* составного ключа и таблица *Заказы2* не находится во 2НФ.

Таблицу *Заказы2* можно преобразовать без потери данных в таблицу 3.3, удовлетворяющую требованиям 2НФ («исчезла» ненужная кодировка по содержанию заказа).

Таблица 3.3 – Заказы3

Заказы3			
КодЗаказа*	СодержаниеЗаказа*	КодПокупателя	Количество
1	молоток	4	5
1	дрель	4	3
1	ключ	4	6
2	молоток	23	1
2	ключ	23	4
3	ключ	15	2

У таблицы *Заказы3* составной ключ содержит 2 поля: *КодЗаказа* и *СодержаниеЗаказа*, а неключевые или описательные поля *КодПокупателя* и *Количество* функционально зависят от *всего* составного ключа.

Из данного примера также видно, как уменьшается повторение одинаковых данных при переходе от 1НФ ко 2НФ («исчезает» один столбец из двух, содержащих повторяющиеся данные в таблице Заказы2).

Таблица находится в *третьей нормальной форме* (3НФ), если она находится во 2НФ, и каждое неключевое поле *нетранзитивно зависит от первичного ключа*. *Транзитивная зависимость полей* имеет место в том случае, если одно из двух описательных полей зависит от ключа, а другое описательное поле зависит от первого описательного поля. Таким образом, таблица соответствует 3НФ, если она соответствует 2НФ, и все неключевые столбцы *взаимно независимы*. Например, таблица

Студенты2 = (Номер*, Фамилия, Имя, Дата, Группа, Староста)

не находится в 3НФ, так как описательное поле *Староста* зависит от описательного поля *Группа* (определенное значение поля *Группа* требует соответствующего значения поля *Староста*, т.е. имеется транзитивная зависимость этих полей).

Поэтому таблица Студенты2 должна быть *расщеплена* на две связанные (по полю *Группа*) таблицы, каждая из которых находится в 3НФ:

Студенты = (Номер*, Фамилия, Имя, Дата, Группа),

Старосты = (Группа*, Староста).

Транзитивные зависимости полей создают проблемы при *добавлении, обновлении и удалении* записей из таблицы. Например, если в какой-то группе поменялся староста, то для этой группы в каждую запись таблицы Студенты2 (всего таких записей столько, сколько студентов в группе) нужно внести изменение в поле *Староста*. А для расщеплённых таблиц Студенты, Старосты, удовлетворяющих 3НФ, в этом случае нужно изменить *только одну запись* в таблице Старосты.

Таблица Заказы3 также находится в 3НФ – так как неключевые поля *КодПокупателя* и *Количество* взаимно независимы, то в этой таблице нет транзитивных зависимостей полей.

Таким образом, «плата» за нормализацию таблиц – добавление в таблицы новых полей, возможное увеличение количества таблиц в процессе нормализации. «Выгода» от нормализации – более простая и быстрая обработка данных; экономия памяти и машинного времени для хранения и обработки данных.

3.3 Система управления базами данных MS Access

3.3.1 Основные характеристики СУБД ACCESS

Microsoft Access (MS Access) - одна из наиболее популярных СУБД. Данная СУБД входит в широко распространенный интегрированный пакет фирмы Microsoft - MS Office Pro. MS Access применяется для разработки относительно небольших информационных систем, содержащих, как правило, не более 50000 записей и функционирующих в локальных вычислительных сетях, объединяющих 1-15 компьютеров. MS Access работает в операционной системе Windows. Имеются версии данной СУБД: 2.0 – для Windows 3.1; 7.0 (95), 97, 2000 – для Windows 95, 98, 2000 и Windows NT. Большим преимуществом MS Access является наличие средств разработки информационных систем для пользователей различной квалификации: от начинающих до профессионалов.

На рисунке 3.14 показана структура и взаимосвязи основных элементов СУБД MS Access.

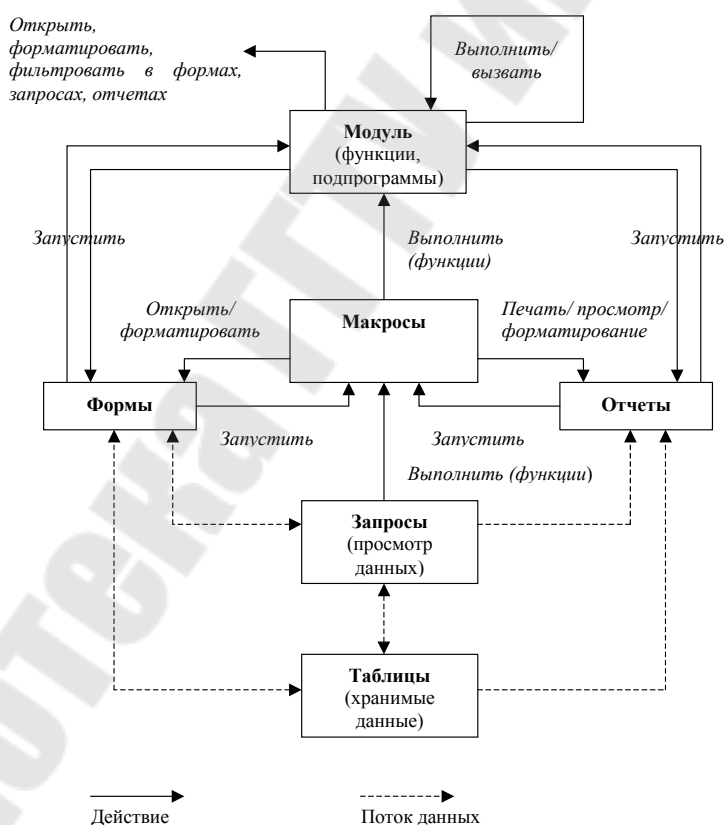


Рисунок 3.14 – Схема структуры и взаимосвязи основных функциональных элементов СУБД MS Access

Кратко охарактеризуем основные структурные элементы СУБД Access.

Таблица является основным элементом всякой реляционной базы данных, в том числе и СУБД MS Access. Эти объекты предназначены для хранения данных. Возможные источники данных для таблиц: 1) непосредственный ввод; 2) импорт из других баз данных (FoxPro, dBASE и др.) или электронных таблиц (Excel, Lotus).

Типы данных, используемых при заполнении таблиц СУБД MS Access, описаны в таблице 3.4.

Таблица 3.4 – Типы данных таблиц СУБД MS Access

<i>Тип данных</i>	<i>Содержимое поля</i>	<i>Размер</i>
1. Текстовый	Значение по умолчанию. Текст или числа, не требующие проведения расчетов, например, номера телефонов	Число символов, не превышающее 255. MS Access не сохраняет пустые символы в неиспользуемой части поля
2. Поле Мемо	Длинный текст или комбинация текста и чисел	До 65535 символов
3. Числовой	Числовые данные, используемые для проведения расчетов	1, 2, 4, 8 байт
4. Дата / время	Даты и время, относящиеся к годам с 100 по 9999, включительно	8 байт
5. Денежный	Денежные значения и числовые данные, используемые в математических расчетах, проводящихся с точностью до 15 знаков в целой и до 4 знаков в дробной части	8 байт
6. Счетчик	Уникальные последовательно возрастающие (на 1) или случайные числа, автоматически вводящиеся при добавлении каждой новой записи в таб-	4 байта

	лицу. Значения полей типа Счетчик обновлять нельзя	
7. Логический	Логические значения, а также поля, которые могут содержать одно из двух возможных значений (Да / Нет, Истина / Ложь, Вкл / Выкл)	1 бит
8. Поле объекта OLE	Объект (например, электронная таблица MS Excel, документ MS Word, рисунок, звукозапись или другие данные в двоичном формате), связанный или внедренный в таблицу MS Access	До 1 Гбайт (ограничивается объемом диска)
9. Гиперссылка	<p>Строка, состоящая из букв и цифр, и представляющая адрес гиперссылки. Адрес гиперссылки может состоять максимум из трех частей:</p> <ul style="list-style-type: none"> ▪ <i>текст</i> - текст, выводимый в поле или в элементе управления; ▪ <i>адрес</i> - путь к файлу (в формате пути UNC) или странице (адрес URL); ▪ <i>дополнительный адрес</i> – смещение внутри файла или страницы. <p>Для вставки адреса гиперссылки в поле или в элемент управления, нужно выбрать из меню Вставка команду Гиперссылка</p>	Каждая из трех частей в типе Гиперссылка может содержать до 2048 символов
10. Мастер подстановок	Создает поле, в котором предлагается выбор значений из списка, или из поля со списком, содержащего набор постоянных значений или значений из другой таблицы. Выбор этого параметра в списке в ячейке запускает мастера подстановок, который определяет тип поля	Тот же размер, что и у ключевого поля, используемого в подстановке (обычно 4 байт)

Запросы служат источниками для построения других запросов, форм и отчетов. Запросы позволяют изменять и анализировать данные. Наиболее распространенный тип запроса - запрос на выборку - представляет собой набор правил, по которым отбираются данные из

одной или нескольких связанных таблиц. Результаты выполнения запроса на выборку представляются в виде таблицы.

Формы – основное средство обработки, анализа и преобразования данных. Формы используются для:

- ввода данных в таблицу;
- отображения отобранных данных в заданном виде на экране и при выводе на печать;
- в качестве панели управления для открытия других форм, отчетов, таблиц, запросов;
- в качестве окна диалога.

Источниками данных для форм служат: таблицы; запросы; данные, заложенные в макете формы. По внешнему виду формы делятся на табличные, ленточные, простые. В простой форме видны поля, относящиеся только к одной записи. Табличная и ленточная формы выводят сразу несколько записей. Ленточная форма обладает более разнообразными возможностями отображения данных по сравнению с табличной.

Все сведения в форме или отчете содержатся в элементах управления. *Элементы управления* – это объекты формы или отчета, которые служат для вывода данных на экран, выполнения макрокоманд или оформления формы или отчета.

Отчеты – это средство для организации вывода данных на печать. С помощью отчета имеется возможность вывести необходимые сведения в нужном виде. Возможен предварительный просмотр отчета перед выводом на печать. Источниками данных для отчетов служат таблицы, запросы, данные, заложенные в макет. Основные элементы управления отчетом такие же, что и у формы.

Макросы – это набор стандартных инструкций (макрокоманд), позволяющих управлять элементами базы данных. Примеры макрокоманд: открыть/закрыть таблицу; открыть/закрыть форму; открыть/закрыть запрос; печать; предварительный просмотр; выход и т.д.

Макросы имеют имена и вызываются к выполнению при наступлении событий. Например, макрос Autoexec - это набор инструкций, обрабатываемый при открытии базы данных.

Макросы являются простым и эффективным способом автоматизации приложения.

События – любые изменения состояния объектов MS Access. События возникают при выполнении действий, например, при открытии/закрытии формы, нажатии на кнопку.

Модули – это объекты, содержащие программы, написанные на языке Visual Basic for Applications (VBA). Модули позволяют разбить некоторый процесс на несколько небольших процедур и найти ошибки, которые нельзя было бы найти при использовании макросов.

В MS Access встречаются два вида модулей: 1) независимые объекты, содержащие функции, вызываемые из любого места приложения; 2) непосредственно “привязанные” к отдельным формам или объектам для реакции на те или иные происходящие в них изменения (события).

Модули, в отличие от макросов, применяются при разработке более сложных приложений.

Меню применяются для управления объектами баз данных. Типы меню: строка меню; панель инструментов; контекстное меню. Меню могут быть общими либо привязанными к форме или отчету. По внешнему виду меню представляют собой набор кнопок, расположенных в строку или на панели. MS Access позволяет создавать свои собственные и изменять стандартные меню.

3.3.2 Практическая работа с MS Access

Рассмотрим работу некоторого научно-консультативного центра, который приглашает высококвалифицированных специалистов для чтения лекций и проведения консультаций по различным проблемам экономики, юриспруденции и т.п.

Очевидно, администрация центра должна иметь информацию о каждом эксперте – *фамилия, ученая степень, должность, место работы, телефоны, электронный адрес, адрес, область компетенции*. Необходимо также знать сведения об организациях, где они работают – *название, ведомство, адрес, телефон, факс* и т.п. Кроме того, необходимо накапливать *сведения о работе, выполненной экспертами по заявкам центра, а также об оплате их услуг*.

При желании всю эту информацию можно разместить в одной большой таблице.

При этом в столбцах таблицы (полях) будет собрана информация определенного типа (фамилия, степень, должность...), а строки будут содержать всю информацию о специалисте. Но такая таблица

будет иметь существенный *недостаток* – многие сведения будут повторяться. Так, в строках, содержащих информацию о специалистах, работающих в одной организации, будут повторяться сведения об этой организации.

Очевидно, такой недостаток ведет не только к избыточности информации, но и к ошибкам, которые неизбежны при вводе повторяющихся данных, а это будет причиной ошибок при обработке информации.

Такие проблемы можно избежать, если разбить таблицу на несколько таблиц, содержащих сведения об отдельных объектах.

Итак, приступим.

Создание новой базы данных.

Необходимо выполнить следующие действия:

- запустить MS Access;
- выбрать *Создать файл*
- выбрать *Новая база данных*
- ввести имя базы данных и выбрать ее место расположения на диске.

Создание первой простой таблицы.

Основной и наиболее универсальный путь создания таблицы - использование режима конструктора (рисунок 3.15) Для создания таблицы в этом режиме необходимо:

- определить поля таблицы – задать их имена и типы данных;
- задать свойства полей;
- дать имя таблице.

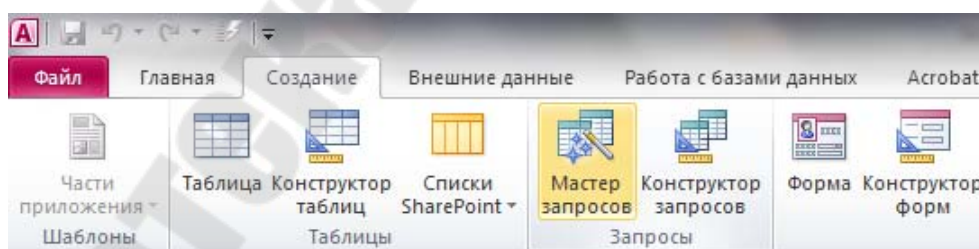


Рисунок 3.15 – Вкладка «Создание»

Просмотр данных.

Просмотр данных, содержащихся в таблице, осуществляется в режиме таблицы (выделить необходимую таблицу и нажать кнопку Открыть). В этом режиме можно не только просматривать, но и фор-

матировать данные, вставлять и удалять записи, редактировать данные.

Вернемся к своему примеру.

Для полей первой таблицы 3.5 «Организация» установлены следующие типы данных в режиме «Конструктор» (рисунок 3.16).

Таблица 3.5 – «Организация»

Поля	Типы данных
Код организации	Счетчик
Организация	Текстовый
Ведомство	Текстовый
Адрес	Текстовый
Телефон	Текстовый, с маской ввода
Факс	Текстовый

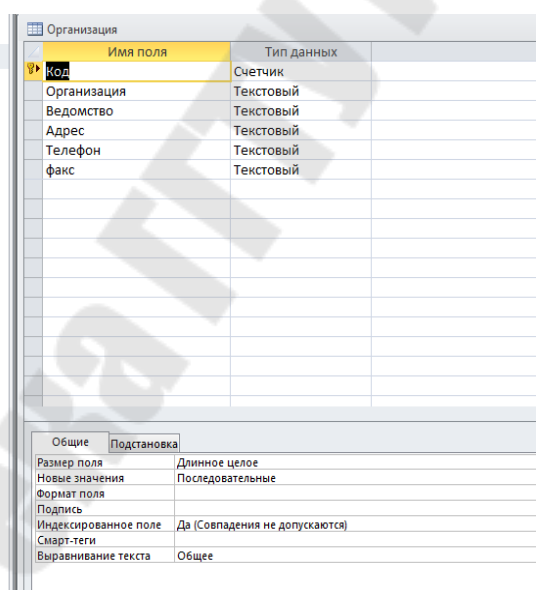


Рисунок 3.16 – Таблица «Организация» в режиме «Конструктор»

Для полей второй таблицы 3.6 «Эксперты» установлены следующие типы данных в режиме «Конструктор» (рисунок 3.17).

Таблица 3.6 – «Эксперты»

Поля	Типы данных
Код эксперта	Счетчик
Фамилия	Текстовый
Ученая степень	Текстовый
Код_Организации	Мастер подстановок
Подразделение	Текстовый
Должность	Текстовый
Телефон	Текстовый
E-mail	Текстовый
Тариф	Числовой

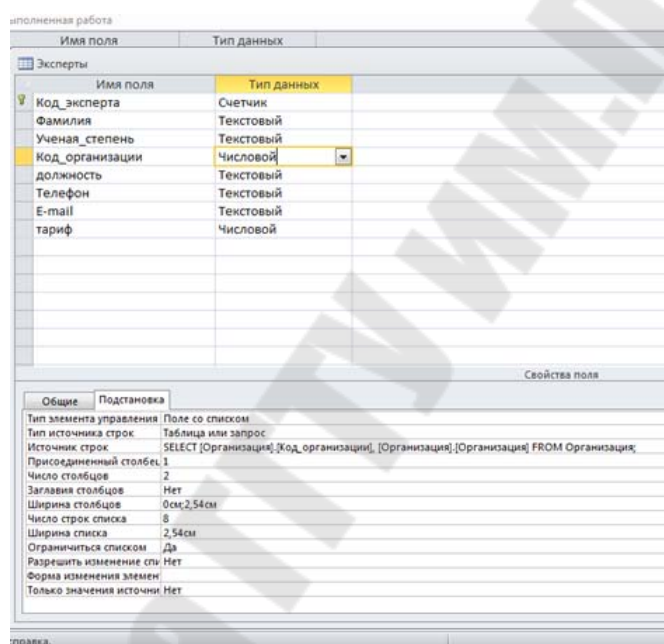


Рисунок 3.17 – Таблица «Эксперты» в режиме «Конструктор»

Тип данных поля *Код_Организации* – *Мастер подстановок* состоит из следующих шагов (рисунки 3.18–3.20).

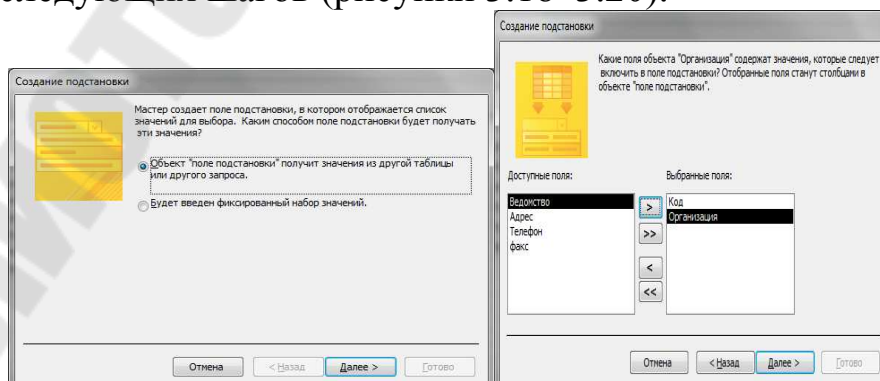


Рисунок 3.18 – Мастер подстановок

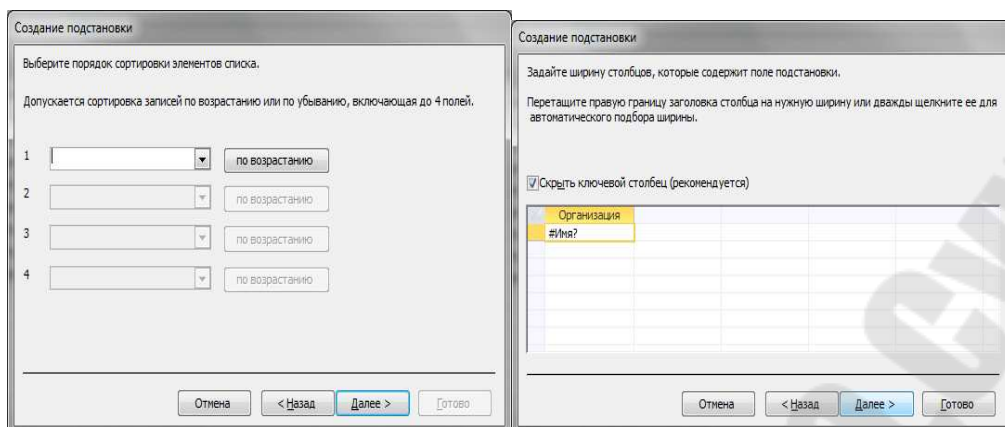


Рисунок 3.19 – Мастер подстановок (продолжение)

Для полей третьей таблицы «Выполненная работа» установлены следующие типы данных в режиме «Конструктор» (рисунок 3.17).

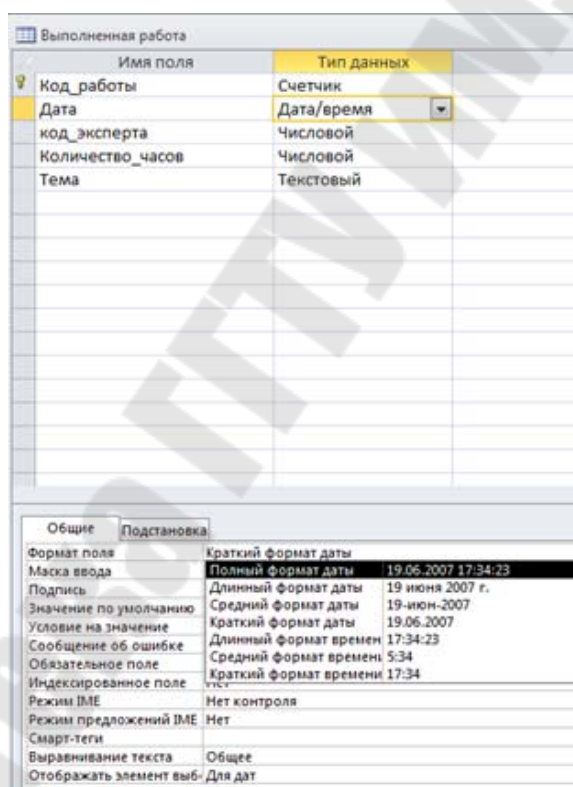


Рисунок 3.20 –Таблица «Выполненная работа» в режиме «Конструктор»

Создание связи между таблицами.

Глобальная связь задается командой **Сервис / Схема данных** или нажатием кнопки *Схема данных* на панели инструментов (рисунок 3.21). При этом связующее поле перетаскивается из таблицы, на-

ходящейся на стороне отношения «один», в таблицу на стороне отношения «многие» (рисунок 3.22).

В диалоговом окне «Связи», устанавливаются следующие опции: *Обеспечение целостности данных*; *Каскадное обновление связанных полей*; *Каскадное удаление связанных полей*; *Объединение* (рисунок 3.23).

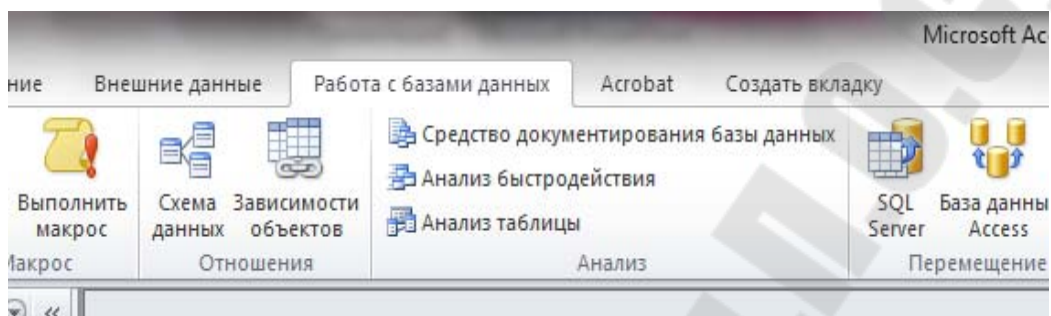


Рисунок 3.21 – Работа с базами данных

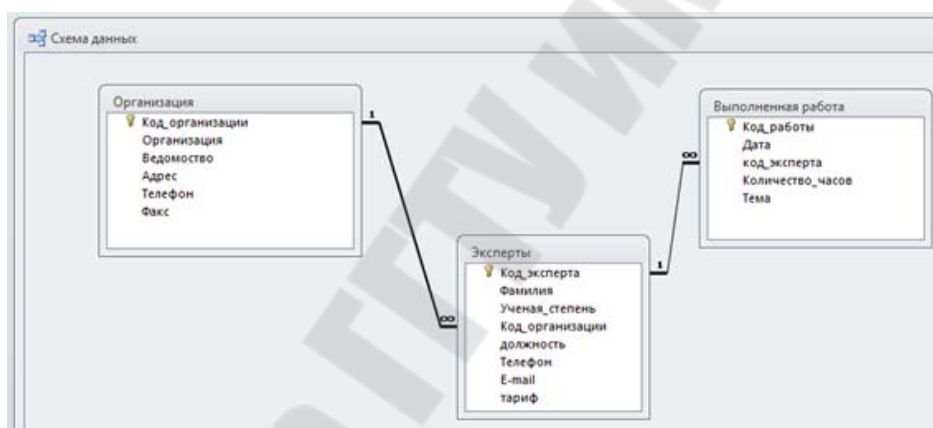


Рисунок 3.22 – Схема данных

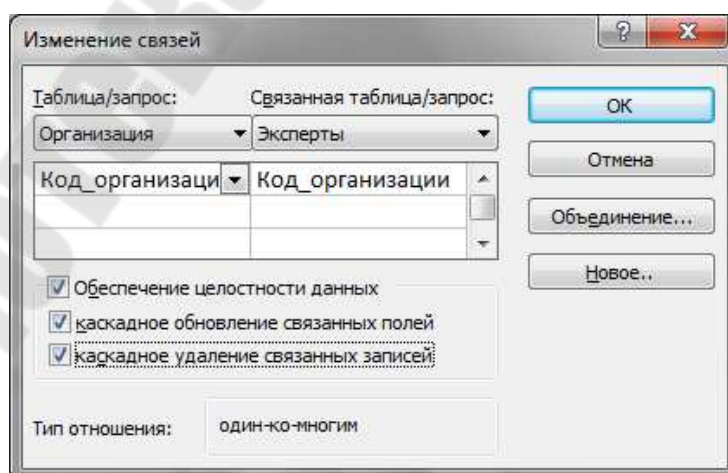


Рисунок 3.23 – Изменение связей

Ввод данных.

В окне базы данных нужно:

- выбрать вкладку *Таблицы*;
- выделить нужную таблицу;
- нажать кнопку *Открыть*;
- заполнить таблицы данными в следующем порядке: *Организация, Эксперты, Выполненная работа*

Работа с запросами.

Запрос – это объект, позволяющий пользователю получить нужные данные из одной или нескольких связанных таблиц. Для создания запроса можно использовать бланк QBE (Query by example - запрос по образцу), показанный на рис. 14, или непосредственно вводить текст инструкции SQL (Structured Query Language – структурированный язык запросов, используемый для обработки данных таблиц в MS Access).

Отметим, что при использовании бланка QBE инструкция SQL автоматически генерируется MS Access без участия пользователя.

Для создания нового запроса нужно щелкнуть в панели инструментов на кнопке *Новый объект* и выбрать кнопки *Запрос / Создать*. При этом появляется диалоговое окно «Новый запрос», предлагающее пять основных типов запросов и методов их создания:

- *Конструктор* (вызывается чистый бланк запроса QBE);
- *Простой запрос* (создает запрос к выбранным полям);
- *Перекрестный запрос* (запускается мастер для построения особого типа итогового запроса, который суммирует данные в табличной форме);
- *Повторяющиеся записи* (запускается мастер для построения запроса на выборку, который обнаруживает повторяющиеся записи в таблице);
- *Записи без подчиненных* (запускается мастер для построения запроса на выборку, который обнаруживает в таблице записи, не имеющие соответствующей записи в связанной таблице).

Создание запроса в режиме конструктора.

Создание запроса на выборку в режиме конструктора выполняется с помощью бланка запроса в следующем порядке:

- вкладка *Запросы*;
- кнопка *Создать*;
- кнопки *Конструктор, ОК*;

- добавить нужные таблицы (одну или несколько) с помощью диалогового окна «Добавление таблицы»;
- установить связи между таблицами (если таблиц несколько);
- с помощью мыши (по технологии «drag and drop») переместить в бланк запроса необходимые поля из таблиц;
- задать тип сортировки для тех полей, где это необходимо;
- задать, если необходимо, условия отбора: ограничения, налагаемые на значения полей;
- добавить, при необходимости, вычисляемые выражения (поля);
- отметить, какие поля нужно выводить на экран;
- сохранить запрос под некоторым именем.

