

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА

В.П. ДЕРЯБКИН

ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННЫХ
СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ
И УПРАВЛЕНИЯ

Курс лекций для студентов заочной формы обучения

САМАРА 2001

Проектирование автоматизированных систем обработки информации и управления: Курс лекций / В.П. Дерябкин. Самар. гос. аэрокосм. ун-т. Самара, 2001. 120 с.

ISBN 5-7883-0168-8

Изложены основные методы концептуального и логического проектирования автоматизированных систем с применением методологии структурного системного анализа Гейна / Сарсона. На конкретных примерах объясняются основные приемы моделирования и проектирования процессов сбора, обработки и хранения информации в системе с использованием инструментальных средств. Рассмотрены вопросы структурной организации интегрированных автоматизированных систем предприятий. Даются рекомендации по методам обработки информации и предварительной оценки характеристик автоматизированных систем при выборе комплекса технических и программных средств. Кратко изложены требования к терминологии и документированию проектов в соответствии с государственными стандартами.

Предназначен для студентов заочной формы обучения по специальности 220200 «Автоматизированные системы обработки информации и управления». Данный курс лекций будет также полезен студентам других специальностей, связанных с анализом, разработкой и использованием автоматизированных систем, обучающимся как по очной, так и заочной формам обучения. Разработан на кафедре информационных систем и технологий.

Печатается по решению редакционно-издательского совета Самарского государственного аэрокосмического университета имени академика С.П. Королева

Рецензенты: зав. кафедрой информатики Самарской гуманитарной академии, канд. техн. наук, доц. А. В. Б а л а н д и н;
зав. кафедрой компьютерных систем Самарского государственного аэрокосмического университета, д-р техн. наук, проф. А. А. К а л е н т ь е в.

ISBN 5-7883-0168-8 © В.П. Дерябкин, 2001

© Самарский государственный

аэрокосмический университет, 2001

Оглавление

Введение.....	4
1. Общие понятия. Терминология.....	5
1.1 Классификация автоматизированных систем. Виды обеспечения.....	5
1.2. Тиражируемые компоненты АСОИУ.....	8
1.3. Этапы создания и функционирования АСОИУ.....	9
1.4. Элементы технического, программного и информационного обеспечения АСОИУ.....	10
2. Анализ предметной области и разработка концепции построения системы.....	12
2.1. Методология структурного системного анализа Гейна-Сарсона. Схема анализа.....	12
2.2. CASE-системы, поддерживающие методологию Гейна-Сарсона	14
2.3. Анализ задач предметной области. Выделение внешнего окружения. Контекстные диаграммы системы.....	17
2.4. Функции, реализуемые системой. Диаграммы потоков данных верхнего уровня.....	23
2.5. Анализ входных и выходных сигналов системы.....	30
2.6. Диаграммы потоков управления верхнего уровня.....	35
2.7. Концепция построения системы.....	39
3. Детальная проработка логического проекта и базы данных системы.....	41
3.1. Состав логического проекта.....	41
3.2. Анализ информационных запросов пользователей. Отчеты системы.....	42
3.3. Анализ документооборота. График документооборота.....	45
3.4. Структурограммы данных.....	49
3.5. Разработка структуры базы данных. Модель «сущность-связь» (ER-модель).....	54
3.6. Семантическая иерархическая модель (SHM-модель).....	58
3.7. Переход к реляционной модели хранения данных.....	60
3.8. Детализация диаграмм потоков данных.....	64
3.9. Детализация диаграмм потоков управления для систем реального времени.....	65
3.10. Описание логики процессов. Миниспецификации.....	67
4. Техническое проектирование АСОИУ.....	73
4.1. Структура интегрированной АС предприятия.....	73
4.2. Структура и функции АСУП – ядра интегрированной АС.....	76
4.3. Структура и функции АСУТП.....	89
4.4. Оценка характеристик проектируемой системы.....	100
4.5. Особенности реализации системного и прикладного програм- много обеспечения.....	105
4.6. Документирование проекта системы.....	113
Список рекомендуемой литературы.....	119

Введение

Проектирование автоматизированных систем обработки информации и управления (АСОИУ) является сложной задачей, в процессе решения которой приходится рассматривать широкий круг вопросов, связанных с моделированием предметной области, анализом информационных потоков, разработкой схем баз данных и алгоритмов сбора и обработки информации, выбором комплекса технических и программных средств, документированием проекта. В настоящее время проектирование АСОИУ ведется коллективами разработчиков с использованием специальных инструментальных программных систем – CASE-средств (Computer-Aided Software/System Engineering) [1-4]. В качестве теоретического базиса проектирования большинство CASE-технологий используют методы структурного системного анализа, наиболее популярным из которых является методология Гейна-Сарсона, иногда еще называемая DFD-технологией (DFD – Data Flow Diagram, диаграмма потоков данных). Нотация Гейна-Сарсона фактически стала международным стандартом при проектировании и моделировании информационных систем как удобное и понятное средство для описания и задания спецификации будущей системы на логическом уровне, то есть без излишних деталей технической реализации, свойственной документации технического проекта и, тем более, рабочей документации.

В курсе рассмотрены основы концептуального моделирования хранимых данных предметной области двумя различными способами: с помощью ER и SHM-моделей (моделей Чена и Смита соответственно), методика перехода от этих моделей к реляционной модели хранения данных. Достаточно подробно рассмотрена методика структурного системного анализа Гейна-Сарсона, поддержанная рядом инструментальных средств, знакомство с которыми предполагается в лабораторном практикуме.

Этап технического проектирования следует после этапов анализа и логического проектирования и моделирования. Особое внимание уделено структурам и функциям современных интегрированных АСОИУ, включающим в себя различные виды автоматизированных систем. Приведены краткие рекомендации по выбору комплекса технических и программных средств, предварительной оценке технических характеристик проектируемой системы. В заключении рассмотрены вопросы документирования проекта.

1. Общие понятия. Терминология

1.1. Классификация автоматизированных систем. Виды обеспечения

Автоматизированная система обработки информации и управления (АСОИУ) – это обобщенное понятие автоматизированных систем, используемых в различных сферах деятельности (учет, управление, выдача справочной информации и информационных отчетов, анализ данных, исследование, проектирование и др.), основными процессами в которых является хранение, передача и переработка информации.

Согласно ГОСТ 34.003-90 "АС. Термины и определения" *автоматизированная система* (automated system) – система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций. Таким образом, АСОИУ относится к классу человеко-машинных систем, при проектировании которых следует учитывать эргономические требования, т.е. учитывать физические и психологические особенности людей, участвующих в работе системы.

В зависимости от сферы автоматизируемой деятельности выделяют различные частные виды АСОИУ:

- Автоматизированные системы управления (АСУ);
- Системы автоматизированного проектирования (САПР);
- Автоматизированные системы научных исследований и комплексных испытаний образцов новой техники (АСНИ);
- Автоматизированные системы технологической подготовки производства (АСТПП).

Этот перечень можно было бы продолжить, причем каждый частный вид АСОИУ имеет свои особенности и методику проектирования. Однако есть общая теоретическая база для проектирования АСОИУ в целом, особенно на начальных этапах. Этой базой являются методы структурного системного анализа.

Весьма важным в настоящее время является понятие *интегрированной АС* (integrated AS) как совокупности двух или более взаимосвязанных АС, в которой функционирование одной из них зависит от результатов функционирования другой (других) так, что эту совокупность можно рассматривать как единую АС. В интегрированную систему входят различные частные виды АС,

взаимодействующих между собой. Зачастую такие системы территориально рассредоточены и связаны компьютерными сетями различной конфигурации. По существу, АСОИУ любого предприятия среднего и крупного масштаба нужно рассматривать как интегрированную систему.

В процессе своего функционирования АСОИУ реализует определенные функции путем решения различных задач. Под *функцией* АС (AS function) понимается совокупность действий АС, направленных на достижение определенной цели. *Задача* АС (AS problem) – функция или часть функции АС, представляющая собой формализованную совокупность автоматических действий, выполнение которых приводит к результату заданного вида. Например, какие-либо задачи расчетного характера, решаемые по заданию конструктора, приводят к выполнению функции проектирования определенного объекта. Точно также можно рассматривать задачу определения текущих остатков материальных ценностей на складе для выполнения сложной функции управления снабжением (закупками). При моделировании АСОИУ удобно представлять выполнение функций и решение задач процессами обработки данных. На практике вначале на верхнем уровне процессы получения всех результатов рассматривают как функции и лишь затем, по мере уточнения метода получения результата, начинают выделять отдельные задачи, решаемые автоматически, без вмешательства человека. Как правило, на стадиях технического и рабочего проектирования задачи реализуются аппаратно или программно отдельными устройствами или процедурами или их комбинациями, работающими после их запуска автоматически. Логика процессов задается описанием алгоритмов работы с данными, при этом сложные процессы делятся на простые, выполняемые последовательно или одновременно (параллельно).

В качестве основных составных частей АСОИУ рассматриваются следующие:

- пользователи ;
- эксплуатационный персонал;
- организационно-правовое обеспечение;
- методическое обеспечение;
- техническое обеспечение;
- программное обеспечение;
- информационное обеспечение.

В случае необходимости для сложных систем могут выделяться дополнительные виды обеспечения: математическое, лингвистическое (языковое),

эргономическое, метрологическое. Метрологическое обеспечение обычно выделяется из состава других видов обеспечения в случае, когда система включает большое число измерительных каналов, требующих аттестации по показателям точности измерений. Все виды обеспечения, за исключением технического, эргономического и метрологического, представляют собой документы либо файлы на машинных носителях. Техническое, метрологическое и эргономическое обеспечение кроме документов и файлов могут включать технические устройства, компьютеры и аппаратуру связи, необходимую для нормального функционирования системы. В совокупности комплекс видов обеспечения без пользователей и эксплуатационного персонала называют *комплексом средств автоматизации*.

Так как согласно государственным стандартам Российской Федерации документирование и проектирование системы ведется по видам обеспечения, поясним эти понятия более подробно.

Пользователь АС (AS user) – это лицо, участвующее в функционировании АС или использующее результаты ее функционирования.

Эксплуатационный персонал АС (AS maintenance staff) – лица, осуществляющие техническое обслуживание системы во время эксплуатации (инженеры и техники по вычислительной технике, по связи и коммуникациям, системные программисты и т. д.)

Организационно-правовое обеспечение АС (AS organization support) – совокупность документов, устанавливающих организационную структуру, права и обязанности пользователей и эксплуатационного персонала системы. Сюда входят штатное расписание, приказы и положения, инструкции и т. п. Для небольших систем, не имеющих собственного эксплуатационного персонала, может включаться в документацию технического и программного обеспечения.

Методическое обеспечение АС (AS methodical support) складывается из документов или файлов с описанием методов и технологии работы пользователей и решения задач в системе при ее функционировании. Для относительно небольших систем может отдельно не выделяться, а встраиваться в эксплуатационную документацию других видов обеспечения, например, в инструкции по эксплуатации и обслуживанию.

Техническое обеспечение АС (AS hardware) – совокупность всех технических средств, используемых при функционировании системы и сопутствующей технической документации.

Программное обеспечение АС (AS software) – программы на машинных носителях и программные документы (описания, руководства, листинги, контрольные примеры и т. д.). Различают общее (широкого использования) и специальное (прикладное) программное обеспечение.

Информационное обеспечение АС (AS information support) – формы и альбомы документов, классификаторы, описания схем баз данных, способов их взаимодействия и защиты, наконец, сами данные на бумажных и машинных носителях, необходимые для нормального функционирования системы. Для небольших по масштабам систем возможно включение разделов информационного обеспечения в документацию и машинные носители программного обеспечения.

1.2. Тиражируемые компоненты АСОИУ

Поскольку тиражировать АСОИУ в целом весьма затруднительно из-за большого количества специфических особенностей в каждой организации, обычно тиражируют отдельные функционально законченные части системы, как правило, решающие отдельные задачи или группы взаимосвязанных задач. В связи с этим стандартом определены понятия *программного* и *информационного изделия*, *программно-технического комплекса АС (ПТК АС)* как продукции производственно-технического назначения, прошедшей испытания установленного вида и являющейся предметом самостоятельной поставки. Если программное и информационное изделие поставляется на машинных носителях (например, система бухгалтерского учета «1С:Бухгалтерия», юридическая справочная система «Гарант» и др.), то в состав ПТК АС входят средства вычислительной техники, сбора информации, выдачи управляющих воздействий, связи и коммуникаций, программное и информационное обеспечение. Часто отдельные виды ПТК АС называют *автоматизированными рабочими местами (АРМ)* (automated workplace). Например, АРМ оператора-технолога, АРМ бухгалтера, АРМ проектировщика и т. д. Все виды АРМ предназначены для автоматизации деятельности определенного вида и могут являться составными частями АСОИУ.

В информационном обеспечении АСОИУ различного назначения особую роль играют специальным образом организованные и хранимые данные, которые обычно называют *информационной базой АС* (другое название - *информационный фонд*).

В информационной базе сосредоточены данные, несущие полезную информацию для пользователей, обращающихся к системе с *информационными запросами*. Удовлетворение информационных потребностей пользователей - одна из основных и важнейшая функция базы данных наряду со стандартными функциями ввода, вывода, защиты, поиска и редактирования нужных данных.

Информационная база АС делится на *машинную и немашинную информационные базы*. Первая представляет собой совокупность хранимых данных на машинных носителях, вторая – то же, но на документах, непосредственно читаемых человеком без применения средств вычислительной техники (справочные книги, таблицы нормативов и т. д.). Конечно, такое деление является достаточно условным и в последнее время наблюдается тенденция к уменьшению объема немашинной информационной базы. Информационная база системы может тиражироваться отдельно как информационное средство или программно-технический комплекс (в зависимости от комплектности поставки).

В дальнейшем информационную базу системы мы будем называть просто *базой данных*. Проектирование базы данных – самостоятельная и серьезная задача проектирования АСОИУ, Более подробно она рассмотрена в разделе 2.

1.3. Этапы создания и функционирования АСОИУ

Создание и функционирование АСОИУ в совокупности составляет *жизненный цикл системы*, вплоть до ее полной замены или демонтажа. Под *процессом создания АСОИУ* понимается совокупность работ, начиная от формирования исходных требований к системе и до ввода в действие.

ГОСТ 34.601-90 "АС. Стадии создания" устанавливает следующие стадии создания, которые, в свою очередь, могут подразделяться на этапы:

- формирование требований к АС;
- разработка концепции АС;
- техническое задание;
- эскизный проект (стадия не обязательная);
- технический проект;
- рабочая документация;
- ввод в действие.

Первые две стадии могут быть объединены и включают работы по обследованию объекта автоматизации и обоснованию необходимости создания АС. Формируются требования пользователей к АС, проводится сравнительный анализ вариантов концепций построения АС и даются рекомендации по наиболее выгодному и перспективному варианту. Стадии оформляются отчетами о проведенных работах. Названия остальных стадий соответствуют типам разрабатываемой документации. Более подробно состав и содержание документов по стадиям обсуждаются в разделе 4. Основные работы по изготовлению нестандартного оборудования, монтажу технических средств, изготовлению программного и информационного обеспечения, автономной и комплексной отладке компонентов системы и системы в целом ведутся на стадии рабочей документации.

Одновременно ведется подготовка персонала и будущих пользователей системы, разрабатываются программа и методика испытаний.

Ввод в действие подразумевает проведение приемочных испытаний и сдачу системы в эксплуатацию. Для сложных систем вводится понятие *опытной эксплуатации* в течение определенного периода с тщательной регистрацией всех замечаний и их устранением и в последующем с повторным проведением приемочных испытаний.

Во время эксплуатации системы (функционирования) со стороны разработчика системы ведется *сопровождение АС*. Последнее представляет собой деятельность по оказанию услуг, необходимых для обеспечения устойчивого функционирования или развития АС.

Сложные АСОИУ могут вводиться в действие очередями. *Очередь АС* – часть АС, для которой в техническом задании на создание системы в целом установлены отдельные сроки ввода и набор реализуемых функций.

1.4. Элементы технического, программного и информационного обеспечения АСОИУ

Для систем, активно взаимодействующих с технологическими объектами управления или объектами исследования (испытания), например, АСУТП, АСНИ и др., в составе технического обеспечения выделяется *устройство связи с объектом (УСО)*. УСО (data acquisition and control system, DACS) – устройство, предназначенное для ввода сигналов с объекта в АС и вывода сигналов на

объект. В большинстве случаев УСО собирается из стандартных отдельно аттестуемых элементов, однако может включать и нестандартную часть, специально разрабатываемую в рамках данной системы. Основными элементами УСО являются измерительные каналы и каналы выдачи управляющих воздействий, часто включающих в себя датчиковую аппаратуру, усилители, коммутаторы, аналого-цифровые и цифроаналоговые преобразователи, микропроцессорные устройства управления и первичной обработки информации в реальном масштабе времени.

Программное обеспечение АСОИУ делится на *общее и специальное (прикладное)*. Общее программное обеспечение, иногда называемое *общесистемным*, представляет собой совокупность программных средств, разработанных вне связи с созданием данной системы и предназначенных для организации вычислительного процесса и решения часто встречающихся задач обработки информации. Прежде всего это автономные и сетевые операционные системы и системы управления базами данных (СУБД). Специальное программное обеспечение – это совокупность программ, разработанных при создании данной системы, часто называемых *приложениями*.

АСОИУ и ее компоненты работают с информацией различного вида.

Входная информация АС (AS input information) – информация, поступающая в АС в виде документов, сообщений, данных, сигналов, необходимых для выполнения функций АС.

Выходная информация АС (AS output information) – информация, получаемая в результате выполнения функций АС и выдаваемая на объект ее деятельности, пользователю и в другие системы.

Оперативная информация АС (AS rapid information) – информация, отражающая на данный момент времени состояние объекта, на который направлена деятельность АС.

Нормативно-справочная информация АС (AS normative-reference information) – информация, заимствованная из нормативных документов и справочников и используемая при функционировании АС.

Дополнительные сведения об элементах видов обеспечения АСОИУ приведены в разделе 4.

2. Анализ предметной области и разработка концепции построения системы

2.1. Методология структурного системного анализа Гейна-Сарсона. Схема анализа

Основные положения методологии структурного системного анализа информационных систем были сформулированы американскими учеными К. Гейном и Т. Сарсоном в конце 70-х годов [6]. Методология базируется на следующих принципах:

- нисходящая поэтапная разработка;
- диаграммная техника;
- иерархичность описаний;
- строгая формализация описания проектных решений;
- первоначальная проработка проекта на логическом уровне без деталей технической реализации;
- концептуальное моделирование в терминах предметной области для понимания проекта системы заказчиком;
- технологическая поддержка инструментальными средствами (CASE-системами).

Методология использует в процессе разработки особую нотацию (систему условных обозначений), которую мы будем называть нотацией Гейна-Сарсона.

Известен ряд других методологий проектирования АСОИУ (Йордана, Росса, Буча и др.), опирающихся на эти принципы, но со своей нотацией. Среди различных методологий методология Гейна-Сарсона прежде всего в силу простоты, наглядности и эффективности наиболее популярна, стандартизована за рубежом и поддерживается большинством современных CASE-систем, ориентированных на разработку информационных систем.

Результатом работы в среде CASE-системы является информационно-логическая модель АСОИУ, представляемая совокупностью иерархических диаграмм, структурограмм, текстовых описаний и документов. Предполагается, что степень детализации графических и текстовых описаний достаточна для четкого, ясного и однозначного понимания проекта АСОИУ как проектировщиком, так и

заказчиком (будущим пользователем АСОИУ), а также группой технической реализации проекта.

По существу, информационно-логическая модель представляет собой илюстрированную развернутую подробную функциональную спецификацию будущей системы с описанием логики процессов обработки данных и структур передаваемых и хранимых данных. Понятность проекта обеспечивается большим количеством диаграмм, вложенных друг в друга по нарастающей степени детализации, и записью текстов описаний на формализованном языке, близком к естественному русскому языку. При этом детали реализации (формы и размеры экранов, цвета, шрифты, тип протокола связи и т. п.) в информационно-логической модели опускаются, их уточнение происходит в дальнейшем.

Основными компонентами информационно-логической модели системы являются [4-6]:

- контекстная диаграмма (одна или несколько);
- диаграмма потоков данных (одна или несколько);
- структурограмма данных (для каждого потока данных и накопителя данных);
- миниспецификация (для каждого элементарного процесса).

В соответствии с методологией модель системы определяется как иерархия контекстных диаграмм и 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 АСОИУ в целом и основные подсистемы АСОИУ 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 процессы становятся элементарными и детализировать их далее невозможно. Логика элементарных процессов описывается миниспецификациями.

При моделировании информация о всех компонентах проекта заносится в базу данных проекта, часто называемую словарем данных или репозиторием.

Информационно-логическая модель, хранящаяся в базе данных проекта, является достаточно полным описанием АСОИУ независимо от того, является ли она существующей или проектируемой вновь. Это описание освобождено,

насколько это возможно, от деталей реализации и является полной логической функциональной спецификацией системы, понятной как заказчику, так и разработчику. На основе этой спецификации может быть составлено достаточно обоснованное техническое задание на создание или модернизацию системы и проведено техническое проектирование.

Построение информационно-логической модели проводится в несколько этапов, которые могут выполняться повторно по мере уточнения представлений о системе:

- построение контекстной диаграммы верхнего уровня;
- разбиение на подсистемы и построение контекстных диаграмм следующих уровней (этап необязательный и выполняется только для сложных АСОИУ, реализующих большое число функций);
- построение детализирующих диаграмм потоков данных (один или несколько уровней в зависимости от степени сложности системы);
- построение структурограмм потоков данных и накопителей;
- построение ER или SHM-моделей хранимых данных и переход к реляционной модели, уточнение состава и структуры накопителей;
- разработка описаний логики элементарных процессов в виде миниспецификаций.

2.2. CASE-системы, поддерживающие методологию Гейна-Сарсона

В настоящее время известно большое число коммерческих CASE-средств и систем, поддерживающих методологию Гейна-Сарсона, т.е. нотацию DFD-диаграмм. Отметим наиболее популярные (по мере наращивания функциональных возможностей и, соответственно, стоимости системы):

- MetaDesign фирмы Meta Software Corp. (США);
- CASE.Аналитик фирмы Эйтекс (Россия);
- Silverrun фирмы Computer Systems Advisers (США);
- Bpwin фирмы PLATINUM technology (США);
- Vantage Team Builder фирмы Cayenne Software (США);
- Designer/2000 фирмы Oracle (США);
- Visible Analyst Workbench фирмы Visible Systems (США);

- ARIS фирмы IDS Prof. Sheer (Германия);
- PRO-IV WORKBENCH фирмы McDonnell Douglas Information Systems (США)

MetaDesign является недорогим удобным компактным графическим редактором (1 инсталляционная дискета 1,44 Мб) для рисования совокупности иерархически связанных диаграмм в различных условных обозначениях. Имеется несколько шаблонов (инструментальных линеек) для выбора нотации, в том числе нотации по Гейну-Сарсону. Размеры графических обозначений можно менять в широких пределах, количество уровней иерархии диаграмм не ограничено. С помощью специальных средств можно связать с каждым объектом и связью диаграмм текст с настройкой расположения и шрифта . Можно менять внешний вид связи и ее кривизну Однако в системе отсутствует понятие проекта автоматизированной системы и поэтому основные операции над проектом (представление, верификация, построение структурограмм и описаний логики процессов, документирование по стандартам и т.д.) не поддерживаны. На базе этого редактора можно создавать свои собственные CASE-системы, используя средства импорта-экспорта и другие системы программирования.

Инструментальная система CASE.Аналитик [4,5,7] явилась первой отечественной коммерческой CASE-системой, обеспечившей поддержку процесса моделирования и разработки АСОИУ различного назначения на концептуальном и логическом уровнях представления информации. Подробно CASE.Аналитик рассмотрен в учебном пособии [5], здесь отметим только некоторые особенности работы с системой. В системе введено понятие проекта АС, организована база данных хранения проекта и всех его компонентов и введены средства защиты проекта от несанкционированного доступа (проверка фамилий, паролей, кодирование проекта). Данные хранятся в формате СУБД Paradox, однако наличие самой СУБД у пользователя не предполагается. Система компактна (1 инсталляционная дискета 1,44 Мб) и поддерживает основные операции создания проекта в нотации Гейна-Сарсона, включая рисование контекстных диаграмм, диаграмм потоков данных, создание структурограмм данных и миниспецификаций. С самим проектом и его компонентами может быть связана текстовая информация , сгруппированная по разделам (участники проекта, цели, источники финансирования, комментарий, синонимы и т.д.). Эта информация используется документатором автоматически при создании шаблонов документов по ГОСТ 34 XXX В учебной версии допускается только 3 уровня детализации и до 5

сущностей каждого типа на одном уровне. В настоящее время система CASE.Аналитик используется в учебном процессе, однако в перспективе заменяется собственной разработкой ИИС-ПроектГС-02 с более широкими функциональными возможностями и более удобным интерфейсом. В следующих подразделах наряду с общетеоретическим материалом приводятся методические указания по использованию системы CASE.Аналитик.

CASE-система Vрwіn в качестве основной поддерживает методологию Росса SADT (стандарт США IDEF0-функциональная модель) [8] . Однако система позволяет строить так называемую смешанную модель проектируемой АС с дополнительным использованием нотаций Гейна-Сарсона (DFD) и стандарта IDEF3 - WorkFlow Diagrams - диаграмм описания сценариев выполнения работ и их детализации. Диаграммы работ IDEF3 могут экспортироваться в систему имитационного моделирования BPSimulator фирмы Systems Modeling Corporation (США), в которой при задании времени выполнения работ исследуются очереди работ и задержки их выполнения. Для моделирования данных с использованием реляционной модели фирма Platinum предлагает отдельное CASE-средство Erwin, имеющую связь с Vрwіn. Erwin может использоваться самостоятельно или в сочетании с Vрwіn для детальной проработки таблиц базы данных проектируемой АС в нотации, близкой к нотации Чена (ER-модель), с последующим автоматическим формированием SQL-запроса на генерацию структуры базы данных в одной из выбранных целевых СУБД: SQL Server, Oracle и др. Таким образом, диаграммы по Гейну-Сарсону в Vрwіn играют вспомогательную роль и служат для уточнения и лучшего понимания остальных моделей. По нашему мнению, смешение моделей и методологий усложняет систему и делает ее более трудной для понимания и освоения, хотя и расширяет функциональные возможности. Изучение систем Vрwіn и Erwin производится в спецкурсах по проектированию информационных систем и баз данных и знаний. Полностью методология Гейна-Сарсона в Vрwіn не поддерживается.

CASE-система Silverrun поддерживает методологию Гейна-Сарсона и по своим функциям соответствует системе CASE.Аналитик, но, конечно, без поддержки отечественных стандартов в режиме документирования. В редакторе диаграмм допускаются произвольные изменения размеров условных обозначений по вертикали и горизонтали, что может при неаккуратной работе проектировщика привести к искажению вида диаграмм. Относительная компактность и дешевизна способствует использованию этой системы российскими предприятиями. Однако

средства прототипирования, конструирования интерфейса и запросов, имитационного моделирования, реализации и отладки проектируемой АС в Silverrun отсутствуют.

