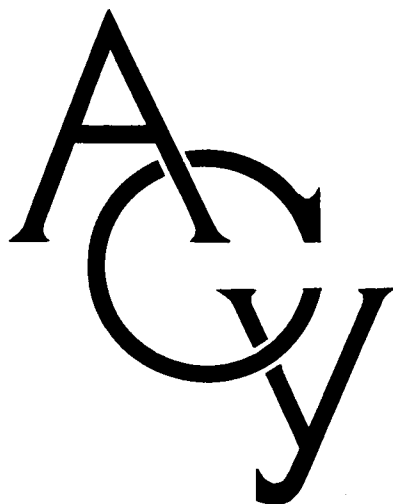


А. Г. МАМИКОНОВ

Проектирование



Допущено
Министерством высшего и среднего
специального образования СССР
в качестве учебника
для студентов вузов,
обучающихся по специальности
"Автоматизированные системы управления"



Москва «Высшая школа» 1987

ББК 65.050.9(2) 2
М22
УДК 658.5.011.56

Рецензенты:

кафедра автоматизированных систем управления Московского института нефти и газа им. И.М. Губкина (заведующий кафедрой проф. Л.А. Овчаров);
проф. Б.Я. Советов (Ленинградский электротехнический институт им. В.И. Ульянова (Ленина)).

Мамиконов А.Г.

М22 Проектирование АСУ: Учебник для спец. "АСУ" вузов. — М.:
Высш. шк., 1987. — 303 с.: ил.

Рассмотрены вопросы проектирования автоматизированных систем управления предприятиями, объединениями и отраслями. Описаны основные стадии создания АСУ, проектирование систем на макро- и микроуровнях, проектирование функциональных подсистем. Изложены вопросы проектирования информационного, программного и технического обеспечения, построения имитационных моделей. Рассмотрены психологические аспекты проектирования. Даны основные сведения о технической документации при проектировании АСУ.

М 0604020101(4309000000) — 335 6—87
001(01)—87

ББК 65.050.9(2) 2
6Ф6.5

© Издательство "Высшая школа", 1987

Задача учебного курса "Проектирование АСУ", по которому выпущен данный учебник, заключается в том, чтобы на основе предшествующих курсов учебного плана дать студентам завершающие знания в области современных научных и практических методов и моделей управления сложными организационными системами, проектирования и функционирования АСУ. По окончании изучения курса студенты должны уметь проектировать и эксплуатировать АСУ в самых различных сферах человеческой деятельности, владеть соответствующими навыками.

В силу специфики и сложности АСУ процесс проектирования во многом пока не формализован. Умение правильно поставить задачу, выявить и корректно записать критерии и ограничения приходит с опытом. Существующие ГОСТы и методические материалы определяют организационные вопросы и регламентируют состав и содержание проектной документации, но не содержат рекомендаций и указаний, раскрывающих сущность процесса создания АСУ.

Это предопределило значительные трудности подготовки данного учебника, содержание которого базируется как на освещении отдельных вопросов курса в отечественной и зарубежной литературе, так и на опыте автора по созданию научных основ проектирования АСУ, практических разработок АСУ различного назначения и чтения соответствующих лекций в высшей школе.

В первой главе рассмотрены основные стадии создания АСУ. Официальные материалы определяют эти стадии аналогично традиционным этапам проектирования и конструирования технических систем: техническое задание, техническое и рабочее проектирование, ввод в эксплуатацию. Однако специфика АСУ, заключающаяся в эволюционном, постепенном переходе от существующей системы к автоматизированной, приводит к тому, что в ряде случаев выделение упомянутых выше стадий является в известной мере искусственным. В частности, в них важнейшая для АСУ стадия сопровождения и развития системы, непрерывной ее адаптации к изменяющимся условиям управления, модернизации и совершенствования разработчиками, вовсе не предусмотрена. Поэтому наряду с изложением официальных материалов в этой главе приводятся сведения о некоторых неформальных этапах разработки АСУ, об особенностях создания систем, не имеющих аналогов.

Вторая глава посвящена внешнему проектированию системы, вопросам взаимодействия с внешней средой и определения эффективности. Здесь предпринята попытка заполнить пробел в литературных и методических материалах, относящийся к постановке задачи, умению понять ее содержание, сформулировать на описательном уровне и затем перейти к формальной записи. Это один из важнейших моментов, в котором сочетаются искусство и инженерный подход, характеризующий квалификацию разработчика.

В четырех последующих главах рассмотрены вопросы проектирования функциональных подсистем, информационного, математического и технического обеспечения. В силу ограниченности объема рассмотрены не все функциональные подсистемы и тем более не все содержащиеся в них задачи. Тем не менее разнообразие рассмотренных задач и подходов к методам их решения позволяет надеяться, что читатель получит основные сведения, необходимые разработчику, о характере задач, построении моделей и практической реализации их решения.