На рисунке 3.24 показан запрос на выборку сведений об экспертах, отсортированных в порядке возрастания

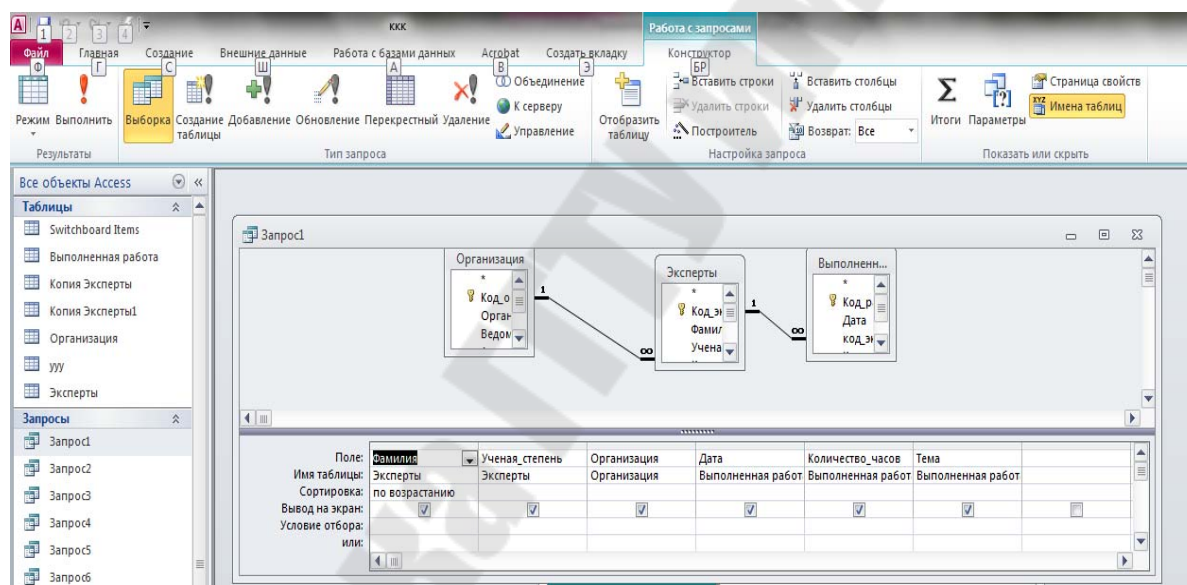


Рисунок 3.24 – Выбор сведений об экспертах

Для установки свойств полей запроса нужно: 1) щелкнуть по ячейке поля; 2) выбрать команду Вид / Свойства; 3) в диалоговом окне «Свойства поля» задать свойства на вкладках «Общие», «Подстановка».

Для задания условий отбора, накладывающих ограничения на значение поля, в соответствующую ячейку строки «Условие отбора» вводится нужное выражение. Это выражение состоит из операторов сравнения и операндов (значений). Если выражение не содержит оператора, то Access будет использовать оператор “=” (равно). Выражения могут соединяться логическими операторами AND и OR.

Приведем некоторые операторы сравнения.

< меньше; > больше; <= меньше или равно; <> не равно;
>= больше или равно.

IN - задает используемый для сравнения список значений; например, при условии отбора IN (“Склад1”, “Склад2”, “Склад3”) будут отобраны только данные полей Склад1, Склад2, Склад3.

BETWEEN – определяет диапазон значений, например, BETWEEN 10 And 20 означает то же самое, что и выражение >= 10 And <= 20; BETWEEN #01.01.97# AND #01.09.97# - подходят все даты между 1 января и 1 сентября 1997 года.

LIKE – оператор для поиска образцов в текстовых полях; при этом используются следующие символы шаблона: “?” – любой символ в данной позиции; “*” – любое количество символов в данной позиции; “#” – в данной позиции должны стоять цифры. Например, в случае условия отбора LIKE “С*К” подходят все значения, начинающиеся с буквы С и заканчивающиеся на букву К; LIKE “?00###” – задается строка, состоящая из 7 символов, первые 2 из которых являются произвольными, за ними следуют два нуля, а затем – три любые цифры.

На рисунке 3.25 показан запрос на выборку сведений об экспертах, которые являются кандидатами любых наук.

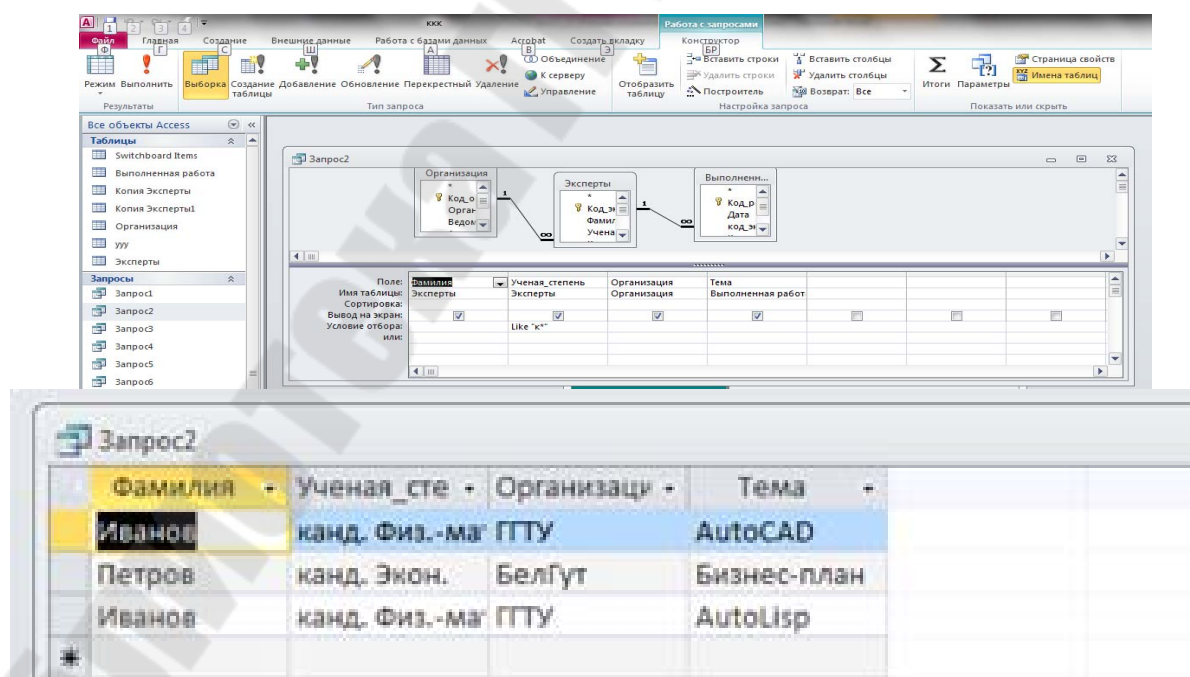


Рисунок 3.25 – Выбор сведений о кандидатах любых наук

Параметрические запросы. Для введения конкретных условий отбора в диалоговом режиме используются запросы с параметром. Чтобы определить параметр, нужно ввести в строку «Условия отбора» вместо конкретного значения (операнда) имя или фразу, заключенную в квадратные скобки []. То, что заключено внутри квадратных скобок, Access рассматривает как имя параметра. Это имя выводится в окне диалога при выполнении запроса, поэтому в качестве имени параметра используют содержательную фразу. В одном запросе можно указать несколько параметров, при этом имя каждого должно быть уникальным и информативным.

На рисунке 3.26 представлен параметрический запрос сведения о заданном эксперте, а на рисунке 3.27 – информация о выполненной в заданном интервале времени.

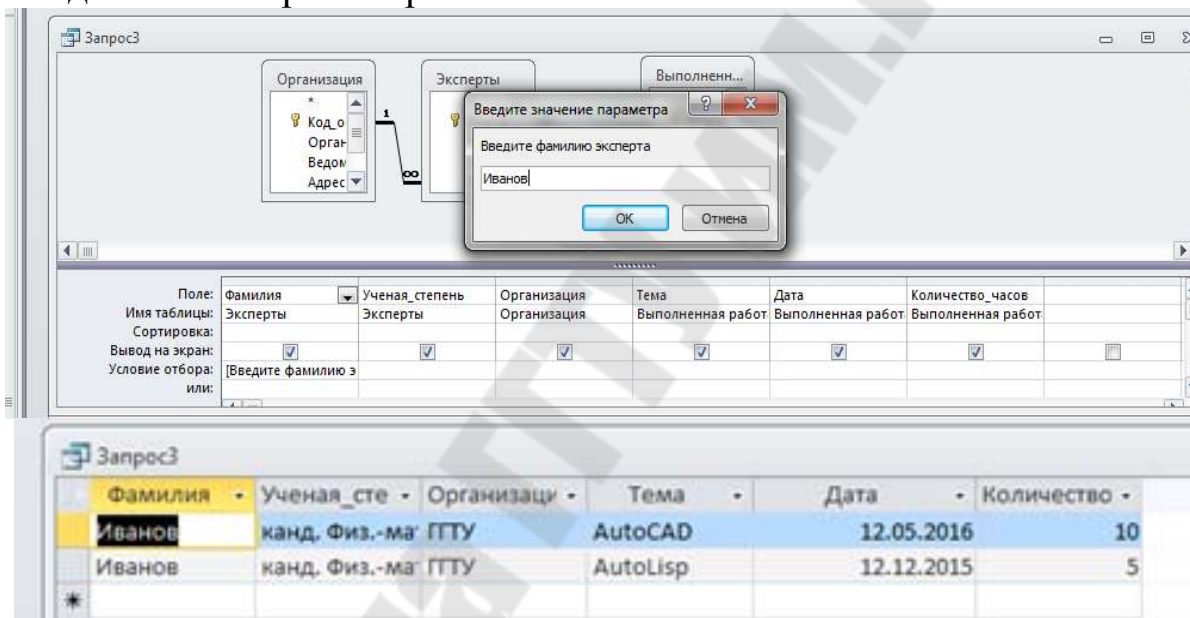


Рисунок 3.26 – Выбор сведений о заданном эксперте

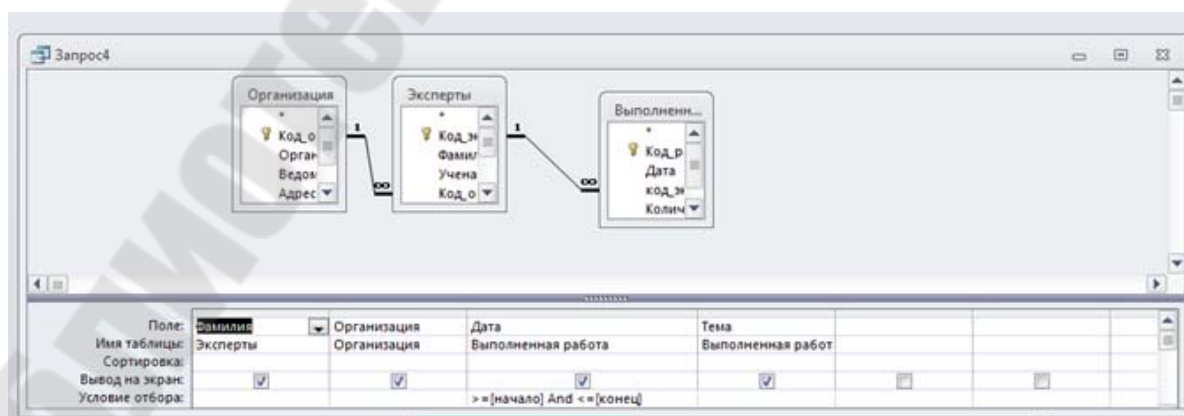


Рисунок 3.27 – Информация о выполненной в заданном интервале времени

Вычисляемые поля. В запросе можно выполнять вычисления с любыми полями таблицы, а также делать вычисляемое выражение новым полем в наборе записей.

При создании таких полей можно использовать: 1) встроенные функции Access; 2) арифметические операции над полями таблицы.

Например, для создания нового поля с именем «Зарплата» воспользуемся построителем выражений (рисунок 3.28).

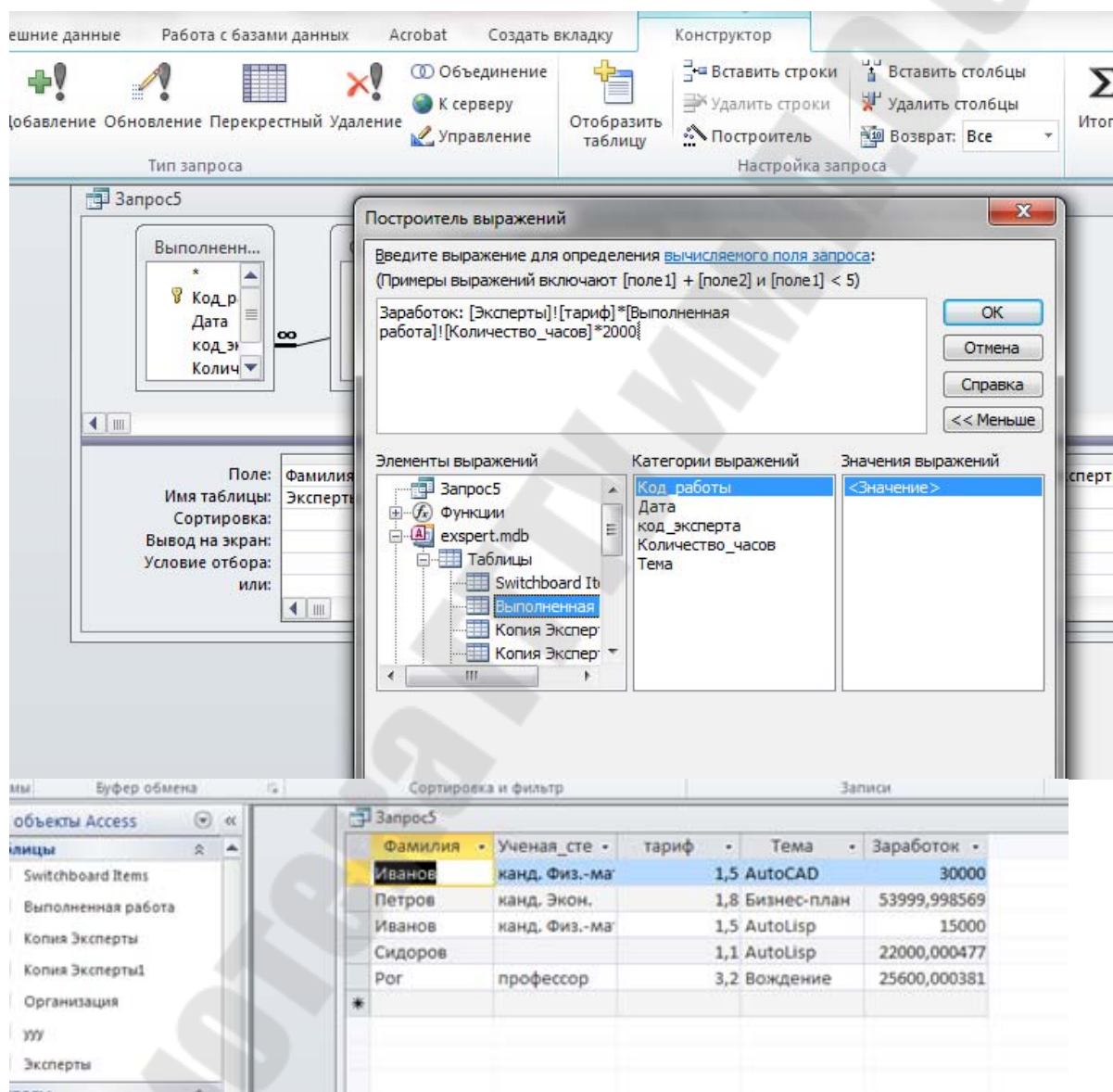


Рисунок 3.28 – Вычисляемое поле «Зарплата»

Итоговые запросы на выборку.

Эти запросы предназначены для вывода не отдельных записей таблицы, а итоговых значений по группам данных.

Например, для вычисления в поле общей суммы заработка эксперта (рисунок 3.29), нужно нажать на панели инструментов кнопку со значком суммы (Σ) – Групповые операции. При этом в бланке запроса появляется строка «Групповая операция» с установкой «Группировка» для каждого поля, занесенного в бланк запроса.

Для получения итоговых значений в нужных полях установка «Группировка» заменяется на конкретную итоговую функцию, которую можно ввести с клавиатуры или выбрать из раскрывающегося списка.

Некоторые итоговые функции Access перечислены ниже:

- **Sum** – вычисляет сумму всех значений заданного поля в каждой группе. Используется только для числовых или денежных полей;
- **Avg** – вычисляет среднее арифметическое всех значений данного поля в каждой группе. Используется только для числовых или денежных полей. Не учитывает в вычислениях нулевые значения (*Null*);
- **Min (Max)** – возвращают наименьшее (наибольшее) значение, найденное в этом поле внутри каждой группы. Для текстовых полей возвращают наименьшее (наибольшее) из символьных значений независимо от регистра. Игнорируют значение *Null*;
- **Count** – возвращает число записей, в которых значения данного поля отличны от нулевых значений. Для подсчета числа записей с учетом значений *Null* в строку «Поле» бланка запроса нужно ввести выражение **Count(*)**;
- **First (Last)** – возвращают первое (последнее) значение этого поля в группе.

При создании запросов можно пользоваться встроенными функциями. Например, на рисунке 3.30 показан процесс создания запроса о выборе работ, выполненных три года назад.

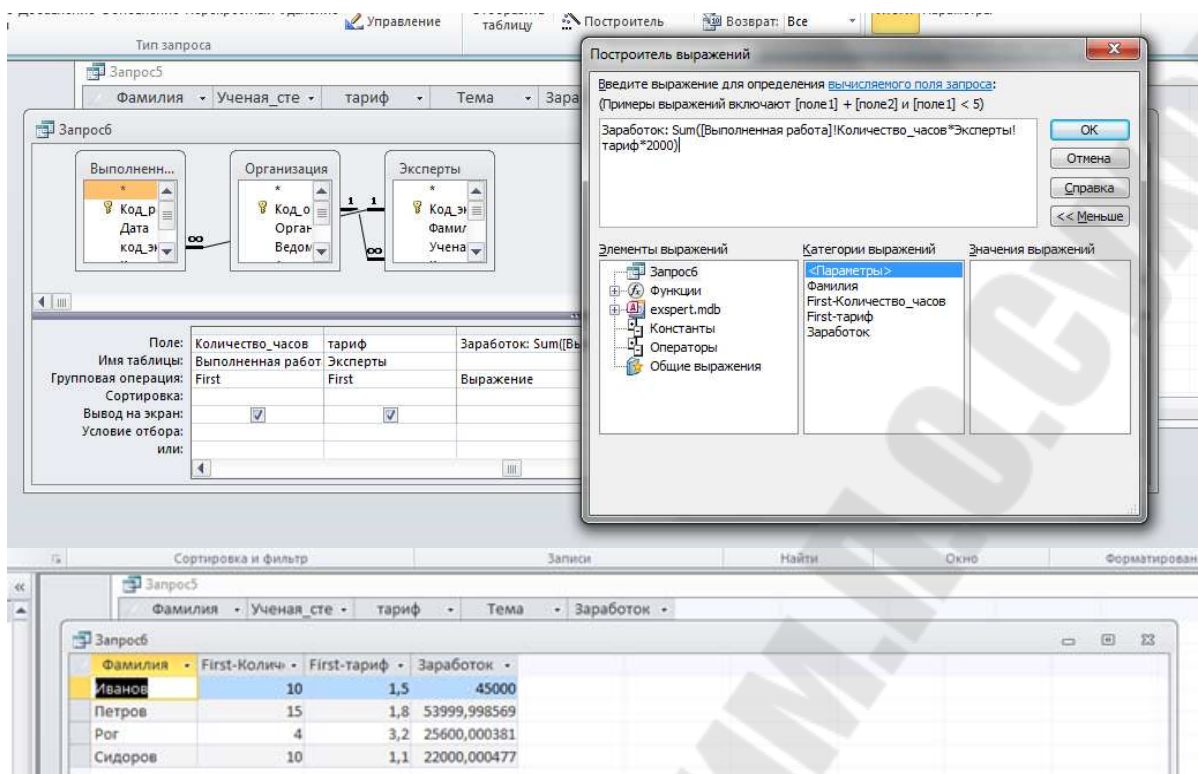


Рисунок 3.29 – Вычисление зарплаты

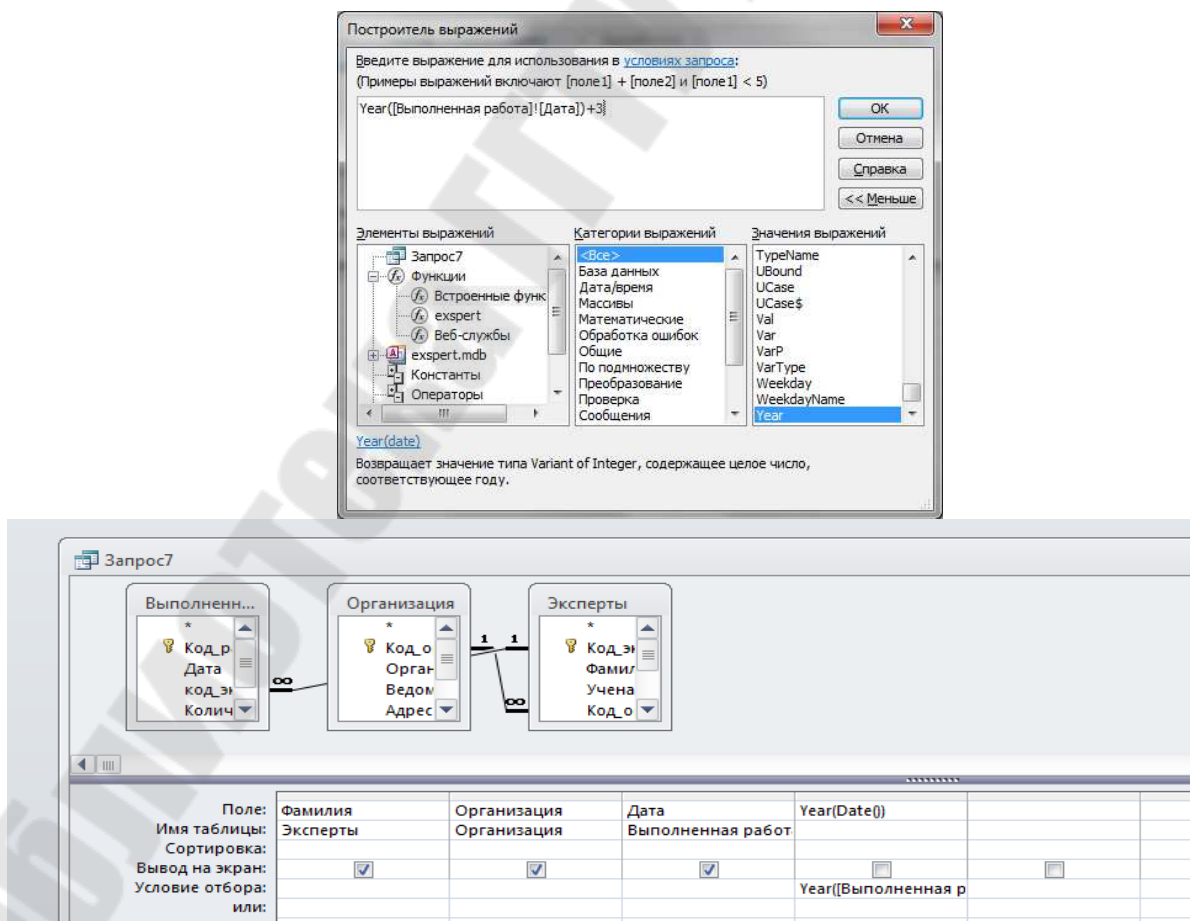


Рисунок 3.30 – Выполненная работа три года назад

Запросы действия. К запросам действия относятся запросы на обновление, добавление, удаление, создание таблицы.

Общая технология создания запроса действия.

1. Создать запрос на выборку, где в качестве условий внести выражения, отбирающие данные, предназначенные для обновления, удаления, добавления или формирования новой таблицы.

2. Выбрать тип запроса в пункте меню Запрос. Для запроса на обновление заполнить строку «Обновление», то есть задать выражения, которые будут заноситься в таблицу вместо старых (рисунок 3.31). Для запроса на добавление или формирование новой таблицы в открывшемся окне ввести имя таблицы, в которую будут добавляться данные, или имя создаваемой таблицы.

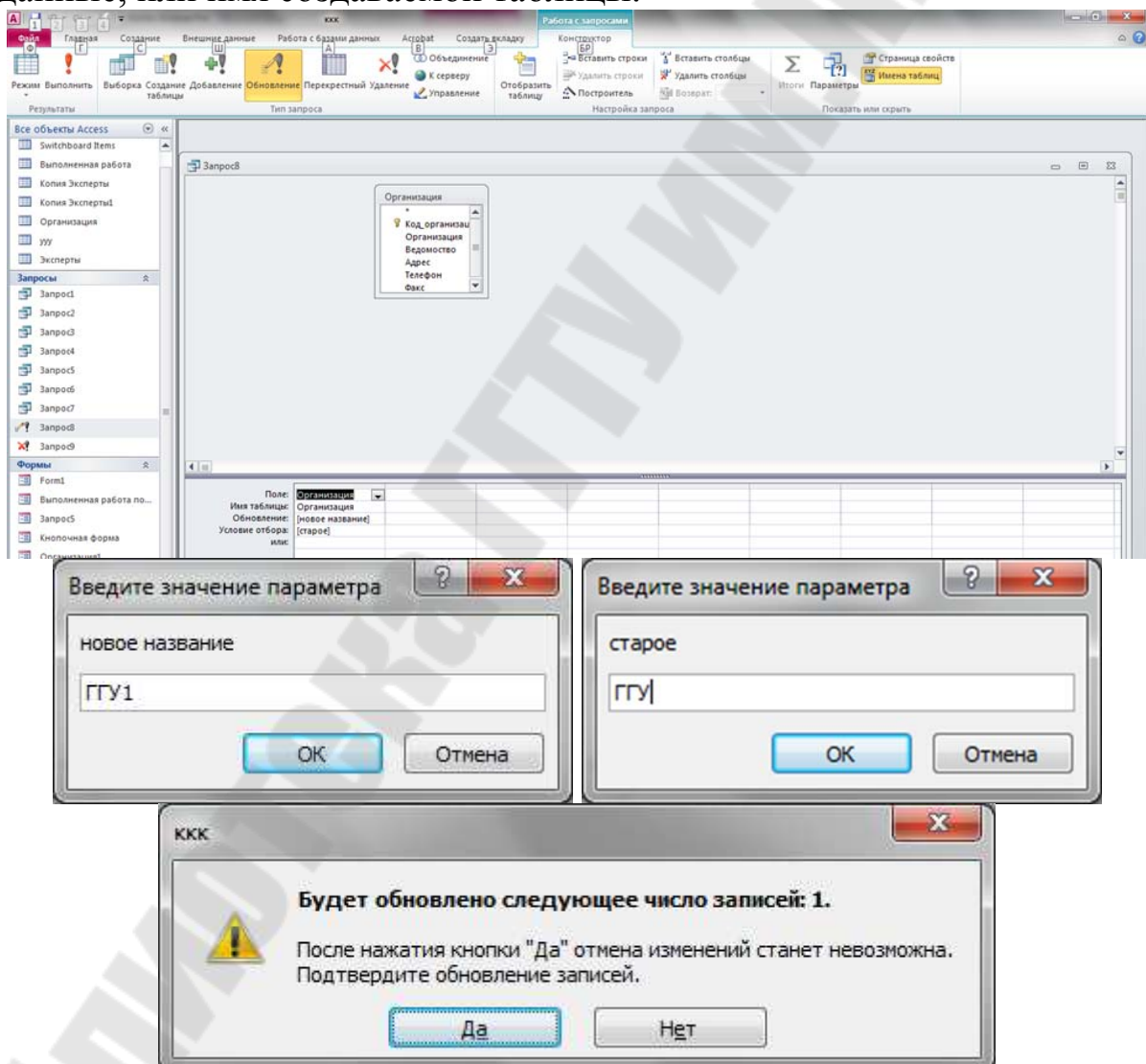


Рисунок 3.31 – Запрос на обновление

4 МЕТОДЫ И АЛГОРИТМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В САПР

4.1 Оптимизация технологических процессов в САПР ТП

4.1.1 Проектирования сложного технического объекта

При создании систем автоматизации проектирования (САПР) в основу общего подхода к процессу проектирования закладывается алгоритм, включающий три этапа (см рисунок 4.1):

- синтез;
- анализ;
- и принятие решения.