Остальные CASE-системы представляют собой сложные дорогостоящие (свыше 10 000 долларов) программные комплексы, устанавливаемые на мощных рабочих станциях и серверах и автоматизирующие все этапы проектирования и реализации АСОИУ, включая концептуальное и логическое моделирование и проектирование, прототипирование (быстрое изготовление действующих макетов), ведение репозитория (базы данных проекта), управление проектом, конструирование интерфейса и запросов, кодогенерацию по описанию проекта на языке проектирования, автономную и комплексную отладку, документирование и изготовление инсталляционных дискетов и компакт-дисков. Как правило, системы поддерживают смешанное моделирование с использованием большого количества методологий и нотаций с последующей унификацией результатов в репозитории проекта, допускают параметрическую настройку на большое число СУБД и средств программирования. Например, система ARIS допускает использование 83 нотаций моделирования, хотя такая универсальность в большинстве случаев проектирования даже для сложных АСОИУ не нужна. Система PRO-IV WORKBENCH поддерживает язык проектирования высокого уровня PRO-IV, допускающий последующую автоматическую кодогенерацию программного обеспечения на одном из выбранных языков программирования.

2.3. Анализ задач предметной области. Выделение внешнего окружения. Контекстные диаграммы системы

Формирование требований к АС и разработка концепции АС являются начальными стадиями создания любой автоматизированной системы и, согласно методологии Гейна-Сарсона, выполняются с применением методов структурного системного анализа. В соответствии с принципами этой методологии строится информационно-логическая модель системы, которая и подвергается анализу.

Для принципиально новых производств и технологий создается модель будущей системы. Для действующих производств и технологий предварительно строится модель существующей системы, которая подвергается критическому анализу. На основе результатов этого анализа далее создается модель будущей

системы. Основные требования к модели – строгость определений, не допускающая неоднозначности их толкований, достаточная полнота и наглядность (понятность) для всех участников проекта, включая заказчиков, пользователей и проектировщиков системы. Построение такой модели без использования CASE-систем весьма трудоемко и в настоящее время применяется только в учебных целях на упрощенных примерах. Хорошо построенная модель в дальнейшем существенно уменьшает вероятность дорогостоящих переделок проекта и самой системы на последующих стадиях жизненного цикла АС.

Создание информационно-логической модели проводится в несколько этапов, которые могут выполняться повторно по мере уточнения представлений о системе:

- построение контекстной диаграммы верхнего уровня;
- разбиение на подсистемы и построение контекстных диаграмм следующих уровней (этап необязательный и выполняется только для сложных АСОИУ, реализующих большое число функций);
- построение детализирующих диаграмм потоков данных (один или несколько уровней в зависимости от степени сложности системы);
- построение структурограмм потоков данных и накопителей;
- построение ER или SHM-моделей хранимых данных и переход к реляционной модели, уточнение состава и структуры накопителей;
- разработка описаний логики элементарных процессов в виде миниспецификаций.

Создание информационно-логической модели начинается с определения контекста системы, то есть выявления внешнего окружения и границ действующей или проектируемой системы. С этой целью строится контекстная диаграмма верхнего уровня, в которой присутствует анализируемая система (обозначается единственным символом системы(подсистемы)), связанная потоками данных с внешними сущностями – внешними по отношению к системе источниками/приемниками информации.

Указание. При использовании системы CASE Аналитик необходимо начать новый проект и ввести минимально необходимые сведения о проекте: названия проекта и системы, фамилию, имя, отчество разработчика, пароль и директорию на диске, в которой будет располагаться проект. После этого автоматически создается нужная директория и включается окно редактирования контекстных диаграмм.

Контекстная диаграмма включает в себя следующие компоненты:

- поток данных;

- поток управления (не обязательно);
- система /подсистема;
- внешняя сущность;
- информационный канал (не обязательно).

Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам. Те, в свою очередь, преобразуют информацию и порождают новые потоки, которые переносят информацию к другим подсистемам или внешним сущностям - потребителям информации, возможно, с использованием информационных каналов. Каждый компонент контекстной диаграммы имеет свое условное обозначение и связанный с ним пояснительный текст определенной структуры.

Система/подсистема изображается, как показано на рис. 1.

В поле имени указывается наименование системы или подсистемы в виде предложения с подлежащим и с соответствующими определениями и дополнениями, например: «Рабочее место бухгалтера», «АС предприятия», «Подсистема учета работы сотрудников». При построении модели простой АСОИУ система в целом отображается на контекстной диаграмме одним символом. Для сложных (и, как правило, пространственно распределенных) систем осуществляется разбиение системы на подсистемы и АСОИУ будет отображаться на контекстной диаграмме несколькими символами подсистем.

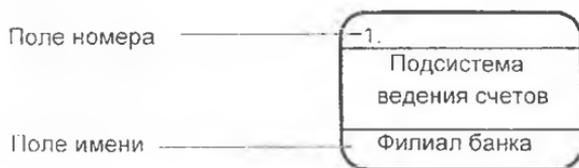


Рис. 1. Условное обозначение системы/подсистемы

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

сущность. Основным признаком внешней сущности является то, что она не выполняет никакой работы по обработке данных внутри системы и по отношению к системе ведет себя только как источник или приемник данных, например, обращается к системе с запросами или получает из нее отчеты.

Внешняя сущность обозначается квадратом (рис. 2), расположенным как бы "над" диаграммой и бросающим на нее тень, для того чтобы можно было выделить этот символ среди других обозначений:



Рис. 2. Изображение внешней сущности на диаграмме

Поток данных изображается линией с горизонтальными и вертикальными участками (или дугой), заканчивающейся стрелкой. Направление стрелки указывает направление потока. Вдоль стрелки проставляется содержательное имя потока. По существу, поток – это логическая структура данных, которыми обмениваются между собой основные компоненты контекстной диаграммы: подсистемы с подсистемами и внешние сущности с системой или подсистемами.

Поток управления используется для анализа систем реального времени и рассматривается в подразделе 2.6.

Информационный канал (рис.3) логически отображает на диаграмме среду передачи информации для пространственно распределенных АСОИУ. Он не производит никаких действий по обработке данных и просто передает логическую структуру данных на определенное расстояние без изменения ее содержания. Информационный канал может реализоваться в виде, например, почты, курьерской службы, магистрали или шины данных, канала сети Интернет и т.д.

Информационный канал именуется подлежащим с соответствующими определениями. Для уменьшения числа пересечений линий потоков данных на диаграмме может создаваться несколько копий одного и того же информационного канала.

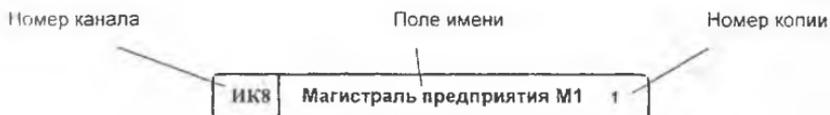


Рис. 3. Условное обозначение информационного канала

Контекстная диаграмма строится на основе предварительного анализа предметной области путем изучения естественных текстовых описаний, требований заказчика, условий работы пользователей, решающих рассматриваемые задачи своими способами, или на основе предположений о решении этих задач, если система будет реализована и запущена в эксплуатацию.

Обычно при проектировании или анализе относительно простых АСОИУ создается единственная контекстная диаграмма первого уровня с топологией звезды, в центре которой находится символ системы, соединенный с источниками и приемниками информации, посредством которых с АСОИУ обмениваются информацией пользователи и другие системы.

Для сложных систем такая контекстная диаграмма неприменима, так как будет содержать большое количество внешних сущностей, которые трудно и даже невозможно расположить на листе бумаги разумного формата. В этом случае АСОИУ разбивается на ряд подсистем, соединенных потоками данных. На рис. 4 приведен пример контекстной диаграммы, нарисованной в CASE.Аналитике. Директор, обращающийся с информационным запросом к АСОИУ о выполнении заказа, является внешним источником-приемником информации и изображается на диаграмме символом внешней сущности.

На верхнем уровне в контекстной диаграмме внешние сущности могут обобщаться (например, «руководство», «заказчики» и т. п.) и более детально показываться на контекстных диаграммах следующих уровней иерархии, раскрывающих структуру подсистем верхнего уровня. Индекс «ДПД» в поле номера системы на рис. 4 означает, что на следующем уровне иерархии АСОИУ детализируется диаграммой потоков данных. Если на следующем уровне детализация идет через контекстную диаграмму, то в этом поле проставляется индекс «КД». Других способов детализации системы/подсистемы не существует.

Знак вопроса означает, что на следующем уровне детализация не произведена. Таким образом, в информационно-логической модели должна присутствовать обязательно как минимум одна контекстная диаграмма системы.

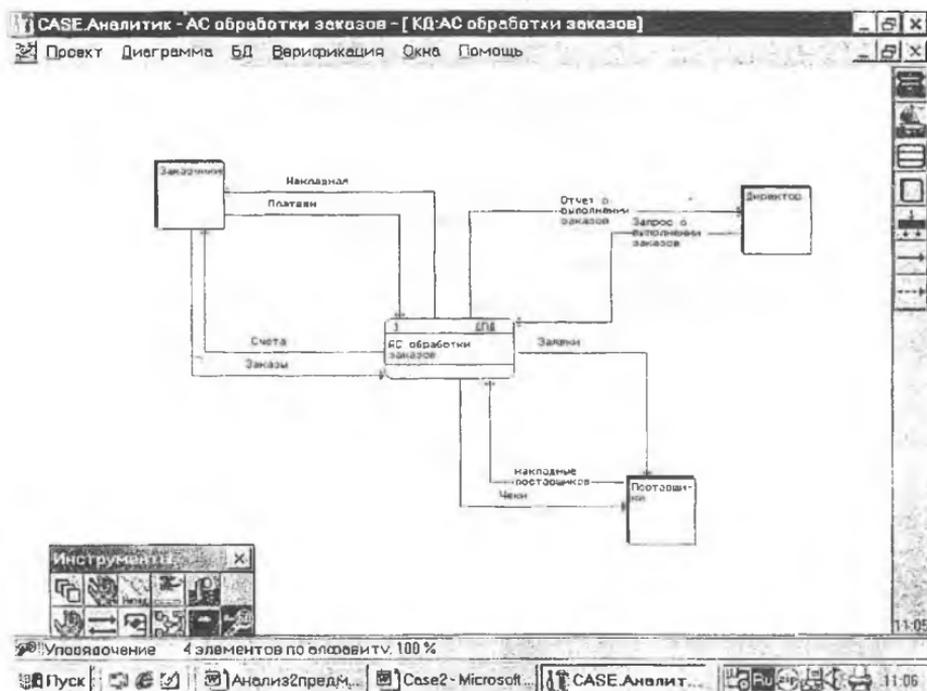


Рис. 4. Пример контекстной диаграммы системы

Указание. В системе CASE.Аналитик окно редактора контекстных диаграмм справа вверху содержит набор кнопок (рис. 4, сверху вниз) : инструменты редактирования, окно навигации по диаграмме, система(подсистема), внешняя сущность, информационный канал, поток данных, поток управления. Рисование диаграммы осуществляется методом «drag-and-drop» - «перетаскиванием» символа при нажатой левой кнопке мыши на поле диаграммы и последующем отпускании кнопки. Инструменты редактирования (набор кнопок в левом нижнем углу слева направо по рядам сверху) включают создание копий (внешних

сущностей и информационных каналов), пересвязывание потока данных с новым объектом, возврат на один шаг редактирования, обзор общего вида диаграммы, подъем на один уровень вверх, захват объекта с целью его переноса на новое место, изменение направления потока данных, удаление объекта, возврат к обычному виду диаграммы из общего обзора, погружение на уровень вниз, включение окна редактирования детальной информации об объекте.

Для сложных систем на следующем уровне снова строится контекстная диаграмма с разбиением на подсистемы. Выделение подсистем обычно происходит либо по функциональному признаку (выполнение относительно обособленной крупной функции, например, подсистема технологической подготовки производства), либо по территориальному расположению (например, подсистема управления филиалом фирмы). Рекомендуемое количество подсистем – не более 10, иначе следует создавать контекстные диаграммы следующих уровней детализации.

Указание. В учебной версии CASE.Аналитик допускается на втором уровне не более 5 подсистем (третий уровень детализации – последний и заполняется диаграммами потоков данных).

2.3. Функции, реализуемые системой. Диаграммы потоков данных верхнего уровня

При моделировании системы важной является классификация внешних источников информации (внешних сущностей - источников) на *источники оперативной информации, источники нормативно-справочной информации и источники информационных запросов* (см. раздел 1.3). В модели потоки данных от этих источников должны быть отделены друг от друга.

Оперативная информация отражает текущее состояние объекта автоматизации и может вводиться *синхронно* в заранее определенные моменты времени (по периодическим запросам системы, например, по опросам датчиковой аппаратуры, периодическому получению информации от ответственных лиц и т.п.) и *асинхронно* – в случайные моменты времени, по мере готовности источника и наступления каких-либо событий, связанных с изменением состояния объекта. Особенностью потоков оперативной информации является их высокая интенсивность, особенно для систем реального времени. Часто оперативная информация вводится в систему в виде документов или сообщений по различным

каналам связи. В модели должны быть отражены операции по контролю и вводу такой информации в информационную базу системы, при этом некоторые операции могут выполняться вручную (операторами или другими ответственными лицами). Отражается это введением соответствующих процессов в диаграмму потоков данных, либо усложнением логики выполнения какого-либо процесса, связанного с такой операцией.

Нормативно-справочная информация обычно вводится администратором базы данных системы или другими ответственными лицами. Только они имеют право изменять эту информацию, так как произвольное ее изменение может привести к некорректной работе системы. При настройке и запуске системы в эксплуатацию справочники заполняются и настраиваются в первую очередь. Потоки нормативно-справочной информации обладают обычно меньшей интенсивностью, чем остальные потоки, однако должны быть предприняты специальные меры по защите и сохранению такой информации в базе данных системы. Логика процессов обработки этой информации сводится к выполнению стандартных функций работы с записями.

Информационные запросы пользователей системы подразделяются на две категории: с выводом результатов только на экран (*экранные*) и с дополнительным выводом на печать (*документальные*). Последние сложнее в реализации, так как требуют разработки и отладки формы документа и использования специального генератора отчетов. По логике процессов подготовка этих отчетов как одной и той же структуры данных практически одинакова. Ответом системы на информационные запросы являются потоки соответствующих отчетов. В большинстве случаев запросы носят асинхронный (случайный во времени) характер, хотя и возможны *регулярные* отчеты, предоставляемые системой определенным внешним приемникам в заранее запланированные моменты времени (например, к 8 утра ежедневно). В этом случае входной поток информационных запросов может не указываться.

Для построения более детальной модели должны быть изучены или намечены функции, задачи и операции системы. Они должны быть связаны с документооборотом в системе. Регламент документооборота определяет регламент выполнения основных процессов системы и ее взаимодействие с внешним окружением.

Диаграмма потоков данных (ДПД или DFD) является дальнейшей детализацией контекстной диаграммы и раскрывает функциональное содержание символа системы/подсистемы.

Диаграмма потоков данных включает в себя следующие компоненты:

- процесс;
- поток данных;
- внешняя сущность;
- накопитель данных (не обязательно);
- информационный канал (не обязательно).

Процесс – это преобразование входных потоков данных в выходные потоки в соответствии со своей внутренней логикой. Поскольку с логической точки зрения потоки данных представляют собой некоторые именованные структуры данных с конкретными значениями элементов данных, то в результате преобразования должны меняться либо значения элементов данных, либо состав элементов и способы их связи в структуре данных. Недопустимы процессы, которые не имеют входных либо выходных потоков данных. Также недопустимы процессы, которые ничего не меняют во входном потоке. Коммутаторы и маршрутизаторы рассматриваются как процессы, меняющие во входном потоке адресные элементы. Устройства задержки во времени в данной методологии не рассматриваются, так как все процессы считаются активными после поступления на их входы хотя бы одного из входных потоков, а все возможные задержки учитываются внутренней логикой самого процесса. Кроме того, считается, что АСОИУ работает в бесконечном цикле. Это позволяет отвлечься в информационно-логической модели от деталей технической реализации запуска и останова системы. Для большинства АСОИУ, в которых не очень важными являются показатели быстродействия (допустимое время реакции при выдаче отчетов порядка нескольких секунд или даже минут), это допущение вполне правомерно.

Анализ АСОИУ реального времени более сложен и сводится к анализу схемы событий и переходов системы из состояния в состояние. Элементы такого анализа рассмотрены в разделе.

Условное обозначение процесса на диаграмме потоков данных приведено на рис. 5 (его не следует путать с символом системы/подсистемы).



Рис. 5. Условное обозначение процесса

Номер процесса включает в себя номера компонентов диаграмм разных уровней, детализацией которых является процесс. Номера располагаются последовательно по уровням и разделяются точкой. Последний номер является порядковым номером компонента в диаграмме, в которой присутствует данный процесс. Например, 1.3.2 при трехуровневой детализации и т.д. Таким образом, можно по номеру определить, к какому компоненту относится данный процесс на любом уровне.

Имя процесса следует представлять в форме предложения с глаголом в неопределенной форме (вычислить, определить, рассчитать) или отглагольной формой (вычисление, определение, расчет), т.е. четко подчеркнуть действие, выполняемое процессом.

Поле физической реализации следует заполнять в том случае, если есть ясность о способе физической реализации процесса (исполнитель, программный модуль и т.д.).

Накопитель данных логически представляет собой некоторое хранилище информации, куда ее можно поместить и через некоторое время взять. Данный компонент диаграммы является, как и информационный канал, пассивным, т.е. не ведущим никакой обработки данных. Поэтому логическая структура накопителя должна полностью соответствовать структурам входных потоков данных. Более того, поскольку хранение лишней информации недопустимо, должен соблюдаться *принцип баланса накопителя*: все, что помещается в накопитель входными потоками, должно браться из накопителя выходными потоками данных. Отсюда следует, что сумма структур входных потоков в накопитель должна быть равна сумме структур выходных потоков из накопителя и равна структуре самого накопителя. Если в системе хранение данных не используется, то накопитель может отсутствовать. Для современных АСОИУ с большим объемом хранимых данных диаграммы могут включать большое число накопителей на разных уровнях. Обозначается накопитель, как показано на рис. 6.

Рис. 6. Условное обозначение накопителя данных

Внешняя сущность, поток данных и информационный канал здесь используются в том же смысле и с теми же условными обозначениями, что и в контекстных диаграммах. Однако при детализации подсистемы или процесса все связанные с ними компоненты на следующем уровне становятся внешними сущностями.

Указание. В CASE.Аналитик, чтобы не усложнять понимание диаграмм, для них условные обозначения сохраняются, но становятся блеклыми.

Пример диаграммы потоков данных второго уровня, детализирующей АС обработки заказов (рис. 4), приведен на рис. 7.

Символы «ДПД», «МС» и «?», стоящие в поле номера, означают способ дальнейшей детализации процесса (диаграмма потоков данных, миниспецификация, пока не определен - соответственно). Данная диаграмма сбалансирована с контекстной диаграммой рис. 4 по внешним сущностям и потокам данных.

В системе выделен центральный процесс-диспетчер 1 «Управлять обработкой данных». На его вход поступают все входящие в систему потоки данных. Процесс анализирует виды запросов, осуществляет ряд контрольных операций и распределяет данные запросов по другим процессам. Здесь на него же возложены функции ведения базы данных системы. Реализация данного процесса сводится к реализации главного меню системы и интерфейса с пользователями в виде главного программного модуля или процедуры. Остальные процессы выполняют функции, необходимые для работы АС: оформление и отсылку счета заказчику, взаимодействие с поставщиком, формирование накладной на отпуск товара заказчику после приема сведений о произведенных им платежах.

Для хранения оперативной и справочной информации в АС предусмотрены два накопителя: БД1 «Заказы» и БД2 «Заявки поставщикам». Для того чтобы избежать многочисленных пересечений линий информационных потоков, в диаграмме создана копия условного обозначения накопителя БД1.

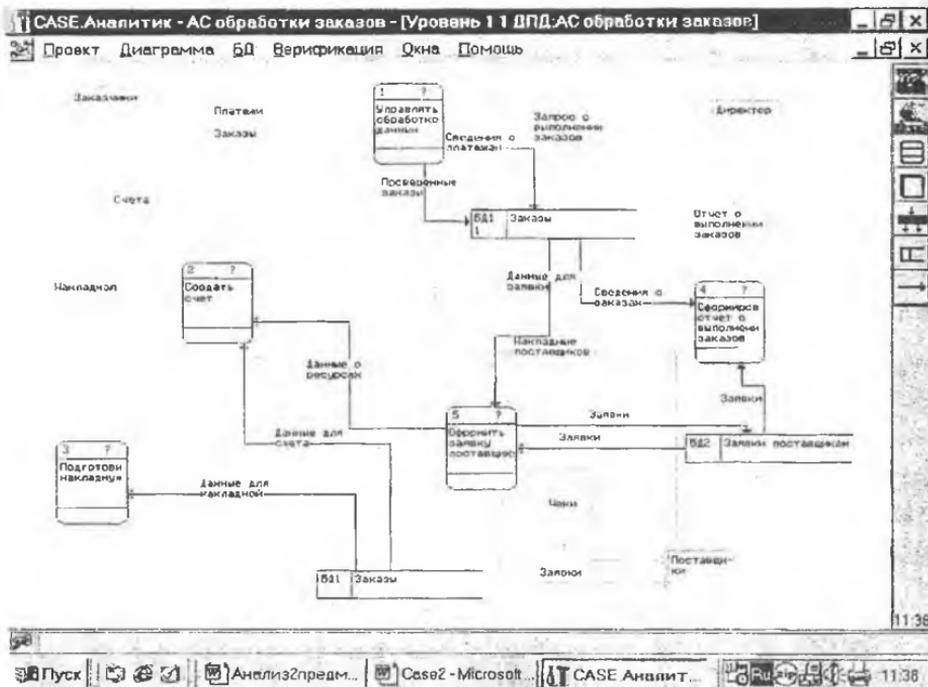


Рис. 7. Диаграмма потоков данных, детализирующая АС обработки заказов

Рекомендации по построению диаграмм:

1. Имена входных и выходных потоков процессов должны быть различны.
2. Миниспецификации создаются для элементарных процессов (объем миниспецификации не более 2-3 страниц текста). Иначе требуется дальнейшая детализация сложного процесса с помощью ДПД.
3. Недопустимы следующие соединения потоками данных: внешняя сущность - внешняя сущность, внешняя сущность - накопитель, накопитель-накопитель, информационный канал - информационный канал (эти объекты должны связываться между собой через процессы).
4. Множественные потоки данных от внешней сущности или процесса, идущие в одном направлении, на верхних уровнях лучше объединять под одним названием и направлять на обобщенный процесс типа «Анализировать вид запроса» или «Распределять запросы», логика которого более детально определяется на нижних уровнях (это делает диаграммы более понятными).

5. Для уменьшения числа пересечений линий потоков данных внешние сущности, накопители и информационные каналы могут копироваться с указанием номера копии в нижнем правом углу.
6. Имена компонентов диаграмм должны быть уникальными.
7. С каждым компонентом рекомендуется связывать дополнительную поясняющую информацию (комментарий и др.).

Указание. Редактирование диаграммы потоков данных в системе CASE.Аналитик происходит в окне редактирования аналогично редактированию контекстной диаграммы (см. выше). Однако палитра объектов ДПД отличается от палитры объектов КД: кнопка «Подсистема» заменена на кнопку «Процесс», добавлена кнопка «Накопитель данных». Операции с диаграммой в целом могут быть выполнены через раздел меню *Диаграмма*. При выборе *Диаграмма\Загрузить* активизируется окно «Дерево проекта», в котором отображается иерархическая структура всего проекта. При выборе *Диаграмма\Сохранить* текущая диаграмма запоминается в базе данных проекта. Эта функция дублируется при завершении работы. Команда меню *Диаграмма\Удалить* удаляет текущую диаграмму, но только в том случае, если все ее элементы не являются детализированными. Иначе следует удалить все детализирующие диаграммы, начиная с самого нижнего уровня. При помощи команды *Диаграмма\Спецификация* можно посмотреть общие данные о диаграмме. Командой *Диаграмма\Выбор* экранного шрифта можно задать тип, стиль и размер шрифта, которым будут выводиться надписи на объектах диаграммы. По умолчанию система CASE.Аналитик использует шрифт Fontsez. Для распечатки на принтере диаграммы используется команда меню *Диаграмма\Распечатать*. Предварительно командой *Диаграмма\Настройка* печати настраивается принтер и указывается формат диаграммы.

Следует помнить, что процесс на следующем уровне может раскрываться либо диаграммой потоков данных, либо миниспецификацией. Потоки данных и накопители раскрываются структурограммами (см. раздел 2.4).

2.4. Анализ входных и выходных сигналов системы

Для отдельных видов АСОИУ (например, АСНИ, АСУ ТП, диспетчерские и тренажерные комплексы и др.) важным является получение информации о значениях характеристик объектов в определенные моменты времени с помощью технических средств (датчиковая аппаратура, аналого-цифровые преобразователи, сигнализаторы и т.д.). Как правило, получение такой информации осуществляется в реальном времени с учетом скорости изменения характеристик. АСОИУ может также формировать и выдавать на объекты управляющие воздействия с помощью технических устройств (цифроаналоговые преобразователи, реле, сельсины, потенциометры и т.д.) также с привязкой к определенным моментам реального времени. Такие потоки данных называются *входными* и *выходными сигналами* соответственно. С логической точки зрения они представляют последовательности значений, связанных с моментами времени, т.е. они по смыслу аналогичны математической функции от времени. Одной из основных задач, решаемой при этом системой, является задача измерения физических величин. Для решения этой задачи в зависимости от сложности применяемых алгоритмов обработки информации могут выделяться в составе АСОИУ отдельные подсистемы. В общем случае измерение рассматривается как отображение состояния характеристик реального объекта в множество действительных чисел. Диапазон чисел зависит от алгоритма или способа отображения, всегда условен, зависит от выбора эталонной единицы измерения и коэффициента линейного масштабирования.

Важным является то, чтобы при любых изменениях состояний характеристик этот диапазон укладывался в разрядную сетку ЭВМ, а количество значений в единицу времени (интенсивность потока) не превышало максимального значения, которое может обработать и сформировать АСОИУ за период решения задачи без каких-либо существенных потерь информации. Даже в том случае, если измерение носит разовый характер, мы условно рассматриваем его как сигнал, т.к. предполагаем повторное неоднократное использование системы на разных объектах или в изменяющихся условиях.

