

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**Пермский национальный исследовательский
политехнический университет**

Чайковский филиал
Кафедра автоматизации, информационных и инженерных технологий

МП 12.8-2022

**Методические предписания
К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ**

По дисциплине «Проектирование автоматизированных систем управления»

Направление 09.03.01 Информатика и вычислительная техника
для всех форм обучения

Чайковский, 2022

Русских, Т.И. Методические предписания к выполнению курсовой работы по дисциплине «Проектирование автоматизированных систем управления» для всех форм обучения направления 09.03.01 Информатика и вычислительная техника / Т.И. Русских. – Чайковский, 2022. –48с.

Методические предписания к выполнению курсовой работы по дисциплине «Проектирование автоматизированных систем управления» предназначены для студентов, обучающихся по направлению 09.03.01 Информатика и вычислительная техника. В методических предписаниях содержатся рекомендации по выполнению курсовой работы, требования к оформлению пояснительной записки, графического материала, а также порядок защиты выполненной работы.

Методические предписания обсуждены и одобрены на заседании кафедры автоматизации, информационных и инженерных технологий» ЧФ ПНИПУ «05» декабря 2022 г., протокол №14

Методические предписания к выполнению курсовых работ рекомендованы методической комиссией ЧФ ПНИПУ для использования в учебном процессе (протокол № 4 от 29.12.2022)

©Пермский национальный исследовательский
политехнический университет
Чайковский филиал, 2022
©Русских Т.И., 2022

Содержание

1	Цели и задачи курсовой работы	4
2	Общая характеристика процесса проектирования АСУ	6
3	Разработка функциональной модели АСУ	8
4	Исходные данные для проектирования, стандарты	12
5	Проектирование пользовательского интерфейса	188
6	Задачи информационного обеспечения, структура информационно-логической модели АСУ	21
7	Распределенная обработка данных	25
8	Структура программных модулей, разработка алгоритмов, логический анализ структур АСУ	28
9	Разработка моделей защиты данных.....	322
10	Автоматизация проектирования CASE-средствами.....	355
11	Задание на курсовую работу и общие указания по выполнению курсовой работы.....	38
12	Оформление курсовой работы.....	411
13	Рекомендуемая литература	444
	Приложение А. Образец оформления титульного листа	466
	Приложение Б. Образец оформления задания на курсовую работу	477

1 Цели и задачи курсовой работы

Курсовая работа по дисциплине «Проектирование автоматизированных систем управления» выполняется в соответствии с учебным планом направления подготовки 09.03.01 Информатика и вычислительная техника, профиль «Автоматизированные системы обработки информации и управления». Она предназначена для практического усвоения студентами основных разделов дисциплины «Проектирование автоматизированных систем управления», закрепления знаний по методам объектно-ориентированного и структурного проектирования различных видов автоматизированных систем управления, развития практических навыков комплексного решения задач исследования и проектирования систем на современных электронных вычислительных машинах (ЭВМ). Курсовая работа должна показать знания студента, полученные им в процессе изучения дисциплины «Проектирование автоматизированных систем управления».

Выполняя курсовую работу, студент должен проявить полученный им в процессе обучения опыт разработки конкретных информационно-программных изделий и свои творческие способности.

Курсовая работа готовит студентов к решению более сложной задачи, завершающей обучение, – выполнению выпускной квалификационной работы.

В результате выполнения курсовой работы по дисциплине «Проектирование автоматизированных систем управления» обучающийся должен демонстрировать следующие результаты:

1) Знать:

- основы методики структурного анализа и проектирования систем и возможности оценки их качества и эффективности функционирования;
- способы и процедуры выработки и принятия решений в автоматизированных системах управления;
- текущий уровень научно-технического потенциала для создания и применения автоматизированных систем управления;

2) Уметь:

- обследовать объекты автоматизации, строить для них схемы информационных потоков или документопотоков и формулировать предложения по их усовершенствованию;

- разрабатывать материал для включения его в технико-экономическое обоснование конкретной автоматизированной системы управления, а также рабочий материал для раздела технического задания на систему;

- давать предложения для предварительных оценок положительного эффекта и возможных негативных последствий создания и функционирования автоматизированных систем обработки информации и управления;

3) Владеть:

- навыками выполнения работ на стадии обследования объектов автоматизации;

- навыками работы с многообразным техническим материалом из области автоматизированных систем и информационных технологий и обширной научной литературой по профилю с целью поиска нужных фактов для решения возникающих задач и пополнения своих знаний;

- навыками проектирования автоматизированных систем управления.

Выполнение курсовой работы способствует формированию следующих компетенций (согласно ФГОС ВО по направлению подготовки 09.03.01 Информатика и вычислительная техника и рабочей программе по дисциплине «Проектирование автоматизированных систем управления»):

ПК-1.2: способность разрабатывать прототипы информационных систем на базе типовой информационной системы

2 Общая характеристика процесса проектирования АСУ

Современные информационные технологии предоставляют широкий набор способов реализации АСУ, выбор которых осуществляется на основе требований со стороны предполагаемых пользователей, которые, как правило, изменяются в процессе разработки. Для теории принятия решений процесс проектирования системы – это процесс принятия проектно-конструкторских решений, направленных на получение версии системы, удовлетворяющей требованиям заказчика.

Под *проектом* будем понимать проектно-конструкторскую и технологическую документацию, в которой представлено описание проектных решений по созданию и эксплуатации системы в конкретной программно-технической среде.

Под *проектированием системы* понимается процесс преобразования входной информации об объекте проектирования, о методах проектирования и об опыте проектирования объектов аналогичного назначения в соответствии с ГОСТом в проект АСУ. С этой точки зрения проектирование АСУ сводится к последовательной формализации проектных решений на различных стадиях жизненного цикла системы: предпроектного анализа требований, технического и рабочего проектирования, внедрения и эксплуатации АСУ.

Осуществление проектирования системы предполагает использование проектировщиками определенной технологии проектирования, соответствующей масштабу и особенностям разрабатываемого проекта.

Технология проектирования системы – это совокупность методологии и средств проектирования системы, а также методов и средств организации проектирования (управление процессом создания и модернизации проекта системы).

В основе технологии проектирования лежит технологический процесс, который определяет действия, их последовательность, состав исполнителей, средства и ресурсы, требуемые для выполнения этих действий. *Технологический процесс* проектирования системы в целом делится на совокупность последовательно-параллельных, связанных и соподчиненных цепочек действий, каждое из которых

может иметь свой предмет.

Проектирование системы – трудоемкий, длительный и динамический процесс. Технологии проектирования, применяемые в настоящее время, предполагают поэтапную разработку системы. Этапы по общности могут разделяться в стадии. Совокупность стадий и этапов, которые проходит система в своем развитии от момента принятия решения о создании системы до момента прекращения функционирования системы, называется *жизненным циклом системы*.

Суть содержания жизненного цикла разработки системы в различных подходах одинакова и сводится к выполнению следующих стадий:

Планирование и анализ требований (предпроектная стадия) – системный анализ. Исследование и анализ существующей системы, определение требований к создаваемой системе, оформление технико-экономического обоснования и технического задания на разработку системы.

Проектирование (техническое проектирование, логическое проектирование). Разработка в соответствии со сформулированными требованиями состава автоматизируемых функций и состава обеспечивающих подсистем, оформление технического проекта системы.

Реализация проекта (рабочее проектирование, физическое проектирование, программирование). Разработка и настройка программ, наполнение баз данных, создание рабочих инструкций для персонала, оформление рабочего проекта.

Внедрение (тестирование, опытная эксплуатация). Комплексная отладка подсистем, обучение персонала, поэтапное внедрение системы в эксплуатацию, оформление акта о приемо-сдаточных испытаниях системы.