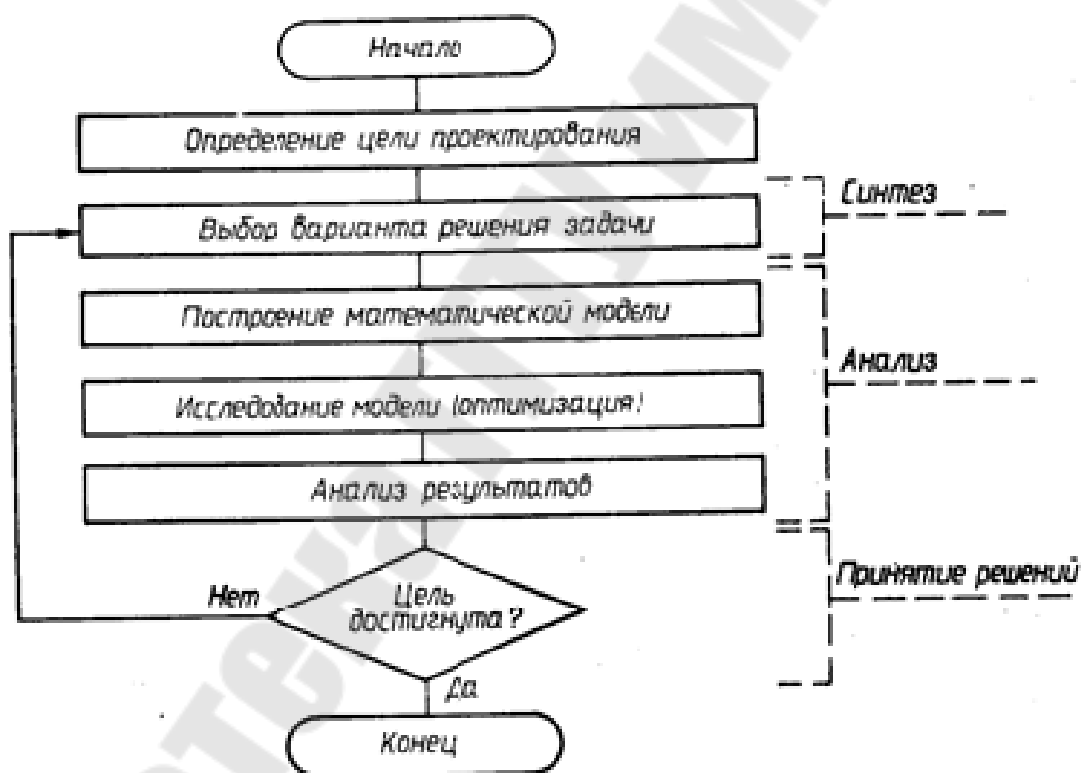


Рисунок 4.1 – Схема алгоритма процесса проектирования сложного технического объекта

Рассмотрим данный алгоритм. После определения цели проектирования происходит формирование (генерирование) возможных вариантов (альтернатив) решения проектной задачи. Этот этап называется *синтезом* и охватывает наиболее творческие виды работ по

созданию объекта. В современных САПР на этом этапе могут генерироваться принципиально новые технические решения.

Следующий этап процесса проектирования – *инженерный анализ* – направлен на детализацию намеченных вариантов, решение задачи (определение структуры и отдельных параметров проектируемого объекта) и проверку возможных условий функционирования объекта.

Определяющими видами работ на данном этапе являются математическое моделирование объекта и его исследование на основе этой модели с целью выявления основных функциональных свойств в рассматриваемой области.

Информация, полученная в результате анализа, позволяет перейти к третьему этапу процесса проектирования – *принятию решения*.

Это наиболее ответственный этап, цель которого – выявление единственного решения задачи среди возможных вариантов. На этом этапе наиболее универсальными являются многошаговые методы принятия решения, при которых каждый последующий шаг сужает область поиска и ограничивает число альтернатив.

Рассмотренная схема (рисунок 4.1) показывает, что основные задачи, решаемые на втором этапе – этапе анализа – связаны с оптимизацией технических объектов.

Оптимизация (от лат. optimum – наилучший) – это процесс нахождения экстремума некоторой количественной величины (параметра) проектируемого объекта, представляемой в виде функции (функционала).

Если эта функция характеризует положительное свойство объекта, то ищется максимальное ее значение, если отрицательное – то минимальное.

Обычно в инженерной практике используется термин «*оптимальное решение*», или «*оптимальный проект*», под которым в этом случае понимается наилучшее из некоторого множества решение, удовлетворяющее всем требованиям, предъявляемым к проектируемому объекту.

Теория оптимизации в современном представлении включает совокупность фундаментальных математических результатов и численных методов, ориентированных на нахождение наилучших вариантов из множества альтернатив и позволяющих избежать полного перебора и сравнения возможных вариантов.

Процесс оптимизации лежит в основе инженерной деятельности, направленной на проектирование новых, более эффективных и менее дорогостоящих технических объектов.

Достижение этих двух основных целей любого процесса проектирования сопряжено, как уже отмечалось, с синтезом различных элементов, анализом множества их состояний и выбором из них такого состояния, при котором обеспечиваются наилучшие показатели функционирования технического объекта.

Большинство используемых методов оптимизации являются по своей сути инвариантными и могут использоваться при решении различных проектных задач.

Поэтому в настоящее время разработаны десятки численных методов оптимизации, оформленных в виде стандартных процедур (алгоритмов) и хранящихся в библиотеках прикладных программ вычислительных центров, которые открыты для доступа различным пользователям.

В этих условиях перед проектировщиком встает задача правильного выбора метода и соответствующих наборов программ.

Так, при расчете оптимальных режимов резания могут использоваться стандартные программы симплекс-метода или других методов линейного программирования. Однако при этом достаточно сложной задачей является стыковка принятой математической модели объекта с соответствующими программами оптимизации.

Поиск оптимальных технических решений в технологии машиностроения затруднен в связи с низким уровнем формализации существующих методов проектирования технологических процессов и сложностью построения соответствующих математических моделей.

Поэтому главным вопросом оптимизации технологических процессов при создании САПР ТП является разработка математических моделей различных технологических объектов и их информационное обеспечение.

Сфера применения методов оптимизации в технологии машиностроения достаточно широка:

- проектирование отдельных структурных элементов технических систем, какими, например, являются режимы резания,
- проектирование более сложных структур, таких, как технологические маршруты и операции обработки,
- и, наконец, проектирование цехов и промышленных предприятий в целом.

4.1.2 Особенности построения структуры математических моделей технологических процессов

Формально технологический процесс можно представить как упорядоченное множество элементов структурной модели, каждый элемент которой выполняет определенную функцию (работу) и находится

- в конструктивной;
- функциональной;
- информационной связи с другими элементами.

В зависимости от поставленной задачи структурная модель процесса строится с той или иной степенью подробности (детализации).

Например, в ТП, выполняемом на поточной линии, при решении вопроса о распределении припусков на послеоперационную обработку элементами структуры могут быть сами станки.

Если исследуются качество и точность деталей на отдельных операциях, то в такой постановке станок расчленяется на соответствующие узлы и структурная модель состоит из этих узлов (элементов).

Таким образом, структурная модель отвечает характеру поставленной задачи.

Технологический процесс как сложная система функционирует в пространстве и во времени, т.е. в каждом элементе системы происходят определенные изменения, а также протекают физические, химические, кинетические и другие процессы, обусловленные назначением элементов в структуре и их взаимодействием.

Построение модели функционирования системы сводится к построению математических моделей процессов, которые, в конечном счете, выражаются дифференциальными, интегральными, алгебраическими и другими типами уравнений или какой-либо логической зависимостью.

Формоизменение заготовки происходит в процессе выполнения операций и переходов. Главная функция цели ТП может быть выражена в виде

$$\Phi: S_0 \rightarrow S_k = C_0,$$

где Φ – оператор формоизменения; S_0 – исходное состояние заготовки; S_k – конечное состояние готовой детали; C_0 – критерий оптимизации.

Пользуясь таким подходом, ТП можно описать математически в виде функционала функции Φ формоизменения обрабатываемой детали S , т.е.

$$F(\Phi, S) = C_0.$$

Весь процесс формообразования может быть представлен как переход из состояния заготовки S_0 в состояние детали S_k посредством выполнения совокупности некоторой последовательности операций.

В этом случае можно показать

$$\begin{aligned} F(\Phi, S) &= \Phi_1 \rightarrow \Phi_2 \rightarrow \Phi_3 \rightarrow \dots \rightarrow \Phi_k; \\ S_1 &\rightarrow S_2 \rightarrow S_3 \rightarrow \dots \rightarrow S_k. \end{aligned}$$

Главным требованием, предъявляемым к математической модели любого объекта, и в частности к ТП, является адекватность отражения модели реальным свойствам объекта.

Математическая модель выражается математическими зависимостями, представляющими собой определенные соотношения между отдельными параметрами, описывающими данный объект, а также множеством ограничений, накладываемых на эти параметры и выражаемых в виде уравнений и неравенств.

Совпадение математической модели процесса с фактическим процессом зависит от квалификации проектировщика и уровня его математической подготовки.

При решении задач оптимизации, возникающих на разных этапах технологического проектирования, используются различные виды математических моделей и методов оптимизации.

Математические модели *по структуре* делятся на две группы:

- без ограничений;
- и с ограничениями, которые могут задаваться как линейными, так и нелинейными функциями.

По виду переменных различают математические модели

- с непрерывными значениями переменных;
- и дискретными значениями переменных.

В последней группе бывают дискретные целочисленные и нецелочисленные.

Методы оптимизации *в зависимости от вида функции цели и ограничений* подразделяются:

- на классический метод дифференцирования,
- линейное,

- квадратичное,
- выпуклое,
- и динамическое программирование.

С точки зрения стратегии поиска оптимума выделяют четыре группы методов оптимизации:

- аналитические;
- рекурсивные;
- итерационные;
- стохастические.

Аналитические методы находят применение при решении классических задач и задач с ограничениями в виде уравнений.

Для решения задач без ограничений используют методы исследования производной функции. Путем приравнивания производной нулю отыскиваются точки экстремума, а затем исследуются точки с помощью второй производной для отыскания максимума.

Таким способом решаются простые технологические задачи, например, выполняется расчет режимов резания, выбор параметров режущего инструмента и др.

Рекурсивные методы относятся к методам, позволяющим определить одну переменную за одну расчетную операцию.

Решение всей задачи осуществляется путем поочередного определения переменных.

Наиболее распространенным среди этих методов является динамическое программирование. Этот метод можно использовать при анализе многоэтапных процессов принятия решения, например, при оптимизации маршрутных ТП.

Однако метод динамического программирования эффективен при небольшом числе ограничений, вводимых в математическую модель, поэтому он пока не получил широкого распространения при решении технологических задач.

Итерационные методы объединяют наибольшую группу методов поиска оптимумов. К ним относятся способы расчета функции цели в одной или нескольких вероятностных точках для определения «лучшей» точки.

Расчет выполняют до тех пор, пока не приблизятся к назначенному критерию на расстояние, меньшее некоторого заданного значения.

Эти методы позволяют устанавливать только локальные оптимумы, однако они могут применяться в случаях, когда оптимизацию проводят в различных исходных точках.

Оптимумы, определяемые этим способом, представляют собой достаточно точное решение относительно абсолютного оптимума.

Различают два больших класса итерационных методов:

- методы линейного программирования;
- нелинейного программирования.

Линейное программирование применяют для решения линейных задач, когда функции цели и ограничения являются линейными, а все переменные – непрерывными функциями.

В основу этого программирования положено утверждение, что точка оптимума целевой функции находится в одной из вершин выпуклого многогранника определяющего область возможных решений.

Наиболее известным итерационным методом решения линейных задач является симплекс-метод.

Для методов *нелинейного программирования* характерно непосредственное отыскание оптимума. Эти методы разделяются на две группы:

- методы, базирующиеся на расчетах градиентов,
- методы, при использовании которых этот расчет не требуется.

К первой группе относится метод наискорейшего спуска, а ко второй – метод Фибоначчи, основанный на отыскании оптимума вдоль произвольно выбранного направления.

Все методы непосредственного поиска оптимума включают операции выбора направления поиска и длины шага.

Отдельные методы имеют разные критерии выбора этих двух параметров.

Большинство методов непосредственного отыскания оптимума не может быть применено к математическим моделям с ограничениями. В этом случае предварительно необходимо привести математическую модель с ограничениями к модели без ограничений. Для этой цели используются специальные математические методы: метод штрафных функций, метод множителей Лагранжа.

Стохастические методы оптимизации (методы случайного поиска решений) включают процедуры накопления и обработки информации, в которые сознательно вводится элемент случайности.

Преимущества этих методов заключаются в их простоте, надежности, достаточной точности и легкости программирования.

В результате методы случайного поиска стали одними из наиболее эффективных методов оптимизации.

Стохастические методы оптимизации применяются для различных задач технологического проектирования процессов изготовления деталей при наличии большого числа случайных факторов, которые не представляется возможным описать в традиционной математической форме.

4.1.3 Обоснование и выбор критериев оптимальности

4.1.3.1 Виды критериев оптимальности

При разработке оптимального технологического процесса наиболее важным является обоснование цели и оценка эффективности технологических операций или ее отдельных элементов, например режимов резания.

Под основной целью технологического процесса или операции в машиностроении обычно понимается обеспечение заданных характеристик качества изделия наиболее производительным путем при минимальных затратах.

В этом случае оптимальность операции можно определить как меру ее соответствия поставленной цели. Чем эффективнее операция, тем выше ее производительность и экономичность. То же можно сказать и о технологическом процессе в целом.

В задачах, которые встречаются в условиях оптимизации ТП, критерии оптимальности могут быть различными, однако все они должны удовлетворять определенным требованиям:

- обладать достаточной полнотой описания объекта;
- иметь определенный физический смысл;
- быть количественными и выражаться однозначно некоторым числом;
- иметь простой математический вид;
- определяться с допустимой точностью.

В зависимости от вида и уровня задач оптимизации (расчет режимов резания, проектирование операции и технологического процесса или оценка работы предприятия в целом) основные используемые критерии оптимальности можно подразделить на следующие виды:

1) *экономические:*

- минимальная себестоимость;
- наименьшие народнохозяйственные приведенные затраты;
- наименьшие приведенные хозрасчетные затраты;
- наибольшая прибыль;
- рентабельность;
- минимальный уровень затрат на производство (минимальные затраты на электрическую и другие виды энергии, на основные и вспомогательные материалы, на фонд заработной платы и др.);

2) *технико-экономические:*

- максимальная производительность;
- наименьшее штучное время;
- основное и вспомогательное время;
- коэффициент полезного действия оборудования;
- надежность работы системы оборудования или отдельных ее элементов;
- станкоемкость изделия;
- стабильность технологического процесса обработки;

3) *технологические:*

- точность изготовления изделия,
- показатели качества поверхности изделия (шероховатость, волнистость, микротвердость, остаточные напряжения и др.);
- физико-химические свойства изделий;
- стойкость инструмента;

4) *эксплуатационные:*

- износостойкость;
- усталостная прочность;
- контактная жесткость и
- другие показатели долговечности изделий;

5) *прочие:* психологические; эстетические; эргономические.

Наибольшее распространение при решении задач оптимизации технологического проектирования получили экономические и технико-экономические критерии оптимальности.

Это связано с тем, что в основе разработки любого ТП или решения более частной задачи, например расчета режимов резания, лежат два принципа: технический и экономический.

В соответствии с первым принципом технологический процесс должен гарантировать выполнение всех требований на изготовление изделия.

Второй принцип определяет условия, обеспечивающие минимальные затраты труда и наименьшие издержки производства.

Первый принцип наиболее полно отражается минимальной себестоимостью из группы экономических критериев, а второй – максимальной производительностью из группы технико-экономических критериев.

Технологические и эксплуатационные критерии оптимальности используются при обеспечении требуемого качества наиболее ответственных изделий (точности, качества поверхности, физико-химических свойств и др.), а также эксплуатационных свойств отдельных деталей, определяющих надежность и долговечность машин.

4.1.3.2 Построение критериев максимальной производительности и наименьшего штучного времени

Одним из широко применяемых критериев оптимальности для решения технологических задач и определения режимов резания, в частности, является максимальная производительность.

Производительностью рабочей машины называется количество обрабатываемого продукта за единицу рабочего времени.

Штучная производительность (шт/мин) на операции определяется величиной, обратной штучно-калькуляционному времени на эту операцию

$$P = (t_{um.-к})^{-1}. \quad (4.1)$$

Как известно, время выполнения операции $t_{um.-к}$ состоит из штучного времени t_{um} и подготовительно-заключительного $T_{n.-3}$ на всю партию деталей N :

$$t_{um.-к} = t_{um} + T_{n.-3} / N.$$

Составными элементами штучного времени являются: основное t_o и вспомогательное t_e время, время перерывов на отдых и естественные надобности $t_{омд}$ и время обслуживания рабочего места $t_{обс}$.

Анализ элементов штучного времени t_{um} показывает, что от режимов резания зависят t_o (или можно принять машинное t_m) и часть времени $t_{обс}$, затрачиваемого на смену и подналадку инструмента:

$$t_{обс} = t'_{обс} + t_{см}$$

Тогда формула (4.1) примет вид

$$\Pi = [(t_m + t_{cm}) + (t_g + t'_{обс} + t_{отд} + T_{н-з}/N)]^{-1} = [t_{шт.-к.р.} + t_{шт.-к.н.}], \quad (4.2)$$

где $t_{шт.-к.р.}$ и $t_{шт.-к.н.}$ – часть штучно-калькуляционного времени, соответственно зависящего и не зависящего от режимов резания. Таким образом, штучная производительность, зависящая от режимов резания, определяется только величиной

$$t_{шт.р.} = t_m + t_{cm}. \quad (4.3)$$

Машинное время в общем виде

$$t_m = t_p + t_x,$$

где t_p – время резания, мин; t_x – время холостого хода, мин. Тогда, если ввести коэффициент резания $\lambda = L_D/L$, представляющий отношение длины детали L_D к длине рабочего хода L , получим

$$t_m = \lambda t_p. \quad (4.4)$$

Для наиболее распространенных методов обработки металлов резанием (точение, сверление, фрезерование) величина t_p может быть найдена по формуле,

$$t_p = \frac{L}{ns} i = \frac{Lh}{nst} \quad (4.5)$$

где n – частота вращения заготовки, об/мин; s – подача, мм/об; t – глубина резания, мм; L – длина обрабатываемой поверхности, мм; h – величина припусков, мм; i – число проходов.

Время смены и подналадки инструмента, приведенной к одной детали,

$$t_{cm} = T_{cm} t_p / T, \quad (4.6)$$

где T_{cm} – время, затрачиваемое на каждую смену инструмента, мин; T – период стойкости инструмента, мин.

Подставив формулы (4.4), (4.5) и (4.6) в формулу (4.3) и введя обозначение

$$k = \frac{1000v}{\pi D},$$

получим

$$t_{шт.р.} = \frac{\pi D L h}{1000 v s t} \left(\lambda + \frac{T_{cm}}{T} \right). \quad (4.7)$$

Эта формула не может в таком виде использоваться для оптимизации режимов резания, так как в нее входит переменная величина –

стойкость инструмента T , зависящая от этих режимов согласно известной формуле (4.8) для определения скорости резания v :

$$v = \frac{C_v}{T^{1/m} \cdot s^{1/n} \cdot K} \quad (4.8)$$

После подстановки этой зависимости в формулу (4.7) и соответствующих преобразований получим окончательное выражение для критерия оптимальности *минимальное штучное время*

$$t_{шт} = \frac{\pi D L k}{1000 v s t} \left(1 + \frac{T_{ин} v^{1/m} s^{1/n} K^m}{C_v^m} \right) \quad (4.9)$$

Анализ формулы (4.9) показывает, что в наиболее общем виде этот критерий по отношению к оптимизируемым параметрам v , s и t является нелинейным.

4.1.3.3 Построение критерия минимальной себестоимости

Показатель «минимальная себестоимость» охватывает широкий круг затрат общественного труда и наряду с затратами живого труда учитывает затраты прошлого труда, овеществленного в средствах производства (амортизация и ремонт оборудования, энергия, вспомогательные материалы, измерительный инструмент, помещения).

Цеховая себестоимость технологической операции без учета затрат на заготовку определяется следующим образом:

$$C_{оп} = C_{з.с} + C_a + C_{рем} + C_{эн} + C_в + C_{пр} + C_{ин} + C_n,$$

где $C_{з.с}$ – заработная плата станочников (с начислением в фонд социального страхования); C_a – амортизационные отчисления на замену станка; $C_{рем}$, $C_{эн}$, $C_в$, $C_{пр}$, $C_{ин}$, C_n – затраты соответственно на ремонт станка, силовую электроэнергию, вспомогательные материалы, амортизацию и ремонт универсальных приспособлений, амортизацию, ремонт и заточку универсальных режущих инструментов, а также затраты, связанные с использованием помещения.

Процесс обработки заготовки резанием характеризуется двумя факторами: временем $t_{шт}$, требуемым для обработки, и зависящей от него себестоимостью. С изменением режимов обработки эти факторы изменяются, однако простой зависимости между ними не существует.

Это связано с тем, что изменение каждого фактора зависит от различных параметров, и в первую очередь от изнашивания инструмента.

Изменение условий обработки в направлении повышения производительности может привести к настолько большим инструментальным расходам, что они превысят сокращение расходов, достигнутых за счет уменьшения времени обработки. И наоборот, недостаточное использование режущих способностей инструмента может резко повысить общие расходы (суммарную себестоимость обработки), связанные с временным фактором.

Характер влияния различных затрат на себестоимость операции показан на рисунке 4.2.

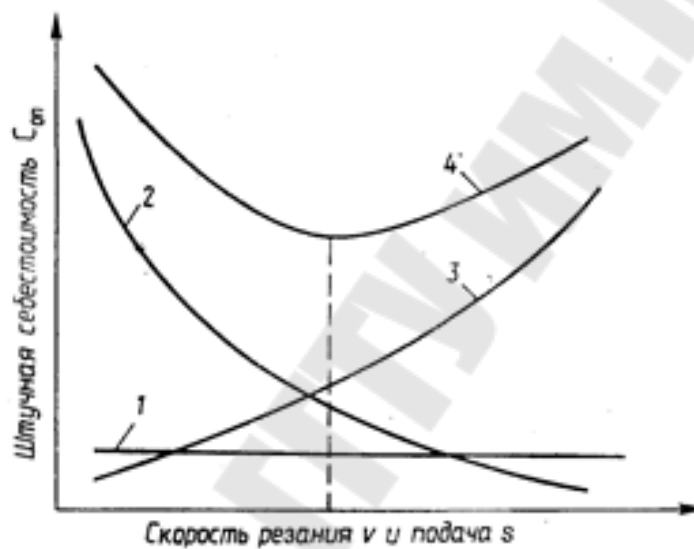


Рисунок 4.2 – Зависимости себестоимости обработки от скорости резания v и подачи s

Суммарная себестоимость обработки (кривая 4) включает три вида затрат:

1) затраты, не зависящие от режимов резания (затраты на вспомогательное время и приходящаяся на заготовку часть подготовительно-заключительного времени $T_{п.-з./N}$);

2) затраты, пропорциональные времени обработки, которые уменьшаются с сокращением машинного времени (все затраты, которые снижаются с уменьшением t_o , за исключением затрат на инструмент);

3) затраты, пропорциональные производительности обработки (инструментальные затраты, зависящие от скорости, подачи, глубины резания, а также инструментального и обрабатываемого материалов).

Кривая суммарной себестоимости обработки имеет минимум, положение которого зависит от характера кривых 2 и 3. В большинстве случаев эта кривая имеет характер гиперболы, зависящей от стоимости станко-минуты и стоимости амортизации рабочего места.

При определении технологической себестоимости операции могут использоваться

- бухгалтерский метод расчета;
- метод определения себестоимости станко-часа;
- и метод поэлементного расчета.

Последний является наиболее точным методом и используется в дальнейших расчетах.

Элементы технологической себестоимости можно условно разделить на две группы,

- одна из которых не зависит ($C_{рем}, C_в$);
- а вторая зависит от режимов резания ($C_{з.с}, C_а, C_{ин}, C_{ин}, C_{эн}, C_n$).

Вторая группа элементов себестоимости операции пропорциональна штучному времени, поэтому ее целесообразно привести к 1 мин работы оборудования ($C'_{з.с}, C'_а, C'_{пр}, C'_{эн}, C'_n$).

Так, затраты на заработную плату станочника

$$C_{з.с} = Z_{мин} K_T t_{шт.р.},$$

где $Z_{мин}$ – тарифная ставка 1-го разряда данной категории рабочих, коп/мин; K_T – тарифный коэффициент; $t_{шт.р.}$ – часть штучно-калькуляционного времени, зависящая от режимов резания, мин.

Произведя замену в рассматриваемой формуле, можно получить

$$C_{з.с} = C'_{з.с} t_{шт.р.}$$

Аналогично могут быть найдены затраты, определяющие элементы себестоимости по амортизации станка $C_а$, эксплуатации приспособлений $C_{пр}$, амортизации помещений C_n и силовой электроэнергии $C_{эн}$, приведенных к 1 мин работы оборудования:

$$C_а = C'_а t_{шт.р.}; \quad C_{пр} = C'_{пр} t_{шт.р.}; \quad C_n = C'_n t_{шт.р.}; \\ C_{эн} = C'_{эн} t_{шт.р.}$$

В этом случае себестоимость операции, зависящая от режимов резания, может быть определена по зависимости

$$C_{on.пер} = C'_{з.с}t_{ум.р} + C'_a t_{ум.р} + C'_{пр}t_{ум.р} + C_{ум.р} + C'_n t_{ум.р} + C_{ин}$$

где $C_{ин}$ – инструментальные расходы, приведенные к одной заготовке.

Введя обозначение $R = C'_{з.с} + C'_a + C'_{пр} + C'_{эн} + C'_n$, определим положение минимума кривой штучной себестоимости $C_{on.пер}$ (рисунок 4.2) расчетным путем, используя выражение

$$C_{on.пер} = Rt_{ум.р} + C_{ин}$$

Инструментальные расходы на одну деталь можно определить по формуле

$$C_{ин} = M/q,$$

где M – инструментальные расходы, приведенные к одному периоду стойкости; q – число деталей, обрабатываемых за период стойкости. При этом

$$q = \frac{T}{t_p}, \quad (4.10)$$

где t_p – время резания инструментом каждой заготовки, мин.

Инструментальные расходы, приведенные к одному периоду стойкости:

$$M = S_{ин}/n_T + C'_{пер} + C'_{зам.ин},$$

где $S_{ин}$ – покупная (начальная) стоимость инструмента; $C'_{пер}$ – стоимость переточки инструмента, приведенная к одному периоду стойкости; $C'_{зам.ин}$ – стоимость замены затупившегося инструмента, приведенная к одному периоду стойкости; n_T – число периодов стойкости.

После подстановки всех зависимостей

$$C_{on.пер} = Rt_{ум.р} + Mt_p/T \quad (4.11)$$

Учитывая, что $t_{ум.р} = t_m + t_{см}$, а $t_m = t_p \lambda$, и введя зависимость (4.5) для времени резания t_p , получим

$$C_{on.пер} = \frac{Lh}{nst} jR + \frac{T_{сст}}{T} \frac{LhR}{nst} + \frac{1}{T} M \frac{Lh}{nst} \quad (4.12)$$

Для получения окончательного вида оценочной функции «минимальная себестоимость» производится замена переменных n и T их зависимостями от скорости резания v , и после несложных преобразований имеем

$$C_{\text{оп.пер}} = \frac{\pi DLk}{10^3 \text{ust}} \left(2R + \frac{(RT_{\text{ср}} + 1)v^1 \cdot \dots \cdot \dots}{C_1^1} \right) \quad (4.13)$$

Как видно из этой формулы, вид критерия оптимальности по параметрам v , s и t является нелинейным.

Это обстоятельство затрудняет разработку методов оптимизации, поэтому в частных случаях стремятся упростить вид критерия оптимальности, что, естественно, вносит определенные погрешности в получаемые результаты.

Таким примером является широко используемый метод линейного программирования для расчета оптимальных режимов резания v и s .

В этом случае критерий «минимальная себестоимость» рассматривается в виде $C_{\text{оп.пер}} = C_1/ns$, где C_1 – некоторая постоянная, характеризующая условие обработки.

4.1.3.4 Построение обобщенных критериев оптимальности

При решении задач оптимизации процессов механической обработки часто возникает необходимость одновременного достижения нескольких противоречащих друг другу целей.

Принимая решения, улучшающие оценки одного критерия, например минимальной себестоимости операции, мы ухудшаем тем самым оценки по другим критериям, например наибольшей производительности и др.

В таких случаях возникает задача оценки и сравнения различных проектных решений при так называемом *векторном критерии эффективности*.

С этой целью используют обобщенные критерии, которые являются скалярными функциями частных критериев и учитывают степень достижения всех целей в совокупности, отражая относительную значимость каждого критерия в отдельности.

Поскольку каждый из частных критериев является фактически функцией управляемых переменных, то и обобщенный критерий в

свою очередь можно рассматривать как некоторую функцию управляемых переменных.

Эту функцию, как отмечалось выше, обычно называют *целевой*.

При таком подходе, называемом *свертыванием векторного критерия*, задача сравнения решений по векторному критерию фактически заменяется задачей выбора способа свертывания и определения значения коэффициентов, участвующих в этом свертывании.

Существуют следующие виды обобщенных (свернутых) критериев.

Аддитивный критерий. В этом случае в качестве обобщенного критерия берется «взвешенная» сумма частных критериев

$$F(x) = \sum_{j=1}^J \alpha_j K_j(x)$$

где x – управляемые переменные; α_j – неотрицательные коэффициенты, значения которых выбираются, исходя из степени важности отдельных целей, и определяются на основании ранее решенных аналогичных задач или методом проб.

В последнем случае «весовые» коэффициенты подбираются при анализе результатов, получаемых при различных значениях этих коэффициентов, α_j является неуправляемой переменной.