Целью структурного системного анализа сигналов на стадии определения концепции построения системы – выявить все внешние сущности (источники/приемники), генерирующие входные сигналы и принимающие выходные сигналы АСОИУ, а также связанные с ними задачи и ограничения на

реализацию. При этом само устройство связи с объектом, датчиковая аппаратура и исполнительные сигнальные элементы могут включаться в состав АСОИУ (на диаграммах это приведет к появлению соответствующих подсистем, процессов и каналов связи). В состав АСОИУ не включается сам объект управления или исследования, пользователи и любые внешние источники и приемники данных, не выполняющие какой-либо работы внутри системы. Все они на диаграммах обозначаются как внешние сущности.

Входные и выходные сигналы делятся на *непрерывные (аналоговые)* и *дискретные (число-импульсные)*. Большинство физических процессов в объектах являются непрерывными функциями времени. Для измерения и анализа таких процессов с использованием ЭВМ их необходимо преобразовать в цифровую форму. *Аналого-цифровое преобразование* включает две операции: *дискретизацию (выборку)* и *квантование* [9]. Первая фиксирует моменты времени, к которым привязывается значение характеристики. Вторая осуществляет преобразование информации из аналоговой формы в цифровую, т. е. в число, которое показывает, сколько раз содержится некоторая эталонная величина в исходном значении сигнала. Каждая пара: значение и момент времени образует один *отсчет*. Количество отсчетов в единицу времени и количество уровней квантования (определяемых величиной эталона) должны быть достаточными для восстановления исходного сигнала с требуемой точностью. В то же время оно не должно быть излишним, так как это предъявляет дополнительные требования к техническому обеспечению, увеличивает объем требуемой памяти, увеличивает трудоемкость и стоимость решения задач.

В соответствии с принципами гармонического анализа произвольная математическая функция может быть представлена суммой синусоидальных составляющих, имеющих различные амплитуды, частоты и фазы. Согласно известной теореме выборки при периоде дискретизации T секунд синусоидальные составляющие с частотами выше $1/2T$ герц не могут быть достоверно восстановлены. Поэтому важна оценка самой высокой частоты F_{max} , входящей в состав спектра сигнала. Система должна обеспечивать не менее двух отсчетов за период $T_{min}=1/F_{max}$, т.е. период дискретизации должен быть равен $T_{min}/2$.

Величина эталона квантования (*шаг квантования*) зависит от вероятностных характеристик совокупности значений сигнала. Практически можно считать, что для нормального и равномерного законов распределения шаг квантования $g=R/8$, где R – диапазон (интервал) изменения сигнала в

соответствующих единицах измерения. Это обеспечивает точное восстановление математического ожидания и дисперсии значений сигнала при обработке квантованных (цифровых) значений. Например, абсолютная ошибка оценки дисперсии по квантованным значениям с этим шагом при нормальном распределении значений в диапазоне от -1 до +1 составит всего 0,000001. Для более сложных законов, например, состоящих из нескольких колоколообразных пиков, шаг квантования выбирается равным 1/8 ширины самого узкого пика. Необходимо отметить, что для построения функции распределения с приемлемым уровнем достоверности статистика значений сигнала должна содержать не менее 100-150 значений. Операция квантования сигнала по уровню осуществляется специальным техническим устройством – *аналого-цифровым преобразователем (АЦП)*.

При выводе управляющих воздействий применяется обратная операция – *цифроаналоговое преобразование*, когда квантованный сигнал, значения которого представлены числами из некоторого дискретного множества (точнее – двоичными кодами чисел), преобразуется в пропорциональную аналоговую величину – непрерывную функцию времени со значениями из соответствующего непрерывного диапазона значений. Цифроаналоговое преобразование выполняется специальными техническими устройствами – *цифроаналоговыми преобразователями (ЦАП)*.

Чем меньше шаг квантования, тем больше двоичных или десятичных разрядов требуется для представления результата квантования – числа. Поэтому *разрядность*, характеризующая максимальное число, которое может сформировать и обработать устройство, является одной из важнейших характеристик АЦП и ЦАП. Например, при разрядности 8 двоичных разрядов максимально возможное число квантов в квантованном значении будет $2^{**}8=256$. Размер кванта в натуральных единицах измерения зависит от диапазона изменения сигнала. В последнем случае при диапазоне изменения напряжения, например от 0 до 5 В, размер кванта составит 5/256 или 0,01953125 В (около 20 милливольт). В пределах этой же величины будет лежать абсолютная методическая погрешность самого аналого-цифрового преобразования.

На преобразование требуется время, поэтому быстродействие АЦП и ЦАП (величина, обратная количеству преобразований в единицу времени) определяет минимальный интервал времени между соседними отсчетами в сигнале.

Одной из основных количественных характеристик процесса измерения является *погрешность* – разность между измеренным значением характеристики и ее истинным значением в тех же эталонных единицах для определенного момента времени. Погрешность складывается из отдельных составляющих:

- основная погрешность;
- дополнительная погрешность.

Основная погрешность – это погрешность измерения характеристики при определенных условиях, называемых *эталонными*. При этом предполагается, что источником входного сигнала является неизменная во времени образцовая мера (эталон) – заранее заданная величина измеряемой характеристики. Внешние условия измерения (характеристики окружающей среды) предполагаются фиксированными и неизменными на протяжении всего процесса измерений. Таким образом, на формирование основной погрешности влияют только внутренние факторы: выбранный метод измерения, неточности дискретизации и квантования, погрешности передачи сигнала по выбранному каналу связи и ошибки в работе комплекса технических, программных и информационных средств подсистемы измерений.

Дополнительная погрешность – это погрешность, возникающая при отличии в процессе измерений условий от эталонных (например, повышенная влажность, давление, наличие электромагнитных полей и излучений, частичное изменение состава и характеристик оборудования, программного обеспечения и т. д.).

Количественно погрешности оцениваются в единицах измерения (*абсолютная погрешность*) или в процентах, или долях относительно истинного значения или верхней границы измерений, или диапазона измерений – шкалы (*относительная погрешность*), каждый случай оговаривается особо. Одновременное и меняющееся во времени действие различных факторов приводит к тому, что погрешности являются случайными процессами, суммируемыми с истинными значениями и искажающими входной сигнал. При анализе входных сигналов важно установить, какое значение погрешности по каждому измерительному каналу является допустимым для нормального функционирования АС, а также какие характеристики внешней среды и АС могут подвергаться изменению во время эксплуатации системы и в каких пределах. Несмотря на то что сигнал в процессе измерения последовательно проходит этапы своего преобразования, погрешности в силу своей случайной природы

арифметически не суммируются. В предположении нормального закона распределения суммарную погрешность оценивают по формуле

$$x = \sqrt{\sum_{i=1}^n x_i^2},$$

где x_i – погрешности отдельных частей измерительного канала.

Для выходных сигналов погрешности вводятся и оцениваются аналогично с заменой термина «измеряемое значение» на «значение выходного сигнала», а «измерительный канал» на «канал выдачи выходного воздействия (сигнала)».

Анализ входных и выходных сигналов включает:

- выявление всех внешних источников и приемников сигналов и их именование (например, «стенд», «камера», «объект исследования». «система стабилизации скорости» и т.д.);
- выявление всех входных и выходных потоков данных, содержащих сигналы, и их именование (например, «результаты измерений», «задание скорости», «измерения температур», «управляющие воздействия», «температура в камере», «давление на входе», «положение узла» и т. п.);
- выявление всех входных и выходных сигналов как элементов данных, их именование и определение основных характеристик и ограничений по каждому сигналу.

Результаты анализа отображаются на контекстных диаграммах и диаграммах потоков данных.

По каждому сигналу на этапе определения концепции построения системы фиксируется:

- имя сигнала;
- тип сигнала (входной, выходной, аналоговый, дискретный);
- единица измерения;
- диапазон изменения R (может быть несколько для сложных функций распределения);
- типичное значение (может быть несколько);
- максимальная частота спектра сигнала F_{\max} ;
- количество состояний дискретного сигнала и типичные значения сигнала в этих состояниях;

- максимальная частота изменения состояния дискретного сигнала.

Указание. В системе CASE Аналитик дополнительные данные вводятся при образовании нового потока данных в диаграммах потоков данных, либо элементов данных в структурограммах данных (раздел 3) в специальных полях, или в поле комментария.

2.5. Диаграммы потоков управления

Анализ и проектирование систем реального времени по методологии Гейна-Сарсона включает в себя возможную детализацию систем и подсистем через диаграмму потоков управления. Если временные характеристики выполнения (функций и решения задач не являются достаточно жесткими, например, более 1 секунды, причем за это время изменений в состоянии внешних источников и системы не происходит, то АСОИУ не относят на данном этапе к системам реального времени и строят исполнение как единую последовательность процессов, считая время формирования результата любым процессом равным 0.

Вообще говоря, любая АСОИУ работает в реальном времени и пренебрежение временем обработки информации и формирования управляющих воздействий и отчетов есть абстракция, которую можно принять, если за указанное время никаких существенных изменений во внешней среде и внутреннем устройстве системы не происходит.

Диаграмма потоков управления включает в себя следующие компоненты:

- управляющий процесс;
- процесс;
- поток данных;
- поток управления;
- внешняя сущность;
- накопитель данных (не обязательно);
- информационный канал (не обязательно).

Диаграмма потоков управления [4] – это направленный граф, узлами которого являются процессы и управляющие процессы, а дугами – информационные потоки и потоки управления.

Управляющий процесс можно логически представлять в виде некоторого командного пункта, который, реагируя на наступление некоторых событий, в

соответствии со своей внутренней логикой выдает команды, выполняемые остальными процессами. Выдача каждой такой команды также является событием, она влечет за собой изменение состояния процесса, на вход которого она поступает. Так как работа АСОИУ на логическом уровне рассматривается в виде бесконечного цикла, то можно говорить о *потоке входных событий* и *потоке выходных событий* и о преобразовании управляющим процессом входного потока событий в выходной поток событий. Управляющий процесс на диаграммах обозначается в виде пунктирного прямоугольника со скругленными углами (рис. 8).



Рис. 8. Условное обозначение управляющего процесса

Поток управления изображается пунктирной линией с горизонтальными и вертикальными участками (или дугой), заканчивающейся стрелкой. Направление стрелки указывает направление потока. Вдоль стрелки проставляется содержательное имя потока. По существу, это – потоки событий, которыми обмениваются между собой основные компоненты контекстной диаграммы: подсистемы с подсистемами и внешние сущности с системой или подсистемами, а также компоненты ДПД: внешние сущности с процессами и управляющими процессами, процессы всех видов между собой.

На диаграммах лучше разделять обычные потоки данных и потоки управления, вводя для потоков управления элементы данных типа событий (например, «старт процесса 1.1» и т.д., см. рис. 10). Обработываться события могут только управляющими процессами, обычные процессы обработки данных могут лишь потреблять или порождать события, связанные с изменением их состояний. Любой процесс может находиться с точки зрения его исполнения в одном из следующих состояний: *пассивное*, *активное* (выполняется) и *пауза* (приостановлен) (рис.9). Состояние процесса может изменяться двумя способами: потоками управления от некоторого управляющего процесса и по инициативе самого процесса. В последнем случае допускается переход только в состояние «Приостановлен» с передачей события управляющему процессу. Все переходы носят характер прерываний. Вместе с событием допустима передача параметров.

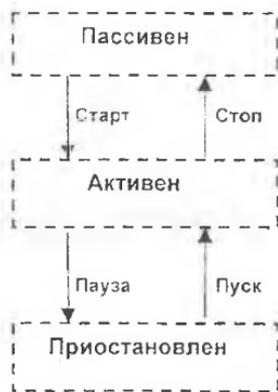


Рис.9. Диаграмма переходов состояний процесса

Остальные компоненты диаграммы потоков управления соответствуют компонентам диаграммы потоков данных. Пример диаграммы потоков управления представлен на рис.10. На диаграмме, выполненной в системе CASE.Аналитик, имеется один управляющий процесс (диспетчер) «Управлять обработкой данных». Поток управления «Инициализация заказа» поступает от внешней сущности «Заказчики». События этого потока обрабатываются диспетчером, который генерирует выходной поток управления «Старт принятия заказа». Под воздействием событий этого потока процесс «Принять заказ» переходит в активное состояние (стартует), принимает поток данных «Заказы» и через некоторое время порождает событие «Заказ принят», обрабатываемое диспетчером. В свою очередь, диспетчер после обработки генерирует событие «Старт оформления счета», под воздействием которого стартует процесс «Оформить счет», вводит поток данных «Данные заказов» из процесса «Принять заказ», подготавливает и отправляет счет заказчику. При наступлении события «Запрос по заказам», порожденного директором, диспетчер, обрабатывая это событие, генерирует выходное событие «Старт отчета». По этому событию запускается процесс «Сформировать отчет по заказам». Завершая формирование отчета, этот процесс переходит в приостановленное состояние и порождает событие «Отчет готов». Обрабатывая это событие, диспетчер генерирует событие «Вывести отчет», переводя процесс «Сформировать отчет по заказам» снова в активное состояние для вывода результата директору. Эту логику генерации и поглощения событий можно воспроизвести программой имитационного моделирования в реальном времени, если известно время, затрачиваемое

каждым процессом на генерацию и поглощение событий. Нарастивая интенсивности потоков «Инициализация заказов» и «Запрос по заказам», можно узнать, при каких предельных интенсивностях система перестает функционировать нормально и какой процесс является узким местом, ограничивая пропускную способность системы.

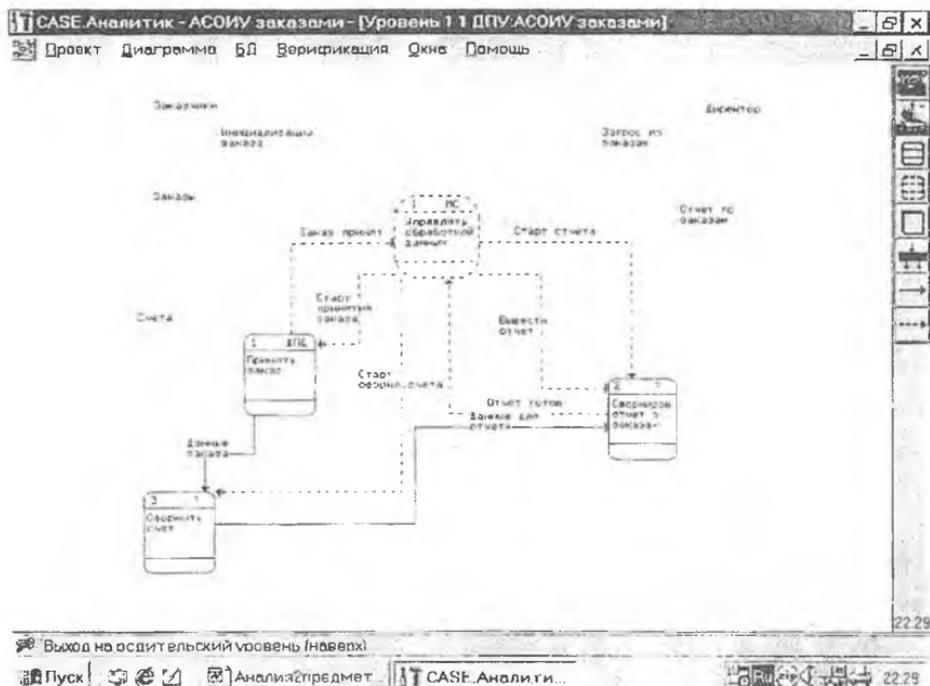


Рис. 10. Пример диаграммы потоков управления

Формальное описание логики процессов управления поясняется в разделе 3.

Указание. При работе с редактором диаграмм потоков управления в системе CASE Аналитик следует учитывать, что палитра объектов не содержит кнопку «Накопитель», т.е. считается, что накопители содержатся внутри процессов и их нужно рассматривать на ДПД следующих уровней детализации. Кроме того, управляющие процессы могут детализироваться только описанием логики (миниспецификацией), что позволяет строить иерархию управлений только детализируя подсистемы контекстными диаграммами. Только подсистема может раскрываться своей единственной диаграммой потоков управления, процесс и диаграмма потоков данных не могут содержать в себе управляющих процессов. Это ограничивает функциональные возможности CASE-системы, а в учебной 38

версии не позволяет эффективно использовать диаграммы потоков управления при малом числе уровней детализации (не более 3) . В этом случае можно отказаться от построения контекстной диаграммы общего вида и начинать проект с контекстной диаграммы системы, подразделенной на подсистемы.

2.3. Концепция построения системы

Системный структурный анализ предполагает построение двух моделей системы:

- существующей системы (AS-IS, как есть);
- новой системы (TO-BE, как должно быть).

Модель AS-IS строится не только на основе анализа должностных инструкций, приказов, официальных положений, знаний и представлений руководителей организации и подразделений. Важно изучить реально сложившуюся ситуацию с исполнением всех формальных предписаний, выявить бесполезные, плохо управляемые и дублирующиеся потоки данных и процессы обработки данных, неэффективные места документооборота (документ либо не нужен, либо не поступает в нужное место). После анализа модель AS-IS преобразуется в модель TO-BE, на базе которой и создается концепция построения системы.

Концепция построения системы включает следующие документы:

- описание предметной области;
- контекстная диаграмма (одна или несколько);
- пояснительная записка к контекстной диаграмме (диаграммам);
- диаграмма потоков управления (одна или несколько, только для систем реального времени);
- пояснительная записка к диаграмме (диаграммам) потоков управления (только для систем реального времени);
- диаграмма потоков данных (одна или несколько, не более 2 уровней детализации);
- пояснительная записка к диаграмме (диаграммам) потоков данных.

Описание предметной области представляет собой набор предложений естественного языка, в котором имена существительные задают типы объектов, а глаголы – отношения (связи) между объектами и виды действий с объектами. Отдельно выделяется раздел информационных запросов с указанием объекта, от которого поступает запрос о необходимости формирования твердой копии отчета

(документа). Описание должно быть кратким, соответствовать диаграммам верхних уровней детализации, но достаточно полным для последующей детализации проекта. Следует избегать неоднозначных формулировок и повторов, не включать в описание объекты и действия, не относящиеся к проектируемой системе. В описании уместны ссылки на официальные документы, регламентирующие деятельность объектов, в том числе по формированию информационных запросов. Полезно использовать одни и те же глаголы для обозначения определенных семантических отношений [10-12]. Например, для обозначения *обобщения* IS-A используется глагол «является»: «легковой автомобиль является автомобилем». Для обозначения *агрегации* PART-OF чаще всего используются глаголы «состоит из» или «характеризуется»: «автомобиль характеризуется госномером, маркой, годом выпуска, названием завода-изготовителя и т.д.». Количественные характеристики и обязательность связей между объектами задаются словами: «один», «каждый», «любой», «несколько», «некоторые», «все».

Пример описания предметной области приведен ниже.

Описание предметной области : ОБУЧЕНИЕ В ВУЗЕ.

ВУЗ состоит из нескольких факультетов, каждый факультет из нескольких кафедр (списки кафедр факультета не пересекаются), каждая кафедра – из преподавателей (списки преподавателей кафедр не пересекаются). Каждый преподаватель характеризуется ФИО преподавателя, должностью, ученой степенью и проводит обучение по одной или нескольким дисциплинам одной или нескольких учебных групп студентов. Каждая группа состоит из студентов и характеризуется уникальным номером, названием специальности, фамилией, именем и отчеством (ФИО) старосты, признаком обучения(дневное, вечернее и заочное) и входит в состав только одного из факультетов. Каждый студент характеризуется уникальным номером зачетной книжки, ФИО студента, адресом проживания и зачисляется в одну из групп. Каждая специальность закреплена за одной из кафедр факультета. Каждая кафедра характеризуется уникальным названием кафедры, ФИО заведующего кафедрой. Каждый заведующий кафедрой является преподавателем. Факультеты территориально размещены в разных корпусах института. Все данные хранятся централизованно на сервере вуза. Справочные данные по факультетам, кафедрам и специальностям обновляются секретарем ректора. Данные по преподавателям и студентам

40

обновляются секретарями соответствующих факультетов. Все данные обновляются по мере их поступления или изменения в случайные моменты времени.

Информационные запросы:

- *Ректора ВУЗА* – выдать на экран и напечатать состав всех преподавателей с указанием ФИО преподавателя, факультета, кафедры, на которой они работают, должности, ученой степени, названия дисциплины, по которой они обучают группу студентов с заданным номером;
- *Декана факультета* – выдать на экран и напечатать список всех студентов, обучающихся по заданной специальности, с группировкой по группам, с упорядочением групп по номерам групп, а внутри групп – по возрастанию ФИО студентов с указанием всех характеристик по каждому студенту. Для каждой группы указать ФИО старосты и вид обучения.

Пояснительные записки к диаграммам должны содержать краткое описание диаграмм, дополнительные характеристики объектов диаграмм (см. разделы 2.3-2.6).

Указание. В системе CASE Аналитик дополнительные сведения могут быть распечатаны в виде спецификации объектов, включая комментарий, для чего во всех диалоговых окнах ввода и редактирования данных об объектах имеется кнопка «Печать».

3. Детальная разработка логического проекта и базы данных системы

3.1. Состав логического проекта

Полностью проработанный логический проект АСОИУ включает:

- описание предметной области;
- контекстная диаграмма (одна или несколько) с пояснительным текстом;

- диаграмма потоков управления (одна или несколько, только для систем реального времени) с пояснительным текстом;
- диаграмма потоков данных (одна или несколько) с пояснительным текстом;
- график документооборота с пояснительным текстом;
- пояснительная записка по детальной разработке базы данных;
- ER-модель хранимых данных;
- SHM-модель хранимых данных;
- структурограммы базы данных;
- структурограммы потоков данных;
- миниспецификации (описания логики процессов).

Часть этих документов является расширенной и дополненной версией документов, образующих концепцию построения системы. Остальные документы формируются в процессе дальнейшего анализа и детализации функционально-логической модели системы. Методика детализации и последующих шагов проектирования изложена в разделах 3.2 – 3.12.

3.2. Анализ информационных запросов пользователей. Отчеты системы

Классификация информационных запросов (асинхронные и периодические) и отчетов (экранные и документальные), приведенная в разделе 2.4, далеко не полная. Важным является сложность запросов, определяющая объем поиска и существенно влияющая на время подготовки отчетов.

Сложность запросов определяется затратами времени и ресурсов, необходимыми АСОИУ для подготовки и вывода информационного отчета или формирования выходного сигнала по данному запросу.

Чем они больше, тем сложнее запрос. На сложность запросов влияют следующие факторы:

- время появления запроса (асинхронные и периодические запросы);
- количество переменных, таблиц и записей, которые нужно обрабатывать при подготовке и выводе отчета или формировании выходного сигнала;

- количество операций доступа к данным и обработки данных, которые необходимо выполнить для подготовки и вывода отчета или формирования выходного сигнала;
- требования по быстродействию (времени реакции системы) к формированию отчета или выходного сигнала;
- требования к *актуальности данных*, т.е. к скорости обновления оперативной информации, достаточной для формирования отчетов и выходных сигналов с требуемым качеством.

Асинхронные запросы всегда сложнее периодических при прочих равных условиях так как время их появления заранее непредсказуемо и быстродействие системы и степень актуальности оперативных данных могут оказаться недостаточными для качественного формирования отчетов и выходных сигналов. Пример асинхронного запроса руководителя: «Каждый раз, когда размер запасов становится меньше определенного безопасного уровня, предоставлять мне отчет о текущем состоянии выполнения заказов». Здесь событие «Размер запасов меньше безопасного уровня» может наступить в любой случайный момент времени и требует немедленной реакции системы.

Количество обрабатываемых таблиц, полей, записей, переменных в запросе, а также количество операций доступа к данным и обработки характеризует *логическую сложность запроса*.

Пусть имеются некоторые объекты (сущности), характеризующиеся определенным набором характеристик (атрибутов). Предполагается, что среди этих атрибутов один является уникальным идентификатором объекта (ключевым атрибутом). Например, процессор с серийным номером 115263 (идентификатор) характеризуется названием (Pentium III), разрядностью (32), тактовой частотой (800 МГц), частотой шины (100 МГц), типом разъема (Socket-462). При использовании реляционной модели данных сведения о процессорах можно хранить в отдельной таблице «Процессоры». Гэйн и Сарсон предлагают ввести для запросов по одному объекту 6 уровней логической сложности (типы запросов 1... 6, в порядке возрастания сложности).

Тип 1. Для данного объекта (O) найти значение определенного атрибута(A).

$$A(O)=?$$

Например: «Какова тактовая частота процессора номер 115263?»

Тип 2. Известно ограничение на значение определенного атрибута. Найти все объекты, удовлетворяющие этому условию.

$$A(?)\{=, \#, >, <\}V,$$

где V – некоторая константа.

Например: «Какие процессоры имеют тактовую частоту 800 МГц и больше?»

Тип 3. Для данного объекта известны ограничения по значениям атрибутов. Найти все атрибуты, удовлетворяющие этому значению.

$$?(O)\{=, \#, >, <\}V,$$

где V – некоторая константа.

Например: «Какие характеристики процессора 115263 имеют значение 100МГц и больше?».

Этот тип запроса характерен для объектов, у которых большое число атрибутов имеет значения, одинаковые по размерности. Например, несколько атрибутов описывают одно и то же свойство в разные моменты времени: зарплата сотрудника по месяцам в течение года, температура воздуха по дням прошедшей недели и т.п.

Тип 4. Аналогичен типу 1, но запрос идет по значениям всех атрибутов.

Например: «Каковы характеристики процессора номер 115263?».

Тип 5. Найти значения определенного атрибута для всех указанных объектов.

$$A(O\{=, \#, >, <\}V)=?$$

Например: «Найти тактовую частоту всех процессоров, номера которых лежат в диапазоне от 100 000 до 300 000».

Тип 6. Известны ограничения по значениям атрибутов. Найти все объекты и все атрибуты, удовлетворяющие поставленным условиям.

$$?\{=, \#, >, <\}V (?)$$

Например: «Найти все процессоры, для которых характеристики имеют значения 100МГц и выше, и указать, какие именно характеристики».

Запросы этого типа представляют собой более сложный вариант запроса типа 3.

В процессе структурного системного анализа необходимо выявить необходимость получения отчетов на запросы каждого из типов и оценить логическую сложность запросов (предварительно). Запросы по нескольким видам объектов, связанных между собой (нескольким таблицам), также

классифицируются по этой схеме, их уровень сложности тем выше, чем больше количество обрабатываемых связанных таблиц.

Уровень логической сложности запросов, связанных с обработкой, может быть оценен по сложности алгоритма обработки.

Чем более жесткими являются требования по быстродействию и актуальности данных, тем сложнее запрос. Эти требования могут быть удовлетворены за счет выбора метода доступа и формирования отчета, также за счет повышения быстродействия комплекса технических средств. Методика оценки времени реакции системы для различных методов доступа приведена в разделе 4.4.