Эксплуатация системы (сопровождение, модернизация). Сбор рекламаций и статистики о функционировании системы, исправление ошибок и недоработок, оформление требований к модернизации системы и ее выполнение.

3 Разработка функциональной модели АСУ

Предпроектная стадия, включающая проведение необходимых исследований, работ по формированию структуры АСУ и получение, в конечном счете, технического задания на проектирование, может проходить по двум принципиально различным вариантам.

Первый – классический фундаментальный подход. Он связан с проведением исследований по вскрытию законов, на основе которых функционирует исследуемая система обработки информации и управления. Далее, вследствие выявленных фундаментальных законов, строится формальная модель, связывающая входы, выходы системы, влияние внешних воздействий на нее. Таким образом получаем формально-математическое представление системы, которое и может в дальнейшем служить основой для функционального, инфологического и техно-рабочего проектирования АСУ.

Второй подход, достаточно часто применяемый при построении АСУ, включает в себя проведение информационного обследования объекта (предметной области), выявление основных информационных потоков, построения, как правило, имитационной модели функционирования объекта и далее выход также на инфологическое и техно-рабочее проектирование.

Первый подход в большей степени гарантирует получение высококачественной разработки, но требует больших интеллектуальных и финансовых затрат. Второй подход обеспечивает, на первый взгляд, меньшие затраты, но вероятней всего за счет неудачных и малоэффективных решений общие затраты при реализации работ по второму варианту могут быть и значительно больше, чем в первом случае.

Однако выбор схемы решения всегда остается за разработчиком.

Функциональным моделированием IDEF0 (структурный анализ) принято называть метод исследования системы с помощью ее графического модельного представления, которое начинается с общего обзора и последующей детализации в иерархическую структуру со все большим числом уровней. Для такого метода харак-

терно:

- разбиение на уровни абстракции с ограничением числа элементов на каждом из уровней (обычно от 3 до 6);

- ограниченный контекст, включающий лишь существенные на каждом уровне детали;

- дуальность данных и операций над ними;

- использование строгих формальных правил записи;

- последовательное приближение к конечному результату.

Анализ является первым этапом создания АСУ, на котором требования заказчика уточняются, формализуются и документируются. Фактически на этом этапе дается ответ на вопрос: «Что должна делать будущая система?».

Структурный анализ начинается с исследования того, как организована система управления предприятием, с обследования функциональной и информационной структуры системы управления. По результатам обследования аналитик на первой стадии анализа строит обобщенную логическую модель исходной предметной области, отображающую ее функциональную структуру, особенности основной деятельности и информационное пространство, в котором эта деятельность осуществляется. Используя специальную терминологию, можно сказать, что аналитик строит модель «как есть» (модель «AS-IS»).

Вторая стадия работы, к которой привлекаются заинтересованные представители заказчика, а при необходимости и независимые эксперты, состоит в анализе модели «как есть», выявлении ее недостатков и узких мест, определение путей совершенствования системы управления на основе выделенных критериев качества.

Третья стадия анализа, содержащая элементы проектирования, – это создание усовершенствованной обобщенной логической модели, отображающей реорганизованную предметную область или ее часть, которая подлежит автоматизации. Эту модель можно назвать моделью «как должно быть» (модель «TO-BE»), т.е. здесь происходит формализация системы.

На ряду со структурным подходом существует и более мощный подход, назы-

ваемый объектно-ориентированным (ООП). Эта методология создана для проектирования больших и сложных систем и имеет ряд преимуществ перед структурным подходом.

ООП базируется на создании объектной модели, которая описывает структуру объектов, составляющих систему, их атрибуты, операции, взаимосвязи с другими объектами. Цель разработки объектной модели – описать объекты, составляющие в совокупности проектируемую систему, а также выявить и указать различные зависимости между объектами. *Декомпозиция проблемы на объекты* – творческий, плохо формализуемый процесс.

В объектной модели должны быть отражены те понятия и объекты реального мира, которые важны для разрабатываемой системы. В объектной модели отражается, прежде всего, прагматика разрабатываемой системы, что выражается в использовании терминологии прикладной области, связанной с использованием разрабатываемой системы.

Объектная модель имеет четыре главных элемента: абстрагирование, инкапсуляция, модульность, иерархия.

Эти элементы являются *главными* в том смысле, что без любого из них модель не будет объектно-ориентированной. Кроме главных имеются еще три дополнительных элемента: типизация, параллелизм, сохраняемость.

Абстракция выделяет существенные характеристики некоторого объекта, отличающие его от всех других видов объектов и, таким образом, четко определяет его концептуальные границы с точки зрения наблюдателя.

Инкапсуляция – это процесс отделения друг от друга элементов объекта, определяющих его устройство и поведение; инкапсуляция служит для того, чтобы изолировать контрактные обязательства абстракции от их реализации.

Модульность – это свойство системы, которая была разложена на внутренне связанные между собой модули.

Иерархия – это упорядочение абстракций, расположение их по уровням.

Типизация – это способ защититься от использования объектов одного класса

вместо другого, или по крайней мере управлять таким использованием.

Параллелизм – это свойство, отличающее активные объекты от пассивных.

Сохраняемость – способность объекта существовать во времени, переживая породивший его процесс, и (или) в пространстве, перемещаясь из своего первоначального адресного пространства.

4 Исходные данные для проектирования, стандарты

Методы, которые используются для изучения предметной области и технологии получения от экспертов сведений о системе, подлежащей описанию, называются сбором данных, а в информатике она больше известна как опрос (интервьюирование) или извлечение знаний. Но как бы она не называлась, способность собрать необходимую информацию, основанную на знаниях экспертов, весьма существенна для построения точной и полезной модели. Поэтому технология сбора информации составляет важную часть методологии IDEF0.

Опрос – это сбор сведений. Первый опрос служит точкой отсчета в процессе моделирования. Чтобы провести опрос, аналитик вначале выбирает наилучший источник информации (документ или конкретного человека), а затем организует его «опрос». Цель опроса – получение порции информации, необходимой для начала либо для продолжения построения определенной части модели. После первого опроса функциональную модель используется для определения той информации, которую необходимо получить в ходе следующего опроса. В соответствии с иерархией модели может быть проведена последовательность опросов для выяснения все более конкретных деталей рассматриваемой области.

Обычно источниками информации служат эксперты. Часто именно они являются наилучшими источниками, потому что им знакомы текущие нюансы и недокументированные аспекты системы. Самое важное – это то, что экспертам известны факты, которые не отражены в документах или которые трудно объяснить. Их можно получить только путем опроса экспертов. Чтобы подготовиться к такому опросу, нужно исследовать другие источники информации, например документы. Существует множество различных стратегий для извлечения информации из этих источников: чтение документов, наблюдение за выполняемыми операциями, анкетирование, использование собственных знаний, составление описания.

Документы – хороший источник информации, потому что они чаще всего доступны и их можно «опрашивать» в удобном для себя темпе. Чтение документов яв-

ляется хорошим способом получить первоначальное представление о системе и сформулировать вопросы к экспертам.

Наблюдение за работой моделируемой системы – хорошая стратегия получения информации. Оно должно проводиться всегда, когда есть такая возможность. Через наблюдение, а возможно, и участие аналитики получают информацию о происходящих день за днем операциях из первых рук. Во время наблюдения за работой системы часто возникают вопросы, которые никогда бы не появились, если бы аналитик только читал документы или разговаривал с экспертами. Однако следует соблюдать осторожность. Слишком долгие наблюдения могут привести к избыточному привыканию к текущему состоянию дел. Из-за потери объективности можно не увидеть альтернативные пути описания функций системы.