Мультипликативный критерий. В отличие от предыдущего в данном случае в качестве обобщенного критерия берется «взвешенное» произведение частных критериев

$$F(x) = \prod_{j=1}^J (K_j(x))^{\alpha_j}$$

Конъюнктивный критерий. По этому критерию оценивается каждое решение с точки зрения цели, степень достижения которой (с учетом «весового» коэффициента) в данном случае наименьшая:

$$F(x) = \min_{j \in J} \alpha_j K_j(x)$$

Дизъюнктивный критерий. Противоположен предыдущему и оценивает решения с точки зрения цели, степень достижения которой (также с учетом «весового» коэффициента) в данном случае максимальна:

$$F(x) = \max_{j \in J} \alpha_j K_j(x)$$

Кроме рассмотренных, могут использоваться и другие способы свертывания критериев.

Например, способ выделения наиболее важного критерия предусматривает определение из набора частных критериев оптимальности одного, который принимается за обобщенный как критерий допустимости.

Для выбора наиболее важного критерия может быть рекомендован метод последовательных уступок, предусматривающий упорядочивание всех количественных целей в порядке убывания их значимости.

4.1.4 Выбор технических ограничений

Оптимизация технологических процессов зависит также от правильного выбора технических ограничений, которые определяют область существования оптимальных решений. Следует иметь в виду, что не может быть речи о каком-либо оптимальном ТП в общем смысле, поэтому поиск оптимального технологического процесса должен быть ограничен определенными технологическими условиями.

Чем точнее сформулированы ограничения, вытекающие из производственных условий, тем меньше вариантов процесса, рассматриваемого в качестве основы для выбора оптимального решения.

В общем виде все параметры (величины, характеризующие элементы процесса обработки), определяющие состояние объекта в произвольный момент времени, могут быть представлены в виде следующих векторов:

1) вектора входных и возмущающих параметров $V = (v_1, v_2, \dots, v_p)$.

К входным параметрам относятся неуправляемые переменные, связанные с объектом обработки и состоянием оборудования.

Возмущающие параметры связаны с проявлением случайных величин, характеризующих изменения неконтролируемых характеристик заготовки или внешней среды;

2) вектора технологических параметров $X = (x_1, x_2, \dots, x_n)$.

Компоненты этого вектора являются управляемыми переменными, позволяющими выбирать необходимые условия обработки. Обязательным условием технологических параметров является возможность их управления и контроля;

3) вектора выходных параметров $Y = (y_1, y_2, \dots, y_m)$.

Выходные параметры, ранее названные производными переменными, определяют основные характеристики качества продукции

и технико-экономические показатели, связанные с рассматриваемым процессом.

Значение каждого из рассмотренных параметров находится в определенном интервале, задаваемом физической природой данного параметра или требованиями к технологическому процессу, поэтому группа ограничений, связанная с диапазоном варьирования параметров, может быть представлена в виде следующей совокупности неравенств:

$$v_{k \min} \leq v_k \leq v_{k \max},$$

$$k = \overline{1, p}$$

$$x_{i \min} \leq x_i \leq x_{i \max},$$

$$i = \overline{1, q}$$

$$y_{j \min} \leq y_j \leq y_{j \max}$$

$$j = \overline{1, m}$$

Решение задач технологического проектирования связано с учетом большого числа факторов, определяющих выполнение какого-либо процесса обработки. Причем в каждой задаче требуется учитывать определенную группу факторов, в наибольшей степени влияющую на принимаемые решения.

Анализ процессов механической обработки показывает, что в большинстве случаев требуется учитывать пять основных групп факторов.

Первая группа факторов характеризует объект обработки (заготовку). Ее составляют:

- вид материала, твердость и другие механические свойства,
- способ получения заготовки, ее размеры, масса,
- точность размеров,
- шероховатость поверхности.

Ко второй группе факторов относятся основные параметры орудий труда (станка, приспособления, инструмента):

- вид, кинематика и динамика станка,
- жесткость, прочность и точность отдельных элементов и системы в целом.

Третью группу факторов составляют выбираемые параметры инструмента для исследуемого процесса обработки:

- физико-механические свойства материала режущей части инструмента,
- геометрические параметры его заготовки,
- размеры и точность,

- степень изнашивания,
- шероховатость поверхности,
- зернистость и вид связки,
- стойкость инструмента.

Четвертая группа факторов характеризует процесс механической обработки. Сюда относятся

- время и глубина обработки,
- скорость,
- подача,
- число проходов,
- усилие резания,
- давление, вид и способ подачи технологических сред.

Пятая группа факторов – это технико-экономические показатели:

- расход и износ инструмента и станка,
- производительность и себестоимость обработки,
- качество изделия (точность, шероховатость поверхности и физико-химические свойства),
- вид и форма детали.

Вектор входных параметров V объединяет первую и вторую группы векторов.

Вектор технологических параметров X формируется из третьей и четвертой групп факторов, а вектор выходных параметров U включает пятую группу факторов.

Выбор необходимого числа параметров обработки связан с требуемой точностью описания математической модели процесса обработки и структурным уровнем отыскиваемых проектных решений. Так, при проектировании маршрута обработки в качестве технических ограничений учитываются вид и материал заготовки, вид и форма детали, ее точность, шероховатость поверхности и физико-химические свойства, вид станка, традиционная форма обработки на заводе, серийность и др.

Наиболее полно описывается математическая модель процесса обработки при выборе оптимальных режимов резания, точность получения которых во многом зависит от числа и достоверности описания технических ограничений.

Большим недостатком используемых в качестве технических ограничений стойкостных и силовых зависимостей для расчета режимов резания является недостаточно высокая точность.

В справочной литературе по резанию металлов отсутствуют данные по диапазонам, в которых эти зависимости справедливы. Хотя известно, например, что зависимости для определения стойкости инструмента и усилия резания применимы в довольно узком интервале скоростей резания.

Поэтому в настоящее время необходимо проводить исследования и обобщать имеющийся материал по созданию точных функциональных зависимостей, отражающий во всей полноте процесс резания металлов.

Причем в основу этих работ должны быть положены результаты современных теоретических и экспериментальных исследований по изучению тепловых явлений и напряженного состояния в зоне резания.

При этом выполняемые исследования закономерностей протекания различных видов ТП механической обработки должны быть направлены на установление количественных зависимостей, позволяющих обоснованно выбирать технические ограничения при создании математических моделей оптимизации решения технологических задач.

4.1.5 Виды оптимизации технологических процессов

Задача оптимизации ТП является комплексной и требует проведения анализа и выбора технологических решений на различных уровнях проектирования. Она обеспечивает минимальные значения приведенных затрат с одновременным соблюдением ряда технических ограничений.

При комплексном подходе следует различать два вида оптимизации ТП, выполняемых на различных этапах технологического проектирования (таблица 4.11):

1) *структурную оптимизацию* – выбор оптимального технологического маршрута, операции, перехода, вида и методов изготовления заготовки, способов базирования, оборудования, приспособлений, инструмента и т. д.;

2) *параметрическую оптимизацию* – выбор оптимальных параметров: допусков на межоперационные размеры, припусков, режимов резания, геометрических размеров режущего инструмента и др.

Комплексный подход к оптимизации усложняет решение задачи. Так, при параметрической оптимизации необходимо иметь реше-

ние о выборе структуры соответствующего уровня. В то же время структурная оптимизация требует знания значений параметров, входящих в соответствующую структуру. Это противоречие может быть устранено при построении алгоритмов оптимизации технологических процессов за несколько итераций.

Таблица 4.1 – Виды оптимизации на различных этапах проектирования технологических процессов

Этапы проектирования ТП	Оптимизация	
	структурная	параметрическая
1. Выбор заготовки и методов ее изготовления	+	–
2. Выбор технологических баз	+	–
3. Составление технологического маршрута обработки	+	–
4. Разработка технологических операций	+	+
5. Нормирование ТП	–	+
6. Расчет экономической эффективности ТП	+	+

С точки зрения структурного описания уровней ТП различают этапы проектирования маршрута, операции и переходов. Здесь возможны два подхода к построению принципиальной схемы технологического процесса:

- 1) маршрут→операция→переход;
- 2) переход→операция→маршрут.

В первом случае производится последовательный синтез сначала вариантов принципиальных схем обработки, а затем вариантов маршрута и операции. На каждом последующем этапе решения предыдущего этапа детализируются (как правило, в нескольких вариантах).

Второй подход основан на анализе отдельных поверхностей и проектировании переходов их обработки. Далее переходы упорядочиваются в операции, а операции упорядочиваются в маршрут обработки детали.

Главной особенностью оптимизации технических решений в рассмотренных подходах является необходимость использования на всех уровнях различных критериев оптимальности.

Анализ этих критериев показывает, что с точки зрения согласования оптимальных решений разных уровней предпочтительнее разработка процессов от наиболее общих вопросов к их детализации, что больше свойственно первому подходу.

При этом возникает задача получения оптимального решения при проектировании ТП в целом за счет оптимизации отдельных технологических решений на всех уровнях проектирования.

Для реализации рассматриваемого процесса проектирования в САПР ТП используется итерационный многоуровневый процесс оптимизации, содержанием которого является многократное повторение процедур анализа, синтеза и оценки.

Анализ исходных данных, условий и ограничений позволяет установить границы области возможных технологических решений.

С помощью процедур синтеза получают технологические решения, допустимые по совокупности граничных условий. Лучшие по некоторому критерию решения отбираются процедурами оценки.

4.1.6 Структурная оптимизация ТП

Принципиальное отличие структурной оптимизации от параметрической оптимизации состоит в сущности оптимизируемых параметров.

При структурной оптимизации они по своей природе являются неупорядоченными переменными.

В параметрической оптимизации параметры представляют собой переменные, для которых существует понятие больше или меньше и которые естественным образом могут быть размещены в координатной системе.

В структурной же оптимизации эти параметры не являются по существу числовыми.

Параметрами структурной оптимизации являются, например, модели станков, типы инструментов, схемы базирования, т.е. варианты типовых решений.

Структурная оптимизация рассматривает последовательно каждую задачу технологического проектирования.

Таким образом, весь процесс проектирования расчленяется на несколько взаимосвязанных уровней.

Процесс проектирования на каждом уровне представляет собой многовариантную процедуру.

В результате проектирования на всех уровнях образуется граф допустимых вариантов ТП, отвечающих заданным ограничениям – рисунок 4.3.

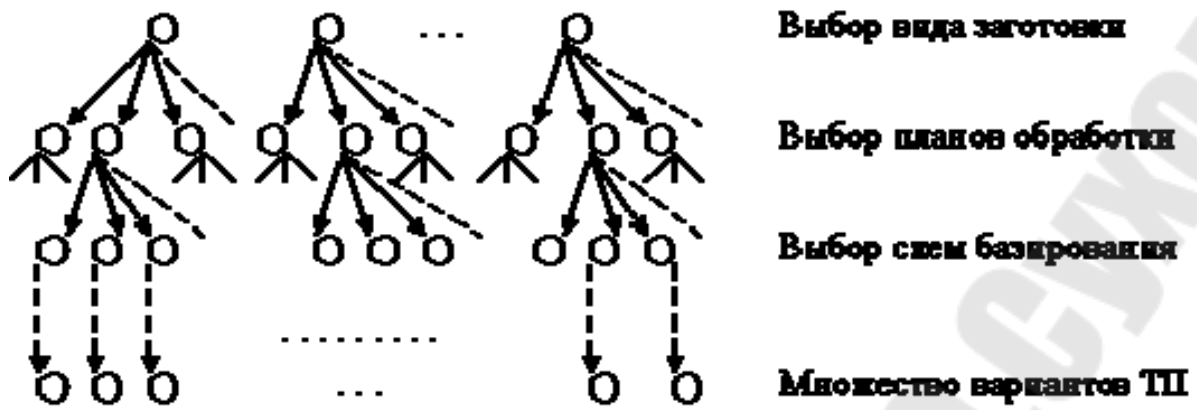


Рисунок 4.3 – Граф допустимых вариантов ТП

Задача структурной оптимизации состоит в поиске ветви графа, обеспечивающей экстремум целевой функции.

В силу неупорядоченности параметров основной метод структурной оптимизации состоит в последовательном переборе возможных вариантов. Чтобы выбрать один оптимальный вариант, необходимо до конца спроектировать очень большое количество допустимых техническими и технологическими ограничениями вариантов ТП.

Для реального ТП изготовления деталей даже средней сложности таких вариантов может быть огромное множество. Перебор всех вариантов даже при помощи современных быстродействующих компьютеров занимает очень большое время. Для уменьшения времени проектирования используются следующие приемы.

Прием 1. Эффективность процесса проектирования можно резко повысить, если организовать отбор рациональных вариантов проектных решений на каждом уровне проектирования.

Однако при этом возникает проблема формирования критериев промежуточного отбора наиболее рациональных вариантов на различных уровнях.

Например, на уровне (этапе) выбора заготовки анализ вариантов можно производить по критерию «себестоимость заготовки». Данный критерий можно достоверно рассчитать на этом этапе.

Но указанный критерий не является до конца объективным. «Дешевая» заготовка (например, круглый прокат для изготовления ступенчатого вала) даст «дорогую» механическую обработку. А «дорогая» заготовка (например, штамповка для изготовления такого же вала) обеспечит более «дешевую» механическую обработку.

Целесообразно, поэтому, использовать в качестве критерия суммарную стоимость заготовки и механической обработки.

Однако стоимость механической обработки можно рассчитать только после разработки всего ТП. Следовательно, пропадает смысл «поэтапной оптимизации».

Но, все-таки, если удачно назначить критерии на каждом уровне проектирования, такой подход имеет смысл. При его применении может оказаться несколько равнозначных вариантов ТП, но среди них уже гораздо легче выбрать оптимальный вариант. Общая модель процесса технологического проектирования с поэтапным отсечением решений на каждом уровне может быть представлена следующим образом – см. рисунок 4.4.

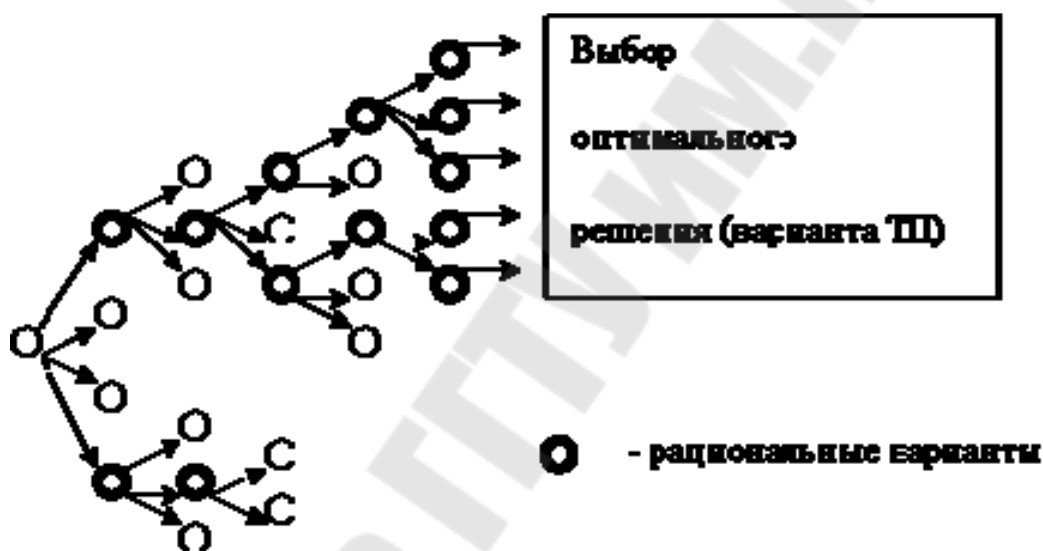


Рисунок 4.4 – Общая модель процесса технологического проектирования с поэтапным отсечением решений на каждом уровне

Прием 2. «Предпроектная оптимизация». Рассмотрим этот прием на примере выбора модели круглошлифовального станка. Множество возможных вариантов моделей круглошлифовальных станков определяется с помощью таблиц соответствий. Фрагмент такой таблицы приведен ниже в таблице 4.2.

Левая часть таблицы, обозначающая ее строки, представляет собой множество типовых решений. Верхняя часть таблицы, обозначающая ее столбцы, – условия применимости и их числовые значения. Центральная часть таблицы – булева матрица соответствий, в которой зафиксированы связи между решениями и определяющими их применимость значениями условий. Наличие связи обозначают

единицей, отсутствие – нулем. Иногда вместо единицы применяют штриховку соответствующей клетки, вместо нуля клетку оставляют незаштрихованной.

Таблица 4.2

Фрагмент таблицы соответствий для выбора возможных моделей круглошлифовальных станков

Модели станков (типовые решения)	Условия применимости							
	Максимальный диаметр детали D, мм			Максимальная длина детали L, мм				...
	≤ 200 (x ₁₁)	≤ 280 (x ₁₂)	...	≤ 300 (x ₂₁)	≤ 630 (x ₂₂)	≤ 900 (x ₂₃)
312М	0	0	...	1	0	0
3Б151	1	0	...	1	1	0
3Б161	0	0	...	1	1	1
...

По имеющемуся комплексу исходных данных из таблицы соответствий принимаются те решения, в строках которых булева матрица имеет единицы для всех значений факторов, входящих в условия применимости.

На базе таблиц соответствий строятся алгоритмы, позволяющие выбирать *множество допустимых решений*, из которых путем последовательного перебора выбираются наилучшие решения согласно тому или иному критерию оптимальности.

Но и при локализованной структурной оптимизации перебор и анализ всех допустимых решений, выбираемых из таблиц соответствий, занимает большое время.

Для сокращения времени счета при структурной оптимизации с использованием таблиц соответствий производят так называемую *предпроектную оптимизацию* на стадии разработки информационного обеспечения. Для этого используют *графики соответствий*.

Построим график соответствий для одного из условий применимости, например, для первого – см. таблицу 4.2.

Критерий оптимизации – себестоимость $C \rightarrow \min$, соответственно, целевая функция. Примем u_i – типовые решения (здесь – модели станков), x_j – диапазоны условий применимости. Пусть количество типовых решений (моделей станков) равняется не трем, а семи, количество диапазонов в первом условии применимости – пять.

График соответствий показан на рисунке 4.5.

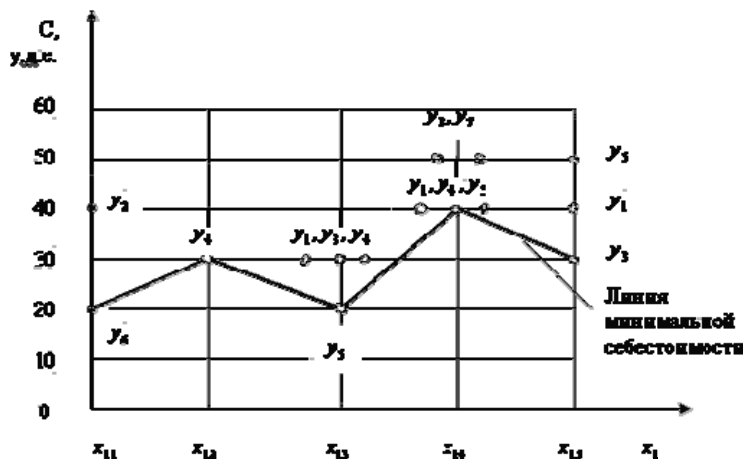


Рисунок 4.5 – График соответствий и линия минимальной себестоимости

Соединяя линией решения, имеющие минимальную себестоимость, получаем линию минимальной себестоимости. Решения, лежащие на этой линии, называют предпочтительными.

Построим теперь таблицу соответствий, в которой единицы заменены штриховкой и предпочтительные решения выделены звездочками – см. таблицу 4.3.

Таблица 4.3 – Таблица соответствий

Таблица соответствий для первого условия применимости

y_i	Диапазоны условия применимости x_i				
	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
y_1				*****	
y_2					
y_3					*****
y_4		*****		*****	
y_5			*****	*****	
y_6	*****				
y_7					

Другими словами в таблице штриховкой показаны технически возможные решения, звездочками – экономически эффективные решения.

Поиск решений в таблице соответствий сначала осуществляется по предпочтительным решениям. В случае отсутствия подходящего предпочтительного решения поиск производится по оставшимся допустимым.

Такой подход эффективен для случаев наличия экстремума целевой функции. Но в ряде случаев решение получается неопределен-

ным. Так, например, в нашем случае для диапазона x_{14} условия применимости имеется несколько эффективных решений.

Прием 3. Следующим шагом в развитии предпроектной оптимизации является переход от булевых матриц соответствий к оценочным матрицам. В этом случае в соответствующих клетках матрицы соответствий проставляются значения себестоимости с графика соответствий – см. таблицу 4.4.

Подобные матрицы заполняются для всех условий применимости.

Алгоритм поиска оптимального решения по оценочной матрице состоит в поиске одноименной строки в оценочных матрицах для всех диапазонов условий применимости, обеспечивающей наименьшую сумму затрат для данного условия задачи.

Рассмотренная процедура повторяется для каждого уровня проектирования, приводя в конечном итоге к варианту с оптимальной структурой

Таблица 4.4 – Оценочная матрица
Оценочная матрица

y_i	Диапазоны условия применимости x_i				
	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
y_1	∞	∞	30	40	40
y_2	40	∞	∞	∞	∞
y_3	∞	∞	30	50	30
y_4	∞	30	30	40	∞
y_5	∞	∞	20	40	50
y_6	20	∞	∞	∞	∞
y_7	∞	∞	∞	50	∞

4.2 Многокритериальная оптимизация

4.2.1 Общая характеристика и классификация методов принятия решений при многих критериях

Все задачи оптимизации и выбора решений можно разделить на однокритериальные и многокритериальные.

Задачи, в которых решение принимается с учетом *одного* критерия (показателя качества), называются однокритериальными, или задачами *скалярной оптимизации*.

В таких задачах требуется найти решение, при котором показатель качества принимает экстремальное (максимальное или минимальное) значение. При этом в задачах скалярной оптимизации, как правило, требуется учитывать и другие критерии; на них обычно накладываются ограничения.

Пример задачи скалярной оптимизации – составление плана производства нескольких видов изделий с целью получения максимальной прибыли (при ограниченных ресурсах).

Как правило, на практике решения принимаются с учетом *нескольких* критериев (*вектора критериев*). Такие задачи называются многокритериальными, или задачами *векторной оптимизации*.

Среди задач оптимизации выделяют также дискретные и непрерывные задачи.

В *дискретных* задачах множество возможных решений (альтернатив) конечно. Типичные примеры таких задач – выбор одного из нескольких товаров при покупке, выбор одного из возможных проектов строительства предприятия и т.д.

В *непрерывных* задачах имеется бесконечное множество возможных решений. Примеры таких задач – выбор оптимальных параметров химической реакции при разработке технологического процесса, выбор оптимального плана производства нескольких сортов бензина и т.д.

Очевидно, что как задачи скалярной оптимизации, так и задачи векторной оптимизации могут представлять собой как непрерывные, так и дискретные задачи.

Например, если требуется распределить сумму в размере 10 млн ден.ед. между несколькими предприятиями, и эффективность вложения средств оценивается по одному критерию (например, только по прибыли), то решаемая задача относится к классу задач скалярной оп-

тимизации. Если учитывается несколько критериев (например, прибыль и срок окупаемости), то решаемая задача представляет собой задачу векторной оптимизации. Если средства могут выделяться только в размерах, кратных одному миллиону, то данная задача – дискретная. Если средства могут выделяться в любых количествах (в пределах имеющейся суммы), то задача относится к числу непрерывных.

Большинство практических задач являются многокритериальными. Можно указать на следующие основные проблемы, возникающие при выборе решений по многим критериям:

- противоречивость критериев: улучшение по одному критерию обычно приводит к ухудшению по каким-либо другим критериям (типичный пример: чем более качественным является товар, тем он дороже);
- невозможность аналитического (в виде формул) выражения связей между оценками по разным критериям;
- оценки по различным критериям имеют разный вид: числовые, качественные («отлично», «хорошо», «да-нет» и т.д.), балльные, в виде ранжирований и т.д.;
- числовые оценки отличаются по размерности (соответствуют разным физическим величинам и измеряются в разных единицах), по направленности (одни критерии требуется минимизировать, другие – максимизировать), по диапазону значений;
- различие критериев по важности.

Основной способ снятия этих проблем в процессе принятия решения – выявление и учет субъективных суждений лица, принимающего решение (ЛПР).

Обычно от ЛПР обычно требуется следующая информация:

- перечень альтернатив, из которых требуется сделать выбор;
- перечень критериев, по которым следует сравнивать альтернативы;
- оценки альтернатив по критериям;
- суждения о важности критериев (т.е. информация о том, какие критерии важнее, какие – менее важны);
- ограничения по отдельным критериям;
- суждения о том, какие значения критериев желательны, а какие – нежелательны;
- суждения о степени допустимости отставания по отдельным критериям, о компенсации одних критериев другими;

- парные сравнения альтернатив (т.е. информация о том, какая из двух рассматриваемых альтернатив лучше, а какая – хуже);
- ранжирования альтернатив по отдельным критериям;
- суждения о возможных состояниях внешней среды.

На основе оценок альтернатив по критериям, а также субъективной информации, полученной от ЛПР, выбирается лучшая альтернатива.

В большинстве методов решения многокритериальных задач для каждой альтернативы рассчитывается некоторая обобщенная оценка, в которой учитываются оценки по всем критериям. Для приведения оценок по различным критериям к единой форме и получения обобщенной оценки альтернативы используются следующие основные методы.

1. Переход от оценок различного вида к экспертным оценкам.

Они могут указываться в виде балльных оценок, в долях единицы, в виде парных сравнений, в виде ранжирований и т.д.

Примеры перехода к экспертным оценкам – *метод анализа иерархий, модифицированный алгоритм Кемени-Снелла*.

2. Для числовых оценок обычно выполняется переход к оценкам, имеющим значения от 0 до 1 и направленных на максимум (т.е. оценок, имеющих смысл «чем больше, тем лучше»).

Обычно лучшей оценке по критерию соответствует оценка, равная единице. Такое преобразование оценок используется, например, в методах на основе функций полезности.

3. Для перевода качественных (словесных) оценок в числовую форму используется шкала Харрингтона.

При этом оценке «отлично» соответствуют числовые оценки от 0,8 до 1; «хорошо» - от 0,63 до 0,8; «удовлетворительно» – от 0,37 до 0,63; «плохо» - от 0,2 до 0,37; «очень плохо» - от 0 до 0,2.

Числовая оценка выставляется человеком (ЛПР или экспертом) исходя из его субъективных суждений.

Например, если по некоторому критерию две альтернативы имеют оценку «хорошо», но одна из них очень хорошая, а другая - немного хуже, то первой из альтернатив (лучшей) можно назначить оценку 0,8, а второй - 0,7.

Такой переход к числовым оценкам применяется, например, в *методике экспресс-анализа альтернатив*.

4. Для оценок, имеющих вид «да-нет» (т.е. выражающих наличие или отсутствие некоторого показателя), используются следующие числовые оценки: «да» - 0,67; «нет» - 0,33 (здесь предполагается, что оценка «да» более желательна, чем «нет»).

Эти числовые оценки могут меняться в зависимости от того, насколько значимо наличие рассматриваемого показателя (в зависимости от конкретной задачи).

Такое преобразование также применяется в *методике экспресс-анализа альтернатив*.

В таблице 4.5 приводится классификация методов и процедур принятия решений при многих критериях. Методы классифицированы в зависимости от основной процедуры, применяемой при сравнении и выборе альтернатив.

Таблица 4.5 – Классификация методов и процедур принятия решений при многих критериях

Классы методов	Примеры	Решаемые задачи
Методы на основе выбора главного критерия	Методы на основе лексикографического упорядочения критериев	Непрерывные и дискретные задачи (с четко выраженным различием критериев по важности)
Методы на основе компенсации критериев	Метод последовательных уступок	Непрерывные и дискретные задачи (с небольшим количеством критериев)
Методы на основе вычисления обобщенных оценок альтернатив	Метод “эффективность-стоимость”, метод комплексной оценки структур, методы на основе функций полезности	Дискретные задачи (в основном – с числовыми критериями)
Методы на основе попарных сравнений альтернатив	Метод анализа иерархий, модифицированный алгоритм Кемени-Снелла, метод ЭЛЕКТРА	Дискретные задачи (с критериями любого вида)
Методы на основе выявления суждений ЛПР	Метод ЗАПРОС	Дискретные задачи (с критериями любого вида)

Ниже рассматриваются основные принципы работы, достоинства и недостатки этих методов.

Методы на основе выбора главного критерия. К этому классу относятся все методы, в которых выбор решения производится на основе одного (главного) критерия; на остальные критерии, как правило, накладываются ограничения.

К этому же классу следует отнести методы, называемые «методами на основе лексикографического упорядочения критериев».

В этих методах сначала выбирается лучшая альтернатива по одному (наиболее важному) критерию; если по данному критерию оказывается несколько одинаковых альтернатив, то используется следующий по важности критерий, и т.д. Такие методы неприменимы для задач, в которых требуется учитывать несколько критериев, близких по важности.

Методы на основе компенсации критериев. Принцип работы этих методов состоит в том, что от ЛПР или эксперта требуется указать следующее:

- какая величина выигрыша по одному критерию компенсирует определенный (заданный) проигрыш по другому критерию;
- какой проигрыш по одному критерию можно считать допустимым, чтобы обеспечить улучшение оценки по другому критерию.

Указание таких величин компенсации достаточно сложно для человека (ЛПР или эксперта). Поэтому применение таких методов ограничено.