3.3. Анализ документооборота. График документооборота

Дальнейшая детализация диаграмм потоков данных требует внимательного изучения документооборота существующей системы, определения в нем критических мест (лишних и необрабатываемых документов, ненужных пересылок и адресаций документов, избыточных по информации и неполных документов). Документы являются носителями информации и имеют логически определенную структуру данных, поэтому потоки документов на диаграммах могут быть представлены потоками данных, в которых в качестве элементов данных могут появляться заголовки всего документа и отдельных разделов, подписи, даты и т.д.

Большинство документов носят официальный юридический характер и подлежат исполнению тем лицом, кому они адресованы. Внедрение АСОИУ обычно имеет целью уменьшить количество бумажных документов, так как практически вся информация, содержащаяся в документах, и даже полные образы (бланки) этих документов могут быть сохранены на машинных носителях и переданы исполнителю в виде экранных форм. Однако средства защиты такой информации пока еще недостаточно совершенны, чтобы перейти на безбумажную технологию. Кроме того, человеку-исполнителю зачастую неудобно работать только с экранной формой представления информации при существующих технических средствах как в силу эргономических, так и в силу юридических соображений. Поэтому наиболее важные распорядительные и отчетные документы обычно поступают в систему и формируются в ней в виде твердой копии на бумажном носителе (вывод на печать).

Все документы делятся на *входные* (по отношению к системе или данному процессу обработки) и *выходные* (результатирующие). Из данных, содержащихся в

этих документах, соответственно формируются входные и выходные потоки данных. Кроме того, документы могут быть *внутренними* (действуют только внутри системы, подсистемы или процесса) и *внешними* (действуют и формируются вне указанных объектов).

При функционировании системы каждый документ, имеющий юридическую силу, может существовать в виде *оригинала* (одного или нескольких экземпляров) – с полным набором личных подписей авторов документа и *копий* – повтора данных документа с отсутствием части или всех подписей авторов документа. Как оригиналы, так и копии имеют свои пути обработки – маршруты. *Маршрут документа* – это перечень узлов (пунктов), последовательно проходимых документом при его обработке вплоть до его уничтожения или передачи в архив на постоянное хранение. Каждый экземпляр может иметь свой маршрут, маршрут может содержать циклически повторяющиеся ветви (циклы с возвратами на какие-либо пункты, итерации), причем количество проходов документа по циклу заранее может быть не известно. За период своего существования документ может создаваться (внутренний) или приниматься (внешний), проходить операции копирования (размножения), контроля, обработки (исполнения) и передачи в архив (или уничтожения).

Схема документооборота должна отражать маршруты прохождения экземпляров документа (в том числе электронных, на машинных носителях) и содержать ограничения по времени выполнения и качеству выполнения основных операций с документом. Во многих случаях эта схема содержит регламент и дисциплину работы с документами и должна быть утверждена руководством организации как самостоятельный документ.

Схема документооборота проектируемой системы позволяет уточнить регламент работы системы, состав и порядок обработки входных потоков, порядок получения и использования отчетов (выходных потоков) при решении задач и выполнении функций. Рекомендуется составлять ее в виде таблиц:

- Создание документов;
- Обработка документов;
- Передача документов в архив

«Создание документа» имеет смысл только для внутренних и исходящих документов и включает в себя: код документа, вид документа, название документа, количество экземпляров, кто отвечает за оформление, кто отвечает

за содержание, порядок и сроки подготовки, визирования и утверждения (окончательной подписи).

«Обработка документа» включает: код документа, вид документа, номер экземпляра, вид экземпляра, номер пункта маршрута обработки, название пункта, название стадии, кто выполняет, порядок и сроки обработки.

«Передача документа в архив» включает: код документа, вид документа, номер экземпляра, кто передает в архив, порядок и сроки передачи в архив, период хранения в архиве.

В обработку документа включаются операции регистрации вновь созданных и поступивших извне документов, отправки (выдачи) документа внешнему приемнику информации. В этом случае документ и соответствующий поток данных является выходным (*исходящим*).

В качестве примера оформления графика документооборота ниже приведены фрагменты таблиц для документооборота в отделении регистрации, учета и анализа инфекционных заболеваний городской службы санэпиднадзора.

Создание документов

Таблица 1

Код	Вид	Название	Кол. экз.	Отв. за оформление	Отв. за содержание	Утв.	Порядок и сроки
RA_Std58	Выходной	Экстренное извещение (форма 58)	12	Оператор	Врач ЛПУ	Зав. отд.	Немедленно по получении телеф. сообщения
RD_Des03	Внутренний отчет	Сведения о проведенных дезинфекциях	2	Врач-статистик	Врач-статистик	Зав. отд.	Еженедельно, итоговый за месяц, квартал, полугодие, год

Обработка документов

Код	Вид	Название	Но- мер экз.	Вид экз.	Отв. за обработ- ку, пункт	Ста дия	Порядок и сроки
RA_Std58	Выходной	Экстренное извещение (форма 58)	3-12	Элект- ронный	Оператор Терминал сети	Рас- сыл- ка вне- шн.	В зависи- мости от вида инфек- ции от 0 до 3 часов после создания электрон- ной почтой в районы и эпидотдел
RA_Std58	Выходной	Экстренное извещение (форма 58)	1	Бумаж- ный	Оператор Пункт приема	Рас- сыл- ка вну- тр.	Немедленно после создания зав. дез- центром
RD_Des03	Внутрен- ний отчет	Сведения о проведен- ных дезин- фекциях	1	Бумаж- ный	Врач- статис- тик	Рас- сыл- ка вну- тр.	В течение суток после создания зав.отделе- нием

Передача документов в архив

Таблица 3

Код	Вид	Название	Но- мер экз.	Вид экз.	Отв. за передачу	Порядок и сроки
RA_Std58	Выходной	Экстренное извещение (форма 58)	2	Бумаж- ный	Оператор	В течение рабочего дня
RD_Des03	Внутрен- ний отчет	Сведения о проведен- ных дезин- фекциях	2	Бумаж- ный	Врач- статис- тик	В течение рабочего дня

Из таблиц видно, что каждый документ может существовать в разных видах в нескольких экземплярах, причем каждый экземпляр может обрабатываться по-разному и иметь свой маршрут обработки. По графику документооборота можно оценить интенсивности входных и выходных потоков данных.

3.4. Структурограммы данных

Потоки данных, накопители данных и информационные каналы детализируются при помощи структурограмм описания данных.

Элементы данных – это далее логически неделимые порции информации (например, дата, фамилия, номер заказа, табельный номер и т.д.). Каждый элемент данных имеет имя и значение из некоторого множества значений.

Структуры данных состоят из элементов данных, из других структур данных или из их комбинаций и имеют свое имя. Набор конкретных значений элементов дает экземпляр структуры.

Считается, что поток данных, накопитель данных и информационный канал логически определяются и детализируются структурами данных с соответствующими именами.

Для каждого элемента данных в системе CASE. Аналитик указывается его вид:

- непрерывные данные (НД);
- дискретные данные (ДД);
- аналоговый сигнал (АС);
- дискретный сигнал (ДС).

Непрерывные данные – это данные, которые могут принимать множество значений, которые трудно перечислить и представить в виде ограниченного списка. Например, температура в диапазоне от 0 до 300 градусов Цельсия с точностью 0.1 градуса и т.д.

Дискретные данные могут быть перечислены и представлены списком, например, название отдела определенного предприятия, семейное положение сотрудника и т.д. Часто такой список хранится в базе данных и используется как справочник для ввода информации и подготовки отчетов.

Аналоговые и дискретные сигналы рассматриваются только для АСОИУ, включающих измерительную и датчиковую аппаратуру, и характеризуют данные автоматических измерений или определения состояний различных объектов.

Формально структура данных изображается структурограммой – блоком прямоугольной формы с заголовком. Допускается вложение структурных блоков друг в друга. Элемент данных также оформляется блоком, но, в отличие от структуры, не может содержать внутри себя других элементов и структур. В системе CASE Аналитик вложенные структуры оформляются отдельными структурограммами и запоминаются в базе данных проекта. В каждой структуре

допустимы итерации (повторяющиеся группы) элементов и структур, условные вхождения (необязательные элементы и структуры) и альтернативы.

Итерация означает неоднократное вхождение элементов и структур в указанном диапазоне.

Условное вхождение означает, что это необязательный компонент структуры.

Альтернатива означает, что в структуру может входить один из ниже перечисленных элементов .

При построении структурограммы рекомендуется указывать тип данных со следующими обозначениями:

- СД – структура данных;
- С10 – символьный (длина 10 символов);
- N6 - числовой (6 позиций, включая десятичную точку);
- I4 – целочисленный (4 разряда);
- D – дата;
- DT – дата-время;
- O –объект.

В реализации могут использоваться и другие типы данных.

Примеры оформления структурограмм представлены на рис. 11,12, 13.

В структуру данных «Платежи» включены две вложенных структуры данных: «Детали оплаты» и повторяющаяся группа «Счет» (итерация от 1 до 999 раз, предполагается, что платеж производится сразу по нескольким счетам). Число 999 отражает максимальное число счетов, которое может быть включено в платеж. Итерация верхнего уровня не показывается, она подразумевается по умолчанию. В извещении о платеже присутствует альтернатива: либо номер чека (при оплате чеком), либо детали оплаты (при оплате непосредственным снятием средств с банковского счета). Комментарий необязателен, поэтому он изображается условным вхождением (необязательным компонентом). На рис.12 изображен фрагмент структурограммы базы данных АСОИУ регистрации, учета и анализа инфекционных заболеваний, состоящий из одних структур данных. Каждая структура данных здесь должна быть раскрыта своей структурограммой. Построение структурограмм возможно средствами текстового редактора Microsoft Word (автофигуры или таблицы) , а также программой работы с электронными таблицами Microsoft Excel. Обе этих программы входят в состав стандартной конфигурации пакета Microsoft Office.

ПОТОК ДАННЫХ: Платежи	
N6	Номер извещения о платеже
D	Дата извещения о платеже
C50	Поставщик
C100	Адрес поставщика
АЛЬТЕРНАТИВА	
N6	Номер чека
СД	Детали оплаты
C100	Банк
C30	Номер счета поставщика
ВХОДИТ ОТ 1 ДО 999 РАЗ	
СД	Счет
N6	Номер счета
D	Дата счета
N8	Сумма счета
УСЛОВНОЕ ВХОЖДЕНИЕ	
C1500	Комментарий
N10	Сальдо платежа

Рис. 11. Пример структурограммы данных «Платежи»

НАКОПИТЕЛЬ ДАННЫХ БД2: База данных ГорЦГСЭН

- СД Экстренные извещения
- СД Типы экстренных извещений
- СД Данные ЛПУ по заболеваемости
- СД Вид обработки экстренных извещений
- СД Журнал обработки экстренных извещений
- СД Статистика по территориям
- СД Территории
- СД Улицы
- СД Группы организаций
- СД Организации
- СД Род занятий
- СД Группы заболеваний
- СД Заболевания/группы
- СД Заболевания

Рис.12. Фрагмент структурограммы базы данных АСОИУ

Указание. В системе CASE Аналитик структурограммы данных создаются в окне редактирования структурограмм, которое открывается из окна редактирования контекстной диаграммы или диаграммы потоков данных при детализации любого потока данных, накопителя данных или информационного канала инструментом «Вглубь» с изображением рыбки.

Окно редактирования структурограмм имеет свое меню, свою палитру объектов и более ограниченный набор кнопок инструментов. С их помощью можно передвигать и удалять объекты, сделать отмену предыдущей операции, ввести и отредактировать детальные сведения об объекте (инструмент «лупа») и перейти на следующий уровень детализации структуры данных (инструмент «погружение» - изображение рыбки). Работа с набором кнопок инструментов аналогична работе в окне редактирования диаграмм, но здесь объектами являются элементы данных и структуры данных, изображаемые строками таблицы. Вертикальный ряд кнопок справа представляет палитру объектов данных:

- непрерывный элемент (пиктограмма: условное изображение диапазона);
- дискретный элемент (три цветных точки);
- аналоговый сигнал (изображение синусоиды);
- дискретный сигнал (импульсы прямоугольной формы);
- структура данных (квадрат с горизонтальными линиями);
- альтернатива (прямоугольники со знаком вопроса);
- условное вхождение (пунктирный прямоугольник);
- итерация (ряд квадратов, наложенных друг на друга).

Для помещения выбранного объекта на структурограмму нужно щелкнуть кнопку, соответствующую объекту, левой кнопкой мыши. Затем переместить курсор мыши в рабочее поле и снова щелкнуть левой кнопкой мыши. Объекты добавляются последовательно друг к другу. Если необходимо поместить объект перед каким-либо объектом, уже имеющимся на диаграмме, нужно щелкнуть непосредственно в поле этого объекта.

Добавление альтернатив, условных вхождений и итераций осуществляется в два шага:

- 1) поместить в нужное место объект;
- 2) заполнить объект.

Заполнение альтернатив, условных вхождений и итераций элементами и структурами выполняется щелчком *правой* кнопки мыши (после выбора элемента) в поле заголовка альтернативы, условного вхождения и итерации соответственно. При этом объект добавляется в конец. Если необходимо поместить объект перед каким-либо объектом, уже имеющимся в альтернативе (условном вхождении, итерации), нужно щелкнуть правой кнопкой непосредственно в поле этого объекта.

При добавлении элемента или структуры данных на экране появляется диалоговый блок спецификации данных, который служит для ввода информации о новом объекте данных.

Для всех объектов единственным обязательным полем ввода является «Имя» (до 36 символов). Все остальные поля являются необязательными. Однако поле «Аннотация» (до 255 символов) желательно заполнять, т.к. зачастую смысл объекта данных остается непонятным или неполностью определенным. Распечатка спецификации осуществляется кнопкой «Печать»

Для сигналов указывается тип датчика, код сигнала и другие параметры.

Система CASE.Аналитик допускает только одноуровневое изображение итераций, альтернатив и условных вхождений. При необходимости вложений этих структур друг в друга их надо выделять в отдельные структуры с собственными именами.

Не вводите в структурограммы новые элементы данных и структуры данных с одним и тем же наименованием!

В этом случае перед вводом выберите пункт меню «Инструменты/Вводить из базы данных» и выберите нужный элемент из предложенного списка. При вводе нового элемента или структуры этот флаг надо сбросить.

В системе CASE.Аналитик все созданные структуры данных хранятся в базе данных проекта и образуют словари системы. Доступ к базе данных проекта возможен из окон редактирования диаграмм и структурограмм через меню «База данных».

На рис.13 представлена структурограмма потока данных «Платежи» (та же, что и на рис. 11), выполненная в системе CASE.Аналитик. Элементы данных «Поставщик» и «Адрес поставщика» считаются дискретными, так как предполагается, что их можно выбрать из списка ограниченной длины. Структуры данных «Детали оплаты» и «Счет» раскрываются отдельными структурограммами.

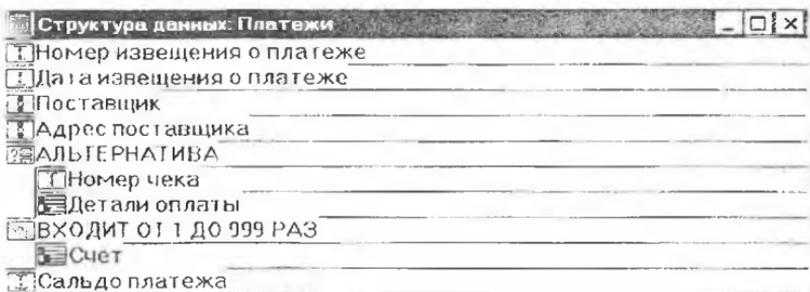


Рис. 13. Структурограмма «Платежи» в системе CASE.Аналитик

3.5. Разработка структуры базы данных. Модель «сущность-связь» (ER-модель)

Проектирование базы данных является одной из важнейших составных частей процесса проектирования АСОИУ. Известно, что база данных, рассматриваемая как сложная система, проектируется с использованием тех же принципов, что и АСОИУ в целом (см. раздел 2.1). В соответствии с принципом нисходящей поэтапной разработки модель хранимых данных рассматривается обычно на трех уровнях детализации: концептуальном, логическом и физическом.

Одним из первых вопросов, которые возникают на стадиях формирования требований и разработки концепции будущей АСОИУ, является вопрос о том, какая информация должна храниться и перерабатываться в системе, какие объекты предметной области и их характеристики представляют для потенциальных пользователей системы информационный интерес. Ответ на этот вопрос во многом определяет структуру системы.

Проектирование информационного обеспечения лучше всего начинать с построения модели предметной области, которую необходимо в течение достаточно длительного периода времени хранить в базе данных с тем, чтобы система выполняла в процессе функционирования все свои функции.

При этом сам процесс проектирования рекомендуется подразделить на 3 этапа, соответствующие 3 уровням представления модели данных.

- 1) концептуальное проектирование;
- 2) логическое проектирование;
- 3) проектирование на физическом (машинном) уровне.

Концептуальная информационная модель – модель данных о предметной области, в которой будет функционировать проектируемая АСОИУ, выраженная в терминах этой предметной области и не учитывающая способ логического хранения данных в памяти ЭВМ. Основные требования к концептуальной модели – полнота, наглядность, использование терминологии, понятной пользователю – специалисту в рассматриваемой предметной области. Концептуальная модель максимально насыщена семантикой (смысловым содержанием).

Логическая информационная модель – модель данных, в которой учитывается способ логического хранения данных в памяти ЭВМ. Известны три классических логических модели хранения данных, рассматриваемых в курсе “Организация баз данных”: иерархическая, сетевая и реляционная. В последнее время к ним прибавилась более сложная объектно-ориентированная модель. Однако, учитывая, что большинство современных СУБД поддерживает реляционную модель данных, будем в последующем изложении ориентироваться на нее. Таким образом, логическая модель будет состоять из совокупности взаимосвязанных таблиц, в которых хранятся конкретные данные о предметной области. Семантика предметной области еще видна, однако названия таблиц и полей таблиц для специалиста в предметной области уже более трудны для понимания.

Физическая модель данных – отражает способ машинного хранения информации на долговременных запоминающих устройствах. Эта модель при использовании стандартного системного программного обеспечения обычно скрыта от пользователя и представляет интерес только для эксплуатационного персонала при настройке системы и анализе ошибочных ситуаций. Семантика предметной области на этом уровне непонятна для обычного пользователя – специалиста в предметной области.

Описание предметной области, входящее в состав концепции построения системы, позволяет сформировать концептуальную модель хранимых данных. Существуют несколько подходов к представлению концептуальной модели. Наиболее распространены диаграммные представления Чена (модель “сущность-связь”) и Смита (иерархическая семантическая модель).

Модель «сущность-связь» (Entity-Relationship, *ER-модель*) предложена Петером Пин-Шен Ченом в 1976 г. [10-13]. Основными понятиями этой модели являются сущность, связь и атрибут.

Сущность – это реальный или абстрактный объект. информация о котором представляет интерес и должна храниться в базе данных. Имя сущности является именем класса или типа объектов. Все экземпляры сущности должны быть отличимы друг от друга.

Связь – некоторое отношение между сущностями, позволяющее сопоставлять между собой пары определенных экземпляров двух сущностей (бинарные связи) по принципу их взаимного соответствия. Связь может быть *обязательной*, когда все экземпляры обеих сущностей участвуют в ней, и *необязательной* (в противоположном случае). Кроме того, связи различают по количеству экземпляров одной сущности, сопоставляемых с одним и тем же экземпляром другой сущности : *один к одному* (1:1) и *один ко многим* (1:M или M:1). Если связь со стороны каждой сущности один ко многим, то говорят, что тип такой связи - *многие ко многим* (M:N).

Атрибут – любая характеристика сущности, связанная только с данной сущностью, которая может принимать конкретные значения из множества значений определенного типа. Набор значений всех характеристик определяет экземпляр сущности. *Идентификатором (ключом) сущности* называется атрибут или набор атрибутов, однозначно определяющих экземпляр сущности (свойство уникальности).

ER-модель изображается диаграммой и широко используется в различных CASE-системах и методологиях. Существуют различные системы обозначений компонентов ER-модели. Предлагается следующая простая нотация:

- сущности обозначаются прямоугольником, содержащим имя сущности;
- связи изображаются линиями, имеют имена; в месте соединения с сущностью ставится соответствующий тип связи (1- единичная, M или N - множественная), необязательность связи обозначается маленьким кружочком на связи;
- атрибуты записываются в виде комментария к сущности, первыми указываются и выделяются ключевые атрибуты.

Пример оформления ER-модели для предметной области, описанной в концепции (раздел 2 7), представлен на рис. 14.



Рис. 14. ER – модель предметной области «Обучение в ВУЗе»

На диаграмме названия сущностей и ключевых атрибутов выделены жирным шрифтом. Типы связей следуют из описания предметной области. Например, только одна кафедра может быть профилирующей (выпускающей) группу по специальности. Однако не все кафедры являются профилирующими, что помечается необязательностью связи «Профилирует» со стороны кафедры. При наличии связи не следует дублировать на диаграмме общие атрибуты, их следует присоединить к одной из связанных сущностей (например, атрибут «Специальность» включен в сущность «Кафедра»). Предмет, по которому преподаватель обучает группу, включен в сущность «Группа», поэтому более точная семантика сущности «Группа» может быть сформулирована как «Группа, обучаемая по определенному предмету данным преподавателем».

Детализация ER-модели осуществляется путем перехода к логической модели представления данных (реляционной, иерархической, сетевой или объектной). Переход к реляционной модели данных рассмотрен в разделе 3.7.

3.6. Семантическая иерархическая модель (SHM-модель)

Другим видом модели хранимых данных на концептуальном уровне является семантическая иерархическая модель, предложенная Смитом (Semantic Hierarchical Model, SHM - модель). Она более тонко отображает семантику сущностей и отношений (связей) между сущностями, чем ER-модель, но и несколько более сложна при построении. Модель использует концепцию иерархии семантических сущностей, важнейшими из которых являются агрегации и обобщения. Эти сущности являются результатом абстрактных структурообразующих операций, свойственных человеку при анализе некоторой предметной области.

Агрегация – объединение разнородных сущностей в одно именованное понятие верхнего уровня только по одному признаку – принадлежности к этому понятию как некоторой характеристики или составной части. Агрегация реализует отношение «часть-целое». Например, «номер зачетной книжки», «ФИО студента», «адрес проживания» объединяются в одно понятие «студент», так как являются характеристиками этого понятия. Понятия «двигатель», «кузов», «колесо», «система зажигания», «рулевое управление» объединяются в понятие «автомобиль», так как являются составными частями этого понятия, и т. д. Связь (отношение) такого типа называется *part_of* и выражается словами «состоит из», «характеризуется».

Обобщение – объединение сходных (имеющих аналогию) сущностей в одно именованное понятие более высокого уровня по одному или нескольким общим признакам, характеризующим все сущности данного понятия. Соответствует построению некоторого класса из частных примеров этого класса. Обобщение – это класс, в котором подклассы могут пересекаться по составляющим их объектам. Например, сущности «группа дневного обучения» и «группа вечернего обучения» обобщаются в понятие более высокого уровня «группа». Сущности «пассажирский самолет» и «транспортный самолет» – в понятие «самолет» и т.д. Связь (отношение) такого типа называется *is_a* и выражается словами «является», «является частным примером». Отличить обобщение от агрегации очень легко: достаточно выяснить, является ли понятие более низкого уровня частным примером понятия более высокого уровня. Если да, то это – обобщение. Обобщение реализует отношение «род-вид». Иногда рассматривают и другие семантические структурообразующие операции:

ассоциация, классификация и др. Однако практически все они могут быть интерпретированы агрегациями и обобщениями.

Сущности нижнего уровня могут приобретать конкретные значения из некоторого множества значений и не иметь следующих уровней (*терминальные сущности*), либо ,в свою очередь, являться агрегациями и/или обобщениями и иметь следующий уровень. Одна и та же сущность может одновременно быть и агрегацией, и обобщением. Набор сущностей нижнего уровня, однозначно определяющий экземпляр агрегации, называется *идентификатором (ключом) агрегации*.

SHM-модель изображается диаграммой и наряду с ER-моделью широко используется в различных CASE-системах и методологиях. Существуют различные системы обозначений компонентов SHM-модели. Предлагается следующая простая нотация:

- сущности обозначаются прямоугольником, содержащим имя сущности, располагаются и выравниваются по уровням иерархии;
- связи между уровнями изображаются линиями с разветвлениями и с пометками «part_of» и «is_a»;
- атрибуты записываются в виде комментария к сущности, первыми указываются и выделяются ключевые атрибуты.

Пример оформления SHM-модели для предметной области, описанной в концепции (раздел 2.7), представлен на рис 15.

Следует отметить, что при построении SHM-модели большую роль играет тип связи между сущностями (1:1, 1:M, M:N). Общим правилом является то, что M-связные сущности располагаются на верхнем уровне, односвязные – внизу; при типе связи 1:1 допустимы оба варианта расположения. При типе связи M:N следует заменить ее двумя связями типов M:1 и N:1, для чего вводится на верхнем уровне дополнительная агрегация связи (например, «Обучение», см. рис.15). На диаграмме нетерминальные (составные) сущности выделены жирным шрифтом, ключевые сущности – курсивом. Все составные сущности раскрыты как агрегации, однако можно сущность «Преподаватель» представить дополнительно как обобщение частных примеров «Обычный преподаватель» и «Завкафедрой», если возникнет необходимость в соответствующих информационных запросах. В этом случае от сущности «Преподаватель» нужно нарисовать отдельную группу связей с пометкой «is_a».



Рис.15. SHM – модель предметной области «Обучение в вузе»

3.7. Переход к реляционной модели хранения данных

В связи с тем что большинство коммерческих СУБД поддерживает реляционную модель данных, дальнейшее уточнение диаграмм и структурограмм данных ведется путем их преобразования в простые табличные формы. Для этого прежде всего необходимо исключить из структурограмм накопителей данных итерации, альтернативы и условные вхождения. Каждая итерация представляется отдельным отношением (таблицей), связанной с основным отношением. Связь устанавливается путем включения ключа основного отношения в дополнительное. Альтернативы задаются отдельными полями таблицы, для ввода пустых (отсутствующих) значений вводится тот или иной способ кодировки, например, «0», «F» или «Отсутствует». Условные вхождения обрабатываются аналогично.

Большую помощь при переходе к реляционной модели данных могут оказать правильно и полно составленные ER или SHM- модели, построенные на

концептуальном уровне. Они же могут помочь при нормализации таблиц реляционной модели [8], [10-13]. Поэтому в полный логический проект АСОИУ необходимо включать обе эти модели, а также результаты нормализации.