Анкетирование проводится для того, чтобы опросить большие группы экспертов в сжатые сроки. Его можно использовать, например, когда необходимо быстро получить сведения о работе какой-либо определенной части системы с разных позиций. Анкетирование при опросе экспертов позволяет выявить, какие части системы более всего нуждаются в улучшении. На практике, однако, информация, полученная от экспертов с помощью анкет, оказывается недостаточно достоверной.

Еще одна полезная стратегия – придумать описание и дать его экспертам для корректировки. Придуманные описания могут дать альтернативные схемы функционирования системы – схемы, о которых эксперты никогда не думали. Однако для реализации таких схем нужна поддержка. Предварительно нужно изучить предметную область и найти хотя бы одну доброжелательно настроенную группу экспертов, прежде чем пытаться придумать описание. В этом случае описание будет иметь шансы отразить реальность. Придуманные описания могут, однако, потерпеть неудачу, если эксперты не готовы воспринимать новые возможности. В идеале, прежде чем прибегать к этой стратегии, автор должен установить надежные контакты с несколькими экспертами.

Создание и внедрение автоматизированных систем различных классов и назначений ведется во многих отраслях промышленности по нормативно-технической

документации, устанавливающей разнообразные организационно-методические и технические нормы, правила и положения, затрудняющие интеграцию систем и эффективное их совместное функционирование.

В период принятия Госстандартом РФ решения о совершенствовании межотраслевых комплексов стандартов действовали следующие комплексы и системы стандартов, устанавливающие требования к различным видам АСОИУ:

1) единая система стандартов автоматизированных систем управления, распространяющаяся на АСУ, АСУП, АСУТП и другие организационно-экономические системы;

2) комплекс стандартов, распространяющихся на системы автоматизированного проектирования;

3) группа стандартов, распространяющиеся на автоматизированные системы технологической подготовки производства (АСТПП).

Практика применения стандартов на АСУ, САПР, АСУТП, АСТПП показала, что в них применяется одинаковый понятийный аппарат, имеется много общих объектов стандартизации, однако требования стандартов не согласованы между собой, имеются различия по составу и содержанию работ, различия по обозначению, составу, содержанию и оформлению документов и пр.

На фоне отсутствия единой технической политики в области создания АСОИУ многообразие стандартов не обеспечивало широкой совместимости АСОИУ при их взаимодействии, не позволяло тиражировать системы, тормозило развитие перспективных направлений использования средств вычислительной техники.

В настоящее время осуществляется переход к созданию сложных автоматизированных систем (за рубежом системы САД – САМ), включающих в свой состав АСУ технологическими процессами и производствами. САПР – конструктора, САПР – технолога, АСНИ и др. системы. Использование противоречивых правил при создании таких систем приводит к снижению качества, увеличению стоимости работ, затягиванию сроков ввода в действие.

Единый комплекс стандартов и руководящих документов должен распространяться на автоматизированные системы различного назначения: АСНИ, САПР, ОАСУ, АСУП, АСУТП, АСУГПС, АСК, АСТПП, включая их интеграцию.

При разработке межотраслевых документов следует учитывать следующие особенности автоматизированных систем как объектов стандартизации:

1) техническое задание является основным документом в соответствии с которым проводят создание АСОИУ и приемку его заказчиком,

2) АСОИУ, как правило, создают проектным путем с комплектацией изделиями серийного и единичного производства и проведением строительных, монтажных, наладочных и пусковых работ, необходимых для ввода в действие системы;

3) в общем случае АСУ состоит из программно-технических (ПТК), программно-методических комплексов (ПМК) и компонентов технического, программного и информационного обеспечения. Компоненты этих видов обеспечения, а также ПМК и ПТК должны изготавливаться, и поставляется как продукция производственно-технического назначения. Компоненты могут входить в систему в качестве самостоятельных частей или могут быть объединены в комплексы;

4) создание АСУ в организациях (предприятиях) требует специальной подготовки пользователей и обслуживающего персонала системы;

5) функционирование АСУ и комплексов обеспечивается совокупностью организационно-методических документов, рассматриваемых в процессе создания как компоненты правового, методического, лингвистического, математического, организационного и др. видов обеспечения. Отдельные решения, получаемые в процессе разработки этого обеспечения, могут реализовываться в виде компонентов технического, программного или информационного обеспечения;

б) совместное функционирование и взаимодействие различных систем и комплексов осуществляется на базе локальных сетей ЭВМ.

Спецификации и соглашения, принятые для локальных сетей ЭВМ, обязательны для обеспечения совместимости систем, комплексов и компонентов.

Стандартизация в области АСУ является составной частью работ по обобщен-

ной проблеме «Информационная технология».

Единый комплекс стандартов (ЕКС) руководящих документов на автоматизированные системы совместно с другими системами и комплексами стандартов должен образовывать полное нормативно-техническое обеспечение процессов создания и функционирования систем.

ЕКС АСУ должен охватывать специфические для автоматизированных систем направления стандартизации и распространять традиционные направления стандартизации на программно-технические, программно-методические комплексы и автоматизированные системы в целом.

Направления и задачи стандартизации при нормативно-техническом обеспечении процессов создания и функционирования АСУ группируют следующим образом:

- 1) установление технических требований к продукции;
- 2) регламентация методов испытаний и правил аттестации и сертификации продукции;
- 3) регламентация правил и порядка разработки;
- 4) установление правил документирования;
- 5) обеспечение совместимости,
- 6) регламентация организационно-методических вопросов функционирования систем.

Направления 1-4 являются традиционными при разработке, изготовлении и поставке продукции. Направления 5, 6 являются специфичными и вытекают из особенностей, присущих АСУ.

Обеспеченность АСУ в целом и их составных частей нормативно-технической документацией (НТД) в рамках принятых направлений и задач стандартизации различна.

Компоненты технического, программного и информационного обеспечения, как продукцию производственно-технического назначения, рассматривают, соответственно, как конструкторские, программные и информационные изделия.

Весь жизненный цикл конструкторских изделий полностью обеспечен нормативно-технической документацией, действующей в машиностроении и приборостроении.

Программные изделия обеспечены НТД, входящей в ЕСПД заказчика. Однако область распространения этих НТД должна быть расширена с целью отражения вопросов, связанных с разработкой, созданием, распространением и эксплуатацией программных изделий.

Информационные изделия в настоящее время не обеспечены НТД, хотя отдельные вопросы проработаны в классификаторах и кодификаторах технико-экономической информации.

5 Проектирование пользовательского интерфейса

Проектирование интерфейса пользователя – сложная многофакторная и многовариантная задача, требующая системного подхода. В настоящее время считается доказанным, что решение данной задачи заключается не в том, чтобы рационально «вписать» человека в контур управления, а в том, чтобы, исходя из задач управления объектом, разработать систему взаимодействия двух равноправных партнеров (человек-оператор и аппаратно-программный комплекс АСУ).

Цель создания эффективного эргономичного пользовательского интерфейса состоит в том, чтобы отобразить информацию настолько эффективно насколько это возможно для человеческого восприятия и структурировать отображение на дисплее таким образом, чтобы привлечь внимание к наиболее важным единицам информации.

Пользователь должен иметь возможность манипулировать объектами в среде приложения. Неплохо, если они (графические элементы) будут ему понятны и станут нести в себе информацию о том, что это такое и что произойдет, если выбрать или произвести действие над каким-то объектом. Иллюстрация этой идеи – панель быстрого доступа Word'a. Что еще, кроме как сохранение файла, можно ожидать от кнопки с изображением дискеты? Необходимо, чтобы пользователь имел наглядное средство отображения информации на различных этапах решения задач, он должен видеть, как его действия влияют на объекты, расположенные на экране.

Создание эффективного интерфейса заключается в быстром, насколько это возможно, развитии у пользователей простой концептуальной модели интерфейса. Концепция согласованности состоит в том, что при работе с компьютером у пользователя формируется система ожидания одинаковых реакций на одинаковые действия, что постоянно подкрепляет пользовательскую модель интерфейса.