Методы на основе вычисления обобщенных оценок альтернатив (обобщенного критерия). Принцип работы этих методов состоит в вычислении обобщенной оценки для каждой из альтернатив на основе их оценок по отдельным критериям.

Достоинство этих методов – небольшой объем информации, требуемый от ЛПР. Эти методы нашли широкое применение и реализованы во многих компьютерных системах поддержки принятия решений (СППР).

В то же время эти методы имеют ряд существенных недостатков:

- методы этого класса не позволяют в достаточной мере учесть субъективные суждения ЛПР о превосходстве альтернатив друг над другом, о желательности (или нежелательности) значений критериев и т.д.;
- применение этих методов затрудняется при использовании критериев с нечисловыми оценками (словесные оценки, оценки «да-нет», оценки в виде ранжирований альтернатив и т.д.).

Методы на основе попарных сравнений альтернатив. При использовании этих методов для каждой пары альтернатив определяется оценка превосходства одной альтернативы над другой; эта оценка может непосредственно указываться человеком или вычисляться на основе оценок альтернатив по отдельным критериям. На основе этих сравнений определяется лучшая альтернатива.

Эти методы обладают следующими достоинствами:

- возможность полного учета суждений ЛПР об альтернативах;
- возможность использования оценок любых видов: числовых, качественных, «да-нет» и т.д.;
- при использовании методов этого класса обычно требуется, чтобы человек (ЛПР или эксперт) выполнил попарное сравнение или ранжирование альтернатив по отдельным критериям; как правило, человек достаточно легко предоставляет такую информацию.

Основные недостатки методов этого класса:

- в некоторых методах от ЛПР или эксперта требуется выполнить большое количество парных сравнений;
- не всегда в полной мере учитывается степень превосходства одной альтернативы над другой по отдельным критериям.

Методы на основе выявления суждений ЛПР. В этих методах от ЛПР требуется информация не о том, какие из имеющихся альтернатив лучше, а о том, какие критерии ЛПР считает более важными, какие оценки – более желательными, и т.д.

Эти методы достаточно перспективны. Их основное достоинство – наиболее полный учет суждений ЛПР.

Недостаток – необходимость получения от ЛПР большого объема информации, а также сложность предоставления этой информации.

4.2.2 Выбор Парето-оптимальных решений

Выбор множества Парето-оптимальных решений (множества Парето) представляет собой отбор перспективных альтернатив, из которых затем отбирается одна (лучшая) альтернатива.

Множество Парето представляет собой множество альтернатив, обладающих следующим свойством: любая из альтернатив, входящих во множество Парето, хотя бы по одному критерию лучше любой другой альтернативы, входящей в это множество.

Выбор множества Парето производится следующим образом. Все альтернативы попарно сравниваются друг с другом по всем критериям. Если при сравнении каких-либо альтернатив (обозначим их как A_i и A_j) оказывается, что одна из них (например, A_j) *не лучше другой ни по одному критерию*, то ее можно исключить из рассмотрения. Исключенную альтернативу (в данном случае - A_j) не требуется сравнивать с другими альтернативами, так как она явно неперспективна.

Как правило, во множество Парето входит несколько альтернатив. Поэтому выбор множества Парето не обеспечивает принятия окончательного решения (выбора одной лучшей альтернативы), однако позволяет сократить количество рассматриваемых альтернатив, т.е. упрощает принятие решения.

Пример 1. Химический комбинат планирует внедрить комплекс средств автоматизации (КСА) для системы управления технологическими процессами.

Имеется возможность выбрать один из семи вариантов КСА (КСА1, КСА2,...,КСА7). При выборе учитываются четыре критерия: затраты, связанные с изготовлением КСА и его вводом в эксплуатацию; срок ввода КСА в эксплуатацию; срок гарантийного обслуживания предприятием-изготовителем; удобство КСА в эксплуатации. Характеристики КСА приведены в таблице 4.6.

Таблица 4.6 – Выбор множества Парето

	КСА1	КСА2	КСА3	КСА4	КСА5	КСА6	КСА7
Затраты, млн ден.ед.	40	30	40	60	45	25	55
Срок ввода в эксплуатацию, мес.	8	8	6	6	7	8	6
Срок гарантийного обслуживания, лет	4	4	5	7	4	4	5
Удобство в эксплуатации	хор	отл	удовл	отл	плохо	очень хор	хор

В этой задаче имеется семь альтернатив (вариантов КСА). Они оцениваются по четырем критериям. Критерий «удобство в эксплуатации» – качественный (словесный) остальные критерии – числовые. Критерии «затраты» и «срок ввода в эксплуатацию» подлежат мини-

мизации, критерий «срок гарантийного обслуживания» – максимизации.

Выберем множество Парето. Для этого выполним попарное сравнение альтернатив по всем критериям

Сравниваем КСА1 и КСА2. По критерию «затраты» лучше КСА2; по критериям «срок ввода в эксплуатацию» и «срок гарантийного обслуживания» эти КСА одинаковы. По критерию «удобство в эксплуатации» лучше КСА2. Таким образом, ни по одному критерию КСА1 не лучше, чем КСА2. Альтернатива КСА1 исключается из рассмотрения, так как она явно не лучшая. Сравнивать другие альтернативы с КСА1 не требуется.

Таким образом, во множество Парето вошли следующие варианты: КСА2, КСА3, КСА4, КСА6, КСА7. Предприятие выберет один из этих вариантов КСА. Для окончательного выбора применяются методы, рассматриваемые ниже.

4.2.3 Методы на основе компенсации критериев

Методы этого класса основаны на улучшении оценок по одним критериям за счет ухудшения оценок по другим, причем допустимая величина ухудшения оценок должна быть задана экспертом или ЛПР. Наиболее распространенным из методов этого класса является *метод последовательных уступок*.

Принцип работы метода следующий. На основе суждений ЛПР (или экспертных оценок) выполняется ранжирование критериев по важности.

В некоторых случаях такое ранжирование может выполняться непосредственно: ЛПР указывает наиболее важный критерий, второй по важности и т.д. В других случаях для ранжирования применяются методы экспертных оценок.

Затем находится лучшее решение по наиболее важному критерию. После этого ЛПР указывает допустимую уступку, т.е. величину, на которую можно ухудшить оценку по наиболее важному критерию, чтобы обеспечить улучшение по другому (второму по важности) критерию.

Находится лучшее решение по второму критерию (при условии, что оценка по первому критерию ухудшается не больше, чем на заданную уступку).

Затем указывается уступка по второму критерию. Находится лучшее решение по следующему (третьему по важности) критерию, при соблюдении ограничений на уступки по первому и второму критерию. Процесс продолжается, пока не будет выполнена оптимизация по всем критериям.

Приведем примеры применения метода последовательных уступок для решения задач дискретной и непрерывной оптимизации.

Пример 2. Предприятие предполагает приобрести новый технологический модуль. Имеется возможность приобрести один из модулей, характеристики которых указаны в таблице 4.7.

По мнению ЛПР (специалиста, ответственного за закупку оборудования), наиболее важный критерий, который необходимо учитывать при выборе модуля – производительность; следующий по важности – стоимость; следующий – удобство в эксплуатации; наименее важный критерий – гарантийный срок.

Таблица 4.7 – Исходные данные для примера 1

	ТМ1	ТМ2	ТМ3	ТМ4	ТМ5	ТМ6	ТМ7
Производительность, изделий/ч	40	45	48	35	50	32	42
Удобство в эксплуатации	удовл	отл	отл	хор	отл	хор	хор
Гарантийный срок, лет	5	3	4	4	5	3	3
Стоимость, тыс ден.ед.	220	240	300	180	320	160	200

Рассмотрим выбор лучшего модуля на основе метода последовательных уступок.

Сначала выбирается лучшая альтернатива по самому важному критерию, т.е. по производительности. Это модуль ТМ5. Его производительность – 50 изделий в час.

Предположим, что ЛПР считает допустимым снижение производительности (для улучшения характеристик по другим критериям), но не более чем на 10 изделий в час. Таким образом, допустимая уступка по критерию «производительность» равна 10. Допустимыми являются альтернативы, отстающие от лучшей (от ТМ5) не более чем на величину уступки, т.е. имеющие производительность не менее 40 изделий в час. Это модули ТМ1, ТМ2, ТМ3, ТМ5, ТМ7.

Выбирается лучший модуль по критерию «стоимость» (из числа модулей, допустимых по критерию «производительность»). Это модуль ТМ7 (стоимость – 200 тыс ден.ед.).

Предположим, что ЛПР считает допустимым увеличение стоимости не более чем на 100 тыс ден.ед. (эта величина представляет собой уступку по критерию «стоимость»). Допустимыми являются модули, превышающие стоимость ТМ7 не более чем на 100 тыс. ден.ед., т.е. имеющие стоимость не более 300 тыс. ден.ед. Это модули ТМ1, ТМ2, ТМ3, ТМ7.

Выбирается лучший модуль по критерию «удобство в эксплуатации» (из числа модулей, допустимых по обоим предыдущим критериям). Таких модулей два: ТМ2 и ТМ3. Они имеют по данному критерию оценку «отлично».

Пусть ЛПР считает допустимым ухудшение оценки по критерию «удобство в эксплуатации» не ниже чем до уровня «хорошо». В этом случае допустимыми являются модули ТМ2, ТМ3, ТМ7.

Выбирается лучший модуль по критерию «гарантийный срок» (из модулей, допустимых по предыдущим критериям, т.е. из ТМ2, ТМ3, ТМ7). Лучший из них – ТМ3.

Таким образом, рациональным решением для предприятия является приобретение технологического модуля ТМ3.

Пример 3. Для цеха по производству изделий бытовой химии разрабатывается план выпуска двух видов стиральных порошков: «Универсал» и «Люкс».

Все виды сырья, используемого при производстве порошков, могут закупаться на рынке в неограниченном количестве, кроме специальной отбеливающей добавки; ее можно закупить не более 2,5 тонн. Расход этой добавки на одну упаковку порошка «Универсал» составляет 100 г, на упаковку порошка «Люкс» – 300 г.

В процессе производства порошков выполняются три операции: очистка сырья, химическая обработка, расфасовка. Затраты времени на эти операции (в минутах) приведены в таблице 4.8.

Таблица 4.8 – Исходные данные для примера 3

	«Универсал»	«Люкс»
Очистка материала для десяти упаковок	20	25
Химическая обработка материала для десяти упаковок	15	40
Фасовка одной упаковки	2	2

Предприятие имеет две установки для очистки материала, три – для химической обработки материала, две – для расфасовки. Фонды времени работы единицы оборудования на плановый период следую-

щие: установка для очистки материала – 200 часов, установка для химической обработки материала – 240 часов, установка для расфасовки – 200 часов.

При производстве порошков применяется реактив, использование которого представляет опасность для работающих и требует специальных мер защиты. При выпуске одной упаковки порошка «Универсал» расходуется 400 г этого реактива, при выпуске упаковки порошка «Люкс» – 200 г.

Себестоимость одной упаковки порошка «Универсал» составляет 6 ден.ед., порошка «Люкс» – 10 ден.ед. Предприятие предполагает продавать порошок «Универсал» по цене 11 ден.ед., «Люкс» – по 18 ден.ед.

Требуется составить план производства порошков, обеспечивающий получение максимальной прибыли. В то же время руководство предприятия желает сократить использование опасного реактива, а также (в рекламных целях) увеличить производство более дешевого порошка «Универсал».

Данная задача представляет собой задачу с тремя критериями. Основным критерий – прибыль, второй по важности – использование опасного реактива, наименее важный – объем производства порошка «Универсал». Рассмотрим решение этой задачи на основе метода последовательных уступок.

Сначала найдем оптимальное решение по основному критерию (прибыли). Составим математическую модель задачи. Обозначим объем выпуска порошков «Универсал» и «Люкс» через X_1 и X_2 . Тогда можно составить следующую систему ограничений.

Ограничения на время работы оборудования:

$$2X_1 + 2,5X_2 \leq 24000$$

$$1,5X_1 + 4X_2 \leq 43200$$

$$2X_1 + 2X_2 \leq 24000.$$

Ограничение на расход отбеливающей добавки:

$$0,1X_1 + 0,3X_2 \leq 2500.$$

Здесь фонды времени работы оборудования указаны в минутах. В первом и втором ограничении (на очистку и химическую обработку) расходы времени приведены с учетом того, что материал обрабатывается партиями по 10 упаковок.

Целевая функция имеет следующий вид:

$$E = 5X_1 + 8X_2 \rightarrow \max$$

Коэффициенты целевой функции определены как разности цены и себестоимости продукции.

Переменные X_1 и X_2 должны быть целочисленными, так как они обозначают количество упаковок.

Решив эту задачу симплекс-методом (и методом ветвей и границ), получим: $X_1 = 2715$, $X_2 = 7428$, $E = 72999$. Таким образом, оптимальное решение по критерию максимизации прибыли состоит в том, что следует выпустить 2715 упаковок порошка «Универсал» и 7428 упаковок порошка «Люкс». Прибыль составит 72999 ден.ед.

Пусть в целях сокращения использования опасного реактива руководство предприятия считает допустимым некоторое снижение прибыли, однако не более чем на 10 %. Найдем величину уступки: она составляет 10 % от 72999, или примерно 7300 ден.ед. Таким образом, прибыль не должна составить менее 65699 ден. ед. ($72999 - 7300 = 65699$). Для удобства расчетов округлим эту цифру до 65700. В систему ограничений требуется включить новое ограничение:

$$5X_1 + 8X_2 \geq 65700$$

Это ограничение устанавливает, что прибыль должна быть не ниже 65700 ден. ед., т.е. она не должна быть ниже оптимальной более чем на 10%.

Таким образом, система ограничений примет следующий вид:

$$5X_1 + 8X_2 \geq 65700$$

$$0,1X_1 + 0,3X_2 \leq 2500$$

$$2X_1 + 2,5X_2 \leq 24000$$

$$1,5X_1 + 4X_2 \leq 43200$$

$$2X_1 + 2X_2 \leq 24000$$

В качестве новой целевой функции используем расход опасного реактива

$$E = 0,4X_1 + 0,2 X_2 \rightarrow \min$$

Решив эту задачу симплекс-методом и методом ветвей и границ, получим: $X_1 = 0$, $X_2 = 8213$. Таким образом, следует выпустить 8213 упаковок порошка «Люкс»; порошок «Универсал» выпускать не

следует. При этом расходуется 1642,6 кг опасного реактива. Легко подсчитать, что прибыль при данном плане производства составит 65704 ден.ед.

Предположим, что с целью увеличения производства порошка «Универсал» руководство предприятия согласно на увеличение использования опасного реактива, однако не более чем на 15 %. Найдем величину уступки: она составляет 15 % от 1642,6, или 246,4 кг. Таким образом, расход опасного реактива не должен превышать $1642,6 + 246,4 = 1889$ кг; для удобства округлим эту величину до 1890 кг. В систему ограничений включается новое ограничение:

$$0,4X_1 + 0,2X_2 \leq 1890$$

Следует обратить внимание, что в системе ограничений сохраняется ограничение на прибыль

$$5X_1 + 8X_2 \geq 65700$$

Таким образом, система ограничений примет следующий вид:

$$5X_1 + 8X_2 \geq 65700$$

$$0,1X_1 + 0,3X_2 \leq 2500$$

$$2X_1 + 2,5X_2 \leq 24000$$

$$1,5X_1 + 4X_2 \leq 43200$$

$$2X_1 + 2X_2 \leq 24000$$

$$0,4X_1 + 0,2X_2 \leq 1890$$

В качестве новой целевой функции используем объем выпуска порошка «Универсал»:

$$E = X_1 \rightarrow \max$$

Решив эту задачу симплекс-методом и методом ветвей и границ, получим: $X_1 = 900$, $X_2 = 7650$. Таким образом, следует выпустить 900 упаковку порошка «Универсал» и 7650 упаковок порошка «Люкс». Прибыль при этом составит 65700 ден.ед., а расход опасного реактива – 1890 кг.

4.2.4 Методика экспресс-анализа альтернатив

Методика предназначена для отбора перспективных альтернатив. При этом перспективными считаются альтернативы, не имеющие существенных недостатков ни по одному из критериев.

Методика рассчитана на применение в задачах, в которых большинство критериев являются числовыми. Методика может применяться и для решения задач, в которых имеются качественные критерии; в этом случае оценки по таким критериям следует выразить по

пятибалльной шкале («отлично», «хорошо», «удовлетворительно», «плохо», «очень плохо»), а затем перейти к числовым оценкам, используя шкалу Харрингтона.

Принцип работы методики следующий. Для каждой альтернативы находится худшая оценка (из всех оценок данной альтернативы по критериям, используемым в задаче). Выбираются альтернативы, у которых худшая оценка не ниже некоторой пороговой величины.

Рассмотрим реализацию этой методики на примере 1 из подпункта 4.2.2.

В таблице 4.9 приведены оценки альтернатив (вариантов КСА), вошедших во множество Парето.

Таблица 4.9 – Множество Парето

	КСА2	КСА3	КСА4	КСА6	КСА7
Затраты, млн ден.ед.	30	40	60	25	55
Срок ввода в эксплуатацию, мес.	8	6	6	8	6
Срок гарантийного обслуживания, лет	4	5	7	4	5
Удобство в эксплуатации	отл	удовл	отл	очень хор	хор

Обозначим оценки альтернатив по критериям как X_{ij} , $i = 1, \dots, M$, $j = 1, \dots, N$. Здесь M – количество критериев, N – количество альтернатив (в данной задаче $M = 4$, $N = 5$).

Выбор множества перспективных альтернатив на основе методики экспресс-анализа реализуется в следующем порядке.

1. Оценки альтернатив по критериям приводятся к безразмерному виду.

Безразмерные оценки альтернатив P_{ij} , $i = 1, \dots, M$, $j = 1, \dots, N$, находятся следующим образом:

- для критериев, подлежащих максимизации: все оценки альтернатив по критерию делятся на максимальную из оценок по данному критерию:

$$P_{ij} = \frac{X_{ij}}{\max_j X_{ij}};$$

- для критериев, подлежащих минимизации: из оценок по данному критерию выбирается минимальная, и она делится на все оценки альтернатив по данному критерию:

$$P_{ij} = \frac{\min_j X_{ij}}{X_{ij}};$$

- для качественных (словесных) критериев: выполняется переход к числовым оценкам по шкале Харрингтона.

Рассмотрим получение безразмерных оценок для данной задачи.

Критерий «затраты» подлежит минимизации. Поэтому для него находится минимальная оценка (в данном примере она равна 25) и делится на все оценки по данному критерию. Например, для КСА2 безразмерная оценка по критерию «затраты» находится следующим образом: $25/30 = 0,83$.

Аналогично находятся безразмерные оценки по критерию «срок ввода в эксплуатацию»: минимальная оценка (в данном примере – 6) делится на все оценки по данному критерию.

Критерий «срок гарантийного обслуживания» подлежит максимизации. Поэтому все оценки по этому критерию делятся на максимальную оценку (в данном примере – на 7).

Безразмерные оценки по критерию «удобство в эксплуатации» назначаются экспертом.

Для данной задачи безразмерные оценки приведены в таблице 4.10.

Таблица 4.10 – Безразмерные оценки

	КСА2	КСА3	КСА4	КСА6	КСА7
Затраты	0,83	0,63	0,42	1	0,45
Срок ввода в эксплуатацию	0,75	1	1	0,75	1
Срок гарантийного обслуживания	0,67	0,83	1	0,67	0,83
Удобство в эксплуатации	1	0,6	0,9	0,8	0,7

Примечание. По критерию «удобство в эксплуатации» эксперт назначил альтернативе КСА2 оценку 1, а КСА4 – оценку 0,9, хотя обе альтернативы оценивались по данному критерию как отличные. Это означает, что, по мнению эксперта, КСА2 по данному критерию немного лучше, чем КСА4.

В результате перехода к безразмерным оценкам устранены различия исходных оценок, затруднявшие сравнение альтернатив. Безразмерные величины не измеряются в каких-либо единицах, поэтому их можно сравнивать друг с другом, складывать и т.д. Безразмерные оценки не отличаются по диапазону значений: все они имеют значе-

ния в пределах от 0 до 1. Они не различаются также по направленности: чем больше безразмерная оценка, тем лучше (по любому критерию), и лучшее значение равно 1.

2. Для каждой альтернативы находится минимальная оценка, т.е. худшая из оценок данной альтернативы по всем критериям:

$$P_j = \min P_{ij},$$

Например, для КСА2 эта оценка равна 0,67; она находится как минимальная из 0,83, 0,75, 0,67 и 1.

Минимальные оценки приведены в таблице 4.11.

Таблица 4.11 – Минимальные оценки

	КСА2	КСА3	КСА4	КСА6	КСА7
P_j	0,67	0,6	0,42	0,67	0,45

3. Выбирается пороговое значение минимальной оценки P_0 . Эта величина назначается ЛПР или экспертом из субъективных соображений, например, в зависимости от количества альтернатив, которые требуется отобрать для дальнейшего анализа.

Пусть в данной задаче назначено $P_0 = 0,5$.

4. Выбирается множество альтернатив, для которых $P_j > P_0$. Таким образом, для дальнейшего анализа отбираются альтернативы, у которых все оценки (в том числе худшая) не ниже предельной величины P_0 .

В данной задаче отбираются альтернативы КСА2, КСА3, КСА6. Окончательный выбор делается на основе одного из методов, рассматриваемых ниже.

4.2.5 Методика скаляризации векторных оценок

Методика предназначена для выбора рациональной альтернативы из множества альтернатив, оцениваемых по нескольким критериям.

Как и методика экспресс-анализа альтернатив, данная методика рассчитана на решение задач, в которых решение принимается на основе числовых критериев (или может быть выполнен переход к таким критериям).

Основное преимущество этой методики – минимальный объем информации, которую требуется получить от ЛПР или эксперта для выбора решения, что позволяет полностью автоматизировать решение

задачи. В то же время недостаточный учет субъективных суждений ЛПР является недостатком этой методики.

Метод основан на вычислении обобщенной оценки каждой альтернативы (с учетом оценок по всем критериям) и сопоставлении этих оценок.

Рассмотрим реализацию методики на примере задачи выбора варианта КСА, рассмотренной в подразделах 4.2.2 и 4.2.4. В таблице 4.12 приведены оценки альтернатив, отобранных на основе выбора множества Парето и методики экспресс-анализа альтернатив.

Таблица 4.12 – Отобранные альтернативы

	КСА2	КСА3	КСА6
Затраты, млн ден.ед.	30	40	25
Срок ввода в эксплуатацию, мес.	8	6	8
Срок гарантийного обслуживания, лет	4	5	4
Удобство в эксплуатации	отл	удовл	очень хор

Методика реализуется в следующем порядке.

1) Оценки альтернатив приводятся к безразмерному виду, как и в методике экспресс-анализа альтернатив. Безразмерные оценки альтернатив для данной задачи приведены в таблице 4.13.

Таблица 4.13 – Безразмерные оценки альтернатив для данной задачи

	КСА2	КСА3	КСА6
Затраты	0,83	0,63	1
Срок ввода в эксплуатацию	0,75	1	0,75
Срок гарантийного обслуживания	0,8	1	0,8
Удобство в эксплуатации	1	0,6	0,8

2. Находятся веса (оценки важности) критериев. В рассматриваемой методике веса находятся на основе разброса оценок. Веса определяются в следующем порядке.

- находятся средние оценки по каждому критерию

$$\bar{P}_i = \frac{1}{N} \sum_{j=1}^N P_{ij}, \quad i=1, \dots, M,$$

где M – количество критериев; N – количество альтернатив P_{ij} – безразмерные оценки.

Для данного примера: $\bar{P}_1 = (0,83+0,63+1)/3=0,82$; $\bar{P}_2=(0,75+1+0,75)/3=0,83$; $\bar{P}_3 =0,87$; $\bar{P}_4=0,8$.

- находятся величины разброса по каждому критерию

$$R_i = \frac{1}{N \cdot \bar{P}_i} \sum_{j=1}^N |P_{ij} - \bar{P}_i|, \quad i=1, \dots, M.$$

Для данного примера:

$$R_1 = \frac{|0,83 - 0,82| + |0,63 - 0,82| + |1 - 0,82|}{3 \cdot 0,82} = 0,16$$

$$R_2 = \frac{|0,75 - 0,83| + |1 - 0,83| + |0,75 - 0,83|}{3 \cdot 0,83} = 0,13$$

$$R_3 = 0,1, \quad R_4 = 0,17$$

- находится сумма величин разброса

$$R = \sum_{i=1}^M R_i$$

Для данного примера $R = 0,16 + 0,13 + 0,1 + 0,17 = 0,56$.

- находятся веса критериев, отображающие разброс оценок

$$W_i = R_i / R, \quad i=1, \dots, M.$$

Для данного примера $W_1 = 0,16/0,56 = 0,29$; $W_2 = 0,13/0,56 = 0,23$; $W_3=0,18$; $W_4 = 0,3$.

Чем больше разброс (различие) в оценках альтернатив по критерию, тем больше вес этого критерия. Таким образом, критерии, по которым оценки альтернатив существенно различаются, считаются более важными. Если оценки альтернатив по какому-то критерию очень близки, то его вес будет небольшим, так как сравнение альтернатив при близких оценках не имеет смысла.

3. Находятся взвешенные оценки альтернатив (путем деления весов критериев на оценки по соответствующим критериям):

$$E_{ij} = W_i / P_{ij}, \quad i = 1, \dots, M, j = 1, \dots, N.$$

Взвешенные оценки для данного примера приведены в таблице 4.13.

Здесь, например, $E_{11} = 0,29/0,83 = 0,35$; $E_{12} = 0,29/0,63 = 0,46$; $E_{13} = 0,29/1=0,29$; $E_{21} = 0,23/0,75 = 0,31$, и т.д.

Чем больше значения безразмерных оценок P_{ij} , тем меньше значения принимают взвешенные оценки. Таким образом, чем *меньше* взвешенные оценки, тем *лучше* альтернатива.

Таблица 4.13 – Взвешенные оценки

	КСА2	КСА3	КСА6
Затраты	0,35	0,46	0,29
Срок ввода в эксплуатацию	0,31	0,23	0,31
Срок гарантийного обслуживания	0,23	0,18	0,23
Удобство в эксплуатации	0,3	0,5	0,38

Чем меньше комплексная оценка, тем лучше альтернатива. Таким образом, в данном примере лучшим является вариант комплекса средств автоматизации КСА2; несколько худший вариант – КСА6, еще хуже – КСА3.

Примечание. Возможны другие реализации данной методики. Например, веса критериев могут определяться не на основе разброса оценок альтернатив, а с помощью методов экспертного анализа. Это позволяет учесть суждения эксперта или ЛПР о том, какие из критериев являются более важными.

4.2.6 Методика сравнительной оценки двух альтернатив по степени доминирования

Методика предназначена для решения задач, в которых требуется выбрать лучшую из двух альтернатив. Такие задачи часто возникают, например, при проектировании технических систем, когда требуется выбрать лучший из двух вариантов системы: базовый (имеющийся) или новый (предлагаемый). В то же время применение данной методики не ограничивается задачами проектирования.

Для применения данной методики все оценки альтернатив должны быть выражены в числовой форме.

Принцип работы методики следующий. Для каждой из двух сравниваемых альтернатив находится обобщенная оценка по всем критериям, по которым она превосходит другую альтернативу; при этом учитывается степень превосходства, а также важность критериев. Полученные обобщенные оценки сравниваются; выбирается альтернатива, имеющая большую оценку.

Рассмотрим реализацию методики на следующем примере.

Пример. В ходе реконструкции порта рассматриваются проекты строительства нового терминала для разгрузки танкеров. Предлагаются четыре проекта. Характеристики проектов приведены в таблице 4.14.

Таблица 4.14 – Характеристики проектов

	П1	П2	П3	П4
Пропускная способность, тонн/день	2600	2000	2000	2500
Затраты на строительство, млрд ден.ед.	3,5	2,5	3	4
Срок строительства, мес	24	30	30	28
Экологическая безопасность	хор	отл	хор	удовл
Стоимость обработки одной тонны нефти, ден.ед.	4,5	3,5	4,5	6

По мнению руководства порта, наиболее важный критерий – пропускная способность, следующие по важности – экологическая безопасность и стоимость обработки одной тонны нефти, еще немного менее важный – затраты на строительство, наименее важный – срок строительства.

Для отбора перспективных проектов найдем множество Парето. Во множество Парето войдут только проекты П1 и П2. Таким образом, требуется выбрать одну из двух альтернатив. Для решения такой задачи целесообразно применить методику сравнительной оценки двух альтернатив по степени доминирования.

По критерию «экологическая безопасность» требуется перейти к числовым оценкам. Для этого воспользуемся шкалой Харрингтона. Пусть для проекта П1 по данному критерию назначена числовая оценка 0,8, а для П2 – оценка 1.

Если по какому-либо критерию альтернативы имеют одинаковые оценки, то этот критерий не учитывается при сравнении альтернатив. В данной задаче таких критериев нет.

Методика реализуется в следующем порядке.

1) Выполняется ранжирование критериев по важности: наиболее важный критерий получает ранг 1, следующий по важности – 2, и т.д. Если какие-либо критерии близки по важности, им рекомендуется назначать одинаковые ранги.

Обозначим ранги как R_i , $i=1, \dots, M$, где M – количество критериев.