Формально переход от ER-модели к реляционной модели можно сделать по следующим правилам, условно называемыми *правилами Джексона*:

- две связанные сущности при типе связи 1:1 и обязательности связи с обеих сторон объединяются в одно отношение, ключ которого должен являться ключом одной из сущностей;
- в предыдущем случае при необязательности связи с одной стороны строятся 2 отношения (по одному на сущность), причем для связи ключ необязательной сущности включается в отношение для обязательной сущности (это позволяет избежать пустых полей в поле связи);
- в предыдущем случае при необязательности связи с обеих сторон строятся 3 отношения : по одному на сущность и отношение связи, в которое включаются ключи обеих сущностей;
- для двух связанных сущностей при типе связи 1:M или M:1 и обязательности связей с обеих сторон строятся 2 отношения (по одному на сущность), причем для связи ключ односвязной сущности включается в отношение для многосвязной сущности (очевидно, это позволит хранить меньше избыточных записей в таблицах, чем в противоположном случае);
- Во всех остальных случаях строятся 3 отношения: по одному на сущность и отношение связи, в которое включаются ключи обеих сущностей.

Конечно, в результате такого формального перехода часть отношений может оказаться лишней, а также не гарантируется, что отношения представлены в нормальных формах выше первой (1НФ). Для нормализации отношений применяются процедуры, описанные в [10-13]. Отметим, что для приведения ко второй нормальной форме (2НФ) из отношений должны быть удалены все частичные функциональные зависимости полей, не входящих в составные ключи, от самих ключей. Для приведения к третьей нормальной форме (3НФ) из отношений должны быть исключены транзитивные (несимметричные) функциональные зависимости для любых наборов из трех

полей или групп полей отношения. В результате нормализации количество таблиц может существенно возрасти, но уменьшается риск ошибочных добавлений или удалений записей в отношениях, нарушающих целостность базы данных. Практически обязательным является приведение отношений к 2НФ. Приведение к более высоким формам должно быть обосновано и подтверждено расчетами памяти и быстродействия АСОИУ (см. раздел 4.4).

Например, для диаграммы рис.14, следуя указанным правилам, получаем следующие отношения (таблицы):

- *Преподаватель* (ФИО_преподавателя: ключ, должность, ученая_степень, название_кафедры(поле связи));
- *Кафедра* (название_кафедры: ключ, факультет, ФИО завкафедрой);
- *Преподаватель_группа* (ФИО_преподавателя, номер_группы);
- *Группа* (номер_группы: ключ, ФИО_старосты, вид_обучения, предмет, специальность);
- *Специализация_группы* (номер_группы, название_кафедры);
- *Студент* (номер_зачетной_книжки: ключ, ФИО_студента, номер_группы(поле связи), адрес_проживания).

В связи с отсутствием составных ключей можно считать, что все эти отношения находятся в форме 2НФ. Отношение *Группа* не находится в 3НФ, так как функциональные зависимости номер_группы-ФИО_старосты-предмет не транзитивны (не симметричны в прямом и обратном направлении, от предмета ФИО_старосты не зависит). Для приведения к 3НФ необходимо разделить отношение *Группа* на два отношения: *Группа* (номер_группы, ФИО_старосты, вид обучения, специальность) и *Исучаемый_предмет* (номер_группы, предмет, ФИО_преподавателя). При этом отношение *Преподаватель_группа* становится излишним и может быть исключено из схемы базы данных.

Переход от SHM-модели к реляционной модели данных производится по следующим правилам:

- при движении снизу вверх по диаграмме заменить все агрегации, отношениями, включая в верхний уровень вместо нижележащих сущностей их ключи(идентификаторы);
- для обобщений вводить в агрегацию дополнительную сущность: признак обобщения;
- исключить избыточные и дублирующие отношения.

Для диаграммы рис.15, следуя указанным правилам, получаем следующие отношения (таблицы):

- *Кафедра* (название_кафедры: ключ, факультет, ФИО завкафедрой);
- *Преподаватель* (ФИО_преподавателя: ключ, должность, ученая_степень, название_кафедры(поле связи));
- *Группа* (номер_группы, название_кафедры, предмет: ключ, ФИО_старосты, вид_обучения, специальность);
- *Обучение_группы* (номер_группы, название_кафедры, предмет, ФИО_преподавателя);
- *Обучение_студента* (номер_зачетной_книжки, номер_группы, название_кафедры, предмет: ключ, ФИО_студента, адрес_проживания).

Мы видим, что схема отношений, построенных по SHM-модели, внешне отличается от схемы, построенной по ER-модели. Это связано с разной семантикой понятий «Группа» и «Студент» в моделях. В SHM-модели их следует понимать как «Специализация групп» и «Обучение студента» соответственно. Большое количество составных ключей требует анализа отношений на степень нормализации. Например, отношение «Группа» не находится в 2НФ в силу частичной зависимости позиций, не входящих в ключ, от ключа (ФИО_старосты однозначно определяется только номером группы и т.д.). Поэтому его можно разбить на два отношения: «Группа»(номер_группы, ФИО_старосты, вид_обучения, специальность) и «Специализация группы»(номер_группы, название_кафедры), связанных между собой по номеру группы. Аналогично «Обучение группы» может быть разбито на два отношения: «Изучаемый предмет»(номер_группы, ФИО_преподавателя, предмет) и «Сведения о преподавателе»(ФИО_преподавателя, название_кафедры). Последнее отношение является лишним, так как оно целиком содержится в отношении «Преподаватель». «Обучение студента» также не находится в 2НФ, для приведения разбиваем его на два отношения: «Студент»(номер_зачетной_книжки, ФИО_студента, адрес_проживания, номер_группы) и «Обучение студента»(номер_зачетной_книжки, название_кафедры, предмет). Так как студент учится в строго определенной группе, то отношение «Обучение студента» в этом варианте является лишним, так как полностью дублирует отношение «Изучаемый предмет». Все отношения представлены и в форме 3НФ в силу отсутствия транзитивных зависимостей.

Окончательно схему базы данных по SHM-модели представляем в виде 6 отношений: «Преподаватель», «Кафедра», «Студент», «Группа», «Изучаемый предмет» и «Специализация группы», что полностью соответствует схеме, построенной по ER-модели.

3.8. Детализация диаграмм потоков данных

После детальной проработки структуры базы данных следует прежде всего обновить состав накопителей. Так как реализация АСОИУ будет вестись на основе реляционной модели данных, то для более точной записи логики процессов необходимо, чтобы каждой таблице базы данных соответствовал накопитель той же структуры. Обычно в концепции АСОИУ база данных содержит небольшое число накопителей, включающих итерации, альтернативы и условные вхождения. В таблицах реляционной модели эти элементы отсутствуют, однако число накопителей заметно возрастает. Соответственно меняется структура потоков данных. Все эти изменения вносятся в диаграммы потоков данных, появляются уточненные варианты ДПД. Процессы уточненной ДПД могут быть детализованы либо при помощи диаграмм потоков данных, либо при помощи миниспецификаций (описаний логики процессов). Последние являются конечной вершиной иерархии диаграмм логического проекта и применяются для описания логики элементарных процессов. Решение о завершении детализации процесса принимает проектировщик с учетом следующих критериев:

- процесс имеет относительно небольшое количество входных и выходных потоков данных (до 3-4 каждого вида);
- преобразование данных процессом можно описать в виде последовательного алгоритма;
- процесс выполняет одну единственную логическую функцию преобразования входов в выходы;
- объем миниспецификации процесса небольшой – не более 2-3 страниц текста.

Рекомендуется переходить к детализации процессов после определения содержания всех потоков данных. Уточнение характеристик потоков данных включает в себя определение его временных характеристик – периода передачи

или интенсивности и дополнительных сведений в зависимости от типа элементов данных, входящих в состав структуры потока (см. раздел 3.4).

Указание. В учебной версии системы CASE. Аналитик при ограничениях на число уровней детализации (до 3) и количество однотипных компонентов диаграмм на каждом уровне (до 5) иногда приходится искусственно объединять некоторые накопители, потоки и процессы, чтобы удовлетворить этим условиям. В этом случае в учебных задачах рекомендуется максимально использовать третий (и последний) уровень детализации, представляя процессы через ДПД с целью максимального приближения процессов к элементарным, а накопители – к таблицам базы данных проектируемой системы.

3.9. Детализация диаграмм потоков управления для систем реального времени

Диаграммы потоков управления, которые являются детализацией диаграмм, описывающих подсистемы и процессы АСОИУ реального времени, далее сопровождаются описаниями логики управляющих процессов. Эти описания могут строиться двумя способами [4]):

- диаграммами переходов состояний;
- таблицами «событие-отклик».

Диаграмма переходов состояний создается для системы (подсистемы) и каждого процесса обработки данных, который может с точки зрения управления им находиться только в одном из трех состояний (рис. 9): пассивен, активен и приостановлен.

Введем понятие «состояние системы». *Состояние системы* в данный момент времени определяется множеством состояний, в которых находятся все процессы обработки данных системы. Очевидно, что максимальное число состояний, в которых может находиться система, равно 2^N , где N – число процессов. Однако далеко не все комбинации состояний являются допустимыми, поэтому число реальных состояний системы обычно намного меньше. Полная диаграмма переходов состояний должна отражать все возможные переходы системы под воздействием событий. События, вызывающие переход системы из состояния в состояние, подразделяются на внешние и внутренние. *Внешние события* происходят вне системы и порождаются внешними сущностями (например, изменение значения параметра управляемого объекта, нажатие клавиши «Пуск» на стенде инженером-испытателем и т. д.). *Внутренние события*

порождаются подсистемами, процессами обработки данных и управляющими процессами внутри системы в связи с изменениями их состояний.

Каждое событие может рассматриваться как реализация некоторых логических условий, например, «кнопка K1 нажата», «значение температуры превысило 65 градусов» и т. п. Таких условий, а соответственно и событий, влияющих на состояние системы, может быть очень много и все они должны быть снабжены идентификаторами, описаны и отражены в детализирующих описаниях.

Рекомендуется в кодах событий учитывать внешние и внутренние события, а также объекты, порождающие эти события. Например, V1_5ST25 –внешнее событие 1: «на стенде №5 параметр №25 (температура) превысил значение 65 градусов».

На диаграммах переходов состояний дуги помечаются событием, вызвавшим переход, а также перечнем процессов, изменяющих состояние, и событий, порождаемых в результате этого перехода. Дополнительно могут быть указаны время или характер наступления входного или выходного события (например, закон распределения для случайных событий). Находясь в некотором состоянии, система может перейти в другое состояние при возникновении событий, помечающих переходы, выходящие из этого состояния. При этом управляющий процесс или другие объекты порождают (генерируют) новые события (внешние и внутренние), указанные на диаграмме. Нотация для диаграмм переходов состояний представлена на рис. 16.

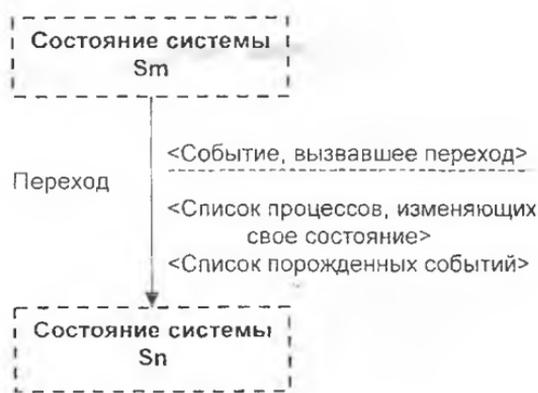


Рис. 16. Диаграмма переходов состояний

Другой формой описания логики управления является таблица «событие-отклик», формат которой представлен ниже.

Событие-отклик АСОИУ

Событие	Процессы, переходящие в состояние			Событие
	Активен	Пассивен	Приостановлен	

В таблице приводятся полный список событий, изменяющих внутреннее состояние системы с точки зрения запуска и останова процессов обработки информации, и список порожденных (генерируемых) событий в системе. Привязка ко времени или нумерация событий позволяет более точно задать алгоритм управления процессами в системе и произвести имитационное моделирование с целью предварительной оценки проектных решений

Рекомендуется использовать обе формы детализации диаграмм потоков управления как взаимно дополняющие друг друга.

Указание. В учебной версии системы CASE.Аналитик диаграммы переходов состояний и таблицы «событие-отклик» не поддерживаются. Их построение можно выполнить средствами редактора Microsoft Word.

3.10. Описание логики процессов. Миниспецификации

Описание логики дается для элементарных процессов в специальной структурной форме, достаточной для выполнения или программирования. Рекомендуемый объем описания – не более 3 страниц формата А4.

Часто описания логики процессов называют *миниспецификациями*, имея в виду, что они являются минимальными указаниями (заданиями) программисту по логике обработки данных, которую ему предстоит реализовать.

Для определения языка введем несколько понятий.

Блок. Блок изображается прямоугольником и имеет заголовок:

<Заголовок>

Блоки можно соединять, образуя последовательность блоков, или вкладывать друг в друга, образуя вложенные структуры.

Предложение. Это – предложение естественного языка, четко и недвусмысленно описывающее некоторое действие или условие. Если предложение описывает некоторое действие, то его нужно начинать со служебного слова **ВЫПОЛНИТЬ**.

Далее следует определение синтаксиса языка.

Описание внутренней логики процесса =

ЛОГИКА ПРОЦЕССА <имя процесса>

```
{<оператор>}
```

Здесь фигурные скобки {} означают возможность многократного повтора.

<оператор> =

```
<предложение>|
<условный оператор>|
<оператор выбора>|
<цикл>|
<оператор ввода\вывода>
```

Вертикальная черта | означает «или».

<условный оператор> =

```
ЕСЛИ <предложение>
  {<оператор>}
ИНАЧЕ
  {<оператор>}
```

Квадратные скобки [] означают, что конструкция внутри них не является обязательной и может отсутствовать. Условный оператор соответствует конструкции IF – THEN – ELSE в языках программирования.

<оператор выбора> =

```
В ЗАВИСИМОСТИ ОТ ЗНАЧЕНИЯ <предложение>
  { = <предложение>
    {<оператор>} }
[В ДРУГИХ СЛУЧАЯХ
  {<оператор>}]
```

Эта конструкция соответствует переключателю в языках программирования высокого уровня (оператор CASE). Альтернатив может быть несколько и каждый

раз выполняется только одна из них (а именно та, для которой условие « = < предложение > » выполнено).

<цикл> =

ПОВТОРЯТЬ ПОКА <предложение>
{<оператор>}

В данном случае перед выполнением оператора всякий раз проверяется условие продолжения цикла, заданное предложением. Если условие выполнено, то оператор выполняется; иначе следует выход из цикла. Эта конструкция соответствует конструкции цикла **DO WHILE** в языках программирования высокого уровня.

Допустима и другая конструкция цикла – с условием окончания:

ПОВТОРЯТЬ
{<оператор>}
ДО <предложение>

Оператор выполняется повторно до выполнения условия, заданного предложением. При выполнении условия следует выход из цикла. Эта конструкция соответствует конструкции цикла с условием окончания **REPEAT – UNTIL** в языках программирования высокого уровня.

Оператор ввода\вывода записывается либо как оператор ввода, либо как оператор вывода.

<оператор ввода> =

ВВЕСТИ <предложение>

<оператор вывода> =

ВЫВЕСТИ <предложение>

Перечисленных конструкций вполне достаточно для описания внутренней логики любого элементарного процесса обработки данных. Далее приводятся примеры описаний и рекомендации по записи логики процессов.

Последовательные операторы. Эта структура охватывает действие или любую последовательность действий, не имеющих повторов или разветвлений в зависимости от выполнения условий. Допустимо объединять в одном блоке ряд

действий, программная реализация которых понятна и не будет вызывать уточняющих вопросов со стороны программиста.

ЛОГИКА ПРОЦЕССА 1.5 Подготовить отчет о продажах

ВЫПОЛНИТЬ	Очистить экран, отключить вывод на экран предупреждающих сообщений и сообщений о результатах промежуточных операций, закрыть все накопители
ВЫПОЛНИТЬ	Открыть накопитель БД1 Продажи
ВЫПОЛНИТЬ	Создать на экране форму для ввода исходных данных к отчету с заголовками : «Введите данные к отчету о продажах», «Нач. дата:», «Кон. дата:»
ВВЕСТИ	Начальная дата, Конечная дата
ВЫПОЛНИТЬ	Очистить экран
ВЫПОЛНИТЬ	Создать на экране форму для вывода отчета с заголовками «Отчет о продажах с», «по», «Дата», «Покупатель», «Наименование товара», «Единица измерения», «Количество», «Сумма»
ВЫПОЛНИТЬ	Скопировать в промежуточную таблицу Продажи1 записи из БД1 по условию: (БД1.Дата>=Начальная дата) И (БД1.Дата<=Конечная дата)
ВЫПОЛНИТЬ	Закрывать накопитель БД1
ВЫВЕСТИ	Начальная дата, Конечная дата, данные таблицы Продажи1
ВЫПОЛНИТЬ	Закрывать и удалить промежуточную таблицу Продажи1 после просмотра отчета пользователем

Операторы выбора решения . Пример условного оператора с вложением условий приведен ниже.

ЕСЛИ Объем заказов > =100000 рублей в год
ЕСЛИ Заказчик является надежным плательщиком
ВЫПОЛНИТЬ Обслужить приоритетно
ИНАЧЕ
ЕСЛИ Заказчик наш клиент >= 10 лет
ВЫПОЛНИТЬ Обслужить приоритетно
ИНАЧЕ
ВЫПОЛНИТЬ Обслужить обычным образом

Здесь необходимым условием приоритетного обслуживания заказчика является большой объем ежегодных заказов, к которому добавляется либо условие надежности по оплате, либо условие продолжительного взаимодействия с заказчиком

Если имеется несколько вариантов условий типа альтернатив, представляющих взаимно исключающие случаи, то лучше использовать оператор выбора.

В ЗАВИСИМОСТИ ОТ ЗНАЧЕНИЯ Вид запроса
= Заказ
ВЫПОЛНИТЬ Добавить к продажам на сегодняшний день
= Возврат
ВЫПОЛНИТЬ Вычесть из продаж на сегодняшний день
= Оплата
ВЫПОЛНИТЬ Добавить к полученным наличным деньгам
= Рекламация
ВЫПОЛНИТЬ Передать в отдел рекламаций
В ДРУГИХ СЛУЧАЯХ
ВЫПОЛНИТЬ Обратиться к менеджеру

Здесь переменная Вид запроса может принимать в данный момент только одно из указанных значений.

Операторы циклов. Используются во всех случаях, когда действия повторяются до тех пор, пока некоторое условие истинно (цикл **ПОКА**), либо до достижения истинности некоторым условием (цикл **ДО**). Например, при обработке данных счета на продажу, содержащего несколько наименований товара, для получения общей суммы счета может использоваться следующая логика:

ПОВТОРЯТЬ ПОКА Не все строки счета учтены
ВЫПОЛНИТЬ Сумма строки = Количество * Цена единицы товара, Сумма счета = Сумма счета + Сумма строки

Тот же результат мы можем получить с помощью цикла **ДО**:

ПОВТОРЯТЬ
ВЫПОЛНИТЬ Сумма строки = Количество * Цена единицы товара, Сумма счета = Сумма счета + Сумма строки
ДО Все строки счета учтены

Хотя запись логики процессов не является программой для ЭВМ, однако она должна содержать все необходимые указания программисту для

составления кода программы. При этом следует выполнять следующие рекомендации.

Точная запись названий. Все имена должны быть уникальными и строго соответствовать именам объектов в диаграммах и структурограммах. Если вводится имя для новой промежуточной переменной, например, «Сумма строки», то это имя также должно быть уникальным и однозначно трактуемым во всех описаниях логики процессов. Элементы структур данных следует указывать операцией разыменования через точку: <имя структуры>.<имя элемента>. Например, Платежи.Дата или БД1.Дата.

Точная запись условий. Следует строго указывать границы диапазонов в условиях, используя знаки логических отношений или соответствующие словосочетания:

> означает «больше»;

> = означает «больше или равно»;

< = означает «меньше или равно»;

< означает «меньше»;

= означает «равно»;

означает «не равно».

При записи условий нужно стремиться к тому, чтобы условия были как можно проще и понятней, содержали не более 2-3 конъюнкций (И) или дизъюнкций (ИЛИ). Более сложные условия представляют вложенными конструкциями (см. пример в предыдущем пункте). При необходимости используются круглые скобки:

ЕСЛИ (Сумма счета > = 10000) И (Заказчик = «ООО ИНВЕСТ»).

Ограничение объема миниспецификации. Рекомендуемый объем описания логики одного процесса в виде миниспецификации - 1-3 страницы текста. В более сложных случаях процесс следует раскрыть на следующем уровне детализации диаграммой потоков данных, состоящей из более элементарных процессов

Указание. В системе CASE Аналитик для редактирования миниспецификаций используется текстовый редактор. Однако возможности этого редактора очень ограничены, в частности, он не позволяет рисовать блочные конструкции. Поэтому для оформления миниспецификаций лучше использовать текстовый редактор Microsoft Word или процессор электронных таблиц Microsoft Excel.

4. Техническое проектирование АСОИУ

4.1. Структура интегрированной АС предприятия

Понятие интегрированной АС было введено в разделе 1.1. Эффективность работы предприятия во многом определяется уровнем информатизации предприятия. Максимально эффективной АС является та, которая охватывает все взаимосвязанные процессы деятельности предприятия, обеспечивает управление внутри предприятия и всестороннюю информационную связь с внешней средой, то есть интегрированная АС. С внедрением правильно спроектированной интегрированной АС сокращается длительность цикла от разработки до освоения и реализации серийно выпускаемой продукции, повышается гибкость производства, способность его функционировать в быстро меняющихся условиях, экономятся трудовые и материальные ресурсы [14]. В связи с этим проектирование даже небольшой по масштабам АС, реализующей ограниченный набор функций и задач, должно проводиться с учетом перспективы включения ее в интегрированную АС предприятия.

В состав интегрированной АС входят АС различного функционального назначения, существенно отличающиеся по времени реакции и интенсивности обрабатываемых информационных потоков. Поэтому управление в ней довольно сложно и носит характер координации принятия решений на разных уровнях иерархии по определенным критериям, важнейшими из которых являются экономические и критерии безопасности производства. Виды АС, входящих в состав интегрированной АС, обусловлены видами процессов деятельности предприятия и показаны на рис. 17.

На рисунке показаны следующие виды АС:

- АСНИ – автоматизированная система научных исследований и комплексных испытаний образцов новой техники;
- САПР – система автоматизированного проектирования;
- АСТПП – автоматизированная система технологической подготовки производства;
- АСУП – автоматизированная система управления производством;
- АСУ ГПС – автоматизированная система управления гибкими производственными системами;

- АСУ ТП – автоматизированная система управления технологическими процессами.

Ядром интегрированной системы, координирующей действия остальных систем, является АСУП. АСНИ, АСУ ГПС и АСУ ТП являются системами реального времени. Проектирование остальных систем может вестись обычными методами, без детального анализа временной схемы событий и анализа сигналов.



Рис. 17. Виды АС в составе интегрированной АС предприятия

Иерархия управления в интегрированной системе тесно связана с иерархией целей и критериев. Могут быть выделены следующие уровни иерархически связанных целей.

Уровень 1. Производственно-технологические цели, которые определяют соблюдение параметров технологии, плановых параметров производственных звеньев. Достижению этих целей способствуют комплекс средств локального управления отдельными технологическими процессами АСУ ТП, АСУ ГПС и решение локальных задач управления в АСУП.

Уровень 2. Техничко-экономические цели предприятия тактического характера. Для их достижения локальных ресурсов недостаточно, требуется согласованное применение ресурсов нескольких систем: АСНИ, САПР, АСУП, АСТПП, АСУ ГПС, АСУ ТП (например, достижение заданного значения себестоимости продукции).

Уровень 3. Стратегические цели, связанные с выбором типа продукции, способа ее производства и реализации, долгосрочным планированием и распределением ресурсов. Достижение этих целей обеспечивается верхним уровнем решения задач АСУП, АСНИ и САПР, анализом и накоплением больших массивов информации. Координация заключается в ранжировании критериев по степени важности и межуровневом распределении ресурсов, которые всегда ограничены.

В автоматизированных системах управления окончательное управляющее решение принимается человеком. Решение задач и реализация информационно-поисковых операций должны обеспечить необходимую достоверность и обоснованность принимаемых решений, их своевременность.

Особенностью системного анализа при построении интегрированной АС является правильная декомпозиция на подсистемы. Обычно учитываются виды систем, территории и подразделения предприятия. АСУП декомпозируется, в свою очередь, на ряд АС (см. раздел 4.2). Последовательно применяемый принцип нисходящей разработки в конечном итоге приводит к проектированию отдельных процессов как реального, так и не реального времени.

Концепция интегрированной АС должна включать основные решения по организационной структуре предприятия и архитектуре комплекса технических средств системы. Обоснованные решения по организационной структуре будущей системы принимаются на основе тщательного изучения документооборота предприятия и должностных обязанностей персонала существующей системы (см. раздел 2).

Интегрированные АС взаимодействуют с большим числом пользователей, как правило, территориально удаленных. Поэтому основной архитектурой технического обеспечения является сетевая. Пример построения сети для интегрированной АС приведен на рис. 18. Буквой Н на схеме обозначены концентраторы сети (Hub) [15]. Общие данные хранятся на сервере, остальные пользователи являются клиентами (архитектура *клиент-сервер*). В системе используются в качестве линий связи коаксиальные кабели и витые пары. В последнее время наметилась тенденция перехода к оптоволоконным линиям связи, однако их стоимость пока остается достаточно высокой. Пропускная способность разделяемой линии связи «сервер-коммутатор» должна быть на порядок выше, чем остальных.

Следует отметить, что централизованное хранение данных на сервере имеет свои преимущества и недостатки. Не следует увлекаться чрезмерной централизацией хранимых данных, особенно в случае большого числа автономных систем, выполняющих ответственные функции в интегрированной системе. Лучше наиболее важную часть информации хранить в нескольких местах, включая сервер, что существенно повышает живучесть системы в случае возникновения аварийных ситуаций. При этом, конечно, усложняется отслеживание изменений в базе данных (необходимо предусмотреть одновременную корректировку информации в нескольких пунктах).



Рис.18. Пример построения сети интегрированной АС предприятия

4.2. Структура и функции АСУП – ядра интегрированной АС

Процесс управления предприятием (организацией), имеющий цель получения прибыли или выполнения государственного заказа, может быть представлен схемой, изображенной на рис. 19 [16]. Процесс движения к поставленной цели является многоступенчатым. Неоднократно циклически корректируются первоначальные планы в зависимости от фактически достигнутых результатов (принцип обратной связи). Достижение стратегических целей уровня 3 зависит от следующих основных факторов:

- наличие новых коммерческих идей и обоснованной стратегии деятельности;
- уровень квалификации персонала и его заинтересованность в достижении конечных целей;
- наличие финансовых, материальных и трудовых ресурсов;
- уровень развития информационной инфраструктуры предприятия.

В условиях конкуренции и динамично меняющихся ситуаций на рынке очень важно иметь своевременную и достоверную информацию о конкурентах и состоянии рынка. Без применения новых информационных технологий эта задача решена быть не может.