Приложение должно проектироваться таким образом, чтобы пользователь в любой момент и на любом этапе работы мог получить помощь, контекстную справку или подсказку.

Безусловно, пользователю нужно дать возможность экспериментировать в

приложении (нажатие любых кнопок, изменение настроек и т. д.). Но при этом необходимо избавить его от тупиковых ситуаций: все последствия экспериментов должны быть исправимы, а в лучшем случае еще и обратимы.

Интерфейс должен предоставлять информацию о том, что происходит в данный момент на компьютере. Нельзя допускать, чтобы пользователь долгое время ожидал реакции приложения на некоторое свое действие.

Цветовая гамма, компоновка элементов, пиктограммы, звуки, анимация – все должно помогать пользователю при выполнении задачи. Но здесь важно не переборщить, т.к. в этом случае внимание человека начнет рассеиваться, у него появится раздражение и, как следствие снизится эффективность работы.

Начальным этапом разработки пользовательского интерфейса являются создание его ассоциативной модели, после чего осуществляется проработка концептуального дизайна. Здесь необходимо разработать необходимый набор интерфейсных элементов, каждый из которых должен обладать определенным цветом, формой, надписью и т. п., и все вместе они должны составлять единую систему, вызывающую стойкую систему ассоциаций у пользователей.

Цвет – мощный визуальный инструмент, который необходимо использовать очень осторожно, чтобы не вызвать неадекватной реакции пользователя.

Целесообразно ограничить число цветов до 4 на экране и до 7 для последовательности экранов. Для неактивных элементов рекомендуется использовать бледные цвета. Необходимо использовать цвета согласно представлениям пользователя. Например, для картографа зеленый цвет – лес, желтый – пустыня, синий – вода. Если цвет используется для кодировки информации, необходимо удостовериться, что код адекватно воспринимается пользователем: красный – опасность/стоп, зеленый – нормально/продолжение работы, желтый – предостережение. Для привлечения внимания наиболее эффективны белый, желтый и красный цвета.

Меню необходимый элемент любой автоматизированной системы, позволяющий пользователю выполнять задачи внутри приложения и управлять процессом решения. Достоинство меню в том, что пользователи не должны помнить название

элемента или действия, которое они хотят выполнить – они должны только распознать его среди пунктов меню.

Сообщения необходимы для направления пользователя в нужную сторону, подсказок и предупреждений для выполнения необходимых действий на пути решения задачи. Они также включают подтверждения действий со стороны пользователя и подтверждения, что задачи были выполнены системой успешно либо по каким-то причинам не выполнены. Сообщения могут быть обеспечены в форме диалога, экранных заставок и т.п.

6 Задачи информационного обеспечения, структура информационно-логической модели АСУ

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Примерами понятий являются "сотрудник", "отдел", "проект", "зарплата". Примеры взаимосвязей между понятиями – "сотрудник числится ровно в одном отделе", "сотрудник может выполнять несколько проектов", "над одним проектом может работать несколько сотрудников". Примеры ограничений – "возраст сотрудника не менее 16 и не более 60 лет".

Логическая модель данных является начальным прототипом будущей базы данных. Логическая модель строится в терминах информационных единиц, но *без привязки к конкретной СУБД*. Более того, логическая модель данных необязательно должна быть выражена средствами именно *реляционной* модели данных. Основным средством разработки логической модели данных в настоящий момент являются различные варианты *ER-диаграмм (Entity-Relationship, диаграммы сущность-связь)*. Одну и ту же ER-модель можно преобразовать как в реляционную модель данных, так и в модель данных для иерархических и сетевых СУБД, или в постреляционную модель данных.

Решения, принятые на предыдущем уровне, при разработке модели предметной области, определяют некоторые границы, в пределах которых можно развивать логическую модель данных, в пределах же этих границ можно принимать различные решения. Например, модель предметной области складского учета содержит понятия "склад", "накладная", "товар". При разработке соответствующей реляционной модели эти термины обязательно должны быть использованы, но различных способов реализации тут много – можно создать одно отношение, в котором будут присутствовать в качестве атрибутов "склад", "накладная", "товар", а можно создать три отдельных отношения, по одному на каждое понятие.

При разработке логической модели данных возникают вопросы: хорошо ли спроектированы отношения? Правильно ли они отражают модель предметной об-

ласти, а, следовательно, и саму предметную область?

Для того чтобы оценить качество принимаемых решений на уровне логической модели данных, необходимо сформулировать некоторые критерии качества в терминах физической модели и конкретной реализации и посмотреть, как различные решения, принятые в процессе *логического* моделирования, влияют на качество *физической* модели и на скорость работы базы данных.

Таких критериев может быть очень много и выбор их произволен. Некоторые из таких критериев являются важными с точки зрения получения качественной базы данных: адекватность базы данных предметной области, легкость разработки и сопровождения базы данных, скорость выполнения операций обновления данных (вставка, обновление, удаление кортежей), скорость выполнения операций выборки данных.

База данных должна адекватно отражать предметную область. Это означает, что должны выполняться следующие условия.

1. Состояние базы данных в каждый момент времени должно соответствовать состоянию предметной области.

2. Изменение состояния предметной области должно приводить к соответствующему изменению состояния базы данных.

3. Ограничения предметной области, отраженные в модели предметной области, должны некоторым образом отражаться и учитываться базе данных.

Практически любая база данных, за исключением совершенно элементарных, содержит некоторое количество программного кода в виде триггеров и хранимых процедур.

Хранимые процедуры – это процедуры и функции, хранящиеся непосредственно в базе данных в откомпилированном виде и которые могут запускаться пользователями или приложениями, работающими с базой данных. Основное назначение хранимых процедур – реализация бизнес-процессов предметной области.

Триггеры – это хранимые процедуры, связанные с некоторыми событиями, происходящими во время работы базы данных. В качестве таких событий выступа-

ют операции вставки, обновления и удаления строк таблиц. Если в базе данных определен некоторый триггер, то он запускается *автоматически* всегда при возникновении события, с которым этот триггер связан. Триггер срабатывает независимо от того, кто из пользователей и каким способом инициировал событие, вызвавшее запуск триггера. Таким образом, основное назначение триггеров – автоматическая поддержка целостности базы данных.

Очевидно, что чем больше программного кода в виде триггеров и хранимых процедур содержит база данных, тем сложнее ее разработка и дальнейшее сопровождение.

На уровне логического моделирования определяются реляционные отношения и атрибуты этих отношений. На этом уровне можно определять какие-либо физические структуры хранения (индексы, хеширование и т.п.). Единственное, чем можно управлять – это распределение атрибутов по различным отношениям. Можно описать немного отношений с большим количеством атрибутов, или сформировать большое количество отношений, каждое из которых содержит мало атрибутов. Таким образом, необходимо попытаться ответить на вопрос – влияет ли количество отношений и количество атрибутов в отношениях на скорость выполнения операций обновления данных. Такая постановка не является достаточно корректной, т.к. скорость выполнения операций с базой данных зависит от физической реализации базы данных. Тем не менее, целесообразно *качественно* оценить это влияние при *одинаковых подходах к физическому моделированию*.

Основными операциями, изменяющими состояние базы данных, являются операции вставки, обновления и удаления записей. В базах данных, требующих постоянных изменений (складской учет, системы продаж билетов и т.п.) производительность определяется скоростью выполнения большого количества небольших операций вставки, обновления и удаления.