Пусть в данной задаче критериям назначены следующие ранги: $R_1=1$, $R_2=3$, $R_3=4$, $R_4=2$, $R_5=2$. Ранги R_4 и R_5 равны, так как (по мне-

нию ЛПР) критерии «экологическая безопасность» и «стоимость обработки одной тонны нефти» примерно одинаковы по важности.

2) Выполняется переход от рангов к весам критериев. Веса находятся следующим образом: из всех рангов выбирается максимальный (в данном примере он равен 4), к нему прибавляется единица, из полученного числа вычитаются ранги:

$$V_i = \max_j(R_j) + 1 - R_i, \quad i=1, \dots, M.$$

Таким образом, чем важнее критерий, тем больше его вес.

Для данной задачи веса критериев следующие:

$$V_1 = 4 + 1 - 1 = 4; V_2 = 4 + 1 - 3 = 2; V_3 = 1; V_4 = 3; V_5 = 3.$$

3) Находятся отношения оценок альтернатив (степени доминирования) путем деления большей оценки по каждому критерию на меньшую:

$$S_i = \max(X_{i1}, X_{i2}) / \min(X_{i1}, X_{i2}), \quad i=1, \dots, M,$$

где X_{i1} , X_{i2} – оценки двух сравниваемых альтернатив по i -му критерию.

Для данной задачи $S_1 = 2600/2000 = 1,3$; $S_2 = 3,5/2,5 = 1,4$; $S_3 = 30/24 = 1,25$; $S_4 = 1/0,8 = 1,25$; $S_5 = 4,5/3,5 = 1,29$.

4) Находятся скорректированные степени доминирования альтернатив путем возведения степеней доминирования в степени, равные весам критериев:

$$C_i = S_i^{V_i}, \quad i=1, \dots, M.$$

Таким образом, учитывается важность критериев: чем больше вес критериев, тем больше соответствующая степень доминирования будет влиять на окончательную оценку.

Для данной задачи $C_1 = 1,3^4 = 2,86$; $C_2 = 1,4^2 = 1,96$; $C_3 = 1,25^1 = 1,25$; $C_4 = 1,25^3 = 1,95$; $C_5 = 1,29^3 = 2,15$.

1. Для каждой из сравниваемых альтернатив находится оценка ее доминирования для другой альтернативой. Эта оценка вычисляется как произведения скорректированных степеней доминирования по всем критериям, по которым данная альтернатива лучше другой.

В данном примере проект П1 лучше проекта П2 по критериям «пропускная способность» и «срок строительства». Оценка доминирования проекта П1 над П2 находится следующим образом:

$$D_1 = 2,86 * 1,25 = 3,575.$$

Проект П2 лучше, чем проект П1, по критериям «затраты», «экологическая безопасность» и «стоимость обработки одной тонны нефти». Оценка доминирования П1: $D_2 = 1,96 * 1,95 * 2,15 = 8,22$.

6. Находится обобщенная оценка доминирования: $D = D_1/D_2$.

Если $D > 1$, то первая альтернатива лучше второй; если $D < 1$, то вторая альтернатива превосходит первую. В данном примере $D = 3,575/8,22 = 0,44$. Таким образом, проект П2 лучше, чем П1.

4.2.7 Алгоритм Саати

Алгоритм Саати основан на сравнении альтернатив, выполняемом одним экспертом.

Для каждой пары альтернатив эксперт указывает, в какой степени одна из них предпочтительнее другой.

Рассмотрим применение этого метода на следующем примере.

Пример. Предприятие выбирает основной вид рекламы для новой продукции. Предлагаются четыре возможных вида:

- реклама на телевидении (обозначим ее как А1),
- на радио (А2),
- в газете (А3),
- на стендах (А4).

Решение о выборе вида рекламы принимается на основе консультации с экспертом.

Принятие решения на основе алгоритма Саати выполняется в следующем порядке.

1. Экспертом заполняется матрица парных сравнений размером $N \times N$, где N – количество альтернатив.

Матрица заполняется по правилам, приведенным в таблице 4.15.

Таблица 4.15 – Правила заполнения матрицы парных сравнений для метода Саати

X_{ij}	Значение
1	i -я и j -я альтернативы примерно равноценны
3	i -я альтернатива немного предпочтительнее j -й
5	i -я альтернатива предпочтительнее j -й
7	i -я альтернатива значительно предпочтительнее j -й
9	i -я альтернатива явно предпочтительнее j -й

Если i -я альтернатива менее предпочтительна, чем j -я, то указываются обратные оценки (1/3, 1/5, 1/7, 1/9). Могут использоваться промежуточные оценки (2, 4, 6, 8 и 1/2, 1/4, 1/6, 1/8); например, если i -я альтернатива совсем немного лучше j -й, то можно использовать оценку $X_{ij} = 2$ (тогда $X_{ji} = 1/2$). На главной диагонали ставятся единицы.

Пусть эксперт заполнил матрицу парных сравнений следующим образом (таблица 4.16).

Таблица 4.16 – Матрица парных сравнений

	A1	A2	A3	A
A1	1	7	3	9
A2	1/7	1	1/5	3
A3	1/3	5	1	5
A4	1/9	1/3	1/5	1

Здесь, например, элемент $X_{14} = 9$ означает, что реклама на телевидении, по мнению эксперта, явно более эффективна, чем реклама на стендах. Элемент $X_{23} = 1/5$ означает, что реклама на радио менее эффективна, чем реклама в газетах. Элемент $X_{24} = 3$ означает, что реклама на радио немного более эффективна, чем реклама на стендах.

2. Находят цены альтернатив – средние геометрические строк матрицы:

$$C_i = \sqrt[N]{\prod_{j=1}^N X_{ij}}, \quad i=1, \dots, N,$$

т.е. элементы строки перемножаются, и из их произведения извлекается корень N -й степени.

Для данного примера:

$$C_1 = \sqrt[4]{1 \cdot 7 \cdot 3 \cdot 9} = 3,71, \quad C_2 = \sqrt[4]{(1/7) \cdot 1 \cdot (1/5) \cdot 3} = 0,54, \quad C_3 = 1,7, \quad C_4 = 0,29$$

3. Находится сумма цен альтернатив:

$$\text{В данном примере } C = 3,71 + 0,54 + 1,7 + 0,29 = 6,24$$

4. Находятся веса альтернатив:

$$V_1 = 3,71/6,24 = 0,595; \quad V_2 = 0,54/6,24 = 0,087; \quad V_3 = 1,7/6,24 = 0,272; \\ V_4 = 0,29/6,24 = 0,047.$$

Наиболее предпочтительный по мнению эксперта, является альтернатива, имеющая максимальный вес.

Таким образом, по мнению эксперта, наиболее эффективной является реклама на телевидении, следующая за ней – реклама в газетах, менее эффективна реклама на радио, наименее эффективна реклама на стендах.

Для данного метода возможна проверка экспертных оценок на непротиворечивость. Проверка позволяет выявить ошибки, которые мог допустить эксперт при заполнении матрицы парных сравнений. Ошибки (противоречия) могут быть следующими: например, эксперт указывает, что 1-я альтернатива хуже 2-й, 2-я хуже 3-й, и в то же время 1-я лучше 3-й.

Рассмотрим проверку на непротиворечивость в задаче о выборе рекламы.

1. Находят суммы столбцов матрицы парных сравнений

$$R_j = \sum_{i=1}^N X_{ij}, \quad j=1, \dots, N.$$

$$R_1 = (1+1/7+1/3+1/9) = 1,588; \quad R_2 = 13,333; \quad R_3 = 4,4; \quad R_4 = 18.$$

2. Рассчитывается вспомогательная величина λ путем суммирования произведений сумм столбцов матрицы на веса альтернатив

$$\lambda = \sum_{j=1}^N R_j \cdot V_j.$$

$$\lambda = 1,588 \cdot 0,594 + 13,333 \cdot 0,087 + 4,4 \cdot 0,272 + 18 \cdot 0,047 = 4,07.$$

3. Находится величина, называемая индексом согласованности (ИС): $ИС = (\lambda - N)/(N - 1)$.

$$\text{Для данного примера } ИС = (4,07 - 4)/(4 - 1) = 0,023.$$

4. В зависимости от размерности матрицы парных сравнений находится величина согласованности (СлС). Значения СлС приведены в таблице 4.17

Таблица 4.17 – Значения СлС

Размерность матрицы	3	4	5	6	7	8	9	10
СлС	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49

В данном примере (для $N = 4$) $СлС = 0,90$.

5. Находится отношение согласованности: $ОС = ИС/СлС$. Если ОС превышает 0,2, то требуется уточнение матрицы парных сравнений.

В данном примере $ОС = 0,023/0,9 = 0,024$. Таким образом, уточнение экспертных оценок не нужно.

4.2.8 Метод ранга

Метод основан на балльных оценках альтернатив, указываемых несколькими экспертами. Каждый из экспертов (независимо от других) оценивает альтернативы по некоторой шкале (обычно – 10-балльной). Чем более предпочтительной (по мнению эксперта) является альтернатива, тем более высокий балл для нее указывается.

Рассмотрим этот метод на примере (оценка влияния факторов на производительность труда)

1. Каждый эксперт указывает оценки альтернатив по 10-балльной шкале. Оценки, указанные экспертами, сводятся в матрицу размером $M \times N$, где M – число экспертов, N – число альтернатив.

Обозначим эти оценки, как X_{ij} , $i = 1, \dots, M$, $j = 1, \dots, N$.

Пусть в рассматриваемом примере получена следующая матрица (оценок) – таблица 4.18.

Таблица 4.18 – Матрица оценок

Эксперты	Альтернативы (факторы)				
	A1	A2	A3	A4	A5
1	10	10	7	2	6
2	10	9	10	4	6
3	10	8	10	3	7
4	9	10	6	2	9

Здесь, например, первый эксперт считает, что наибольшее влияние на производительность труда оказывает уровень профессиональной подготовки рабочих и соблюдение технологической дисциплины; менее важный фактор – эффективность материальных стимулов, еще менее важный – технологическое перевооружение; значительно менее важный фактор – эффективность организации соревнования.

2. Находятся суммарные оценки альтернатив всеми экспертами:

$$C_j = \sum_{i=1}^M X_{ij}, \quad j=1, \dots, N.$$

В данном примере $C_1 = 10 + 10 + 10 + 9 = 39$, $C_2 = 10 + 9 + 8 + 10 = 37$, $C_3 = 33$, $C_4 = 11$, $C_5 = 28$.

3. Находится сумма всех оценок:
 $C = \sum_{j=1}^N C_j$
 В примере $C = 39 + 37 + 33 + 11 + 28 = 148$.

4. Находятся веса альтернатив:

$$V_j = C_j / C, \quad j=1, \dots, N$$

Наиболее предпочтительной, по мнению экспертов, является альтернатива, имеющая максимальный вес.

В данном примере $V_1 = 39/148 = 0,26$; $V_2 = 37/148 = 0,25$; $V_3 = 33/148 = 0,22$; $V_4 = 11/148 = 0,07$; $V_5 = 28/148 = 0,19$.

Таким образом, наиболее важным фактором, влияющим на производительность труда, признается уровень профессиональной подготовки рабочих; следующий по важности фактор (очень близкий к первому) – соблюдение технологической дисциплины; немного менее важный – технологическое перевооружение. Наименее важным фактором (существенно менее важным по сравнению с другими) оказывается эффективность соревнования.

Для данного метода также возможна проверка экспертных оценок на согласованность. Для этого рассчитываются дисперсии (оценки разброса) оценок для каждого эксперта и каждой альтернативы. Расчет выполняется в следующем порядке.

Находятся средние оценки каждой альтернативы:

$$X_j = \frac{1}{M} \sum_{i=1}^M X_{ij}, \quad j=1, \dots, N.$$

В данном примере $X_1 = 39/4 = 9,75$; $X_2 = 37/4 = 9,25$; $X_3 = 33/4 = 8,25$; $X_4 = 11/4 = 2,75$; $X_5 = 28/4 = 7$.

Находятся дисперсии оценок каждого эксперта:

$$D_{si} = \frac{1}{N-1} \sum_{j=1}^N (X_{ij} - X_j)^2, \quad i=1, \dots, M.$$

Эта величина показывает отклонение оценок, указанных i -м экспертом для альтернатив, от средних оценок этих альтернатив. Чем больше эта величина, тем больше отличие i -го эксперта от остальных экспертов.

В данном примере:

$$D_{s1} = 1/4 * ((10-9,75)^2 + (10-9,25)^2 + (7-8,25)^2 + (2-2,75)^2 + (6-7)^2) = 0,94.$$

$$D_{s2} = 1/4 * ((10-9,75)^2 + (9-9,25)^2 + (10-8,25)^2 + (4-2,75)^2 + (6-7)^2) = 1,44.$$

$$D_{s3} = 1,19. \quad D_{s4} = 2,69.$$

Находятся дисперсии оценок каждой альтернативы:

$$D_{aj} = \frac{1}{M-1} \sum_{i=1}^M (X_{ij} - X_j)^2, \quad j=1, \dots, N.$$

Эта величина показывает различие оценок, указанных экспертами для j -й альтернативы. Чем больше эта величина, тем больше расхождение мнений экспертов в отношении данной альтернативы.

В данном примере:

$$D_{a1} = 1/3 * ((10-9,75)^2 + (10-9,75)^2 + (9-9,75)^2) = 0,25.$$

$$D_{a2} = 1/3 * ((10-9,25)^2 + (9-9,25)^2 + (10-9,25)^2) = 0,917.$$

$$D_{a3} = 4,25. \quad D_{a4} = 0,917. \quad D_{a5} = 2.$$

Если величина D_{si} оказывается большей (оценки i -го эксперта сильно отличаются от оценок, указанных другими экспертами), то i -му эксперту предлагается обосновать свои оценки. Если большей оказывается величина D_{aj} (оценки j -й альтернативы у экспертов сильно отличаются), то следует проанализировать причины таких расхождений.

В данном примере, возможно, следует предложить обосновать свои оценки четвертому эксперту. Кроме того, следует обратить внимание на разброс оценок третьей альтернативы.

4.2.9 Метод ЭЛЕКТРА

Метод предназначен для решения задач, в которых из имеющегося множества альтернатив требуется выбрать заданное количество лучших альтернатив с учетом их оценок по нескольким критериям, а также важности этих критериев.

Принцип работы метода следующий. Для каждой пары альтернатив (A_j и A_k) выдвигается предположение (гипотеза) о том, что альтернатива A_j лучше, чем A_k . Затем для каждой пары альтернатив находятся два индекса: индекс согласия (величина, подтверждающая предположение о превосходстве A_j над A_k) и индекс несогласия (величина, опровергающая это предположение). На основе анализа этих индексов выбирается одна или несколько лучших альтернатив («ядро» альтернатив).

Рассмотрим реализацию одного из вариантов метода ЭЛЕКТРА на следующем примере.

Пример. Предприятию требуется приобрести датчики для использования в составе автоматизированной системы управления технологическим процессом механообработки. Имеется возможность приобрести датчики одного из шести типов. Характеристики датчиков приведены в таблице 4.19.

Таблица 4.19 – Характеристики датчиков

	GD360	BE178A	RON200	RON300	RON800	D200
Стоимость, ден.ед.	1600	1600	2000	6000	2100	1800
Наработка на отказ, ч	3200	500	4000	6500	5000	3500
Условия технического обслуживания	удовл	удовл	отл	отл	хор	отл
Точность, количество отсчетов	3600	2500	5000	5000	4000	3500

Имеются также суждения двух специалистов предприятия в отношении важности критериев.

По мнению первого специалиста, наиболее важный критерий – *стоимость*, следующие по важности (и одинаково важные между собой) – *наработка на отказ и точность*, наименее важный критерий – *условия технического обслуживания*.

По мнению второго специалиста, наиболее важный критерий – *наработка на отказ*, немного менее важный – *стоимость*, следую-

щий по важности – *точность*, еще менее важный критерий – *условия технического обслуживания*.

При этом (в зависимости от окончательного варианта проектируемой АСУТП) предприятию могут потребоваться датчики как одного, так и нескольких типов.

Выберем множество Парето. Выполнив попарное сравнение альтернатив получим, что во множество Парето входят все датчики, кроме BE178A.

Для удобства записи введем обозначения критериев: стоимость – K1, наработка на отказ – K2, условия технического обслуживания – K3, точность – K4.

Обозначим также альтернативы GD360, RON200, RON300, RON800 и D200 как A1, A2, A3, A4, A5. В таблице 4.20 приведены оценки альтернатив, вошедших во множество Парето.

Таблица 4.20 – Оценки альтернатив

	A1	A2	A3	A4	A5
K1	1600	2000	6000	2100	1800
K2	3200	4000	6500	5000	3500
K3	удовл	отл	отл	хор	отл
K4	3600	5000	5000	4000	3500

Найдем веса критериев. В данной задаче важность критериев оценивается несколькими (двумя) экспертами. Поэтому для определения весов необходимо применить один из коллективных методов экспертных оценок. Пусть эксперты согласны представить свои суждения в виде балльных оценок. В этом случае можно воспользоваться методом ранга. Предположим, что от экспертов получена следующая матрица оценок (таблица 4.21).

Таблица 4.21 – Матрица оценок

	K1	K2	K3	K4
1-й эксперт	10	8	5	8
2-й эксперт	9	10	6	7

Выполнив обработку экспертных оценок по методу ранга, получим веса критериев: $V_1 = 0,30$; $V_2 = 0,29$; $V_3 = 0,17$; $V_4 = 0,24$.

Выбор лучших альтернатив по методу ЭЛЕКТРА реализуется в следующем порядке.

1. Оценки альтернатив приводятся к безразмерному виду. Эта операция может выполняться разными способами, например, так же, как в методике экспресс-анализа альтернатив. Безразмерные оценки приведены в таблице 4.22.

Таблица 4.22 – Приведение оценок к безразмерному виду

	A1	A2	A3	A4	A5
K1	1,00	0,80	0,27	0,76	0,89
K2	0,49	0,62	1,00	0,77	0,54
K3	0,60	1,00	0,90	0,80	1,00
K4	0,72	1,00	1,00	0,8	0,70

2. Определяются индексы согласия C_{jk} , $j = 1, \dots, N$, $k = 1, \dots, N$ (где N – количество альтернатив). Индекс согласия отражает степень согласия с предположением о том, что j -я альтернатива лучше k -й.

В рассматриваемой реализации метода ЭЛЕКТРА индексы согласия находятся по формуле:

$$C_{jk} = \sum_{i \in K^+} V_i, \quad j=1, \dots, N, k=1, \dots, N,$$

где V_i – веса критериев; K^+ – подмножества критериев, по которым j -я альтернатива не хуже k -й.

Таким образом, индекс согласия C_{jk} находится как сумма весов критериев, по которым j -я альтернатива не хуже k -й. Чем больше индекс согласия, тем более выражено превосходство j -й альтернативы над k -й.

Индексы согласия для данной задачи приведены в таблице 4.23.

Приведем примеры расчета индексов согласия. Найдем, например, индекс согласия C_{12} (оценку согласия с предположением о превосходстве альтернативы A1 над A2). Альтернатива A1 (GD360) не хуже альтернативы A2 (RON200) только по критерию K1 (стоимость). Его вес равен 0,30; таким образом $C_{12} = 0,30$. Аналогично найдем индекс согласия C_{21} . Альтернатива A2 не хуже, чем A1, по критериям K3, K2, K4, поэтому $C_{21} = 0,29 + 0,17 + 0,24 = 0,70$.

Найдем индекс согласия C_{23} . Альтернатива A2 (RON200) не хуже альтернативы A3 (RON300) по критериям K1, K3, K4 (по критери-

ям К1 и К3 альтернатива А2 лучше, чем А3, а по критерию К4 они одинаковы). Таким образом, $C_{23} = 0,30 + 0,17 + 0,24 = 0,71$.

Аналогично найдем индекс согласия С32. Альтернатива А3 не хуже, чем А2, по критериям К2 и К4, поэтому $C_{32} = 0,29 + 0,24 = 0,53$.

Таблица 4.23 – Индексы согласия

	A1	A2	A3	A4	A5
A1	—	0,30	0,30	0,30	0,54
A2	0,70	—	0,71	0,71	0,70
A3	0,70	0,53	—	0,70	0,53
A4	0,70	0,29	0,30	—	0,53
A5	0,46	0,47	0,47	0,47	—

3. Определяются индексы несогласия $D_{jk}, j = 1, \dots, N, k = 1, \dots, N$. Индекс несогласия отражает степень несогласия с предложением о том, что j -я альтернатива лучше k -й. Индексы D_{jk} находятся по формуле

$$D_{jk} = \max_{l \in K^-} (P_{lk} - P_{lj}), j = 1, \dots, N, k = 1, \dots, N,$$

где P_{lk}, P_{lj} – безразмерные оценки альтернатив; K^- – подмножество критериев, по которым j -я альтернатива не превосходит k -ю.

Таким образом, индекс несогласия D_{jk} находится как максимальная из разностей оценок по критериям, по которым j -я альтернатива не лучше k -й. Чем больше индекс несогласия, тем менее выражено превосходство j -й альтернативы над k -й.

Индексы несогласия для данной задачи приведены в таблице 4.24.

Таблица 4.24 – Индексы несогласия

	A1	A2	A3	A4	A5
A1	—	0,40	0,51	0,28	0,40
A2	0,20	—	0,38	0,15	0,09
A3	0,73	0,53	—	0,49	0,62
A4	0,24	0,20	0,23	—	0,20
A5	0,11	0,30	0,46	0,23	—

Приведем примеры расчетов индексов несогласия. Найдем индекс несогласия D_{12} (оценку несогласия с предложением о превосходстве альтернативы А1 над А2). Альтернатива А1 (GD360) не имеет превосходства над А2 (RON200) по критериям К2, К3, К4. Разности

безразмерных оценок по этим критериям следующие: $0,62 - 0,49 = 0,13$; $1 - 0,6 = 0,4$; $1 - 0,72 = 0,28$. Таким образом, $D_{12} = 0,4$.

Аналогично найдем индекс несогласия D_{21} . Альтернатива А2 не имеет превосходства над А1 только по критерию К1. Разность оценок по этому критерию: $1 - 0,8 = 0,2$. Таким образом, $D_{21} = 0,2$.

4. Для каждой альтернативы находится предельное значение индекса согласия:

$$C_j = \min_k C_{jk}, j = 1, \dots, N.$$

Таким образом, предельное значение индекса согласия для j -й альтернативы находится как минимальный элемент j -й строки матрицы индексов согласия. Эта величина отражает степень согласия с предложением о том, что j -я альтернатива имеет превосходство над всеми другими альтернативами.

Для рассматриваемого примера

$$C_1 = 0,3; C_2 = 0,7; C_3 = 0,53; C_4 = 0,29; C_5 = 0,46.$$

5. Для каждой альтернативы находим предельное значение индекса несогласия

$$D_j = \max_k D_{jk}, j = 1, \dots, N.$$

Таким образом, предельное значение индекса несогласия для j -й альтернативы находится как максимальный элемент j -й строки матрицы индексов несогласия. Эта величина отражает степень несогласия с предложением о превосходстве j -й альтернативы над другими альтернативами.

Для рассматриваемого примера

$$D_1 = 0,51; D_2 = 0,38; D_3 = 0,73; D_4 = 0,24; D_5 = 0,46.$$

6. Выделяются лучшие альтернативы («ядро» альтернатив), удовлетворяющие условиям:

$$C_j > C^*, D_j < D^*,$$

где C^*, D^* – пороговые значения индексов согласия и несогласия. Эти величины назначаются в зависимости от того, какое количество альтернатив требуется выбрать. Обычно сначала принимаются пороговые значения $C^* = 0,5$ и $D^* = 0,5$; затем они изменяются в соответствии с количеством отбираемых альтернатив. Выбираются альтернативы, удовлетворяющие обоим условиям.

Пусть в рассматриваемом примере требуется выбрать один тип датчиков. Назначим пороговые значения $C^* = 0,5$ и $D^* = 0,5$. Усло-

вию $C_j > C^*$ удовлетворяют альтернативы А2 и А3, условию $D_j < D^*$ – альтернативы А2, А4 и А5. Таким образом, выбирается альтернатива А2, т.е. датчик RON200.

Пусть требуется выбрать два типа датчиков. Снизим пороговое значение индекса согласия до $C^* = 0,45$, оставив значение $D^* = 0,5$. Тогда условию $C_j > C^*$ будут удовлетворять альтернативы А2, А3 и А5, условию $D_j < D^*$ – альтернативы А2, А4 и А5. Таким образом, выбираются альтернативы А2 и А5, т.е. датчики RON200 и D200.

Примечание. Выбор ядра альтернатив в данном методе не всегда является однозначным. Например, если в рассматриваемом примере назначить пороговые значения $C^* = 0,5$ и $D^* = 0,8$, то условию $C_j > C^*$ будут соответствовать альтернативы А2 и А3, а условию $D_j < D^*$ – все альтернативы. Таким образом, в качестве двух лучших альтернатив будут отобраны А2 и А3 (датчики RON200 и RON300). Отбирая несколько вариантов ядра, ЛПР может самостоятельно выбрать окончательный вариант решения.

4.2.10 Модифицированный алгоритм Кемени-Снелла

Рассматриваемый алгоритм предназначен для ранжирования альтернатив с учетом их оценок по нескольким критериям.

Основное преимущество алгоритма – возможность анализа и выбора альтернатив, оцениваемых по критериям различных видов: числовым, качественным, «да-нет» и т.д. Алгоритм также позволяет учитывать суждения ЛПР о важности критериев.

Алгоритм основан на ранжировании и попарном сравнении альтернатив по каждому критерию.

Пример. Крупная фирма предполагает создать совместное предприятие за рубежом. Рассматривается возможность создания совместного предприятия в одной из пяти стран. Характеристики этих стран приведены в таблице 4.25.

По мнению руководства фирмы, при выборе страны следует прежде всего учитывать законодательство о совместных предприятиях. Немного менее важный критерий – спрос на внутреннем рынке. Еще немного менее важные критерии – наличие сырья и затраты на подготовку персонала.

Таблица 4.25 – Характеристики стран

Страна	C1	C2	C3	C4	C5
Законодательство о совместных предприятиях	неблагоприятное	благоприятное	удовлетворительное	благоприятное	неблагоприятное
Наличие сырья	нет	есть	нет	нет	есть
Спрос на продукцию предприятия на внутреннем рынке, млн ден.ед./год	5	6	4	8	6
Затраты на подготовку персонала, млн ден.ед.	1,2	2,5	1,5	1,8	1,3

В данной задаче для оценки альтернатив (стран) используются критерии различных видов.

Критерий «законодательство о совместных предприятиях» – качественный (причем шкала оценок отличается от пятибалльной, что затрудняет перевод критерия в числовую форму).

Критерий «наличие сырья» имеет вид «да-нет». Остальные критерии – количественные.

Для решения таких задач целесообразно применять модифицированный алгоритм Кемени-Снелла.

Прежде чем приступать к выбору решения с использованием данного алгоритма (как и любого другого метода), следует отобрать множество Парето, т.е. множество перспективных альтернатив.

Выполнив попарное сравнение альтернатив, получим, что во множество Парето входят все пять альтернатив.

Выбор альтернативы на основе модифицированного алгоритма Кемени-Снелла реализуется в следующем порядке.

1. С помощью одного из методов экспертных оценок находятся веса критериев, представляющие собой числовые оценки их важности.

В данном случае имеется только одно суждение о важности критериев. Поэтому следует применить один из индивидуальных методов экспертных оценок.

Воспользуемся алгоритмом Саати. Обозначим критерии:

K1 – законодательство о совместных предприятиях, K2 – наличие сырья, K3 – спрос на внутреннем рынке, K4 – затраты на подготовку персонала.

Выполним попарное сравнение альтернатив, как показано в таблице 4.26.

Таблица 4.26 – Парное сравнение альтернатив

	K1	K2	K3	K4
K1	1	5	2	5
K2	1/5	1	1/3	1
K3	1/2	3	1	3
K4	1/5	1	1/3	1

Выполнив обработку матрицы парных сравнений по алгоритму Саати, найдем веса критериев: $V1 = 0,52$; $V2 = 0,10$; $V3 = 0,28$; $V4 = 0,10$.

2/ Выполняется ранжирование альтернатив по каждому из критериев. При этом лучшая альтернатива по данному критерию получает оценку (ранг) 1, следующая за ней – оценку 2, и т.д. Если альтернативы по данному критерию одинаковы, то они получают одинаковые оценки. Результаты ранжирования сводятся в матрицу. Для данной задачи матрица ранжирований приведена в таблице 4.27.

Таблица 4.27 – Матрица ранжирований

	C1	C2	C3	C4	C5
K1	3	1	2	1	3
K2	2	1	2	2	1
K3	3	2	4	1	2
K4	1	5	3	4	2

3. На основе ранжирования альтернатив по каждому из критериев составляется матрица парных сравнений. Всего составляется M таких матриц, где M – число критериев. Матрицы заполняются по правилам, приведенным в таблице 4.28.

Таблица 4.28 – Матрица парных сравнений

R_{jk}^i	Значение
1	По i -му критерию j -я альтернатива лучше k -й.
-1	По i -му критерию j -я альтернатива хуже k -й.
0	По i -му критерию j -я и k -я альтернативы одинаковы.

Здесь i – номер матрицы (номер критерия).

Для рассматриваемой задачи матрицы парных сравнений по критериям K1-K4 приведены в таблицах 4.29–4.32.