Рис. 19. Схема управления предприятием

Анализ процессов производственной деятельности предприятия показывает, что в составе АСУП целесообразно выделить следующие подсистемы, обеспечивающие выполнение основных функций управления производством:

- подсистема административного управления;
- подсистема централизованного хранения данных и организации сети (база данных общего назначения);
- подсистема управления закупками (материально-техническим снабжением);
- подсистема управления продажами (сбытом продукции и услуг);
- подсистема технологической подготовки производства;
- подсистема оперативного управления производством;

- подсистема управления транспортно-складскими операциями (логистическая подсистема);
- подсистема бухгалтерского учета.

Подсистема административного управления является координирующей и осуществляет стратегическое управление предприятием в целом по целям уровня 3 и в исключительных случаях – тактическое по целям уровня 2 через подсистему оперативного управления производством. Для крупных предприятий с численностью работающих более 500 человек каждая из указанных подсистем представляет фактически самостоятельную АС, которая, в свою очередь, подразделяется на подсистемы в соответствии с выполняемыми функциями.

Подсистема административного управления выполняет следующие функции:

- анализ финансовой и хозяйственной деятельности предприятия за истекшие периоды по различным показателям и критериям;
- управление маркетингом;
- финансовое и хозяйственное планирование для предприятия в целом;
- управление кадрами;
- управление документооборотом;
- координация деятельности всех подсистем и структурных подразделений верхнего уровня (филиалов, служб, отделений, производств, самостоятельных подразделений).

Анализ финансовой и хозяйственной деятельности проводится для обоснования стратегических решений по управлению предприятием со стороны руководства верхнего уровня (генерального директора, директоров филиалов и их заместителей, главного бухгалтера). К ним относятся утверждение и корректировка стратегических долгосрочных (не менее чем на квартал) финансовых и хозяйственных планов предприятия, решения о производстве новых изделий и оказании новых услуг, решения о назначениях, перемещениях, наказаниях и поощрениях руководящего персонала предприятия, решения о распределении ресурсов между подсистемами и структурными подразделениями верхнего уровня и разрешение конфликтных ситуаций между ними. Данные для анализа подготавливаются отдельными подсистемами, причем основная их часть должна пройти предварительную обработку в подсистеме бухгалтерского

учета. Результаты анализа представляются в виде наглядных графических и текстовых отчетов, при подготовке широко используются различные методы статистической обработки информации.

Управление маркетингом дополняет результаты анализа финансовой и хозяйственной деятельности предприятия детальным анализом состояния рынка сбыта продукции и услуг. Основная цель – дать обоснованный прогноз развития рынка и выдать рекомендации по финансовому и хозяйственному стратегическому планированию с учетом ожидаемого спроса товаров и услуг. Кроме того, маркетинг предполагает применение средств активного воздействия на рынок (реклама, работа с потенциальными и фактическими покупателями, организация работы коммивояжеров) с целью создания благоприятной коммерческой ситуации. Реализация этой функции позволит осуществлять более эффективное финансовое и хозяйственное планирование и управление продажами (сбытом) продукции и услуг. Типичными задачами управления маркетингом являются :

- ведение раздела базы данных по продукции, типовым услугам;
- ведение досье на фирмы-конкуренты и товары-аналоги;
- анализ сегментов рынка и ценовой ситуации, складывающейся на рынке;
- сбор отзывов о реализованной продукции, анализ претензий, рекламаций и хода гарантийного обслуживания;
- прогноз состояния рынка и ценовой ситуации по отдельным сегментам рынка;
- рейтинговая оценка перспективности отдельных видов продукции и оказания услуг и прогноз их потребности;
- анализ рынка рекламы и эффективности рекламной деятельности предприятия и работы с покупателями.

Финансовое и хозяйственное планирование предполагает составление стратегических долгосрочных (не менее чем на квартал) финансовых и хозяйственных планов предприятия, на основании которых осуществляется деятельность всех подсистем и структурных подразделений предприятия. Каждая из подсистем, в свою очередь, на основе стратегического плана разрабатывает оперативные планы своей деятельности на период от одной смены до месяца и осуществляет контроль выполнения этих планов. В случае необходимости в стратегические планы вводится коррекция, которая должна

быть немедленно отработана в оперативных планах всех подсистем и подразделений.

В финансовом плане планируются финансовые поступления и затраты в течение определенного периода (обычно от квартала до 5 лет) с группировкой по видам поступлений и затрат и направлениям деятельности предприятия. Предусматривается анализ вариантов плана по различным критериям при изменении факторов, например, изменения курса базовой валюты. Планирование ведется по результатам анализа финансовой и хозяйственной деятельности предприятия за истекшие периоды и выполнения функции управления маркетингом. Важнейшей задачей управления является динамическое отслеживание показателей выполнения плана и своевременное оповещение администрации об отклонениях для принятия стратегических и, при возникновении исключительной ситуации, и тактических (оперативных) решений. Все решения отображаются в базе данных с датой и временем его принятия и оформляются в виде документа бумажного и/или электронного вида с включением в документооборот предприятия.

Хозяйственный план предприятия содержит перечень мероприятий со сроками их выполнения, ответственными лицами и указанием ресурсов, выделяемых на выполнение этого мероприятия. Мероприятия группируются по направлениям деятельности и должны полностью соответствовать финансовому плану предприятия. По плану составляется календарно-сетевой график выполнения работ, который отслеживается по факту выполнения. Хозяйственный план предприятия является основой для оперативно-календарного планирования в подсистеме оперативного управления производством. Поэтому одной из важных задач, решаемых при реализации данной функции, является задача взаимной увязки (согласования, координации) оперативных планов всех подразделений и исполнителей работ с хозяйственным планом предприятия (соблюдение баланса ресурсов и временных ограничений).

Управление кадрами является одной из важнейших управляющих функций в АСУП. При реализации этой функции решаются следующие комплексы задач:

- автоматизированное ведение личных дел сотрудников;
- учет и оформление допусков сотрудников к закрытой информации;
- планирование и управление штатным расписанием и резервом на замещение должностей;

- планирование и учет рабочего времени сотрудников (совместно с другими подсистемами по функциональной принадлежности);
- учет и документальное оформление решений руководства по кадровым перемещениям, приему и увольнению, поощрению и наказанию сотрудников с передачей соответствующей информации в подсистемы и базу данных предприятия;
- формирование и выдача отчетов по кадровой информации по запросам администрации предприятия, руководителей подразделений и сотрудников отдела кадров.

Состав хранимых данных должен соответствовать требованиям личной карточки сотрудника (типовая форма Т-2) и типовой анкеты (утвержденной постановлением Правительства РФ). По данной функции подсистема административного управления должна взаимодействовать с подсистемой бухгалтерского учета в части табельного учета и оплаты труда сотрудников. Для управления штатным расписанием и резервом на замещение должностей в подсистеме должны храниться сведения о занятых и свободных вакансиях, характеристики рабочих мест и требования к обязанностям, уровню квалификации и другим характеристикам персонала для каждого из рабочих мест, список сотрудников, входящих в резерв.

Управление документооборотом предполагает единую схему документооборота на предприятии в целом. Документы, входящие в документооборот, могут быть получены сканированием бумажных документов (электронные полные копии бумажных документов), получены по электронной почте или подготовлены с помощью текстового редактора. Содержание документа может дублироваться записями, хранящимися в базе данных системы и отображающимися на экране в виде специальной видеоформы. График документооборота утверждается руководством предприятия и определяет фактически регламент работы пользователей в системе и ответственность лиц, создающих, проверяющих и обрабатывающих конкретный документ или вид документа. Бумажные оригиналы и заверенные копии документов и порядок их обработки должны регистрироваться в системе в специальном реестре (электронном журнале) с указанием характеристик операции обработки и состояния документа. Завершенные и предназначенные к исполнению и обработке электронные документы также регистрируются в системе с полным контролем всех стадий работы с этими документами. Руководству, лицам,

ответственным за документооборот, и всем остальным пользователям системы предоставляются отчеты по документам с учетом их прав доступа к служебной информации. Обычно система документооборота используется и как дополнительное средство общения пользователей при решении производственных задач. Обязательным условием нормального выполнения функции является общая система классификации и кодирования документов предприятия

Координация деятельности подсистем и структурных подразделений обеспечивает реализацию идеи интеграции. Должно обеспечиваться достижение совокупности целей разного уровня при условии ограниченных ресурсов, в том числе информационных. В интегрированной АС предприятия должны выполняться условия взаимосвязанного управления технологическими и организационно-экономическими процессами, проектированием и исследованием. Для координации должно быть построено дерево целей и определены критерии согласованности применяемых решений. Координация облегчается при применении принципа открытых систем, который является одним из способов решения проблемы совместимости технического, информационного и программного обеспечения систем, входящих в состав интегрированной системы. При реализации функции координации должны четко учитываться уровни управления и межуровневая обратная связь: решение, принятое на верхнем уровне (план), должно приниматься с учетом оценки результатов исполнения предыдущих решений и быть обязательным для исполнения следующим нижестоящим уровнем. Период выдачи воздействий и оценки состояния управляемого объекта уменьшается при переходе на следующий уровень управления. Таким образом, на каждом уровне должна быть задана или построена модель управляемого объекта, которая может быть нелинейной или содержать некоторую базу правил, формируемую по результатам анализа финансовой и хозяйственной деятельности предприятия, подсистем и структурных подразделений за истекший период и маркетинга (анализ истории и прогноз).

Подсистема централизованного хранения данных и организации сети (база данных общего назначения) работает под управлением администратора системы и содержит оперативную информацию, справочники, классификаторы, документы, результаты работы отдельных подсистем, которые используются двумя и более подсистемами. Типовая техническая реализация

включает сервер с усиленной информационной защитой и устройствами сетевой связи, кабели связи, различное коммуникационное оборудование. Общесистемное программное обеспечение включает сетевую операционную систему, достаточно мощную СУБД и различные утилиты, облегчающие эксплуатацию системы. Пример реализации сети предприятия приведен на рис. 18.

Подсистема управления закупками (материально-техническим снабжением) выполняет следующие основные функции:

- ведение картотеки потенциальных поставщиков продукции и услуг, необходимых предприятию;
- планирование закупок в соответствии с финансовыми и хозяйственными планами предприятия и данными других подсистем;
- учет закупок и контроль выполнения плана закупок;
- анализ закупок и формирование заказов на поставку;
- поддержка системы документооборота в части, связанной с закупками;
- формирование заявок на складирование закупок и проведение транспортных операций;
- учет рекламаций и претензий к поставщикам, формирование и контроль исполнения соответствующей документации.

Подсистема взаимодействует со всеми остальными подсистемами, особенно интенсивно с подсистемами бухгалтерского учета, оперативного управления производством, управления транспортно-складскими операциями. Подсистема должна использовать единую для предприятия классификацию и кодирование продукции и услуг, основой которой может быть общероссийский классификатор продукции (ОКП).

Подсистема управления продажами (сбытом продукции и услуг) выполняет функции, аналогичные предыдущей подсистеме, но по продажам продукции и услуг другим предприятиям:

- ведение картотеки потенциальных покупателей продукции и услуг, необходимых предприятию;
- планирование продаж в соответствии с финансовыми и хозяйственными планами предприятия и данными других подсистем;

- учет продаж и контроль выполнения плана продаж;
- анализ продаж и формирование предложений по перспективным поставкам и оказанию услуг, а также по ценовой политике предприятия (поддержка маркетинга);
- поддержка системы документооборота в части, связанной с продажами;
- формирование заявок на отпуск продукции со склада и проведение транспортных операций;
- учет рекламаций и претензий покупателей, формирование и контроль исполнения соответствующей документации.

Подсистема взаимодействует со всеми остальными подсистемами, особенно интенсивно с подсистемами бухгалтерского учета, административного управления, управления транспортно-складскими операциями. Подсистема должна использовать единую для предприятия классификацию и кодирование продукции и услуг, основой которой может быть Общероссийский классификатор продукции (ОКП).

Подсистема технологической подготовки производства использует результаты конструкторской подготовки к выпуску продукции, осуществляемой САПР предприятия. Деятельность подсистемы регламентируется стандартами РФ Единой системы технологической документации (ЕСТД). Пользователями подсистемы являются руководство и специалисты технологических служб и подразделений предприятия, отвечающих за технологическую подготовку и соблюдение технологии.

Основными функциями подсистемы являются:

- разработка технологической документации на производство продукции и оказание услуг, включая документацию пооперационных технологических процессов и технологических маршрутов;
- поддержка поддетально- и пооперационно-специфицированных норм расхода материалов и других ресурсов при производстве продукции и оказании услуги;
- расчеты потребностей в материальных и трудовых ресурсах на программу выпуска продукции и оказание услуги, формирование соответствующих заявок в подсистему управления закупками и подсистему административного управления;

- предварительная оценка себестоимости продукции;
- расчет потребности в оборудовании, оснастке, инструменте и формирование заказов в САПР на проведение соответствующих конструкторских работ и заявок в подсистему управления закупками;
- поддержка документооборота в части технологической документации, в частности, подготовки и формирования извещений на изменение технологии или временных отступлений от нее;
- формирование и выдача отчетов руководству и специалистам-технологам по вопросам, связанным с технологической подготовкой производства и соблюдением технологии изготовления продукции и оказания услуг на предприятии.

На крупных предприятиях данная подсистема выделяется наряду с АСНИ и САПР в отдельную автоматизированную систему технологической подготовки производства (АС ТПП).

Подсистема оперативного управления производством

осуществляет тактическое управление основным и вспомогательным производством по целям уровня 2 и частично для слабоавтоматизированных производств, не управляемых АСУ ТП и АСУ ГПС, по целям уровня 1. Основной формой управляющих воздействий являются оперативные планы и задания на смену, сутки, несколько суток, неделю и месяц по подразделениям, службам, производственным участкам, бригадам и отдельным исполнителям. Планы формируются по финансовым и хозяйственным планам предприятия в соответствии с конструкторской и технологической документацией на выпуск продукции и оказание услуг. Планирование должно охватывать как основное, так и вспомогательное производство, включая службы энергообеспечения, изготовления технологического оборудования и оснастки, обеспечения другими ресурсами, службы ремонта и профилактики оборудования, зданий, сооружений, связи и коммуникаций. В соответствии с этим основными функциями подсистемы являются:

- оперативное планирование основного и вспомогательного производства и рассылка соответствующих документов исполнителям по системе документооборота;
- оперативный учет выполнения планов с формированием отчетов о ходе выполнения по запросам других подсистем и оповещение

руководства и ответственных специалистов о всех отклонениях от плана;

- оперативный учет брака и отклонений от технологии и конструкции изготавливаемой продукции и формирование заявок на разрешение изменений технологии и конструкции в подсистему технологической подготовки производства и САПР;
- оперативный учет расхода материальных ресурсов по основному и вспомогательному производству;
- производственный учет рабочего времени персонала, нарушений трудовой дисциплины, переработок, болезней и невыходов на работу и передача сведений в подсистему административного управления для выполнения функции управления кадрами;
- детальный контроль незавершенного производства;
- управление транспортно-складскими операциями для транспортных единиц и складов автономного использования;
- поддержка документооборота в части оперативного управления основным и вспомогательным производством.

Подсистема управления транспортно-складскими операциями (логистическая подсистема) необходима в том случае, если имеются склады и транспортные единицы общего пользования. Складские помещения могут быть удалены на значительные расстояния, специализированы под хранение продукции и незавершенного производства определенного вида и иметь различные стоимости хранения единиц. Кроме того, транспортное обслуживание складских операций зависит от расположения и уровня потребностей пользователей складов. Задача оптимизации перевозок и складского хранения с учетом требуемых уровней гарантийных запасов, не позволяющих останавливать производство и оказание услуг, является типичной задачей логистики () и может решаться средствами данной подсистемы. В результате ее решения в начале планируемого периода (от смены до месяца) должно быть составлено расписание движения всех транспортных единиц и погрузо-разгрузочных работ, связанных с централизованными перевозками и складированием. Аналогичные задачи для автономно используемых транспортных средств и складских помещений решаются подсистемой оперативного управления производством.

Подсистема выполняет следующие функции:

- оперативное планирование складских, погрузо-разгрузочных и транспортных операций на смену, сутки, несколько суток, неделю, декаду, месяц по видам продукции, пунктам отправления и назначения, бригадам, отдельным исполнителям (водителям, материально-ответственным лицам(МОЛ), специалистам по погрузке и разгрузке) и отдельным техническим единицам и складам;
- поддержка документооборота в части складских, погрузо-разгрузочных и транспортных операций;
- учет фактически выполненных складских, погрузо-разгрузочных и транспортных операций с ведением картотек складского учета,
- динамический пересчет остатков товарно-материальных ценностей (ТМЦ), включая горючесмазочные материалы (ГСМ) и отслеживание критических уровней запасов;
- учет партий товаров, контроль сроков хранения партий, сроков действия сертификатов(лицензий) на хранимые ТМЦ;
- ведение учетных цен, поддержка различных методов списания хранимых единиц ТМЦ;
- формирование оборотных ведомостей и ведомостей движения ТМЦ за период по различным условиям по запросам пользователей;
- поддержка проведения инвентаризации и формирование соответствующей документации;
- учет состояния складов и транспортных единиц и формирование заявок на ремонт и закупки в подсистемы оперативного управления производством и управления закупками;

Данные документов и учетные данные подсистемы должны обрабатываться подсистемой бухгалтерского учета.

Подсистема бухгалтерского учета по своим функциям и результатам деятельности должна соответствовать Положению по ведению бухгалтерского учета и бухгалтерской отчетности в Российской Федерации, утвержденному Министерством финансов РФ. В свою очередь, оно разработано на основе Федерального закона «О бухгалтерском учете». Ответственность за организацию бухгалтерского учета в организации несет руководитель организации.

Подсистема выполняет следующие основные функции:

- учет хозяйственных операций и документов, сопровождающих операции, с отражением результатов операции в плане счетов предприятия (совместно с другими подсистемами);
- формирование по запросу сотрудника бухгалтерии первичных учетных документов и отчетов по ним;
- обработка учетной информации и соответствующих документов, поступающих из других подсистем, с целью наиболее полного отражения хозяйственной деятельности предприятия и оформления платежных документов;
- ведение Главной книги предприятия, подготовка бухгалтерского баланса и приложений к нему, другой бухгалтерской отчетности в соответствии с правилами ее составления и представления;
- поддержка кассовых и финансово-расчетных операций с формированием отчетности по движению денежных средств;
- учет основных средств и нематериальных активов (НМА) предприятия;
- учет товарно-материальных ценностей (ТМЦ), малоценных и быстроизнашиваемых предметов (МБП);
- расчет и начисление зарплаты и других выплат сотрудникам предприятия по данным табельного учета подсистемы административного управления;
- учет фактических затрат по статьям затрат и прибыли предприятия от реализации продукции и услуг, расчет и анализ фактической себестоимости продукции и услуг;
- учет и анализ отношений предприятия с дебиторами-кредиторами;
- поддержка системы документооборота в части бухгалтерской документации.

Известны многие стандартные системы и программные комплексы, автоматизирующие функции бухгалтерского учета: 1С, Парус, БЭСТ, Инфобухгалтер и др. При их использовании следует осуществить их связь с другими подсистемами интегрированной АС предприятия .

4.3. Структура и функции АСУ ТП

В понятие «технологический процесс как объект управления» [22] включается технологическое оборудование кроме датчиков и исполнительных механизмов, которые, оставаясь конструктивными элементами оборудования, обычно относятся к АСУ ТП. В отдельных случаях, когда выбор датчиков и исполнительных механизмов жестко predetermined и связан с отдельной и сложной задачей преобразования состояния внешней среды или оборудования в электрический сигнал, датчики и исполнительные механизмы могут рассматриваться как внешние источники и приемники сигналов и не включаться в состав проектируемой системы. Управление технологическими процессами в большинстве случаев сводится к управлению режимами работы оборудования. Для управляемого технологического процесса должны быть определены входные контролируемые управляющие воздействия, установлены детерминированные или вероятностные зависимости между входными воздействиями и выходными параметрами выпускаемой продукции. Эти зависимости будем называть *моделью управления в АСУ ТП*. Модели управления могут быть заранее известны с той или иной степенью точности или полностью или частично неизвестны. В последнем случае они могут строиться и уточняться самой АСУ ТП в процессе функционирования (*идентификация управляемого объекта*).

Типовая структура технического обеспечения АСУ ТП представлена на рис. 20.

На схеме приняты следующие обозначения: *АРМ* – автоматизированное рабочее место оператора (диспетчера) участка; *СОИ* – система отображения информации (пульт диспетчера); *УСО* – устройство связи с объектом; *ПК* – программируемый контроллер; *Д* – датчик; *Т* – табло; *ИМ* – исполнительный механизм. На производственном участке реализуется процесс производства некоторой продукции на нескольких единицах оборудования.

Некоторые параметры контролируются и управляются в целом для процесса, например, качество готовой продукции. Отдельные единицы технологического оборудования контролируются и управляются по индивидуальным режимам, что требует отдельных УСО, ПК, датчиков и исполнительных механизмов. На каждом виде оборудования может устанавливаться табло отображения его состояния. Центральный пункт управления состоит из диспетчерского пульта и управляющей ЭВМ, входящей в

состав АРМ диспетчера участка. АРМ подключен к вычислительной сети предприятия, обеспечивая входение АСУ ТП в интегрированную АС предприятия.



Рис.20. Типовая структура технического обеспечения АСУ ТП

В общем виде математическая модель управления в АСУ ТП представляет собой зависимость вида

$$y(t+\Delta t) = A \{u(t) + By(t); F[\xi(\tau), \eta(\tau)]\}, \quad (1)$$

где $y = \{y_1, y_2, \dots, y_n\}$ – вектор выходных переменных; Δt – время от начала выдачи воздействия до получения контрольной информации о результатах этого действия; A – оператор действия АСУ ТП в целом;

$u(t) = \{u_1(t), u_2(t), \dots, u_m(t)\}$ – входные контролируемые воздействия; B и F – операторы управляющих и неуправляющих воздействий; $\xi(\tau)$ – вектор контролируемых, но неуправляемых воздействий (помех, возмущений); $\eta(\tau)$ – вектор неконтролируемых воздействий.

В выражении (1) $t_0 \leq t \leq t_0 + T$, $t \leq \tau \leq t + \Delta t$, где t_0 – начало отсчета времени; T – длительность интервала наблюдения за поведением процесса. Таким образом, управление происходит с некоторым запаздыванием, связанным не только с инерционностью самого объекта, но и дополнительным временем, требующимся для измерений и оценки состояния объекта. Поэтому многие алгоритмы управления предусматривают построение или уточнение параметров модели с целью прогноза состояния объекта на период Δt . Если инерционность самого процесса на порядок превосходит время измерений и оценки состояния, то возможно с определенной погрешностью вести управление в реальном масштабе времени.

Рассматривая АСУ ТП как нижний уровень иерархии управления производством, мы должны иметь в виду, что, в свою очередь, управление в АСУ ТП также может быть иерархическим. Будем различать 4 уровня управления. На нижнем уровне иерархии α_0 управление осуществляется отдельными технологическими операциями. Алгоритмы управления на этом уровне относительно независимы и обычно сводятся к стабилизации режима или несложному программному управлению. На более высоком уровне α_1 рассчитываются изменения режима и хода технологической операции и выдаются задания (уставки) на регуляторы уровня α_0 . Еще более высокий уровень α_2 служит для координации и корректировки конечных состояний связанных между собой технологических операций и управления ходом технологического процесса в целом (например, управление технологической линией). Задающие воздействия для этого уровня формируются самым верхним уровнем управления АСУ ТП -- α_3 , который соответствует управлению участком производства или цехом (группой автоматических линий). На этом уровне оптимизируется выпуск изделий по участку или цеху в целом в соответствии с оперативными планами и заданиями подсистемы оперативного управления производством.

На каждом уровне используются свои модели управления. С ростом номера уровня растет и период обработки информации и принятия решения, динамика управления замедляется. Степень сложности АСУ ТП обычно оценивается по количеству контролируемых и управляемых параметров (Вальков): малые (до 40 параметров), средней сложности (до 100), сложные (до 800), большой сложности (свыше 800). Конечно, это деление является условным, так как некоторые измерения и преобразования физических величин

могут потребовать разработки очень сложного нестандартного оборудования, без которого АСУ ТП не сможет нормально функционировать.

Стандартные модели управления предусматривают управление по отклонению и по возмущению. Известно большое число типовых регуляторов по отклонению на базе программируемых контроллеров, реализующих наборы пропорциональных (П), интегральных (И) (астатических), пропорционально-интегральных (ПИ) (изодромных), пропорционально-интегрально-дифференциальных (ПИД) законов управления. Принципиальным здесь является необходимость измерения или вычисления отклонения выходной управляемой величины от заданного значения (ошибки), управляющего воздействия и, при управлении по возмущению, возмущающего воздействия. Управление по возмущению реализуется методами теории инвариантности. Набор вычислительных операций при реализации алгоритмов реализует численные методы вычисления значений интегралов и производных по результатам измерений (отсчетов), решения линейных и, в более сложных случаях, нелинейных систем дифференциальных и интегральных уравнений.

Наиболее сложными в реализации являются алгоритмы оптимального управления верхних уровней. Характер этих алгоритмов полностью определяется спецификой производственного процесса. Поскольку одновременного достижения оптимальных значений нескольких показателей добиться невозможно, среди всех показателей выделяется наиболее важный, а на остальные накладываются ограничения. Возможно введение комплексного критерия в виде линейной или нелинейной свертки критериев. Если модель управления полностью или частично не определена, ее можно доопределить экспериментально методами параметрической или непараметрической идентификации.

Характер производственных процессов также во многом определяет вид модели управления. Для многих производств (энергетика, нефтехимия, металлургия и др.) процессы выпуска продукции носят преимущественно непрерывный во времени характер и их динамика хорошо описывается системами дифференциальных уравнений с непрерывными функциями времени. В то же время существуют производства (машиностроение, горная промышленность, строительство, электроника и др.), в которых явно наблюдается дискретность выполнения операций во времени (типичный пример

– сборочный конвейер). Здесь преобладают логические модели обработки событий, вероятностные и статистические модели.

Например, если характер изменения вектора неконтролируемых возмущений $\eta(\tau)$ неизвестен, выражение (1) при управлении по статистическим алгоритмам примет вид:

$$M_y(t+\Delta t) = M A \{u(t) + B[M_y(t), \xi(\tau)], \quad (2)$$

где M – символ математического ожидания.

Таким образом, здесь реализуется принцип циклического управления с циклом Δt , когда в цикле управления значительная часть времени тратится на статистическую обработку измерений по вектору управления и по вектору выходных величин с целью вычисления математического ожидания случайных процессов по их реализациям. При этом в силу запаздывания оценки результата управления вероятностные характеристики возмущений учитываются в следующем цикле управления в предположении, что за время цикла управления они не успевают существенно измениться. В многооперационных схемах правая часть выражения (2) может быть представлена суммой составляющих математического ожидания по всем операциям.