Обычно, вставка записи производится в одну из свободных страниц памяти, выделенной для данной таблицы. СУБД постоянно хранит информацию о наличии и расположении свободных страниц. Если для таблицы не созданы индексы, то опера-

ция вставки выполняется фактически с одинаковой скоростью независимо от размера таблицы и от количества атрибутов в таблице. Если в таблице имеются индексы, то при выполнении операции вставки записи индексы должны быть перестроены. Таким образом, скорость выполнения операции вставки *уменьшается при увеличении количества индексов* у таблицы и *мало зависит от числа строк* в таблице.

Для операции обновления и удаления записей из таблицы, прежде чем обновить или удалить запись, ее необходимо найти. Если таблица не индексирована, то единственным способом поиска является последовательное сканирование таблицы в поиске нужной записи. В этом случае, скорость операций обновления и удаления существенно увеличивается с увеличением количества записей в таблице и не зависит от количества атрибутов. Но на самом деле неиндексированные таблицы практически никогда не используются. Для каждой таблицы обычно объявляется один или несколько индексов, соответствующий потенциальным ключам. При помощи этих индексов поиск записи производится очень быстро и практически не зависит от количества строк и атрибутов в таблице (хотя, конечно, некоторая зависимость имеется). Если для таблицы объявлено несколько индексов, то при выполнении операций обновления и удаления эти индексы должны быть перестроены, на что тратится дополнительное время. Таким образом, скорость выполнения операций обновления и удаления также *уменьшается при увеличении количества индексов* у таблицы и *мало зависит от числа строк* в таблице.

Одно из назначений базы данных – предоставление информации пользователям. Информация извлекается из реляционной базы данных при помощи оператора SQL – SELECT. Одной из наиболее дорогостоящих операций при выполнении оператора SELECT является операция соединение таблиц. Таким образом, чем больше взаимосвязанных отношений было создано в ходе логического моделирования, тем больше вероятность того, что при выполнении запросов эти отношения будут соединяться, и, следовательно, тем медленнее будут выполняться запросы. Таким образом, увеличение количества отношений приводит к замедлению выполнения операций выборки данных, особенно, если запросы заранее неизвестен.

7 Распределенная обработка данных

Распределенная обработка данных – методика выполнения прикладных программ группой систем.

Сущность ее заключается в том, что пользователь получает возможность работать с сетевыми службами и прикладными процессами, расположенными в нескольких взаимосвязанных абонентских системах. При этом возможны несколько видов работ, которые он может выполнять: удаленный запрос, например, команда, позволяющая посылать одиночную заявку на выполнение обработки данных; удаленная транзакция, осуществляющая направление группы запросов прикладному процессу; распределенная транзакция, дающая возможность использования нескольких серверов и прикладных процессов, выполняемых в группе абонентских систем.

Для распределенной обработки осуществляется сегментация прикладных программ. Передача данных происходит при помощи удаленного вызова процедур либо электронной почты. Первая технология характеризуется высоким быстродействием, а вторая – низкой стоимостью. Известны также программные средства Системы Управления Распределенной Базой Данных (СУРБД), содержатся инструментальные средства распределенной среды обработки данных.

Распределенная среда обработки данных – представляет собой технологию распределенной обработки данных. Эта среда обычно представляет собой набор сетевых служб, предназначенный для выполнения прикладных процессов, рассредоточенных по группе абонентских систем гетерогенной сети. Основные ее компоненты показаны в таблице 1.

Таблица 1 – Основные компоненты распределенной обработки данных.

№ п/п	Служба	Выполняемые функции
1.	Имена	База Данных имен пользователей и средств, предназначенных для доступа пользователей к сетевым службам.
2.	Удаленный доступ	Технология, обеспечивающая взаимодействие двух прикладных программ, расположенных в различных абонентских системах.
3.	Защита данных	Программное Обеспечение разрешения на доступ к ресурсам системы или сети.
4.	Многопоточность	Программы, обеспечивающие одновременное выполнение нескольких задач.

Системы, имеющие программы распределенной среды, соответственно, являются серверами и клиентами. Серверы связаны (рисунок 1) друг с другом логическими каналами, по которым передают друг другу файлы. Каждый сервер имеет свою группу клиентов.

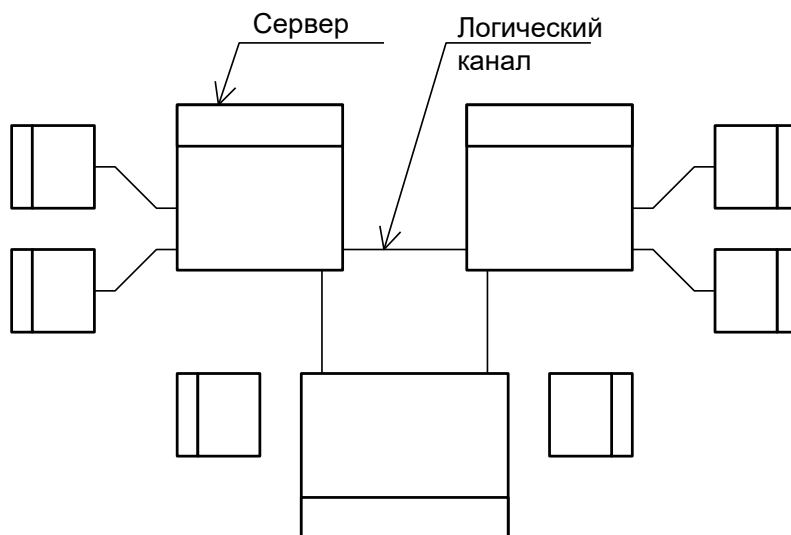


Рисунок 1 – Связь серверов

Среда чаще всего имеет трехступенчатую архитектуру: прикладная программа

– база данных – клиент. Функции, выполняемые средой, включают прикладные службы:

- каталогов, позволяющую клиентам находить нужные им серверы;
- интерфейса многопоточной обработки;
- удаленного вызова процедур;
- обслуживания файлов;
- безопасности данных;
- времени, синхронизирующей часы в абонентских системах.

Программное Обеспечение среды погружается, как правило, в Сетевую Операционную Систему. Серверы могут иметь свои, различные операционные системы. В роли сервера может, также, выступать главный компьютер со своей операционной системой.

Функционирование распределенной среды требует выполнения ряда административных задач. К ним, в первую очередь, относятся средства:

- регистрации и контроля за лицензиями пользователей на работу с прикладными программами;
- унифицированных интерфейсов прикладных программ;
- обеспечения безопасности данных;
- инвентаризации программного и технического обеспечения абонентских систем, работающих в сети.

С точки зрения логического управления среда обработки данных делится на ячейки распределенной среды обработки. В каждую из них может включаться от нескольких единиц до тысяч абонентских систем. Размеры ячеек территориально не ограничены. Входящие в одну и ту же ячейку системы могут быть расположены даже на разных континентах.

8 Структура программных модулей, разработка алгоритмов, логический анализ структур АСОИУ

Современные технологии разработки программного обеспечения информационных систем предполагают большое разнообразие концепций, подходов и методик. Наиболее популярные из них это структурный и объектный подходы.

Структурный метод, о котором шла речь при разработке информационного обеспечения, при создании программного обеспечения базируется на использовании технологии моделирования потоков данных.

Диаграммы потоков данных (DFD) являются основным средством моделирования функциональных требований к проектируемой информационной системе. С их помощью эти требования представляются в виде иерархии функциональных компонентов (процессов), связанных потоками данных. Главная цель такого представления – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

В основе методологии лежит построение модели анализируемой системы – проектируемой или реально существующей. В соответствии с методологией модель системы определяется как иерархия диаграмм потоков данных (ДПД или DFD), описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю. Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы системы с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут такой уровень декомпозиции, на котором процесс становится элементарными и детализировать их далее невозможно.

Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те в свою очередь преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных

или внешним сущностям – потребителям информации. Таким образом, основными компонентами диаграмм потоков данных являются:

- внешние сущности;
- системы/подсистемы;
- процессы;
- накопители данных;
- потоки данных.

Внешняя сущность представляет собой материальный предмет или физическое лицо, представляющее собой источник или приемник информации, например, заказчики, персонал, поставщики, клиенты. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой системы. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой системы, если это необходимо, или, наоборот, часть процессов может быть вынесена за пределы диаграммы и представлена как внешняя сущность.

Процесс представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Физически процесс может быть реализован различными способами: это может быть подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т.д.

Номер процесса служит для его идентификации. В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже, например:

- "Ввести сведения о клиентах";
- "Выдать информацию о текущих расходах";
- "Проверить кредитоспособность клиента".

Использование таких глаголов, как "обработать", "модернизировать" или "отредактировать" означает, как правило, недостаточно глубокое понимание данного

процесса и требует дальнейшего анализа.

Информация в поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

Накопитель данных представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

Накопитель данных в общем случае является прообразом будущей базы данных и описание хранящихся в нем данных должно быть увязано с информационной моделью.

Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока. Каждый поток данных имеет имя, отражающее его содержание.

Первым шагом при построении иерархии ДПД является построение контекстных диаграмм. Обычно при проектировании относительно простых систем строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Если же для сложной системы ограничиться единственной контекстной диаграммой, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и, кроме того, единственный главный процесс не раскрывает структуры распределенной системы.

Признаками сложности (в смысле контекста) могут быть:

- наличие большого количества внешних сущностей (десять и более);
- распределенная природа системы;
- многофункциональность системы с уже сложившейся или выявленной группировкой функций в отдельные подсистемы.

Для сложных систем строится иерархия контекстных диаграмм. При этом

контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

Разработка контекстных диаграмм решает проблему строгого определения функциональной структуры сисси на самой ранней стадии ее проектирования, что особенно важно для сложных многофункциональных систем, в разработке которых участвуют разные организации и коллективы разработчиков.

Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи ДПД. Каждый процесс на ДПД, в свою очередь, может быть детализирован при помощи ДПД или миниспецификации.

Миниспецификация (описание логики процесса) должна формулировать его основные функции так, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу.

При построении иерархии ДПД переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается при помощи структур данных.

Структуры данных конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что данный компонент может отсутствовать в структуре. Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает вхождение любого числа элементов в указанном диапазоне.

Для каждого элемента данных может указываться его тип (непрерывные или дискретные данные). Для непрерывных данных может указываться единица измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования. Для дискретных данных может указываться таблица допустимых значений.

9 Разработка моделей защиты данных

Большое внимание в настоящее время уделяется вопросам формирования принципов построения механизмов защиты информации (ЗИ) и системы требований к ним. На основе имеющегося опыта можно сформулировать следующие фундаментальные принципы организации защиты информации:

- системность;
- специализированность;
- неформальность.

Основные требования принципа *системности* сводятся к тому, что для обеспечения надежной защиты информации в современных АСОИУ должна быть обеспечена надежная и согласованная защита во всех структурных элементах, на всех технологических участках автоматизированной обработки информации и во все время функционирования АСУ.

Специализированность как принцип организации защиты предполагает два аспекта:

1) ввиду специфических особенностей рассматриваемой проблемы надежный механизм защиты может быть спроектирован и организован лишь профессиональными специалистами по защите информации,

2) для обеспечения эффективного функционирования механизма защиты в составе АСОИУ должны функционировать специалисты по защите информации.

В соответствии с данным принципом значительное распространение за рубежом получает специализация по различным аспектам ЗИ. В США, например, на вопросах ЗИ специализируется свыше 100 фирм.

Принцип *неформальности* означает, что методология проектирования механизма защиты и обеспечения его функционирования в основе своей является неформальной. Эта неформальность интерпретируется в том смысле, что в настоящее время не существует инженерной методики проектирования механизма защиты в традиционном понимании этого термина.

Общие требования к механизму защиты следующие:

1) адекватность, т.е. обеспечение требуемого уровня защиты (определяется степенью секретности подлежащей обработке информации) при минимальных издержках на создание механизма защиты и обеспечение его функционирования;

2) удобство для пользователей, основу чего составляет требование, чтобы механизм защиты не создавал для пользователей дополнительных трудностей, требующих значительных усилий для их преодоления; минимизация привилегий в доступе, предоставляемых пользователям, т.е. каждому пользователю должны предоставляться только действительно необходимые ему права по обращению к ресурсам системы и данным;

3) полнота контроля, т.е. обязательный контроль всех обращений к защищаемым данным; наказуемость нарушений, причем наиболее распространенной мерой наказания является отказ в доступе к системе;

4) экономичность механизма, т.е. обеспечение минимальности расходов на создание и эксплуатацию механизма;

5) несекретность проектирования, т.е. механизм защиты должен функционировать достаточно эффективно даже в том случае, если его структура и содержание известны злоумышленнику.

В автоматизированных банках данных должно быть предусмотрено наличие в них средств идентификации пользователей и ресурсов системы с периодической сменой идентифицирующей информации, многоаспектного разграничения доступа к элементам баз данных (по элементам, по разрешенным процедурам, по условиям операций и др.), криптографического закрытия данных, регистрации обращений к защищаемым данным, контроля за использованием защищаемых данных и т.д.

Основные положения по разработке систем ЗИ могут быть сформулированы так:

1) защита информации является не разовым мероприятием и даже не совокупностью мероприятий, а непрерывным процессом, который должен протекать (осуществляться) во все время и на всех этапах жизненного цикла АСУ;

2) осуществление непрерывного процесса защиты информации возможно лишь на базе промышленного производства средств защиты;

3) создание эффективных механизмов защиты может быть осуществлено высококвалифицированными специалистами-профессионалами в области защиты информации;

4) поддержание и обеспечение надежного функционирования механизмов защиты информации в АСУ сопряжено с решением специфических задач и поэтому может осуществляться лишь профессионально подготовленными специалистами.

Сохранность информации может быть нарушена в двух основных случаях: при получении несанкционированного доступа к информации и нарушении функционирования ЭВМ. Система защиты от этих угроз включает следующие основные элементы: защиту АСУ и ее аппаратуры, организационные мероприятия по обеспечению сохранности информации, защиту операционной системы, файлов, терминалов и каналов связи. Следует при этом иметь в виду, что все типы защиты взаимосвязаны и при выполнении своих функций хотя бы одной из них сводит на нет усилия других.

В качестве классификационного признака для схем защиты можно выбрать их функциональные свойства. На основе и этого признака выделяются системы без схем защиты, с полной защитой и с единой схемой защиты.

10 Автоматизация проектирования CASE-средствами

Под термином "CASE-средства" (Computer Aided Software Engineering) понимаются программные средства, поддерживающие процессы создания и сопровождения АСУ, включая анализ и формулировку требований, проектирование прикладного ПО и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы. CASE-средства вместе с системным ПО и техническими средствами образуют полную среду разработки АСОИУ.

Появлению CASE-технологии и CASE-средств предшествовали исследования в области методологии программирования. Программирование обрело черты системного подхода с разработкой и внедрением языков высокого уровня, методов структурного и модульного программирования.

CASE-технология представляет собой методологию проектирования АСОИУ, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения АСОИУ и разрабатывать приложения в соответствии с информационными потребностями пользователей. Большинство существующих CASE-средств основано на методологиях структурного (в основном) или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств.

Наиболее трудоемкими этапами разработки АСОИУ являются этапы анализа и проектирования, в процессе которых CASE-средства обеспечивают качество принимаемых технических решений и подготовку проектной документации. Графические средства моделирования предметной области позволяют разработчикам в наглядном виде изучать существующую АСОИУ, перестраивать ее в соответствии с поставленными целями и имеющимися ограничениями.