Таблица 4.29 – По критерию К1 Таблица 4.30 – По критерию К2

	C1	C2	C3	C4	C5
C1	—	-1	-1	-1	0
C2	1	—	1	0	1
C3	1	-1	—	-1	1
C4	1	0	1	—	1
C5	0	-1	-1	-1	—

	C1	C2	C3	C4	C5
C1	—	-1	0	0	-1
C2	1	—	1	1	0
C3	0	-1	—	0	-1
C4	0	-1	0	—	-1
C5	1	0	1	1	—

Таблица 4.31 – По критерию К3 Таблица 4.32 – По критерию К4

	C1	C2	C3	C4	C5
C1	—	-1	1	-1	-1
C2	1	—	1	-1	0
C3	-1	-1	—	-1	-1
C4	1	1	1	—	1
C5	1	0	1	-1	—

	C1	C2	C3	C4	C5
C1	—	1	1	1	1
C2	-1	—	-1	-1	-1
C3	-1	1	—	1	-1
C4	-1	1	-1	—	-1
C5	-1	1	1	1	—

Здесь, например, в таблице 4.29 элемент $R_{12}^1 = -1$ означает, что по критерию «законодательство о совместных предприятиях» страна C1 хуже, чем C2 (в стране C1 Оно неблагоприятное, в C2 – благоприятное). Элемент $R_{23}^1 = 1$, означает, что по законодательству о совместных предприятиях страна C2 лучше, чем C3; $R_{24}^1 = 0$ означает, что по этому критерию страны C2 и C4 одинаковые (в обеих странах законодательство благоприятное)/

4. Составляем матрицу потерь. Размерность матрицы – NхN, где N – количество альтернатив. Элементы матрицы потерь рассчитываются по следующей формуле:

$$R_{jk} = \sum_{i=1}^M V_i \cdot |R_{jk}^i - 1|, \quad j=1, \dots, N, k=1, \dots, N.$$

Матрица потерь рассматриваемой задачи приведена в таблице 4.33.

Таблица 4.33 – Матрица потерь

	C1	C2	C3	C4	C5
C1	—	1,80	1,14	1,70	1,28
C2	0,20	—	0,20	1,28	0,58
C3	0,86	1,80	—	1,70	0,96
C4	0,30	0,72	0,30	—	0,40
C5	0,72	1,42	1,04	1,60	—

Приведем примеры расчета некоторых элементов матрицы.

$$R_{12} = V_1 \cdot |R_{12}^1 - 1| + V_2 \cdot |R_{12}^2 - 1| + V_3 \cdot |R_{12}^3 - 1| + V_4 \cdot |R_{12}^4 - 1| = \\ = 0,52 \cdot |-1 - 1| + 0,10 \cdot |-1 - 1| + 0,28 \cdot |-1 - 1| + 0,10 \cdot |1 - 1| = 1,80.$$

$$R_{25} = V_1 \cdot |R_{25}^1 - 1| + V_2 \cdot |R_{25}^2 - 1| + V_3 \cdot |R_{25}^3 - 1| + V_4 \cdot |R_{25}^4 - 1| = \\ = 0,52 \cdot |1 - 1| + 0,10 \cdot |0 - 1| + 0,28 \cdot |0 - 1| + 0,10 \cdot |-1 - 1| = 0,58.$$

Смысл элементов матрицы потерь следующий: чем больше элемент R_{jk} , тем больше отставание j -й альтернативы от k -ой.

5 Выполняется предварительное ранжирование альтернатив. Для этого находятся суммы строк матрицы потерь. Смысл этих сумм следующий: сумма j -й строки представляет собой оценку отставания j -й альтернативы от всех остальных альтернатив.

Альтернатива, которой соответствует минимальная сумма, предварительно считается лучшей. Строка и столбец этой альтернативы исключаются из матрицы потерь.

Суммирование строк матрицы потерь и исключение альтернатив выполняются до тех пор, пока не будет исключена вся матрица. Чем раньше исключена альтернатива, тем она лучше.

$$P_1 = 1,80 + 1,14 + 1,70 + 1,28 = 5,92,$$

$$P_2 = 0,20 + 0,20 + 1,28 + 0,58 = 2,26,$$

$$P_3 = 0,86 + 1,80 + 1,70 + 0,96 = 5,33,$$

$$P_4 = 0,30 + 0,72 + 0,30 + 0,40 = 1,71,$$

$$P_5 = 0,72 + 1,42 + 1,04 + 1,60 = 4,78.$$

Выполним предварительное ранжирование для рассматриваемой задачи. Найдем суммы строк матрицы потерь:

Предварительно лучшей считается альтернатива С4. Она исключается из матрицы потерь. Сокращенная матрица потерь приведена в таблице 4.34.

Таблица 4.34 – Сокращенная матрица потерь

	C1	C2	C3	C5
C1	—	1,80	1,14	1,28
C2	0,20	—	0,20	0,58
C3	0,86	1,80	—	0,96
C5	0,72	1,42	1,04	—

Суммы строк этой матрицы : $P_1=4,22$; $P_2=0,98$; $P_3=3,63$; $P_5=3,17$. Исключается альтернатива С2. Сокращенная матрица потерь приведена в таблице 4.35.

Таблица 4.35 – Сокращенная матрица потерь 2

	C1	C3	C5
C1	—	1,14	1,28
C3	0,86	—	0,96
C5	0,72	1,04	—

Суммы строк этой матрицы $P_1 = 2,42$; $P_3 = 1,83$; $P_5 = 1,76$. Исключается альтернатива C5. Сокращенная матрица потерь приведена в таблице 4.36.

Таблица 4.36 – Сокращенная матрица потерь

	C1	C3
C1	—	1,14
C3	0,86	—

Суммы строк этой матрицы: $P_1=1,14$; $P_3=0,86$. Лучшая альтернатива – C3.

Предварительное ранжирование альтернатив: C4, C2, C5, C3, C1.

Выполняется окончательное ранжирование альтернатив. Для этого альтернативы сравниваются попарно, начиная с конца предварительного ранжирования. Если сравниваются j -я и k -я альтернативы (при этом j -я альтернатива в предварительном ранжировании находится выше k -й) и выполняется условие $R_{jk} \leq R_{kj}$ (R_{jk} , R_{kj} – элементы матрицы потерь), то альтернативы остаются в ранжировании на прежних местах (j -я альтернатива лучше k -ой). Если $R_{jk} > R_{kj}$, то альтернативы меняются местами (j -я альтернатива хуже k -ой).

Выполним окончательное ранжирование для данной задачи.

Сравниваем C3 и C1. $R_{31} = 0,86$; $R_{13} = 1,14$. Так как $R_{31} < R_{13}$, альтернативы остаются на своих местах (C3 выше, чем C1).

Сравниваем C5 и C3. $R_{53} = 1,04$; $R_{35} = 0,96$. Так как $R_{53} > R_{35}$, альтернативы меняются местами: альтернатива C3 признается лучшей, чем C5. Ранжирование теперь имеет следующий вид: C4, C2, C3, C5, C1.

Сравниваем C2 и C3. $R_{23} = 0,20$; $R_{32} = 1,80$. Так как $R_{23} < R_{32}$, альтернативы остаются на своих местах.

Сравниваем C2 и C4. $R_{42} = 0,72$; $R_{24} = 1,28$. Так как $R_{42} < R_{24}$, альтернативы остаются на своих местах.

Таким образом, окончательное ранжирование альтернатив следующее: C4, C2, C3, C5, C1. Лучший вариант действий для фирмы – создание совместного предприятия в стране обозначенной как C4.

Библиотека ГГТУ им. П.О.Сухого

5 БАЗЫ ЗНАНИЙ В САПР

5.1 Модели представления знаний

5.1.1 Знания и их классификация

Знания основаны на данных, полученных эмпирическим путем. Они представляют собой результат мыслительной деятельности человека, направленной на обобщение его опыта, полученного в результате практической деятельности.

Знания извлекаются, например, из книг, статей, мыслей специалистов. В печатных изданиях они обычно представляются текстами на естественном языке, формулами, схемами или графически.

Знания – это закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области.

Знания по степени научности подразделяются на *житейские* (обыденные знания или здравый смысл) и *научные*.

Житейские знания, как правило, сводятся к констатации фактов и их описанию, тогда как научные знания поднимаются до уровня объяснения фактов, осмысления их в системе понятий данной науки, включаются в состав теории.

Научному знанию присущи логическая обоснованность, доказательность, воспроизводимость познавательных результатов.

Научные знания могут быть классифицированы на:

- *поверхностные* (эмпирические) – знания о видимых взаимосвязях между отдельными событиями и фактами в предметной области;
- *глубинные* (теоретические) – абстракции, аналогии, схемы, отображающие структуру и природу процессов, протекающих в предметной области. Эти знания объясняют явления и могут использоваться для прогнозирования поведения объектов.

Знания по местонахождению классифицируют на:

- *личностные (неявные, скрытые)* – знания людей;
- *формализованные (явные)* – например, знания в документах, на компакт-дисках, в Интернете.

Кроме того, знания традиционно делят *по природе* на:

- процедурные;
- декларативные.

Чаще всего *процедурные* знания (программы) создаются программистами и помещаются в библиотеки готовых и отлаженных подпрограмм. Это знания, «растворенные» в алгоритмах разного рода. Процедурные знания дают представления о средствах и путях получения новых знаний, проверки знаний (например, знания, приобретаемые методом мозгового штурма для поиска новых идей). Они имеют активную природу. Для их изменения требуется изменять программы.

Процедурные знания добываются на основе следующих видов знаний:

- *алгоритмических знаний* – используемых в реализациях современных системах автоматизированного проектирования;
- *лингвосемантических* знаний – описательных, плохо формализуемых;
- *семантических (языковых)* знаний – реализуемых через трансляторы или интерпретаторы языков программирования, сложно формализуемых;
- *концептуальных* знаний – понятийных, используемых в системах программирования «вручную»;
- *теоретических (логических)* знаний – используемых также «вручную», плохо формализуемых;
- *фактографических* знаний – данных из базы, используемых через системы управления базами данных и по существу уже формализованных;
- *кибернетических* знаний – технологических, системных или конструкторских, плохо формализуемых, которые учитываются (или не учитываются) во многих процессах организационного проектирования.

Из перечисления видно, что многие виды знаний чаще всего только косвенно учитываются при программировании.

Декларативные знания представляются множеством утверждений, не зависящих от того, где они применяются. Эти знания приближены к данным, фактам. Например, «высшее учебное заведение есть совокупность факультетов, а каждый факультет, в свою очередь, есть совокупность кафедр».

Сегодня знания приобрели чисто *декларативную* форму. То есть, знаниями считаются предложения, записанные на *языках представления знаний*. Это языки, предназначенные для описания предметных областей. Универсальным языком представления знаний является естественный язык. Но его применение для машинного представления знаний наталкивается на ряд препятствий, главным из которых является отсутствие формальной семантики естественного языка. *Семантика* – это смысловое значение единиц языка.

5.1.2 Понятие базы знаний

База знаний (knowledge base) – семантическая модель, описывающая предметную область и позволяющая отвечать на такие вопросы из этой предметной области, ответы на которые в явном виде не присутствуют в базе.

База знаний включает один или несколько специальным образом организованных файлов, хранящих систематизированную совокупность понятий, правил и фактов, относящихся к некоторой предметной области. Например, база знаний в области медицины содержит накопленные медицинскими специалистами сведения о связях между болезнями, их симптомами и порождающими причинами, описания стереотипных ситуаций при течении болезни и рекомендуемых или предпринятых лечебных действиях и их результатах и т. п.

Содержимое баз знаний оформляется, связывается между собой и представляется таким образом, чтобы на его основе можно было с помощью специальных программ осуществлять рассуждения и делать выводы, получая сведения, которые в явном виде могут не присутствовать в базе знаний. Так, на основе базы знаний в области медицины разрабатываются диагностические и прогнозирующие медицинские системы.

Для построения базы знаний применяются методы *искусственного интеллекта*, языки представления знаний и интеллектуальный интерфейс. По мнению английского математика А. Тьюринга искусственный интеллект можно определить как научную дисциплину, которая имитирует интеллектуальные способности человека с помощью ЭВМ.

Искусственный интеллект (artificial intelligence) – раздел информатики, изучающий возможность обеспечения разумных рассуждений и действий с помощью вычислительных систем и иных ис-

кусственных устройств. При этом в большинстве случаев заранее неизвестен алгоритм решения задачи.

К реализации искусственного интеллекта подходят с точки зрения моделирования человеческой интеллектуальности. В рамках искусственного интеллекта различают два основных направления:

- *символьное* (семиотическое, нисходящее) основано на моделировании высокоуровневых процессов мышления человека, на представлении и использовании знаний;
- *нейрокибернетическое* (нейросетевое, восходящее) основано на моделировании отдельных низкоуровневых структур мозга (нейронов).

Сверхзадачей искусственного интеллекта является построение компьютерной *интеллектуальной системы*, которая обладала бы уровнем эффективности решений неформализованных задач, сравнимым с человеческим или превосходящим его. На данный момент не существует систем искусственного интеллекта, однозначно отвечающих основным задачам, обозначенным выше.

База знаний является основным компонентом интеллектуальных систем: информационных, обучающих, систем программирования, а также *экспертных систем*.

Экспертная система (expert system) – компьютерная программа, способная заменить специалиста-эксперта в решении проблемной ситуации.

Экспертная система анализирует ситуацию и, в зависимости от ее направленности, дает рекомендации по разрешению проблемы.

Похожие с экспертной системой действия выполняет *программа-мастер* (Wizard). Как правило, мастера применяются для интерактивного общения с пользователем. Главное отличие мастеров от экспертных систем – отсутствие базы знаний. В них все действия жестко запрограммированы. Это просто набор форм для заполнения пользователем.

Другие подобные программы – *поисковые* или *справочные системы*. По запросу пользователя они предоставляют наиболее подходящие разделы *базы статей*.

Экспертные системы начали разрабатываться исследователями искусственного интеллекта в 1970-х годах, а в 1980-х получили коммерческое подкрепление.

База знаний создается и используется с помощью *системы управления знаниями* – комплекса программных, языковых и интеллектуальных средств.

5.1.3 Продукционная модель

В реализациях искусственного интеллекта основными моделями представления знаний являются следующие:

- продукционные модели;
- семантические сети;
- фреймы;
- формальные логические модели.

Наибольшее распространение на практике получила продукционная модель.

Продукционная модель – модель, основанная на представлении знания в виде правил «Если (условие), то (действие)».

Под «условием» понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под «действием» – действия, выполняемые при успешном исходе поиска.

При использовании продукционной модели исходные факты – данные – хранятся в *базе фактов*, а набора правил – в *базе знаний*. На основе базы фактов запускается *машина вывода* или *интерпретатор правил*. Это программа, управляющая перебором правил из базы знаний.

Машина вывода выполняет две функции:

- просмотр существенных фактов из рабочей памяти (базы фактов), правил из базы знаний и добавление, по мере возможности, в рабочую память новых фактов;
- определение порядка просмотра и применения правил.

Этот механизм управляет процессом консультации, сохраняя для пользователя информацию о полученных заключениях, и запрашивает у него информацию, когда для срабатывания очередного правила в рабочей памяти недостаточно данных.

В большинстве систем, основанных на знаниях, механизм вывода представляет собой небольшую по объему программу и включает два компонента –компонент вывода и управляющий компонент.

Действие *компонента вывода* основано на применении правила, называемого *modus ponens*:

Если известно, что истинно утверждение А и существует правило вида «Если А, то В», тогда утверждение В также истинно.

Правила срабатывают, когда находятся факты, удовлетворяющие их левой части: если истинна посылка, то должно быть истинно и заключение.

Компонент вывода должен функционировать даже при недостатке информации. Полученные решения могут быть и неточными. Однако система не должна останавливаться из-за того, что отсутствует какая-либо часть входной информации.

Управляющий компонент определяет порядок применения правил и выполняет четыре функции:

- *сопоставление* – образец правила сопоставляется с имеющимися фактами;
- *выбор* – если в конкретной ситуации может быть принято сразу несколько правил, то из них выбирается одно, наиболее подходящее по заданному критерию;
- *срабатывание* – если образец правила при сопоставлении совпал с какими-либо фактами из рабочей памяти, то правило срабатывает;
- *действие* – рабочая память подвергается изменению путем добавления в нее заключения сработавшего правила. Если в правой части правила содержится указание на какое-либо действие, то оно выполняется.

Интерпретатор правил работает циклически (рисунок 5.1). В каждом цикле он просматривает все правила, чтобы выявить те, посылки которых совпадают с известными на данный момент фактами из рабочей памяти.

После выбора правило срабатывает, его заключение заносится в рабочую память, и затем цикл повторяется сначала. В одном цикле может сработать только одно правило. Если несколько правил успешно сопоставлены с фактами, то интерпретатор производит выбор по определенному критерию единственного правила, которое срабатывает в данном цикле.

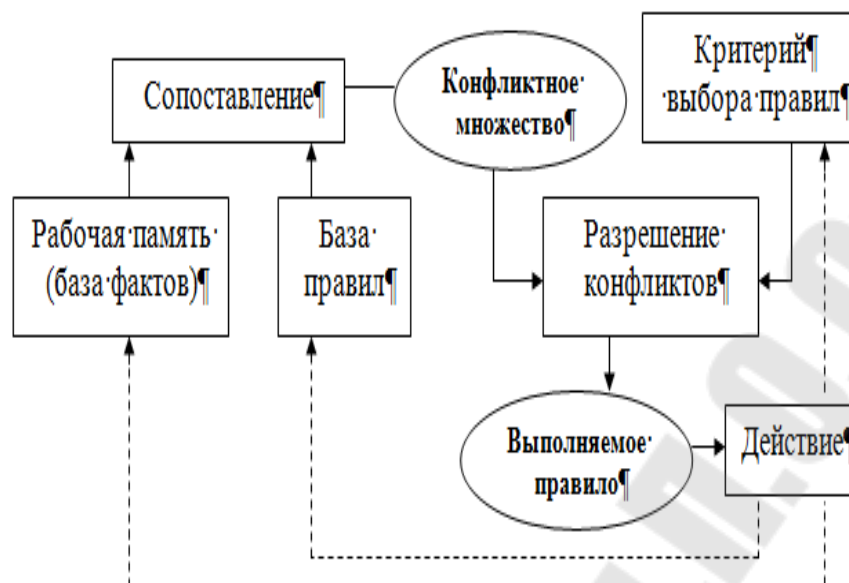


Рисунок 5.1 – Цикл работы интерпретатора

Информация из рабочей памяти последовательно сопоставляется с посылками правил для выявления успешного сопоставления. Совокупность отобранных правил составляет конфликтное множество. Для разрешения конфликта используется критерий. С его помощью выбирается единственное правило. После этого оно срабатывает: факты, образующие заключение правила, заносятся в рабочую память или изменяется критерий выбора конфликтующих правил. Если же в заключение правила входит название какого-нибудь действия, то оно выполняется.

Порядок применения и срабатывания правил зависит от *стратегии вывода*. Она должна определять направление поиска и способ его осуществления.

Направление поиска есть прямое (от данных к поиску цели) и обратное (от цели для ее подтверждения – к данным). *Способ поиска* может быть в глубину, в ширину, по подзадачам, или иначе.

Продукционная модель чаще всего применяется в промышленных экспертных системах. Она *привлекает* разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода.

Продукционная модель обладает тем *недостатком*, что при накоплении достаточно большого числа (порядка нескольких сотен) продукций (правил) они начинают противоречить друг другу.

Имеется большое число программных средств, реализующих продукционный подход. Например, экспертные системы ЭКСПЕРТ, ЭКО и др.

5.1.4 Семантическая сеть

Семантическая сеть реализует идею о том, что любые знания можно представить в виде совокупности объектов (понятий) и связей (отношений между ними).

Семантическая сеть – это ориентированный граф, вершины которого есть понятия, а дуги – отношения между ними.

В качестве *понятий* выступают абстрактные или конкретные объекты.

Наиболее часто в семантических сетях используются следующие *отношения*:

- *связи типа «часть-целое»* (например, «класс-подкласс» «элемент-множество» и т.п.);
- *функциональные связи*, определяемые обычно глаголами (производит, влияет и др.);
- *количественные* (>, <, = и др.);
- *пространственные* (далеко от, близко от, за, под, на и др.);
- *временные* (раньше, позже, в течение и др.);
- *атрибутивные* (иметь свойство, иметь значение и др.);
- *логические* (и, или, не);
- *лингвистические* и др.

Поиск решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети, отражающий поставленный запрос к базе.

Пример семантической сети показан на рисунке 5.2.

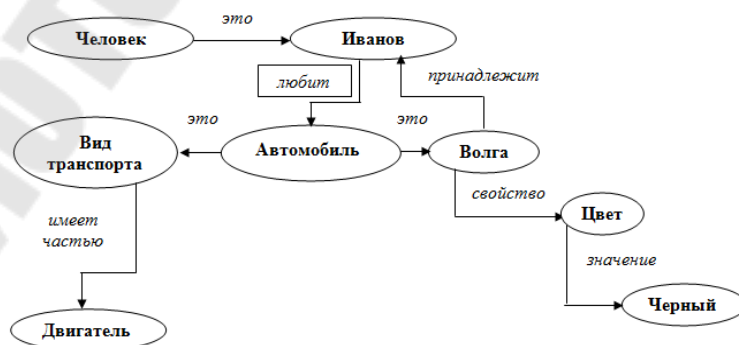


Рисунок 5.2 – Пример семантической сети

Данная модель представления знаний была предложена американским психологом Куиллианом. Основное ее *преимущество* в том, что она более других соответствует современным представлениям об организации долговременной памяти человека. *Недостаток* – это сложность организации процедуры поиска вывода на семантической сети.

Для реализации семантических сетей существуют специальные *сетевые языки*. Например, *NET*, *SIMER+MIR* и др. Широко известны экспертные системы, использующие семантические сети в качестве языка представления знаний, – *PROSPECTOR*, *CASNET*, *TORUS*.

5.1.5 Фреймы

Термин «фрейм» (каркас, рамка) используется в теории представления знаний фреймами, которая была разработана М.Минским в 70-е годы XX века. В ее основе лежит восприятие фактов посредством сопоставления полученной извне информации с рамками, определенными для каждого объекта в памяти человека. *Фрейм – это абстрактный образ для представления некоего стереотипа восприятия.*

Например, произнесенное слово «комната» порождает образ комнаты как жилого помещения с четырьмя стенами, полом, потолком, окнами, дверью, определенной площадью. Из этого описания ничего нельзя убрать (например, убрав окна, получим уже образ чулана). Но в этом описании есть «дырки» или *слоты*. Это незаполненные значения некоторых атрибутов, например, количество окон, цвет стен, высота потолка и др. Такой образ комнаты называется фреймом комнаты.

Фрейм можно определить и как *формализованную модель для отображения образа.*

Различают *фреймы-образцы* (прототипы), хранящиеся в базе знаний, и *фреймы-экземпляры*, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных.

Модель фрейма достаточно универсальна. Она позволяет отобразить все многообразие знаний о мире через:

- *фреймы-структуры*, используемые для обозначения объектов и понятий (заем, залог, вексель);
- *фреймы-роли* (менеджер, кассир, клиент);
- *фреймы-сценарии* (банкротство, собрание акционеров);
- *фреймы-ситуации* (тревога, авария, рабочий режим устройства).

Любой фрейм, представляющий образ, содержит набор атрибутов – слотов, значениями которых являются конкретные данные. Каждый слот имеет имя, уникальное в рамках конкретного фрейма. В качестве примера приведем фрейм «Лекция»:

Лекция	
ПРЕДМЕТ	Информатика
ЛЕКТОР	Крылов И.И.
АУДИТОРИЯ	403
СЛУШАТЕЛИ	107

В данном случае «Лекция» – название фрейма; «ПРЕДМЕТ», «ЛЕКТОР», «АУДИТОРИЯ», «СЛУШАТЕЛИ» – слоты; «Информатика», «Крылов И.И.», «403», «107» – значения слотов.

Кроме того, фрейм может содержать *процедуры*, которые будут выполняться при определенных условиях (при записи или удалении информации из слота, при обращении к слоту, в котором отсутствуют данные и т.д.) С каждым слотом может быть связано любое количество процедур. Процедуры, связанные с определенным слотом фрейма, зависят от конкретной прикладной системы, использующей фреймовые структуры для представления знаний. Если представить, что фрейм «Лекция» используется в системе подготовки расписания занятий, то процедура, вызываемая при внесении значения в слоте «ЛЕКТОР», могла бы уведомлять лектора об изменении расписания; процедуры, вызываемые при изменении значений слотов «АУДИТОРИЯ» и «СЛУШАТЕЛИ», могли бы проверять, вместит ли указанная аудитория всех слушателей и т.д.

Существует несколько *способов получения слотом значений* во фрейме-экземпляре:

- по умолчанию от фрейма-образца;
- через наследование свойств от фрейма, указанного в слоте АКО (A-Kind-Of, это);
- по формуле, указанной в слоте;
- через присоединенную процедуру;
- явно из диалога с пользователем;
- из базы данных.

В качестве значения слота может выступать имя другого фрейма. Тогда образуются *сети фреймов*.

Таким образом, структуру фрейма-экземпляра можно представить в виде следующей таблицы:

Имя слота	Значение слота	Способ получения значения слота	Присоединенная процедура
-----------	----------------	---------------------------------	--------------------------

Существование сетей фреймов обусловлено тем, что между различными объектами можно провести некоторые аналогии. Поэтому фреймы, представляющие такие образы, выстраиваются в иерархическую систему. При этом сложные объекты представляются комбинацией нескольких подчиненных фреймов.

В сетях фреймов (рисунок 5.3) происходит наследование свойств по АКО-связям. Слот АКО указывает на фрейм более высокого уровня, откуда неявно наследуются (переносятся) значения слотов.

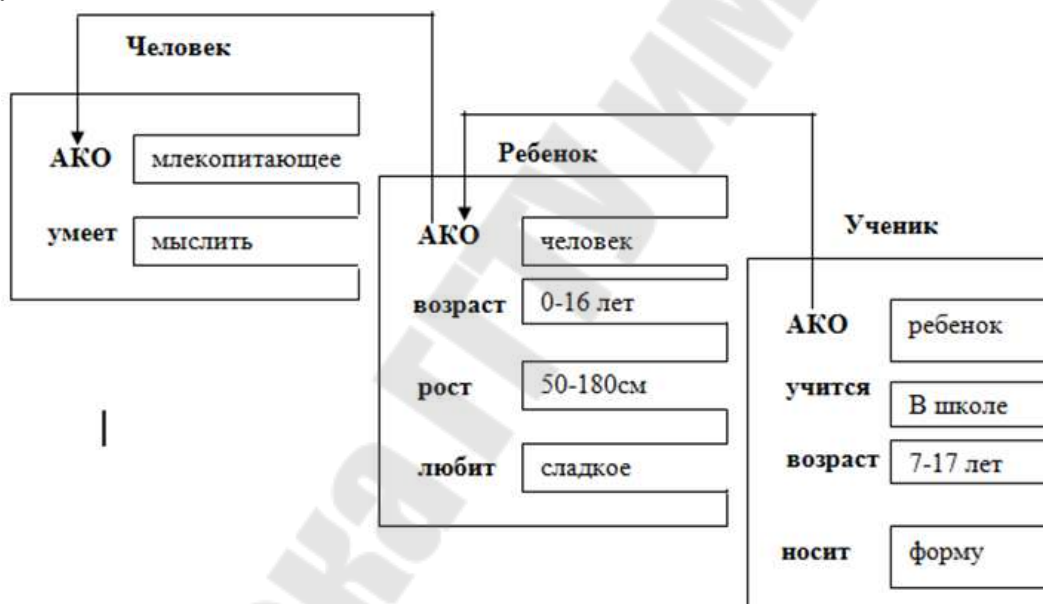


Рисунок 5.3 – Сеть фреймов

Фрейм «Ученик» наследует свойства фреймов «Ребенок» и «Человек», которые на более высоком уровне иерархии. Так, на вопрос «любят ли ученики сладкое?» следует отвечать «да», так как этим свойством обладают все дети, что указано в фрейме «Ребенок».

Наследование свойств может быть частичным. Так, возраст для учеников не наследуется из фрейма «Ребенок», поскольку указан явно во фрейме «Ученик».

Основным *преимуществом* фреймов как модели представления знаний является то, что она отражает концептуальную основу органи-

зации памяти человека, а также ее гибкость и наглядность. Кроме того, отмечают однородность представления знаний и возможность их типового текстового описания с помощью специальных языков.

В сетях фреймов используются такие специальные языки представления знания, как *FRL* (Frame Representation Language) и *KRL* (Knowledge Representation Language). Они позволяют эффективно строить промышленные экспертные системы.

Широко известны такие фрейм-ориентированные экспертные системы, как *ANALYST*, *МОДИС*, *TRISTAN*, *ALTERID*.

5.1.6 Формальные логические модели

Формальная логика – наука об общих структурах и законах правильного мышления, образования и сочетания понятий и высказываний, о правилах умозаключений независимо от их конкретного содержания.