Проектирование АСУ ТП по методологии Гэйна-Сарсона должно начинаться с тщательного анализа технологического объекта управления. Вначале должен быть определен тип производства или технологии выполнения операций (непрерывное, дискретное, смешанное), производство подразделяется на территориальные участки с функционально обособленными технологическими операциями и технологическим оборудованием. В соответствии с целями выполнения каждой технологической операции определяется перечень контролируемых параметров или физических величин, диапазоны их изменения, требуемая точность и быстродействие процесса измерения. По отношению к технологической операции (или оборудованию) контролируемые параметры делятся на две группы: входные (управляющие) и выходные (управляемые и неуправляемые информирующие). Соответственно для АСУ ТП они будут выходными и входными (зеркальное отображение). Рекомендуется отделить сигналы от данных. Будем считать сигналом любую информацию, которую формирует аппаратура (техническое устройство), будь то сложное измерительное устройство или обыкновенная пара контактов реле. Под данными в АСУ ТП будем понимать любую информацию, которая формируется человеком или системой в виде документа или сообщения, вводимого и

выводимого ручным способом. Конечно, это деление является условным и в основном связано с требованиями обработки в реальном масштабе времени. Принципы описания сигналов, данных и их характеристик были изложены ранее, в разделах 2.5, 3.4. В составе технологического оборудования определяют различные системы регулирования и управления, как встроенные в технологическое оборудование, так и функционирующие самостоятельно. Для каждой технологической операции определяется модель локального управления, обеспечивающая требуемое качество и производительность выполнения операции. Для построения моделей можно использовать методы планирования эксперимента или проводить наблюдения с последующей статистической обработкой при проведении контрольных операций. Важными являются вариации (степень разброса) выходных параметров качества и производительности при изменении влияющих факторов. Аналогичный подход применяется для построения модели управления производством в целом. В общем случае строится четырехуровневая модель управления АСУ ТП и ее связь с системой оперативного управления производством АСУП. Если датчиковая и измерительная аппаратура задана жестко, то целесообразно выделить ее во внешние сущности, иначе внешней сущностью следует считать контролируемую точку на технологическом оборудовании или физический параметр (например, температура ротора турбины). В последнем случае задача измерения температуры решается внутри АСУ ТП и представляет часть задачи проектирования АСУ ТП в целом.

Обработка измерительной информации в АСУ ТП делится на первичную и вторичную. Первичная обработка производится всегда в реальном времени протекания технологического процесса и преследует цель повысить достоверность и точность измерений контролируемых параметров. Здесь чаще всего используются стандартные алгоритмы усреднения и фильтрации, причем возможна их аппаратная и программная реализация. Вторичная обработка (расчет оптимальных воздействий, глубокий корреляционный и регрессионный анализ при реализации статистических алгоритмов управления, подготовка и оформление документации и отчетов и т. п.) может осуществляться как в реальном, так и вне реального времени (например, в АСНИ или на верхних уровнях управления АСУ ТП).

В соответствии с проведенным анализом выделяются подсистемы АСУ ТП и строится контекстная диаграмма второго уровня и, при необходимости,

третьего. Каждой технологической операции соответствует своя подсистема. Строится диаграмма потоков управления и детализирующие диаграммы потоков данных, определяется структура накопителей данных.

В качестве примера моделирования технологической операции дискретного типа рассмотрим процесс сборки изделия [22]. Даны две группы деталей, из которых надо собрать изделие. В каждое изделие входят две детали: по одной из каждой группы. Допустим, в худшем случае, что к каждой детали первой группы при сборке по параметрам подойдет только одна деталь из второй группы. Пусть число деталей в группах одинаково. Выборка деталей из групп – случайная с равномерным законом распределения по параметрам. Если вторая деталь при выборке не соответствует первой, она остается на месте сборки, извлекается следующая деталь из второй группы и т.д., пока не будет найдена подходящая, происходит сборка изделия, все неиспользованные детали снова возвращаются во вторую группу. Извлекается очередная первая деталь, и весь цикл повторяется. Технологическая операция завершается, когда все изделия будут собраны.

Вероятность сборки за один шаг обратно пропорциональна количеству деталей второй группы. В первом цикле сборки эта вероятность равна

$$p_n = 1/[N - (n - 1)], \quad (3)$$

где N – начальное число деталей, n – число извлеченных деталей второй группы. Так как с увеличением номера цикла сборки γ на 1 число деталей второй группы уменьшается на 1, вероятность сборки деталей на каждом шаге каждого цикла

$$p_{n\gamma} = 1/\{[N - (\gamma - 1)] - (n - 1)\}. \quad (4)$$

Полная сборка всех изделий в наихудшем случае, когда в каждом цикле сборка осуществляется на последнем шаге, будет произведена за L шагов:

$$L = \sum_{i=1}^{i=N} i = N(N + 1)/2. \quad (5)$$

Например, при $N=10$ понадобится не более 55 шагов выбора деталей.

Если изделие при сборке требует селекции нескольких (больше одной) деталей, то формула (5) приобретает вид:

$$L = m \sum_{i=1}^{i=N} i = m[N(N + 1)/2], \quad (6)$$

где m – число деталей, подлежащих подборке для изделия

Минимальное число шагов сборки, очевидно, равно mN , когда сборка осуществляется наиболее удачно, с первого шага.

Формулы (5) и (6) дают крайние оценки затрат на подборку деталей, реально будет более объективной оценкой среднее значение $m [N (N + 3)/4]$. Однако, применяя различные стратегии, например, введя предварительную группировку деталей по типоразмерам, можно сократить длительность сборки. Конечно, при этом нужно учитывать дополнительные затраты на группировку и контроль деталей. Моделирование и рекомендации по использованию той или иной стратегии АСУ ТП может выработать на основе экспериментальных наблюдений за реальным процессом сборки. На диаграммах потоков данных должны быть отражены входные потоки – характеристики процесса сборки, и выходные – рекомендации по изменению стратегии сборки и введению дополнительных технологических операций контроля и сортировки деталей.

Одной из основных задач, решаемых системой управления, является получение оценки состояния объекта управления. Для оценки состояния по экспериментальным наблюдениям могут использоваться методы наименьших квадратов, максимального правдоподобия, байесовых оценок и ранжирования, хорошо известные в математической статистике и теории планирования эксперимента. Оптимизация состояния объекта управления (оптимальное управление) проводится с использованием методов вариационного исчисления, принципа максимума Понтрягина, динамического программирования, различных градиентных методов поиска.

Для технической реализации АСУ ТП и АС в целом фирмы-производители оборудования выпускают стандартные наборы конструктивов с использованием средств микропроцессорной вычислительной техники. Одним из примеров такого набора является система ПРОСКОН 2100 (PROSCON 2100) фирмы Outokumpu [17]. В состав системы входят прикладные станции управления (АС-станции), реализующие функции УСО и программируемых контроллеров; станции управления процессом (РМ-станции) и заводские станции оператора (РО-станции), реализующие функции АРМ диспетчеров и системы отображения информации. Все станции связаны друг с другом через стандартную локальную вычислительную сеть (ЛВС) Ethernet. Сами станции, а также ЛВС могут быть продублированы с целью повышения надежности. РМ-станции представляют собой относительно простые терминальные ЭВМ (процессорный блок, подключенный к сети, монитор, клавиатура, принтер) и могут располагаться

территориально в любых помещениях предприятия (диспетчерских, кабинетах, складах и т.д.).

При необходимости сложной индивидуальной обработки данных к системе ПРОСКОН 2100 через сеть Ethernet может быть подключено любое число инженерных станций (EW-станций), т. е. высокопроизводительных персональных ЭВМ со своей конфигурацией и прикладным программным обеспечением для образования АРМ инженеров и исследователей. На уровне предприятия организуется центральный диспетчерский пункт с разделяемыми ресурсами на базе нескольких РО и РМ-станций, а также сервер баз данных (операционная система NT4 и СУБД Microsoft SQL Server). РО-станция имеет процессорный блок меньшей мощности, чем РМ-станция, однако более расширенный состав мониторов, позволяющий создавать многоместные системы отображения информации и наблюдения. В принципе, к серверу можно обращаться с любого АРМ. Стандартный протокол сетевого обмена TCP/IP и программная поддержка РМ и РО-станциями сетевой работы с файлами по стандартам NFS и FTP позволяет организовать связь с подсистемами, работающими с различным сетевым программным обеспечением (Novell NetWare, Windows NT, LAN Manager). Эти же протоколы и стандарты используются для взаимодействия с различными *программируемыми логическими контроллерами* (ПЛК, PLC) как автономными, так и входящими в состав какого-либо оборудования.

АС-станции системы используют ПЛК типа Series 90-70 производства GE Fanuc Automation, работающие с шинами данных VME и Intel 80486 CPU. Для реализации элементарных стандартных функций измерения и локального управления фирма Outokumpu предлагает программные модули на языке С, настраиваемые под конкретные производственные условия. Конструктивно АС-станции собираются из секций (корзин), состоящих из блока питания и 9 вставных модулей стандартных размеров со стандартными разъемами. Секции по 4 собираются в стойки, подключаемые к кроссовым шкафам предприятия, к которому подходят кабели и провода от датчиков, измерительного оборудования и исполнительных механизмов. Конструктивно секция объединяется с кроссовым шкафом и соответствует ему по высоте.

Для подключения удаленных точек используется специальное оборудование подключения типа Genius или Field, обладающее малыми утечками электрического тока. Для особо ответственных датчиков и механизмов (например, механизм управления горелкой, аварийные выключатели и т.п.)

может быть вблизи установлен отдельный ПЛК. Возможна группировка АС-станций в отдельных помещениях со связью с остальными станциями по сети Ethernet с использованием волоконно-оптического кабеля .

Модули ввода/вывода Genius также являются разработкой GE Fanuc Automation и устанавливаются по одному или в группах в полевые шкафы непосредственно около точек измерения или управления. Блоки связаны с АС-станциями витой парой и могут быть удалены от них на расстояния до нескольких километров. Применяется тройное резервирование шины связи, для особо ответственных участков может использоваться волоконно-оптический кабель.

Модули ввода/вывода Field – другое конструктивное решение той же фирмы для подключения удаленных точек измерения и управления к АС-станциям. Они также размещаются в полевых шкафах, однако компонуются в локальные группы, управляемые специальным процессором, связанным шиной Genius с АС-станцией. При большом количестве точек измерения и управления использование этой технологии групповых преобразователей может дать экономию за счет уменьшения суммарной длины магистральных кабелей. На случай непосредственного подключения Field-модулей к АС-станции в последней предусмотрен контроллер шины Fieldbus.

Система ПРОСКОН 2100 снабжена системой функциональной самодиагностики и возможностями организации горячего резерва (немедленного включения резерва) для любого ПЛК в составе УСО.

РМ-станции представляют собой достаточно оснащенные АРМ с ЭВМ Intel Pentium или Hewlett Packard, работающих под управлением операционных систем UNIX или Windows NT. Станции собирают информацию от АС-станций и архивируют ее в форме рапортов для долгосрочного хранения . Кроме того, они реализуют управление по критериям верхних уровней и обеспечивают необходимую сигнализацию и сервис оператору. Для повышения надежности системы РМ-станции обычно дублируются, причем вторая станция находится в горячем резерве и немедленно активируется при сбое основной станции. Обе станции одновременно обслуживают терминалы оператора. Для дополнительной обработки архивированных данных все станции снабжены программными системами Excel (электронные таблицы) и APS (расширенное изучение процессов), а также могут передавать данные в виде рапортов в более мощные специализированные инженерные EW-станции. Данные архивируются в файлах

98

фиксированного размера и формата. По каждой переменной могут быть установлены циклы опроса от 1 секунды до 1 суток, при этом количество хранимых значений определяется емкостью дискового накопителя и может включать несколько сотен тысяч значений (это количество задается как параметр и после его достижения самые старые по времени значения обновляются). При необходимости может быть организован архив второго уровня. Например, при периоде опроса 1 минута за 2 месяца накапливается по одной переменной 44690 отсчетов, их можно сбросить в архив длительного хранения второго уровня. Для управления по статистическим критериям и алгоритмам предлагается набор программных модулей SPC (статистическое управление производственным процессом).

РО-станции с двумя и тремя мониторами наблюдения могут содержать как обычные мониторы, так и большие цветные мониторы с высоким разрешением. При необходимости РО-станции могут быть помещены в шкафы со степенью защиты класса IP65 или использовать дисплейные панели с той же степенью защиты для работы в загрязненных средах. Для управления и ввода информации в этом случае используются или контактный экран, или пленочная вспомогательная клавиатура. Таким образом, система ПРОСКОН 2100 позволяет организовать самые различные диспетчерские пункты на самых разных уровнях управления, начиная от отдельной единицы оборудования и до предприятия в целом. Каждая РО-станция может контролировать любые РМ-станции, находящиеся в сети.

В системе ПРОСКОН 2100 цикл обновления всех переменных измерений и управления, кроме логических, кратен 100 миллисекундам (от 1 до 32000 раз). Для логических операций (включено/выключено) цикл обновления менее 50 мс, обычно от 20 до 30 мс. Эти цифры характеризуют быстроедействие системы на нижнем уровне управления.

Теоретически пропускная способность каждой ЛВС Ethernet 10 миллионов бит в секунду (10 Мбит/с). Практически она не превосходит 2 Мбит/с в реальных производственных условиях, что означает, что полный обмен между станциями происходит в наихудшем случае в течение 1 секунды, это и определяет реальный минимальный период опроса контролируемых параметров со стороны РС-станций. Обновление изображений на мониторах не превышает 2 секунд. На случай отключения питания все АС-станции имеют автономное батарейное питание, а система в целом – питание от специального источника UPS с

подзарядкой резервного аккумулятора. Это обеспечивает нормальное функционирование системы не менее 380 минут при полностью отключенной электросети.

Возможна связь с корпоративными сетями и глобальной сетью Internet. Это делает этот комплекс удобным для реализации не только АСУ ТП, но и для реализации интегрированной системы предприятия в целом.

4.4. Оценка характеристик проектируемой системы

После построения информационно-логической модели будущей системы и предварительного выбора комплекса технических средств следует произвести расчеты требуемых ресурсов памяти и времени реакции (быстродействия) системы. Эта оценка является приближительной и должна характеризовать порядок величин при наилучших условиях функционирования системы.

Расчет требуемых ресурсов памяти производится отдельно для внешней (долговременной) и оперативной памяти.

Для расчета необходимой *внешней памяти* воспользуемся формулой (7):

$$V_{ВП} = V_{ОС} + V_{СУБД} + V_{данных} + V_{программы}, \quad (7)$$

где $V_{ВП}$ – общий объем внешней памяти, Мбайт;

$V_{ОС}$ – объем внешней памяти, требуемый для хранения файлов операционной системы и ее нормальной работы, Мбайт;

$V_{СУБД}$ – объем внешней памяти, требуемый для хранения файлов СУБД, Мбайт;

$V_{данных}$ – объем внешней памяти, требуемый для хранения записей базы данных и результатов выполнения функций, Мбайт;

$V_{программы}$ – объем внешней памяти, необходимой для хранения текстов и библиотек приложений. Мбайт.

Аналогичная формула используется для расчетов требуемых ресурсов оперативной памяти.

В качестве примера рассмотрим расчет требуемых ресурсов памяти для АС учета единиц складского хранения, работающей под управлением операционной системы Microsoft Windows 98 с использованием СУБД Visual FoxPro 6.0.

По экспериментальным данным Windows 98 требует $V_{OC} = 166$ Мб. Объем памяти, необходимый для Visual FoxPro 6.0, составляет примерно 60 Мб. Для расчета объема хранимых данных предположим наихудший случай: максимальное заполнение базы данных на 1 миллион единиц хранения. Данные расчетов сведены в таблицу 4. Индексный файл ориентировочно принят в размере 15% от основного.

Итак, мы получили, что $V_{данных} = 1590.75$ Мб. Объем памяти, требуемый для программы приложения, составляет 8 Мб.

Сложив полученные данные, будем иметь:

$$V_{ВП} \approx 1,6 \text{ Гб.}$$

Наиболее распространены устройства внешней памяти типа «винчестер» емкостью от 3,2 Гб до 60 Гб, причем материнская плата позволяет подключать несколько (обычно до 4-х) устройств. Таким образом, даже минимальная конфигурация ЭВМ будет обеспечивать требуемые ресурсы долговременной памяти.

Теперь рассчитаем объем *оперативной памяти*.

По экспериментальным данным и данным разработчиков имеем следующие требования к оперативной памяти:

$$V_{OC} = 16 \text{ Мб; } V_{СУБД} = 4 \text{ Мб; } V_{программы} = 4 \text{ Мб.}$$

Результаты расчета объема кэша для хранения данных в оперативной памяти приведены в таблице 5.

Таким образом, $V_{данных} = 7,68$ Мб.

Подсчитаем общий объем оперативной памяти:

$$V_{ОЗУ} = 16 + 4 + 4 + 7.68 = 31.68 \text{ Мб}$$

Следовательно, объем оперативной памяти, необходимой для нормального функционирования программы:

$$V_{ОЗУ} = 32 \text{ Мб.}$$

Оперативная память комплектуется модулями стандартного размера. В настоящее время получили распространение модули 16, 32, 64, 128, 256 Мб, которые можно в зависимости от типа материнской платы собирать в наборы, обеспечивая тем самым необходимый объем памяти. Каждая материнская плата имеет ограничения как по количеству разъемов подключения, так и по общему объему оперативной памяти, что следует учитывать при выборе процессорного блока компьютера.

Расчет объема данных

Таблица 4

Имя файла БД	Размер записи, байт	Максимальное количество записей	Размер индекса, байт	Итого, байт
edhran.dbf	600	999999	89999910	689999310
harka.dbf	126	99	1871,1	14345,1
adres.dbf	44	99999	659993,4	5059949,4
uchet.dbf	768	999999	115199885	883199116,8
gabarit.dbf	12	999999	1799998,2	13799986,2
class.dbf	206	99	3059,1	23453,1
superclass.dbf	206	50	1545	11845
znach.dbf	42	999999	6299993,7	48299951,7
class_ed.dbf	12	999999	1799998,2	13799986,2
edhar.dbf	12	999999	1799998,2	13799986,2
clsupel	12	50	90	690
mol	97	50	727,5	5577,5
urdost	17	50	127,5	977,5
spravoch	22	50	165	1265
obekt	22	50	165	1265
Итого				1668017705

Таблица 5

Расчет для объема кэша

Имя файла БД	Размер записи, байт	Максимальное количество записей	Размер индекса, байт	Итого, байт
edhran.dbf	600	5000	450000	3450000
harka.dbf	126	99	1871,1	14345,1
adres.dbf	44	1000	6600	50600
uchet.dbf	768	5000	576000	4416000
gabarit.dbf	12	1000	1800	13800
class.dbf	206	50	1545	11845
superclass.dbf	206	50	1545	11845
znach.dbf	42	1000	6300	48300
class_ed.dbf	12	1000	1800	13800
edhar.dbf	12	1000	1800	13800
clsupel	12	50	90	690
mol	97	25	363,75	2788,75
urdost	17	25	63,75	488,75
spravoch	22	25	82,5	632,5
obekt	22	25	82,5	632,5
Итого				8049567,6

Расчетные таблицы удобно обрабатывать как электронные таблицы Excel, входящие в пакет Microsoft Office. Поскольку оценивается лишь порядок величин, округление результата производится с точностью до мегабайта.

Для распределенных систем с архитектурой «клиент-сервер» необходимо оценить объем требуемой внешней и оперативной памяти на сервере и клиенте отдельно. Например, при использовании операционной системы Microsoft Windows 2000 Advanced Server на сервере потребуется примерно 1400 Мб внешней памяти. Кроме того, при взаимодействии с клиентским слоем требуется не менее 20 Мб для обеспечения функционирования утилиты ADO и 80 Мб для размещения СУБД Microsoft SQL Server 6.5 и выше. При расчете внешней памяти под данные следует учесть, что кроме хранения индексных файлов и файлов данных во внешней памяти сервера необходимо хранить журнал транзакций, который по своему объему иногда превосходит базу данных. Для оценки объема журнала транзакций следует задаться максимальной интенсивностью потока транзакций и периодом его хранения. Для ориентировочных расчетов длину одной транзакции можно принять равной максимальной длине записи базы данных.

Часть базы данных может храниться на клиентских ЭВМ – это решение принимает проектировщик. Если вся база данных хранится на сервере, то для клиента учитывается только память, требуемая для размещения системных программ и клиентских приложений.

При расчете оперативной памяти сервера следует учитывать, что администрирование сети и обработка транзакций требуют увеличения ресурсов памяти пропорционально количеству обслуживаемых клиентов. Например, для СУБД SQL Server можно использовать формулу: $M = 5 + (2\% \text{ объема данных}) + (0.1 \cdot \text{число пользователей})$, где M – необходимый объем оперативной памяти в Мб (минимально 16, лучше 32 Мб).

Кроме того, следует предусмотреть резерв памяти под операционную систему: например, для Microsoft Windows 2000 Advanced Server 64 Мб, для Microsoft Windows NT4 минимально 24 Мб, лучше 32 Мб. Следует отметить, что чем больше ресурс оперативной памяти, тем выше быстродействие системы.

Расчет времени реакции системы должен дать оценку быстродействия системы. *Временем реакции системы по какой-либо функции* называется время от момента начала запроса на выполнение этой функции от внешнего источника запросов до момента окончания формирования результата по данной функции.

Если результатом выполнения функции является печатный документ, то

учитывая, что устройство печати является весьма медленно действующим, время реакции оценивается отдельно без печати и с печатью документа.

Общее время реакции системы на выполнение запроса рассчитывается по формулам (8):

$$\begin{aligned}
 t_{\text{реакции}} &= t_{\text{ввода}} + t_{\text{считывания}} + t_{\text{создания}} + t_{\text{вычисления}} \\
 t_{\text{ввода}} &= k_{\text{вв}} \cdot L_{\text{симв}} \cdot t_{\text{симв}} \\
 t_{\text{считывания}} &= N_{\text{бл}} * (t_{\text{поз}} + t_{\text{сч.бл}}) \\
 t_{\text{создания}} &= \frac{N_{\text{опер}} * K_1}{f} \\
 t_{\text{записи}} &= t_{\text{создания}} + t_{\text{счр}} * N_{\text{счр}} + t_{\text{обл}} \\
 N_{\text{бл}} &= \frac{V_{\text{табл}} * N_{\text{табл}}}{2 \cdot V_{\text{бита}}}
 \end{aligned} \tag{8}$$

где $t_{\text{ввода}}$ - время на ввод входных данных запроса;

$k_{\text{вв}}$ - коэффициент ошибок при вводе, для расчетов можно принять равным 1.5;

$L_{\text{симв}}$ - количество символов, вводимых в качестве исходных данных запроса;

$t_{\text{симв}}$ - время ввода одного символа, при ручном вводе с клавиатуры в некоторую экранную форму можно принять в среднем равным 2 с;

$t_{\text{считывания}}$ - время, затрачиваемое на считывание физических блоков при работе с накопителем;

$N_{\text{бл}}$ - количество считываемых физических блоков, зависит от количества обрабатываемых таблиц (файлов) и объема таблиц (файлов);

$t_{\text{поз}}$ - время позиционирования головок дискового накопителя;

$t_{\text{сч.бл}}$ - время считывания физического блока в дисковом накопителе;

$t_{\text{вычислений}}$ - время, затрачиваемое процессором на обработку информации с учетом выполнения циклов;

$N_{\text{опер}}$ - количество операций высокого уровня, необходимых для формирования результата;

K_1 - среднее количество тактов машинных команд на 1 операцию, для большинства случаев можно принять $K_1=60$;

f - тактовая частота процессора в Гц;

$V_{\text{табл}}$ - средний объем таблицы в байтах;

$N_{\text{табл}}$ – количество таблиц, обрабатываемых в запросе;

$V_{\text{блока}}$ – объем физического блока носителя;

$t_{\text{вывода}}$ – время на вывод результата на устройство вывода или отображения, для принтера оценивается отдельно. Для дисплея можно принять 0.5 с (зависит от типа видеокарты и дисплея).

Оценим время реакции той же АС учета единиц складского хранения.

Для расчета возьмем случай создания наиболее сложного ежемесячного отчета. Запрос требует ввода примерно 20 символов, что займет 40 секунд времени. В запросе происходит поиск записей по 3 таблицам, общим объемом 6882000 байт. В среднем просматривается половина записей.

Произведем расчеты по формулам, приняв стандартный минимальный размер физического блока 512 байт:

$$N_{\text{зап}} = \frac{6882000 * 3}{2 * 512} = 20162;$$

$$t_{\text{ввода}} = 20 * 2 = 40\text{с};$$

$$t_{\text{системный}} = 20162 * (0.006 + 0.001) = 141\text{с};$$

$$t_{\text{процессинг}} = \frac{13000 * 60}{200 * 10^6} \approx 0.004\text{с};$$

$$t_{\text{вывода (экрин)}} = 0.5\text{с}.$$

Таким образом, для полной подготовки и вывода отчета на экран в самом худшем случае может потребоваться примерно 3 минуты, причем основная часть этого времени тратится на ввод исходных данных и работу с внешним запоминающим устройством.

Теперь подсчитаем время печати по формуле

$$t_{\text{печати}} = t_{\text{печати страницы}} * N, \quad (9)$$

где N — количество страниц отчета.

Согласно формуле (9) для струйного принтера с временем печати 0,4 минуты на страницу формата А4 при $N=10$

$$t_{\text{печати}} = 0,4 * 10 = 4 \text{ минуты.}$$

Полное время реакции системы при формировании самого сложного отчета с печатью на струйном принтере 7 минут.

Итак, согласно проведенным расчетам можно сделать вывод о технических требованиях, предъявляемых системой: процессор класса Pentium с тактовой частотой 200 МГц и выше, объем оперативного запоминающего устройства (ОЗУ) 32Мб, жесткий диск типа «винчестер» емкостью не менее 1,7 Гб (стандартные значения поставляемых компонент).

4.5. Особенности реализации системного и прикладного программного обеспечения

Программное обеспечение интегрированной АС и всех подсистем, входящих в ее состав, состоит из общего и специального программного обеспечения. Общее программное обеспечение включает в себя универсальные программные компоненты:

- операционная система серверов;
- операционные системы автономных и клиентских АРМ;
- системы управления базами данных;
- программные средства просмотра и подготовки документов;
- средства электронной почты;
- средства управления системой;
- средства организации Web-серверов;
- средства подготовки графических и мультимедийных материалов
- инструментальные средства, необходимые для поддержки и дальнейшего развития автоматизированной системы.

Несмотря на большое количество коммерческих программных систем общего назначения и проблемно-ориентированных инструментальных средств, разработка специального программного обеспечения для решения конкретных производственных задач остается актуальной и наиболее трудоемкой задачей реализации проекта (если не учитывать относительно редкие случаи разработки нестандартной аппаратуры для отдельных измерений в АСУ ТП). Специальное программное обеспечение включает:

- программные модули реализации функций АРМ основных должностных лиц в АС;

- программы сбора оперативной информации и управления технологическими процессами;
- программы обеспечения документооборота в системе, в частности, в подсистемах административного управления, бухгалтерского учета и оперативного управления производством;
- сервисные программы диагностики и восстановления системы в случае возникновения аварийных ситуаций.