CASE-средства обладают следующими основными особенностями:

- а) имеют мощные графические средства для описания и документирования

АСУ, обеспечивающие удобный интерфейс с разработчиком и развивающие его творческие возможности;

б) осуществляют интеграцию отдельных компонент CASE-средств, обеспечивающую управляемость процессом разработки систем;

в) используют специальным образом организованное хранилище проектных метаданных (репозитория).

Интегрированное CASE-средство должно содержать следующие компоненты:

а) репозиторий, являющийся основой CASE-средства. Он должен обеспечивать хранение версий проекта и его отдельных компонентов, синхронизацию поступления информации от различных разработчиков при групповой разработке, контроль метаданных на полноту и непротиворечивость;

б) графические средства анализа и проектирования, обеспечивающие создание и редактирование иерархически связанных диаграмм (DFD, ERD и др.), образующих модели АСУ;

в) средства разработки приложений, включая языки 4GL и генераторы кодов;

г) средства конфигурационного управления;

д) средства документирования;

е) средства тестирования;

ж) средства управления проектом;

з) средства реинжиниринга.

Современный рынок программных средств насчитывает около 300 различных CASE-средств, наиболее мощные из которых используются практически всеми ведущими западными фирмами.

Все современные CASE-средства могут быть классифицированы в основном по типам и категориям. Классификация по типам отражает функциональную их ориентацию на те или иные процессы ЖЦ.

Классификация по категориям определяет степень интегрированности по выполняемым функциям и включает следующее:

а) отдельные локальные средства, решающие небольшие автономные задачи

(tools);

б) набор частично интегрированных средств, охватывающих большинство этапов жизненного цикла систем (toolkit);

в) полностью интегрированные средства, поддерживающие весь ЖЦ систем и связанные общим репозиторием.

Помимо этого CASE-средства можно классифицировать по следующим признакам:

а) применяемым методологиям и моделям систем и БД;

б) степени интегрированности с СУБД;

в) доступным платформам.

Классификация по типам в основном совпадает с компонентным составом CASE-средств.

На сегодняшний день российский рынок программного обеспечения располагает следующими наиболее развитыми CASE-средствами: Vantage Team Builder (Westmount I-CASE), RAMUS, Designer/2000, Silverrun, ERwin+Bpwin, S-Designor, ARIS EXPRESS , CASE /4/0, PRO-IV, System Architect, Visible Analyst Workbench, EasyCASE.

11 Задание на курсовую работу и общие указания по выполнению курсового проекта

Курсовая работа представляет собой самостоятельную разработку программного, аппаратного или технологического компонента АСОИУ.

Работа включает в себя постановку задачи с представлением предметной области задачи проектирования, анализ существующих или возможных решений поставленной задачи с кратким обзором литературных источников, алгоритмическую проработку решений, выбор среды реализации с использованием средств автоматизации проектирования (CASE-средства ERwin, BPwin, Diagram Designer или DIA).

В курсовой работе должны быть представлены результаты отладки проектируемых компонент в средствах выбранной CASE-среды.

Задание на курсовую работу для студентов выдается, как правило, по тематике выпускной квалификационной работы.

Тема курсовой работы утверждается в начале семестра.

По мере изучения дисциплины «Проектирование автоматизированных систем управления» студент выполняет разделы курсовой работы, соответствующие программе курса:

- постановка задачи, анализ решений и функциональная разработка системы – (2-5)-я недели;
- разработка информационного обеспечения, функциональных модулей, интерфейсов – (6-12)-я недели;
- отладочные работы, оформление записки, графических материалов и подготовка к защите – (13-14) недели.

Защита курсовой работы проводится на зачетной неделе.

Структура курсовой работы приведена в таблице 2.

Таблица 2 – Структура курсовой работы

Разделы	Ориентировочный объем
Титульный лист	
Реферат	3-5 строк
Ключевые слова	5-10 слов (словосочетаний)
Содержание	
Введение	2 стр.
Задание на курсовое проектирование	1 стр.
1. Общая характеристика объекта автоматизации	5 стр.
2. Анализ известных подходов к решению проблемы	5 стр.
3. Описание функциональной модели деятельности объекта (модель “как есть”)	10 стр.
4. Разработка функциональной модели желаемой системы (модель “как должно быть”)	10 стр.
5. Разработка инфологической модели системы	10 стр.
6. Разработка технического задания на информационную систему	15-20 стр.
7. Заключение	2 стр.
Список используемых источников	не менее 10 источников
Приложения	без ограничения

Содержание разделов

Реферат. Краткое содержание работы.

Ключевые слова. Основные понятия, характеризующие содержание работы.

Введение. Описание актуальности работы. Объект исследований, цель работы, задачи исследований.

Задание на курсовую работу. Описание основных задач на курсовое проектирование.

1. Общая характеристика объекта автоматизации. Описание организационной

структуры объекта автоматизации. Характеристика вида деятельности.

2. Анализ известных подходов к решению проблемы. Описание существующих программ, методов и средств, позволяющих решить данную проблему.

3. Функциональное моделирование деятельности объекта (модель “как есть”). Описание объекта автоматизации и информационных процессов. Схемы модели привести в приложении. Здесь привести их описание. Описание входных, выходных и нормативно-справочных информационных совокупностей и сами документы и экранные формы.

4. Разработка функциональной модели системы (модель “как должно быть”). Описание функционала будущей системы.

5. Разработка инфологической модели системы. Привести логическую структуру модели данных.

Объем пояснительной записки должен составлять не менее 30 страниц (без учета приложений). При защите проекта должен быть представлен графический материал (не менее 5 листов), в котором отражены основные схемы и модели, построенные в различных CASE-средствах.

12 Оформление курсовой работы

Пояснительная записка является основным документом, предъявляемым студентом при защите курсового проекта. В ней отражаются этапы работы и результаты, полученные при выполнении курсового проекта.

Оформление пояснительной записки осуществляется на основании ГОСТ 7.32-2017

Пояснительная записка к курсовой работе печатается студентом на одной стороне листа бумаги формата А4 (210 x 297 мм). При этом необходимо оставлять поля: слева – 20 мм, справа – 10 мм, сверху – 20 мм, снизу – 20 мм. Цвет текста должен быть черным, высота букв и других знаков не менее 1,8 мм (кегель не менее 12). Текст должен быть напечатан с полуторным интервалом. Изложение текста должно быть кратким, четким и вестись от первого лица множественного числа.

Весь текст пояснительной записки делят на разделы. Каждый раздел следует начинать с новой страницы с абзацного отступа. Разделы в пределах всей пояснительной записки, а также подразделы и пункты имеют порядковые номера, обозначенные арабскими цифрами, например: 1 – первый раздел; 2 – второй раздел; 2.1 – первый подраздел второго раздела; 2.1.1 – первый пункт первого подраздела второго раздела. После номера раздела, подраздела, пункта и подпункта в тексте точку не ставят. Введение и заключение не нумеруются.

Заголовки подразделов пишут с абзаца, отступая слева 12,5 мм, строчными буквами (кроме первой прописной). В заголовке не допускаются переносы слов. Пробелы над заголовками разделов и под ними – 1 пустая строка. Точку в конце заголовка не ставят. Если заголовок состоит из двух предложений, то их разделяют точкой. Заголовок подчеркивать нельзя.

В записке необходимо выдержать единые обозначения и размерности для используемых параметров, переменных и характеристик. Сокращения должны быть расшифрованы после первого их упоминания в тексте.