Система искусственного интеллекта в определенном смысле моделирует интеллектуальную деятельность человека и, в частности, логику его рассуждений. Логические построения при этом сводятся к следующей схеме: из одной или нескольких посылок (которые считаются истинными) следует сделать «логически верное заключение» (вывод, следствие). Для этого необходимо, чтобы и посылки, и заключение были представлены на понятном языке, адекватно отображающем предметную область, в которой проводится вывод.

В обычной жизни это наш естественный язык общения. В математике, например, это язык определенных формул. Наличие же языка предполагает, во-первых, наличие алфавита (словаря), отображающего в символьной форме весь набор базовых понятий (элементов), с которыми придется иметь дело и, во-вторых, набор синтаксических правил, на основе которых, пользуясь алфавитом, можно построить определенные выражения.

Логические выражения, построенные в данном языке, могут быть истинными или ложными. Некоторые из этих выражений, являющиеся всегда истинными, объявляются *аксиомами* (или *постулатами*). Они составляют ту базовую систему посылок, исходя из которой и пользуясь определенными правилами вывода, можно получить заключения в виде новых выражений, также являющихся истинными.

Если перечисленные условия выполняются, то говорят, что система удовлетворяет требованиям *формальной теории* и такую систему называют *формальной* или *аксиоматической*.

Таким образом, всякая формальная теория $F = (A, V, W, R)$, определяющая некоторую аксиоматическую систему, характеризуется:

- наличием алфавита (словаря) – A ;
- множеством синтаксических правил – V ;
- множеством аксиом, лежащих в основе теории, – W ;
- множеством правил вывода – R .

Классическими примерами аксиоматических систем являются *исчисление высказываний* и *исчисление предикатов*.

Исчисление высказываний – область математической логики, называемая *булевой алгеброй*. Булева алгебра изучает высказывания и операции над ними. *Высказывание* – это предложение, которое может быть истинно или ложно. Наиболее известными операциями булевой алгебры являются конъюнкция и дизъюнкция.

Так, *дизъюнкция высказываний* – новое высказывание:

- сконструированное из двух и более исходных высказываний;
- истинное в тех случаях, когда истинно хотя бы одно из исходных высказываний.

Конъюнкция высказываний – новое высказывание:

- сконструированное из двух и более исходных высказываний;
- истинное в тех случаях, когда истинны все исходные высказывания.

В алгебре логики высказывания рассматриваются как нераздельные целые и только с точки зрения их истинности или ложности. Ни структура высказываний, ни их содержание не затрагиваются. В то же время и в науке, и в практике используются заключения, существенно зависящие как от структуры, так и от содержания используемых в них высказываний.

В связи с этим возникает необходимость в расширении логики высказываний и в построении такой логической системы, средствами которой можно было бы исследовать структуру тех высказываний, которые в рамках логики высказываний рассматриваются как элементарные.

Такой логической системой является *логика предикатов*, содержащая всю логику высказываний в качестве своей части. Логика предикатов расчленяет элементарное высказывание на *субъект* (буквально – подлежащее, хотя оно может играть и роль дополнения) и

предикат (буквально – сказуемое, хотя оно может играть и роль определения). Субъект – это то, о чем что-то утверждается в высказывании; *предикат* – это то, что утверждается о субъекте. Например, в высказывании «7 – простое число», «7» – субъект, «простое число» – предикат. Это высказывание утверждает, что «7» обладает свойством «быть простым числом». Такой логической системой является *логика предикатов*, содержащая всю логику высказываний в качестве своей части. Логика предикатов расчленяет элементарное высказывание на *субъект* (буквально – подлежащее, хотя оно может играть и роль дополнения) и *предикат* (буквально – сказуемое, хотя оно может играть и роль определения). Субъект – это то, о чем что-то утверждается в высказывании; *предикат* – это то, что утверждается о субъекте. Например, в высказывании «7 – простое число», «7» – субъект, «простое число» – предикат. Это высказывание утверждает, что «7» обладает свойством «быть простым числом».

Если в рассмотренном примере заменить конкретное число 7 переменной X из множества натуральных чисел N , то получим высказывательную форму « X – простое число». При одних значениях X (например, $X = 13$, $X = 17$) она дает истинные высказывания, а при других значениях X (например, $X = 10$, $X = 18$) – ложные высказывания.

Если в рассмотренном примере заменить конкретное число 7 переменной X из множества натуральных чисел N , то получим высказывательную форму « X – простое число». При одних значениях X (например, $X = 13$, $X = 17$) она дает истинные высказывания, а при других значениях X (например, $X = 10$, $X = 18$) – ложные высказывания.

Таким образом, эта высказывательная форма определяет функцию одной переменной X , определенной на множестве N , и принимающую значения из множества $\{1;0\}$. Здесь предикат становится функцией субъекта и выражает свойство субъекта.

Достоинство аксиоматических систем – исчисление высказываний и исчисление предикатов – в том, что они хорошо исследованы и имеют прекрасно разработанные модели логического вывода. Поэтому все, что может и гарантирует каждая из этих систем, гарантируется и для прикладных формальных систем как моделей конкретных предметных областей. В частности, это гарантии непротиворечивости вывода, алгоритмической разрешимости (для исчисления высказываний) и полурешимости (для исчислений предикатов).

Формальные системы имеют и *недостатки*, главный из которых – это их закрытость, негибкость. Модификация и расширение здесь всегда связаны с перестройкой всей формальной системы, что для практических систем сложно и трудоемко. В них очень сложно учитывать происходящие изменения. Поэтому формальные системы как модели представления знаний могут использоваться только в тех предметных областях, которые хорошо локализуются и мало зависят от внешних факторов.

К тому же, очень высокие требования к предметной области – полнота и непротиворечивость «базового аксиоматического набора» – обусловили то, что в промышленных экспертных системах формальные логические модели практически не используются.

5.2 Экспертные системы

5.2.1 Назначение экспертных систем

В начале 80-х гг. XX в. в исследованиях по созданию искусственного интеллекта сформировалось новое самостоятельное направление, получившее название *экспертных систем*. Цель этих новых исследований по экспертным системам состоит в разработке специальных программ, предназначенных для решения особых видов задач. Что это за особый вид задач, потребовавший создания целой новой инженерии знаний? К этому особому виду задач могут быть отнесены задачи из абсолютно любой предметной области. Главное, что отличает их от задач обычных, – это то, что человеку-эксперту решить их представляется очень сложным заданием. Тогда и была разработана первая так называемая *экспертная система* (где в роли эксперта выступал уже не человек, а машина), причем экспертная система получает результаты, не уступающие по качеству и эффективности решениям, получаемым обычным человеком – экспертом. Результаты работы экспертных систем могут быть объяснены пользователю на очень высоком уровне. Данное качество экспертных систем обеспечивается их способностью рассуждать о собственных знаниях и выводах. Экспертные системы вполне могут пополнять собственные знания в процессе взаимодействия с экспертом. Таким образом, их можно с полной уверенностью ставить в один ряд с вполне оформившимся искусственным интеллектом.

Исследователи в области экспертных систем для названия своей дисциплины часто используют также уже упоминавшийся ранее тер-

мин «инженерия знаний», введенный немецким ученым Е. Фейгенбаумом как «привнесение принципов и инструментария исследований из области искусственного интеллекта в решение трудных прикладных проблем, требующих знаний экспертов».

Однако коммерческие успехи к фирмам-разработчикам пришлѣ не сразу. На протяжении четверти века в период с 1960 по 1985 гг. успехи искусственного интеллекта касались в основном исследовательских разработок. Тем не менее, начиная примерно с 1985 г., а в массовом масштабе с 1987 по 1990 гг. экспертные системы стали активно использоваться в коммерческих приложениях.

Заслуги экспертных систем довольно велики и состоят в следующем:

1) технология экспертных систем существенно расширяет круг практически значимых задач, решаемых на персональных компьютерах, решение которых приносит значительный экономический эффект и существенно упрощает все связанные с ними процессы;

2) технология экспертных систем является одним из самых важных средств в решении глобальных проблем традиционного программирования, таких как продолжительность, качество и, следовательно, высокая стоимость разработки сложных приложений, вследствие которой значительно снижался экономический эффект;

3) имеется высокая стоимость эксплуатации и обслуживания сложных систем, которая зачастую в несколько раз превосходит стоимость самой разработки, а также низкий уровень повторной используемости программ и т. п.;

4) объединение технологии экспертных систем с технологией традиционного программирования добавляет новые качества к программным продуктам за счет, во-первых, обеспечения динамичной модификации приложений рядовым пользователем, а не программистом; во-вторых, большей «прозрачности» приложения, лучшей графики, интерфейса и взаимодействия экспертных систем.

По мнению рядовых пользователей и ведущих специалистов, в недалекой перспективе экспертные системы найдут следующее применение:

1) экспертные системы будут играть ведущую роль на всех стадиях проектирования, разработки, производства, распределения, отладки, контроля и оказания услуг;

2) технология экспертных систем, получившая широкое коммерческое распространение, обеспечит революционный прорыв в ин-

теграции приложений из готовых интеллектуально-взаимодействующих модулей.

В общем случае экспертные системы предназначены для так называемых *неформализованных задач*, т. е. экспертные системы не отвергают и не заменяют традиционного подхода к разработке программ, ориентированного на решение формализованных задач, но дополняют их, тем самым значительно расширяя возможности. Именно этого и не может сделать простой человек-эксперт.

Такие сложные неформализованные задачи характеризуются:

- 1) ошибочностью, неточностью, неоднозначностью, а также неполнотой и противоречивостью исходных данных;
- 2) ошибочностью, неоднозначностью, неточностью, неполнотой и противоречивостью знаний о проблемной области и решаемой задаче;
- 3) большой размерностью пространства решений конкретной задачи;
- 4) динамической изменчивостью данных и знаний непосредственно в процессе решения такой неформализованной задачи.

Экспертные системы главным образом основаны на эвристическом поиске решения, а не на исполнении известного алгоритма. В этом одно из главных преимуществ технологии экспертных систем перед традиционным подходом к разработке программ. Именно это и позволяет им так хорошо справляться с поставленными перед ними задачами.

Технология экспертных систем используется для решения самых различных задач. Перечислим основные из подобных задач.

1. Интерпретация.

Экспертные системы, выполняющие интерпретацию, чаще всего применяют показания различных приборов с целью описания положения дел.

Интерпретирующие экспертные системы способны обрабатывать самые различные виды информации. Примером может послужить использование данных спектрального анализа и изменения характеристик веществ для определения их состава и свойств. Также примером может служить интерпретация показаний измерительных приборов в котельной для описания состояния котлов и воды в них.

Интерпретирующие системы чаще всего имеют дело непосредственно с показаниями. В связи с этим возникают затруднения, которых нет у других видов систем. Что это за затруднения? Эти затруд-

нения возникают из-за того, что экспертным системам приходится интерпретировать засоренную лишним, неполную, ненадежную или неверную информацию. Отсюда неизбежны либо ошибки, либо значительное увеличение обработки данных.

2. Прогнозирование.

Экспертные системы, осуществляющие прогноз чего-либо, определяют вероятностные условия заданных ситуаций. Примерами служат прогноз ущерба, причиненного урожаю хлебов неблагоприятными погодными условиями, оценивание спроса на газ на мировом рынке, прогнозирование погоды по данным метеорологических станций. Системы прогнозирования иногда применяют моделирование, т. е. такие программы, которые отображают некоторые взаимосвязи в реальном мире, чтобы воссоздать их в среде программирования, и потом спроектировать ситуации, которые могут возникнуть при тех или иных исходных данных.

3. Диагностика различных приборов.

Экспертные системы производят такую диагностику, применяя описания какой-либо ситуации, поведения или данных о строении различных компонентов, чтобы определить возможные причины неисправно работающей диагностируемой системы. Примерами служат установление обстоятельств заболевания по симптомам, которые наблюдаются у больных (в медицине); определение неисправностей в электронных схемах и определение неисправных компонентов в механизмах различных приборов. Системы диагностики довольно часто являются помощниками, которые не только ставят диагноз, но и помогают в устранении неполадок. В таких случаях данные системы вполне могут взаимодействовать с пользователем, чтобы оказать помощь при поиске неполадок, а потом привести список действий, необходимых для их устранения. В настоящее время многие диагностические системы разрабатываются в качестве приложений к инженерному делу и компьютерным системам.

4. Планирование различных событий.

Экспертные системы, предназначенные для планирования, проектируют различные операции. Системы предопределяют практически полную последовательность действий, прежде чем начнется их реализация.

Примерами такого планирования событий могут служить создания планов военных действий как оборонительного, так и наступа-

тельного характера, предопределенного на определенный срок с целью получения преимущества перед вражескими силами.

5. Проектирование.

Экспертные системы, выполняющие проектирование, разрабатывают различные формы объектов, учитывая сложившиеся обстоятельства и все сопутствующие факторы.

В качестве примера можно рассмотреть генную инженерию.

6. Контроль.

Экспертные системы, осуществляющие контроль, сравнивают настоящее поведение системы с ее ожидаемым поведением. Наблюдающие экспертные системы обнаруживают контролируемое поведение, которое подтверждает их ожидания по сравнению с нормальным поведением или их предположением о потенциальных отклонениях. Контролирующие экспертные системы по своей сути должны работать в режиме реального времени и реализовывать зависящую как от времени, так и от контекста интерпретацию поведения контролируемого объекта.

В качестве примера можно привести слежение за показаниями измерительных приборов в атомных реакторах с целью обнаружения аварийных ситуаций или оценку данных диагностики пациентов, находящихся в блоке интенсивного лечения.

7. Управление.

Ведь широко известно, что экспертные системы, осуществляющие управление, весьма результативно руководят поведением системы в целом. Примером служит управление различными производствами, а также распределением компьютерных систем. Управляющие экспертные системы должны включать в себя наблюдающие компоненты, для того, чтобы контролировать поведение объекта на протяжении длительного времени, но они могут нуждаться и в других компонентах из уже проанализированных типов задач.

Экспертные системы применяются в самых различных областях: финансовых операциях, нефтяной и газовой промышленности. Технология экспертных систем может быть применена также в энергетике, транспортном хозяйстве, фармацевтическом производстве, космических разработках, металлургической и горной промышленности, химии и многих других областях.

5.2.2 Структура экспертных систем

Разработка экспертных систем имеет ряд существенных отличий от разработки обычного программного продукта. Опыт создания экспертных систем показал, что использование при их разработке методологии, принятой в традиционном программировании, либо сильно увеличивает количество времени, затраченного на создание экспертных систем, либо вообще приводит к отрицательному результату.

Экспертные системы в общем случае подразделяются на *статические* и *динамические*.

Для начала рассмотрим статическую экспертную систему.

Стандартная *статическая экспертная система* состоит из следующих основных компонентов:

- 1) рабочей памяти, называемой также базой данных;
- 2) базы знаний;
- 3) решателя, называемого также интерпретатором;
- 4) компонентов приобретения знаний;
- 5) объяснительного компонента;
- 6) диалогового компонента.

Рассмотрим теперь каждый компонент более подробно.

Рабочая память (по абсолютной аналогии с рабочей, т. е. оперативной памятью компьютера) предназначена для получения и хранения исходных и промежуточных данных решаемой в текущий момент задачи.

База знаний предназначена для хранения долгосрочных данных, описывающих конкретную предметную область, и правил, описывающих рациональное преобразование данных этой области решаемой задачи.

Решатель, называемый также *интерпретатором*, функционирует следующим образом: используя исходные данные из рабочей памяти и долгосрочные данные из базы знаний, он формирует правила, применение которых к исходным данным приводит к решению задачи. Одним словом, он действительно «решает» поставленную перед ним задачу;

Компонент приобретения знаний автоматизирует процесс заполнения экспертной системы знаниями эксперта, т. е. именно этот компонент обеспечивает базу знаний всей необходимой информацией из данной конкретной предметной области.

Компонент объяснений разъясняет, как система получила решение данной задачи, или почему она это решение не получила и какие знания она при этом использовала. Иначе говоря, компонент объяснений создает отчет о проделанной работе.

Данный компонент является очень важным во всей экспертной системе, поскольку он значительно облегчает тестирование системы экспертом, а также повышает доверие пользователя к полученному результату и, следовательно, ускоряет процесс разработок.

Диалоговый компонент служит для обеспечения дружественного интерфейса пользователя как в ходе решения задачи, так и в процессе приобретения знаний и объявления результатов работы.

Теперь, когда мы знаем, из каких компонент в общем состоит статическая экспертная система, построим диаграмму, отражающую структуру такой экспертной системы (рисунок 5.4).

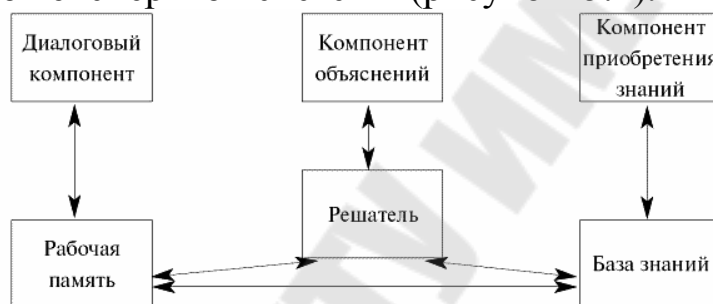


Рисунок 5.4 – Структура статической экспертной системы

Статические экспертные системы чаще всего используются в технических приложениях, где можно не учитывать изменения окружающей среды, происходящие во время решения задачи. Любопытно знать, что первые экспертные системы, получившие практическое применение, были именно статическими.

Итак, на этом закончим пока рассмотрение статической экспертной системы, перейдем к анализу экспертной системы динамической.

Ограничимся разбором только самых основных отличий динамической экспертной системы от статических.

В отличие от статической экспертной системы в структуру динамической экспертной системы дополнительно вводятся два следующих компонента:

- 1) подсистема моделирования внешнего мира;
- 2) подсистема связей с внешним окружением.

Подсистема связей с внешним окружением как раз и осуществляет связи с внешним миром. Делает она это посредством системы специальных датчиков и контроллеров.

Помимо этого, некоторые традиционные компоненты статической экспертной системы подвергаются существенным изменениям, для того чтобы отобразить временную логику событий, происходящих в данный момент в окружающей среде.

Это главное различие между статической и динамической экспертными системами.

Пример динамической экспертной системы – управление производством различных медикаментов в фармацевтической промышленности.

5.2.3 Участники разработки экспертных систем

В разработке экспертных систем участвуют представители различных специальностей. Чаще всего конкретную экспертную систему разрабатывают трое специалистов. Это, как правило:

- 1) эксперт;
- 2) инженер по знаниям;
- 3) программист по разработке инструментальных средств.

Разъясним обязанности каждого из приведенных здесь специалистов.

Эксперт – это специалист в той предметной области, задачи которой и будут решаться при помощи этой конкретной разрабатываемой экспертной системы.

Инженер по знаниям – это специалист по разработке непосредственно экспертной системы. Используемые им технологии и методы называются технологиями и методами инженерии знаний. Инженер по знаниям помогает эксперту выявить из всей информации предметной области ту информацию, которая необходима для работы с конкретной разрабатываемой экспертной системой, а затем структурировать ее.

Любопытно, что отсутствие среди участников разработки инженеров по знаниям, т. е. замена их программистами, либо приводит к неудаче всего проекта создания конкретной экспертной системы, либо значительно увеличивает сроки ее разработки.

И, наконец, *программист* разрабатывает инструментальные средства (если инструментальные средства разрабатываются заново),

предназначенные для ускорения разработки экспертных систем. Эти инструментальные средства содержат в пределах все основные компоненты экспертной системы; также программист осуществляет сопряжение своих инструментальных средств с той средой, в которой она будет использоваться.

5.2.4 Режимы работы экспертных систем

Экспертная система работает в двух основных режимах:

- 1) в режиме приобретения знаний;
- 2) в режиме решения задачи (называемом также режимом консультаций, или режимом использования экспертной системы).

Это логично и понятно, ведь сначала необходимо как бы загрузить экспертную систему информацией из той предметной области, в которой ей предстоит работать, это и есть режим «обучения» экспертной системы, режим, когда она получает знания. А уже после загрузки всей необходимой для работы информации следует и сама работа. Экспертная система становится готовой для эксплуатации, и ее теперь можно использовать для консультаций или для решения каких-либо задач.

Рассмотрим более подробно *режим приобретения знаний*.

В режиме приобретения знаний работу с экспертной системой осуществляет эксперт при посредничестве инженера по знаниям. В этом режиме эксперт, используя компонент приобретения знаний, наполняет систему знаниями (данными), которые, в свою очередь, позволяют системе в режиме решения уже без участия эксперта решать задачи из данной предметной области.

Следует отметить, что режиму приобретения знаний в традиционном подходе к разработке программ соответствуют этапы алгоритмизации, программирования и отладки, выполняемые непосредственно программистом. Отсюда следует, что в отличие от традиционного подхода в случае экспертных систем разработку программ осуществляет не программист, а эксперт, естественно, с помощью экспертных систем, т. е. по большому счету человек, не владеющий программированием.

А теперь рассмотрим второй режим функционирования экспертной системы, т. е. *режим решения задач*.

В режиме решения задачи (или так называемом режиме консультаций) общение с экспертными системами осуществляет непо-

средственно конечный пользователь, которого интересует конечной итог работы и иногда способ его получения. Необходимо отметить, что в зависимости от назначения экспертной системы пользователь не обязательно должен быть специалистом в данной проблемной области. В этом случае он обращается к экспертным системам за результатом, не имея достаточных знаний для получения результатов. Или все же пользователь может обладать уровнем знаний, достаточным для достижения необходимого результата самостоятельно. В этом случае пользователь может сам получить результат, но обращается к экспертным системам с целью либо ускорить процесс получения результата, либо возложить на экспертные системы монотонную работу. В режиме консультации данные о задаче пользователя после обработки их диалоговым компонентом поступают в рабочую память. Решатель на основе входных данных из рабочей памяти, общих данных о проблемной области и правил из базы данных формирует решение задачи. Экспертные системы при решении задачи не только исполняют предписанную последовательность конкретной операции, но и предварительно формирует ее. Это делается для случая, если реакция системы не совсем понятна пользователю. В этой ситуации пользователь может потребовать объяснения о том, почему данная экспертная система задает тот или иной вопрос или почему данная экспертная система не может выполнить данную операцию, как получен тот или иной результат, поставляемый данной экспертной системой.

Литература

1. Авлукова, Ю. Ф. Основы автоматизированного проектирования : учебное пособие / Ю. Ф. Авлукова. – Минск : Выш. школа, 2013. – 216 с.
2. Андронов, С. А. Методы оптимального проектирования / С. А. Андронов. – СПб. : СПб ГУАП, 2001. – 169 с.
3. Технологии анализа данных / А. А. Барсегян [и др.]. – СПб. : БХВ-Петербург, 2007. – 384 с.
4. Берлинер, Э. М. САПР в машиностроении : учеб. для вузов / Э. М. Берлинер, О. В. Таратынов. – М. : Форум, 2008. – 447 с.
5. Гаврилова, Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб. : Питер, 2000. – 384 с.
6. Девятков, В. В. Системы искусственного интеллекта : учеб. пособие для вузов / В. В. Девятков. – М. : МГТУ им. Н. Э. Баумана, 2001. – 352 с.
7. Дворецкий, С. И. Компьютерное моделирование и оптимизация технологических процессов и оборудования : учеб. пособие / С. И. Дворецкий, А. Ф. Егоров, Д. С. Дворецкий. – Тамбов : ТГТУ, 2003.
8. Норенков, И. П. Введение в автоматизированное проектирование технических устройств и систем : учеб. пособие для вузов / И. П. Норенков . – М. : Высш. шк., 1986. – 304 с.

Содержание

Введение.....	3
1 ИНФОРМАЦИОННЫЕ СИСТЕМЫ	4
1.1 Основные понятия.....	4
1.2 Этапы развития ИС.....	6
1.3 Структура информационной системы.....	8
1.4 Состав ИС	11
1.5 Классификация ИС	13
<i>1.5.1 Классификация информационных систем по функциональному признаку и уровням управления</i>	<i>13</i>
<i>1.5.2 Прочие классификации информационных систем</i>	<i>14</i>
1.6 Типы ИС.....	20
1.7 Жизненный цикл ИС	20
1.8 Основы применения инструментальных средств ИТ.....	25
1.9 Создание автоматизированных информационных систем ..	27
2 СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ (САПР) И ИХ МЕСТО СРЕДИ ДРУГИХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ	31
2.1 Системы управления жизненным циклом в современном машиностроении	31
<i>2.1.1 Жизненный цикл изделия (ЖЦИ). Этапы жизненного цикла изделия</i>	<i>31</i>
<i>2.1.2 Информация об изделии</i>	<i>32</i>
<i>2.1.3 Понятие PLM-технологии</i>	<i>34</i>
<i>2.1.4 Понятие CALS-технологии</i>	<i>37</i>
2.2 Основные принципы создания САПР	41
2.2.1 Основные понятия теории САПР	41
2.2.2 Подсистемы САПР.....	41
2.2.3 Типовые проектные процедуры	42
2.2.4 Иерархические уровни проектирования	44
2.2.5 Классификация САПР. Структура	47
2.2.6 Виды обеспечения САПР.....	48
2.2.7 Основные принципы создания САПР.....	61
2.2.8 Системное проектирование технологических процессов	63
2.2.9 Стратегии проектирования технологических процессов	65
2.3 Состав и назначение интегрированных САПР.....	68
2.3.1 Интеграция систем проектирования и изготовления ...	68

2.3.2 Требования к интегрированным САПР.....	69
2.3.3 Состав интегрированных САПР.....	70
2.3.4 CAD /CAM /CAE системы	71
2.3.5 Функции, характеристики и примеры CAE/CAD/CAM-систем.....	72
3 ОРГАНИЗАЦИЯ ИНФОРМАЦИОННОГО ФОНДА САПР С ИСПОЛЬЗОВАНИЕМ БАЗ ДАННЫХ	73
3.1 Банки данных в САПР	73
3.1.1 Основные понятия	73
3.1.2 Виды архитектуры информационных систем на основе баз данных	74
3.1.3 Классификация систем управления базами данных: признаки классификации и виды СУБД. Аппаратное и программное обеспечение СУБД. Функции различных видов СУБД	77
3.1.4 Принципы разработки и выполнения приложений для информационных систем на основе СУБД	82
3.1.5 Система управления базой данных (СУБД), информационные потоки в САПР	84
3.1.6 Особенности и требования к базе данных САПР	88
3.1.7 Виды представления базы данных.....	89
3.1.8 Структура построения базы данных.....	91
3.1.9 Обобщенная технология работы пользователей в среде СУБД.....	91
3.2 Нормализация данных.....	94
3.2.1 Понятие нормализации данных.....	94
3.2.2 Основные нормальные формы реляционных отношений и их взаимосвязь	94
3.3 Система управления базами данных MS Acces.....	97
3.3.1 Основные характеристики СУБД ACCESS	98
3.3.2 Практическая работа с MS Access	102
4 МЕТОДЫ И АЛГОРИТМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В САПР.....	116
4.1 Оптимизация технологических процессов в САПР ТП	116
4.1.1 Проектирования сложного технического объекта.....	116
4.1.2 Особенности построения структуры математических моделей технологических процессов.....	119
4.1.3. Обоснование и выбор критериев оптимальности	123
4.1.4 Выбор технических ограничений.....	133

4.1.5	<i>Виды оптимизации технологических процессов</i>	136
4.1.6	<i>Структурная оптимизация ТП</i>	138
4.2	Многокритериальная оптимизация	144
4.2.1	<i>Общая характеристика и классификация методов принятия решений при многих критериях</i>	144
4.2.2	<i>Выбор Парето-оптимальных решений</i>	149
4.2.3	<i>Методы на основе компенсации критериев</i>	151
4.2.4	<i>Методика экспресс-анализа альтернатив</i>	156
4.2.5	<i>Методика скаляризации векторных оценок</i>	159
4.2.6	<i>Методика сравнительной оценки двух альтернатив по степени доминирования</i>	162
4.2.7	<i>Алгоритм Саати</i>	165
4.2.8	<i>Метод ранга</i>	168
4.2.9	<i>Метод ЭЛЕКТРА</i>	171
4.2.10	<i>Модифицированный алгоритм Кемени-Снелла</i>	176
5	БАЗЫ ЗНАНИЙ В САПР	183
5.1	Модели представления знаний	183
5.1.1	<i>Знания и их классификация</i>	183
5.1.2	<i>Понятие базы знаний</i>	185
5.1.3	<i>Продукционная модель</i>	187
5.1.4	<i>Семантическая сеть</i>	190
5.1.5	<i>Фреймы</i>	191
5.1.6	<i>Формальные логические модели</i>	194
5.2	Экспертные системы	197
5.2.1	<i>Назначение экспертных систем</i>	197
5.2.2	<i>Структура экспертных систем</i>	202
5.2.3	<i>Участники разработки экспертных систем</i>	204
5.2.4	<i>Режимы работы экспертных систем</i>	205
	Литература	207

Мурашко Валентина Семеновна

ИНФОРМАЦИОННЫЕ СИСТЕМЫ В САПР

ПОСОБИЕ

**для студентов специальности
1-53 01 01 «Автоматизация производственных
процессов и производств (по направлениям)»
дневной формы обучения**

Подписано к размещению в электронную библиотеку
ГГТУ им. П. О. Сухого в качестве электронного
учебно-методического документа 16.03.21.

Рег. № 17Е.
<http://www.gstu.by>