Специальное программное обеспечение АС функционирует на основе и совместно с общесистемным программным обеспечением, реализуя функции, которые не могут быть эффективно выполнены с помощью последнего.

Выбор системного программного обеспечения относительно невелик и определяется в основном традициями и опытом работы разработчика и наличием лицензий на использование той или иной операционной системы. Наибольшее распространение получили в настоящее время продукты фирмы Microsoft (США). Сейчас проблема выбора сетевой операционной системы фактически свелась к дилемме: NetWare фирмы Novell или Windows NT фирмы Microsoft (Назаров, Администрирование), причем все чаще предпочтение отдается Windows NT. Для сетей очень крупного масштаба могут использоваться серверы, работающие под операционной системой Unix и ее вариантами. Причина успеха Windows NT кроется в эффективных и постоянно прогрессирующих решениях и продуманной маркетинговой политике разработчика. В Windows NT осуществлена тесная связь с клиентскими операционными системами, она интегрируется в сети с NetWare, Unix, Pathwork, IBM Lan Server, IBM SNA, AS/400, VINES, т.е. практически со всеми распространенными сетевыми средами. На основе комбинации с мощным пакетом BackOffice этой же фирмы можно разрабатывать достаточно универсальное ядро информационной системы, в которое легко интегрируется специальное программное обеспечение. Windows NT Server устойчиво работает на многопроцессорных моделях серверов.

Особенно ценным для системы документооборота является возможность применения в среде Windows NT пакета GroupWise, который изолирует пользователей от файловой системы и позволяет им работать непосредственно

с документами вне зависимости от того, в каком пункте сети и в каком файле они хранятся.

Архитектура клиент-сервер, в которой различные ресурсы с помощью вычислительной сети могут использоваться многими удаленными пользователями для решения своих задач, позволяет осуществлять реализацию самых разнообразных вариантов распределения прикладного программного обеспечения между клиентами и серверами. Создание распределенных приложений сложнее создания автономных [15]: требуется решать множество дополнительных задач – на сколько частей разбить приложение, какие функции возложить на каждую часть, как организовать взаимодействие этих частей и т.д. Однако будущее принадлежит распределенным приложениям.

Создание распределенных приложений для интегрированных АС предприятия рекомендуется вести с использованием концепции многослойной архитектуры клиент-сервер. Все программное обеспечение АС разделяется на клиентскую и серверную части. Клиентская часть обеспечивает средства взаимодействия АС с пользователями и устройствами связи с объектами. Серверная – организует централизованное хранение данных, поступающих из клиентских компонент, ведет централизованную обработку запросов клиентов и выдает результаты запросов клиентам, а также обеспечивает целостность и непротиворечивость данных в соответствии с требованиями предметной области. В трехслойной архитектуре вводится дополнительный серверный слой (уровень), осуществляющий централизованную обработку данных при решении классов типовых задач, что разгружает верхний, центральный серверный уровень хранения данных и делает систему более гибкой и развиваемой. Количество серверных слоев может быть увеличено.

Концепция трехслойной архитектуры Windows DNA (Distributed interNet Application) фирмы Microsoft [18] схематично изображена на рис. 21.

В концепции выделены три основных слоя: представления данных, централизованной обработки запросов клиентов (бизнес-логики) и слой данных. Слой представления данных отвечает за взаимодействие с пользователем и устройством связи с объектом; слой данных отвечает за хранение, доступ и защиту данных, разделяемых приложениями (подсистемами АС).

Вводить промежуточный, средний, слой централизованной обработки запросов клиентов имеет смысл, если удастся выделить процессы решения

задач, результаты которых нужны многим, в том числе территориально удаленным клиентам. Обычно эти процессы реализуются в виде набора так называемых COM объектов (от слова «components», «компоненты»), работающих под управлением программного обеспечения сервера промежуточного слоя (например, Microsoft Transaction Server 2.0 (MTS-сервер, Component Manager)).

COM объекты пишутся на языках Visual C++, Visual Basic и других, поддерживаемых средой разработки Visual Studio фирмы Microsoft. Однако реализация трехслойной системы не означает полного отказа от хранимых процедур на самом нижнем и на самом верхнем слое. В некоторых случаях более эффективно реализовывать работу с данными либо непосредственно на клиенте (средний, толстый клиент), либо на самом верхнем уровне, избегая дополнительных межпроцессных пересылок.

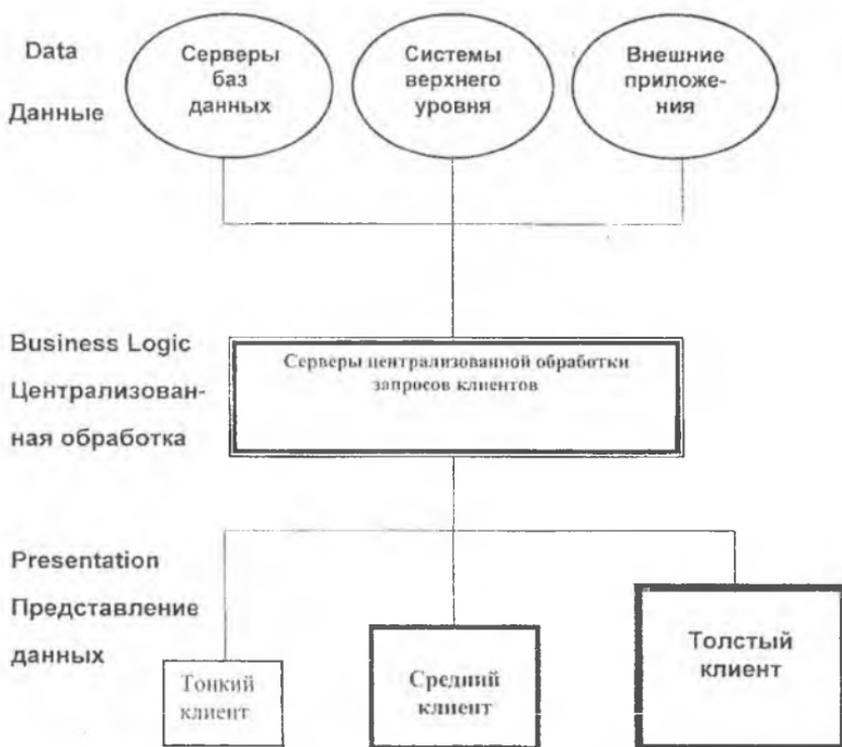


Рис. 21. Концепция трехслойной архитектуры фирмы Microsoft

Отсутствие среднего слоя ведет к вариантам построения двухслойных (двухуровневых) схем. В них нагрузка на сервер существенно возрастает, так как даже при отсутствии запроса для поддержки соединения с клиентом на сервере должен образовываться отдельный процесс с выделением ресурсов. Кроме того, лицензионное соглашение ограничивает количество одновременно открытых соединений с сервером базы данных, что может привести в двухслойной архитектуре к покупке более дорогой лицензии. В трехслойной схеме легче расширять функциональные возможности системы, достаточно только добавлять новые COM объекты с минимальными изменениями в программном обеспечении двух других слоев.

Взаимодействие с компонентами среднего слоя обеспечивается посредством распределенной модели COM объектов DCOM (Distributed Component Object Model). Она позволяет связать между собой объекты, реализованные на различных языках и в различных средах [19]. Запросно-ответная схема взаимодействия для тонкого клиента требует высокой пропускной способности линий связи (100Мбит/с и выше), поэтому для автоматизации локальных подразделений и решения автономных задач лучше использовать варианты со средним и толстым клиентами, самостоятельно реализующими свои функции обработки.

Для удаленных пользователей целесообразно организовать дополнительный четвертый серверный слой, связывающий тонкого клиента со вторым слоем централизованной серверной обработки данных через сеть Internet. Это – сервер Internet-приложений, ведущий прием первичных оперативных данных и запросов тонких удаленных клиентов, обработку данных и пересылку результата в виде HTML-страниц. Компонентами четвертого слоя являются ASP-страницы (Active Server Pages), работающие под управлением сервера Internet-приложений, например, Microsoft Internet Information Server 5.0. Удаленные тонкие клиенты должны быть снабжены каким-либо Web-браузером и средствами подготовки HTML-страниц, например, Microsoft Internet Explorer 4.0 и выше. Преобразование HTML-страниц в ASP-страницы осуществляется элементарно путем изменения расширения имени файла. Но в него теперь можно вставлять команды серверу на специальном скриптовом языке, интерпретируемом компонентой четвертого слоя ActiveX Server Scripting Engine. Четвертый слой может реализовывать программы первичной обработки и через

DCOM-интерфейс обращаться к COM объектам второго слоя для более сложной обработки и доступа к центральному серверу базы данных. Кроме того, для передачи данных от удаленных клиентов могут использоваться средства электронной почты, например, Microsoft Exchange Server 2000.

Программное обеспечение реализуется методами объектно-ориентированного программирования [20]. Все компоненты подразделяются на визуальные (формы, меню, элементы управления и т.п.), видимые на экране, и невидимые (все остальные). Компоненты собираются из объектов, которые принадлежат определенным классам и наследуют свойства классов. Процессы реализуются методами классов и объектов. Объектно-ориентированные среды программирования предлагают обширный набор заготовок (классов) типовых визуальных и невидимых объектов, а также средства для создания новых классов и объектов с различными свойствами и методами. Широко используется технология ActiveX компонент с унифицированными интерфейсами и сгруппированных в библиотеки для использования различными системами программирования.

Для создания программной реализации невидимых компонентов в АС на клиентском уровне определим класс ядра приложения. Данный класс является центральным ядром любой задачи. Он содержит все системные процедуры и функции, которые могут понадобиться при разработке задач: получение нового идентификатора, ограничения пользователя на данные, проверка прав пользователей и др. Через этот класс происходит взаимодействие клиента с сервером баз данных через COM объекты среднего слоя обработки. Если при настройке системы и ее функционировании широко используются справочники, полезно определить класс универсального справочника со свойствами и методами, позволяющими закрепить справочник за конкретным приложением, обратиться к серверу баз данных за конкретной строкой справочника, сделать выборку из этой строки и использовать ее для решения какой-либо задачи.

На базе свойств и методов класса универсального справочника могут быть построены классы специальных справочников со своими специфическими свойствами и методами, например, справочник сотрудников, справочник оборудования, справочник по плану счетов и т.д.

В библиотеку не визуальных классов клиентского уровня включаются также вспомогательные классы объектов работы с наборами данных, работы с калькулятором, отображения прогресса выполнения операции и др.

В библиотеку визуальных компонентов клиентского уровня входят классы навигационных линеек; объектов ввода данных с визуальным отображением, объектов выбора вида, предварительного просмотра и печати отчетов; объектов отображения, ввода и редактирования документов и др.

Средний слой представляет собой реализацию типовых операций обработки данных, хранящихся и выполняемых сервером централизованной обработки запросов в виде компонентов. Взаимодействие компонентов среднего слоя с сервером баз данных осуществляется через объекты библиотеки ADO DB (Microsoft ActiveX Data Object). Для обращения к серверу базы данных используется специальный объект. Отдельный объект служит для создания SQL-запросов с параметрами; другой объект хранит результат выборок с сервера базы данных, он имеет все методы, необходимые для просмотра и редактирования записей. Все компоненты созданы как COM объекты, являются классами и хранятся в одной библиотеке ActiveX DLL, работающей под управлением сервера промежуточного слоя MTS.

При вызове клиентом какого-либо метода компонента среднего уровня на сервере MTS создается автоматически экземпляр соответствующего COM объекта, метод исполняется и все внутренние вызовы происходят в пределах одной родительской транзакции. Процессы обработки данных на среднем уровне называются *бизнес-процессами*, для работы с ними служат методы специального класса среднего уровня работы с процессами : проверка состояния бизнес-процесса, проверка прав на выполнение бизнес-процесса, открытие, закрытие бизнес-процесса(отметка выполнения).

В качестве программного обеспечения сервера баз данных верхнего уровня наиболее часто используются СУБД: Oracle, Microsoft SQL Server, Informix, Sybase, IBM DB/2. Более предпочтительным является использование Microsoft SQL Server 7.0 и выше, обеспечивающего тесную интеграцию с операционной системой Windows NT и другими приложениями пакета BackOffice. Кроме того, им можно управлять программами, написанными на любом языке программирования, поддерживающими технологию ActiveX Server Components,

более известную как OLE Automation-серверы. В частности, к ним относятся Visual Basic, Visual C++, Visual FoxPro. ActiveX Server компоненты располагаются на сервере, к пользователям не пересылаются и на них не накладывается особых ограничений по вводу-выводу и пересылке. Их библиотека непрерывно пополняется независимыми разработчиками и позволяет реализовать многие типовые функции работы с серверными данными в режиме разделения между многими пользователями (клиентами и серверами промежуточных слоев). Одними из важнейших являются компоненты доступа к источникам данных ActiveX Data Object (ADO), на базе которых осуществляется взаимодействие всех слоев системы. Некоторым недостатком Microsoft SQL Server является его жесткая привязка к операционной системе Windows NT. Впрочем, этот недостаток компенсируется возможностью тиражирования и обмена данными со всеми вышеперечисленными СУБД при наличии соответствующего ODBC-драйвера. Язык программирования и запросов Transact-SQL отвечает входному уровню международного стандарта ANSI SQL-92 и постепенно доводится до среднего уровня этого стандарта. Из кода этого языка могут вызываться ActiveX Server компоненты, которые исполняются как хранимые процедуры.

В Microsoft SQL Server имеется встроенная поддержка обмена данными по сети Internet. Специальная утилита SQL Server Web Assistant позволяет интерактивно или программно определить набор данных, передаваемых в HTML-документ на Web-сервер. Данные могут передаваться в запланированные моменты времени или в моменты действительного изменения данных.

Компания Wind River Systems в 2000 г. предложила первое интегрированное решение по применению приложений реального времени в рамках стандартной модели DCOM, упоминавшейся выше [21]. Технология VxDCOM позволяет передавать данные из задачи реального времени в приложение, работающее на стандартном персональном компьютере под управлением операционной системы Windows NT. Возможность включения в стандартную DCOM-модель новых встраиваемых систем реального времени существенно расширяет сферу применения распределенных систем управления и позволяет использовать уже наработанные программные компоненты, созданные в рамках стандарта DCOM.

4.6. *Документирование проекта системы*

Все этапы создания АСОИУ (раздел 3.1) должны быть документированы. Основными группами отечественных стандартов, в соответствии с которыми происходит документирование проекта, являются:

- Информационная технология. Комплекс стандартов и руководящих материалов на автоматизированные системы, группа ГОСТ 34.XXX;
- Единая система конструкторской документации (ЕСКД), группа ГОСТ 2.XXX;
- Единая система программной документации (ЕСПД), группа ГОСТ 19.XXX.

Основополагающей является первая группа стандартов, она ссылается на две остальные группы и постепенно заменяет частично действующие стандарты на автоматизированные системы управления (АСУ) группы ГОСТ 24.XXX. Кроме того, могут использоваться действующие комплексы стандартов СРПП (*Система разработки и постановки продукции на производство*), УСД (*Унифицированная система документации*), классификаторы и кодификаторы технико-экономической информации. Все это вместе называется *нормативно-технической документацией (НТД)*.

ГОСТ 34.201-89 устанавливает виды, наименование, комплектность и обозначение документов, разрабатываемых на различных стадиях создания АС. Различают следующие виды документов :

- ведомость (В) – перечисление в систематизированном виде объектов, предметов и т.д.;
- схема (С) – графическое изображение форм документов, частей, элементов системы и связей между ними в виде условных обозначений;
- инструкция (И) – изложение состава действий и правил их выполнения персоналом;
- обоснование (Б) – изложение сведений, подтверждающих целесообразность принимаемых решений;
- описание (П) – пояснение назначения системы, ее частей, принципов их действия и условий применения;
- конструкторский документ – в соответствии с ГОСТ 2.102 и по ГОСТ 2.601 в части эксплуатационных документов;

- программный документ – в соответствии с ГОСТ 19.101.

В скобках указаны коды видов документов, используемые в условных обозначениях документов.

Результаты первых двух стадий (Формирование требований к АС, Разработка концепции АС) оформляются в виде отчетов с соблюдением требований ГОСТа к оформлению научно-технических отчетов. Содержание отчетов обсуждалось в разделе 3.1 .

На следующей стадии разрабатывается «Техническое задание (ТЗ) на создание автоматизированной системы» в соответствии с требованиями ГОСТ 34.602.

ТЗ на АС должно содержать следующие разделы, которые могут быть подразделены на подразделы:

- 1) общие сведения;
- 2) назначение и цели создания (развития) системы;
- 3) характеристики объектов автоматизации;
- 4) требования к системе;
- 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.

В ТЗ на АС могут включаться приложения.

Раздел «Требования к системе» состоит из следующих подразделов:

- 1) требования к системе в целом (к структуре и функционированию системы; к численности и квалификации персонала системы и режиму его работы; показатели назначения, надежности, безопасности; требования к эргономике и технической эстетике; к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы; к защите информации от несанкционированного доступа; требования по сохранности информации при авариях; к защите от влияния внешних воздействий; к патентной чистоте; требования по стандартизации и унификации; дополнительные требования),
- 2) требования к функциям (задачам), выполняемым системой,
- 3) требования к видам обеспечения.

В стандарте подробно описаны содержание каждого раздела и подраздела ТЗ, правила его оформления, согласования и утверждения.

На стадиях эскизного и технического проектирования на АС в целом и отдельные виды обеспечения разрабатывается до 30 видов документов. Поскольку стадия эскизного проектирования не является обязательной и в большинстве случаев пропускается, кратко рассмотрим состав документации технического проекта (ТП). Полный состав документации перечисляется в конструкторском документе - ведомости технического проекта (код «ТП»). В целом к техническому проекту должна быть составлена пояснительная записка (код «П2»). Она должна включать план мероприятий по подготовке объекта к вводу системы в эксплуатацию. Проектные решения подробно излагаются в описаниях:

- автоматизируемых функций (П3);
- постановки задач (комплекса задач) (П4) – можно включить в документы П2 или П3;
- информационного обеспечения системы (П5);
- организации информационной базы (П6);
- систем классификации и кодирования (П7);
- массива информации (П8);
- комплекса технических средств (П9);
- программного обеспечения (ПА);
- алгоритма (проектной процедуры) (ПБ) – можно включить в документы П2, П3 или П4;
- организационной структуры (ПВ).

На стадии технического проекта разрабатываются схема структурного комплекса технических средств, схема организационной структуры, схема функциональной структуры, схема автоматизации, план расположения и производится локальный сметный расчет. Дополнительно создаются технические задания на разработку новых технических средств и проведение электротехнических и строительных работ. В технический проект они включаются только в виде списка и утверждаются отдельно. Важную роль играют различные ведомости и перечни (покупных изделий, входных сигналов и данных, выходных сигналов (документов), оборудования и материалов).

На стадии «Рабочая документация» разрабатывается полный комплект документов, содержащих сведения, достаточные для выполнения работ по

вводу АС в действие и ее эксплуатации, а также для поддержания уровня эксплуатационных характеристик (качества) системы в соответствии с принятыми проектными решениями.

Полный перечень документов приводится в конструкторском документе «Ведомость держателей подлинников» (код «ДП»). Формируется полный комплект эксплуатационной документации, перечень которой приводится в конструкторском документе «Ведомость эксплуатационных документов» (код «ЭД»). Рабочая документация включает ведомости оборудования, потребности в материалах, а также схемы и таблицы деления системы, соединения внешних проводов, подключения внешних проводов, внутренних соединений и подключений, схему принципиальную и схему структурную комплекса технических средств. Разрабатываются локальная смета и чертежи общего вида, установки технических средств, план расположения оборудования и проводов. Для проведения испытаний создается программа и методика испытаний компонентов, подсистем и системы в целом. Все остальные документы включаются в *комплект эксплуатационной документации*, а именно:

- чертеж формы документа (видеокадра) (С9);
- ведомость машинных носителей информации (ВМ);
- массив входных данных (В6);
- каталог базы данных (В7);
- состав выходных данных (сообщений) (В8);
- методика (технология) автоматизированного проектирования (И1);
- технологическая инструкция (И2);
- руководство пользователя (И3);
- инструкция по формированию и ведению базы данных (набора данных) (И4);
- инструкция по эксплуатации КТС (ИЭ);
- описание технологического процесса обработки данных (включая телеобработку) (ПГ);
- общее описание системы (ПД);
- формуляр (ФО);
- паспорт (ПС).

Требования к содержанию документов изложены в руководящем документе РД 50-34.698-90.

Поскольку одним из основных эксплуатационных документов является «Руководство пользователя», рассмотрим его содержание подробнее.

Документ содержит 6 разделов. Во *введении* указываются область применения системы, краткое описание возможностей, требуемый уровень подготовки пользователя, документация, с которой он должен ознакомиться. В разделе *«Назначение и условия применения»* описываются автоматизируемые функции и виды деятельности; условия нормального функционирования системы (например, вид ЭВМ и конфигурация технических средств, операционная среда и общесистемные программные средства, входная информация, носители данных, база данных и т.п.). В разделе *«Подготовка к работе»* указывается состав и содержание дистрибутивного носителя данных; порядок загрузки данных и программ; порядок проверки работоспособности системы. Следующий раздел – *«Описание операций»* – содержит описание всех выполняемых функций, задач, комплексов задач, процедур с полным описанием технологических операций обработки данных. Для каждой операции указываются наименование, условия выполнения, подготовительные действия, основные и заключительные действия, ресурсы, расходуемые на операцию. В описании действий допускаются ссылки на файлы подсказок, размещенные на магнитных носителях. Действия иллюстрируются соответствующими экранными и документальными формами. В разделе *«Аварийные ситуации»* приводятся действия пользователя при возникновении отказов технических средств, по восстановлению программ и данных, при сбоях и ошибках в процессе функционирования системы, при различных возможных аварийных ситуациях. Последний раздел *«Рекомендации по освоению»* должен содержать рекомендации по освоению и эксплуатации, включая описание контрольного примера, правила его запуска и выполнения.

На стадии «Ввод в действие» осуществляется подготовка объекта автоматизации и персонала, комплектация АС поставляемыми техническими, программными и информационными изделиями. Производятся строительно-монтажные и пусконаладочные работы. После их завершения проводятся предварительные испытания в соответствии с программой и методикой испытаний и система вводится в стадию опытной эксплуатации. По окончании опытной эксплуатации устраняются замеченные недостатки и проводятся

приемочные испытания. Система вводится в постоянную эксплуатацию и сопровождается гарантийным и послегарантийным обслуживанием и сопровождением со стороны изготовителя системы.

На стадии «Ввод в действие» разрабатываются организационно-распорядительные документы (план-график работ, акт завершения работ, акт приемки в опытную эксплуатацию, акт приемки в промышленную эксплуатацию, приказ о составе приемочной комиссии, протоколы предварительных и приемочных испытаний).

Список рекомендуемой литературы

1. Калянов Г.Н. CASE. Структурный системный анализ (автоматизация и применение). — М.: ЛОРИ, 1996.
2. Вендров А.М. CASE - технологии. Современные методы и средства проектирования информационных систем. — М.: Финансы и статистика, 1998.
3. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. — М.: Финансы и статистика, 2000.
4. CASE. Аналитик. Версия 1.1. Руководство аналитика. — М.: Научно-техническое предприятие ЭЙТЕКС, 1992.
5. Бутко В.Р., Дерябкин В.П. CASE-технологии моделирования и проектирования информационных систем: Учебное пособие. - Самара: СГЭА, 2001.
6. Гэйн К., Сарсон Т. Структурный системный анализ: средства и методы. В двух частях / Пер. с англ.; Под ред. Козлинского А.В. — М.: Научно-техническое предприятие ЭЙТЕКС, 1993.
7. CASE. Аналитик. Версия 1.1. Руководство пользователя. — М.: Научно-техническое предприятие ЭЙТЕКС, 1993.
8. Маклаков С.В. Vpwin и Erwin. CASE-средства разработки информационных систем. — М.: Диалог-МИФИ, 1999.
9. Соучек Б. Мини-ЭВМ в системах обработки информации / Пер. с англ.; Под ред. Доброва Е.В. — М.: Мир, 1976.
10. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ / Пер. с англ. — М.: Мир, 1991.
11. Баландин А.В., Сидоров А.А. Основы построения баз данных: Учебное пособие. — Самара: Университет Наяновой, 1996.

12. Корнеев В.В. и др. Базы данных. Интеллектуальная обработка информации — М.: Нолидж, 2000.
13. Чен П. Модель “сущность-связь” – шаг к единому представлению о данных // Системы Управления Базами Данных. № 3.1995. С. 137-158.
14. АСУ на промышленном предприятии: Методы создания: Справочник, 2-е изд. / Михалев С.Б. и др. — М.: Энергоатомиздат, 1989.
15. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник. — СПб: Питер, 2000.
16. Программный комплекс “Галактика”. Рекламные материалы. — М.: Корпорация “Галактика”, 1997.
17. Краткий обзор системы автоматизации ПРОСКОН 2100. 710 0135-4VR \ Рекламный проспект OUTOKUMPU MINTEC 710 0135-4VE / P13.
18. Microsoft Corporation. MSDN Online / Публикация Internet, <http://www.microsoft.ru/msdn/windna>, 2001.
19. Microsoft Corporation. Разработка распределенных приложений на Visual Ba 6.0 Учебный курс: Официальное пособие Microsoft для самостоятельной подготовки / Пер. с англ. — М.: Издательско-торговый дом “Русская редакция”, 2000.
20. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд. / Пер. с англ. — М.: Бином; СПб: Невский диалект, 1999.
21. Кузнецов С. Applied Computing – новая стратегия Intel \ PCWeek, № 9(231),: 27 марта 2000 г., с. 27-28.
22. Вальков В.М., Вершин В.Е. Автоматизированные системы управления технологическими процессами. 3-е изд. — Л.: Политехника, 1991.

Учебное издание

Дерябкин Валентин Павлович

**ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННЫХ СИСТЕМ
ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

Курс лекций для студентов заочной формы обучения

Редактор Т. К. К р е т и н и н а

Лицензия ЛР №020301 от 30.12.96 г.

Подписано в печать 10.10.2001 г. Формат 60×84 1/16.

Бумага газетная. Печать офсетная.

Усл. печ. л. 7,20. Усл. кр.-отт. 7,32. Уч.-изд. л. 7,75.

Тираж 100 экз. Заказ

Самарский государственный аэрокосмический
университет имени академика С.П. Королева
443086 Самара, Московское шоссе, 34.

Отпечатано в УПЦ

443001 Самара, ул. Молодогвардейская, 151.