Иллюстрации (рисунки, таблицы, схемы) располагаются на отдельных страницах пояснительной записки. Согласно ЕСКД иллюстрации в пояснительной за-

писке, кроме таблиц, имеют подпись «Рисунок». Допускается нумеровать иллюстрации в пределах раздела. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например, «Рисунок 3.2 – Название рисунка» (второй рисунок третьего раздела). Иллюстрации снабжаются кратким подрисуночным текстом. Длинные записи переносят в текст пояснительной записки. Рисунок располагают после той страницы, где на него дана первая ссылка.

Иллюстрация должна быть расположена так, чтобы ее было удобно рассматривать (без поворота записи или поворачивая по часовой стрелке). Количество иллюстраций должно быть минимальным, но доступным для пояснения излагаемого материала.

Таблицы применяют для лучшей наглядности и удобства сравнения показателей. Они проводятся после первого упоминания о них в тексте. При ссылке следует писать слово «Таблица» с указанием ее номера. Каждая таблица должна иметь порядковый номер и название. Заголовок и слово «Таблица» начинают с прописной буквы, точка между ними не ставится. Таблицу размещают таким образом, чтобы ее можно было читать без поворота записки или с поворотом по часовой стрелке. При большом количестве строк допускается перенос таблицы на другой лист. При этом заголовок помещают только над ее первой частью. Схемы алгоритмов и программ следует оформлять в пояснительной записке в соответствии с требованиями ЕСПД.

При ссылке в тексте на используемую литературу указывают порядковый номер, выделенный двумя квадратными скобками по списку источников, например [2]. Литературу следует располагать в списке в порядке появления ссылок в тексте. Источник описывается по следующей форме: фамилия и инициалы автора, полное название книги или статьи, место и год издания, объем (для журнала — название журнала, год издания, номер страницы).

Приложения оформляют как продолжение пояснительной записки на последующих ее страницах, располагая их в порядке появления ссылок в тексте. Каждое приложение следует начинать с нового листа с указанием по центру слова «Прило-

жение А», «Приложение Б» и т. д., написанные прописными буквами, далее следует тематический заголовок.

В пояснительной записке все страницы, в том числе титульный лист, содержание, листы с таблицами, рисунками, графиками, нумеруются арабскими цифрами. На титульном листе номер не ставят, на последующих страницах номер проставляют в центре нижней части листа без точки.

К оформлению графического материала к курсовой работе следует приступать после написания пояснительной записки и составления плана доклада по защите работы, что позволит вынести на слайды и в графический материал наиболее интересные иллюстрации, схемы и модели, которые помогут рационально построить доклад при защите курсового проекта.

13 Рекомендуемая литература

1) Автоматизированные информационные технологии в экономике: учебник / под ред. проф. Г.А. Титаренко. – М.: ЮНИТИ, 2007. – 400 с.

2) Белов, В.В. Проектирование информационных систем: учебник / В.В. Белов, В.И. Чистякова; под ред. В.В. Белова. – М.: Издательский центр «Академия», 2013.–352с.

3) Вендров, А.М. Практикум по проектированию программного обеспечения экономических информационных систем / А.М. Вендров. – 2-е изд., перер. и доп. – М.: Финансы и статистика, 2006. – 192 с.

4) Вендров, А.М. Проектирование программного обеспечения экономических информационных систем: учебник / А.М. Вендров.– М.: Финансы и статистика, 2006. – 545с.

5) Виснаул Б.Д. Технология разработки программного обеспечения / Б.Д. Виснаул, Л.Г. Гагарина. – М.: Форум,2008. – 400с.

6) Емельянова, Н.З. Основы построения автоматизированных информационных систем: учебное пособие / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. – М.: ФОРУМ: ИНФРА-М, 2007. – 416с.

7) Избачков, Ю.С. Информационные системы: учебник для вузов / Ю.С. Избачков, В.Н. Петров. – 2-е изд. – СПб.: Питер, 2006. – 656с.

9) Информационные технологии: учебник для академического бакалавриата / под ред. В.В. Трофимова.– М.: Издательство Юрайт, 2014.–624с.

10) Калянов, Г.Н. CASE-технологии. Консалтинг в автоматизации бизнес – процессов / Г.Н. Калянов. – 3-е изд. – М.: Горячая линия-Телеком, 2002. – 320с.

11) Леоненков, А.В. Самоучитель UML / А.В. Леоненков. – СПб.: БХВ-Петербург, 2002. – 304с.

12) Максимов, Н.В. Технические средства информатизации: учебник / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. – 2-е изд., перераб. и доп. – М.: ФОРУМ: ИНФРА-М, 2008. – 592с.

13) Мельников, В.П. Информационные технологии: учебник /

В.П.Мельников.–2-е изд., стер. – М.: Издательский центр «Академия», 2009.–432с.

14) Орлов С.А., Цилькер Б.Я. Технология разработки программного обеспечения: учебник для вузов / С.А.Орлов, Б.Я. Цилькер. – 4-е изд. Стандарт третьего поколения. – СПб.: Питер, 2012. – 608 с.: ил.

15) Советов, Б.Я. Информационные технологии: учебник для прикладного бакалавриата / Б.Я. Советов, В.В. Цехановский.–6-е изд., перераб. и допол. – М.: Издательство Юрайт, 2015.–263с.

Приложение А. Образец оформления титульного листа

Министерство науки и высшего образования Российской Федерации
Чайковский филиал
федерального государственного автономного образовательного учреждения
высшего образования

«Пермский национальный исследовательский политехнический университет»

Кафедра Автоматизации, информационных и инженерных технологий
Направление: 09.03.01 Информатика и вычислительная техника
Направленность: Автоматизированные системы обработки информации и управления

КУРСОВАЯ РАБОТА

по дисциплине

«Проектирование автоматизированных систем управления»

**Тема: Разработка программного/аппаратного/технологического компонента
автоматизированной системы обработки информации и управления**

Выполнил:

Студент группы _____

(Фамилия Имя Отчество)

Руководитель:

(должность, ученая степень, ученое звание)

(Фамилия Имя Отчество)

Оценка

« _____ » _____ 20 ____ г

(подпись)

Чайковский 20_____

Приложение Б.
Образец оформления задания на курсовую работу
Задание

на выполнение курсовой работы
по дисциплине «Проектирование автоматизированных систем управления»

Выдано _____

Студенту группы _____

Тема курсовой работы:

Разработка программного/аппаратного/технологического компонента автоматизированной системы обработки информации и управления

Разработать программный компонент автоматизированной системы обработки информации и управления для предметной области _____.

Осуществить постановку задачи с представлением предметной области задачи проектирования, анализ существующих или возможных решений поставленной задачи с кратким обзором литературных источников, произвести алгоритмическую проработку решений, выбор среды реализации с использованием средств автоматизации проектирования (использовать CASE-инструмент ERwin, BPwin, Diagram Designer или DIA). В курсовой работе должны быть представлены результаты отладки проектируемых компонент в средствах выбранной CASE-среды.

Этапы выполнения курсовой работы:

1-ый этап

Анализ полученного задания на курсовую работу. Постановка задачи с представлением предметной области задачи проектирования, анализ существующих или возможных решений поставленной задачи с кратким обзором литературных источников.

Срок выполнения: «__» ____ 202__ г.

2-ой этап

Алгоритмическая проработка решений. Выбор среды реализации с использованием средств автоматизации проектирования. Моделирование бизнес-процессов для заданной предметной области с помощью CASE-средств.

Срок выполнения: «__» ____ 202__ г.

3-ий этап

Анализ полученных результатов. Оформление пояснительной записки и графического материала к курсовому проекту.

Срок выполнения: «__» ____ 202__ г.

Сдача работы на проверку преподавателю: «__» ____ 202__ г.

Защита курсового проекта

Сроки защиты: «__» ____ 202__ г.

Дата выдачи задания: «__» ____ 202__ г.

Подпись студента: _____/_____

Подпись преподавателя: _____/Т.И. Русских