Развитие математических методов, постоянное появление новых алгоритмов оправдывают такую направленность на изучение методологии проектирования и некоторых примеров, а не на детальное изложение большого множества конкретных задач.

Седьмая глава посвящена психологическим аспектам проектирования. Несмотря на их исключительную важность для человеко-машинных систем, в обычной практике им уделяется недостаточно внимания. Здесь выделены два аспекта — учет при создании АСУ психологии руководителя, как лица, принимающего решения, и психологические факторы, влияющие на сам процесс проектирования.

В самостоятельную восьмую главу выделены вопросы имитационного моделирования, имеющие большое значение как при разработке, так и при функционировании АСУ. Материал этой главы ориентирован на практическое применение имитационных моделей.

В последней, девятой главе изложены официальные требования к технической документации на систему, оформляемой в процессе создания АСУ.

Учебник предназначен для студентов вузов, обучающихся по специальности "Автоматизированные системы управления". Его содержание соответствует типовой учебной программе по одноименному курсу. Предполагается, что студенты в соответствии с учебным планом уже изучили курсы "Основы построения АСУ", "Исследование операций", "Моделирование систем", "Базы и банки данных", "Операционные системы" и др. Вместе с тем учебник построен таким образом, чтобы его могли использовать в качестве пособия при проектировании специалисты — разработчики АСУ, не имеющие большого опыта разработки.

При подготовке рукописи большую помощь оказали сотрудники автора, а некоторые разделы написаны практически совместно с ними: с канд. техн. наук А.Б. Шелковым (гл. 4), канд. техн. наук Е.Б. Соколо-

вой (гл. 4, 5), М.П. Шариковой (гл. 5), С.Э. Калугиным (гл. 6). В подборе материалов и оформлении рукописи активно участвовала Т.С. Ефимова. Всем им автор приносит глубокую благодарность.

Особую благодарность автор выражает д-ру техн. наук, проф. Б.Я. Советову, д-ру техн. наук, проф. Л.А. Овчарову и сотрудникам руководимых ими кафедр за внимательное рецензирование рукописи и высказанные при этом замечания, во многом способствовавшие ее улучшению.

Все замечания и пожелания просьба направлять по адресу: 101430, Москва, ГСП-4, Неглинная ул., 29/14, издательство "Высшая школа".

Автор

Состояние советского общества на современном этапе в значительной мере определяется уровнем науки и техники, высоким и стабильным темпом развития социалистической экономики, быстрым увеличением производительности общественного труда, повышением благосостояния и культуры народа.

Принятые XXVII съездом КПСС "Основные направления экономического и социального развития СССР на 1986–1990 годы и на период до 2000 года" указывают, что "... высшей целью экономической стратегии партии был и остается неуклонный подъем материального и культурного уровня жизни народа. Реализация этой цели в предстоящем периоде требует ускорения социально-экономического развития, всемерной интенсификации и повышения эффективности производства на базе научно-технического прогресса"*.

Основой роста материального производства является переход на преимущественно интенсивный путь развития, всемерное ускорение научно-технического прогресса, применение его результатов в производстве и управлении. Развиваются концентрация и специализация производства, осуществляются его техническое перевооружение и реконструкция, намечен рост производительности общественного труда в 2,3–2,5 раза. На более высокую ступень предусматривается поднять социалистическую экономическую интеграцию, развивать специализацию и кооперирование производства, новые эффективные формы сотрудничества.

В этих условиях существенно возрастает сложность задач управления, без решения которых невозможно сколь-нибудь эффективное функционирование производства.

Карл Маркс писал: "Всякий непосредственно общественный труд, осуществляемый в сравнительно крупном масштабе, нуждается в большей или меньшей степени в управлении, которое устанавливает согласованность между индивидуальными работами и выполняет общие функции, возникающие из движения всего производственного организма в отличие от движения его самостоятельных органов". (Маркс К., Энгельс Ф. Соч. 2-е изд. Т. 23. С. 342.)

Партия и правительство постоянно ставят в качестве первоочередных задачи совершенствования управления народным хозяйством, в том чис-

* Материалы XXVII съезда КПСС. – М.: Политиздат, 1986. С. 271.

ле за счет автоматизации. На широкое применение автоматизированной системы плановых расчетов, ее взаимодействие с отраслевыми и ведомственными АСУ нацеливают решения XXVII съезда КПСС.

За последние годы АСУ всех уровней получили в нашей стране широкое распространение и показали высокую эффективность, в особенности АСУ технологическими процессами. Большое значение имеет их развитие для создания комплексно-автоматизированных производств в двенадцатой пятилетке. Вместе с тем многие АСУ, главным образом организационного типа, оказались недостаточно эффективными, а в некоторых случаях просто нежизнеспособными. Основной причиной такого положения чаще всего является игнорирование разработчиками в процессе проектирования некоторых принципов создания АСУ.

Использование вычислительной техники в процессе управления не может являться самоцелью. Созданная на основе квалифицированно выполненного системного анализа АСУ ориентирована на конечный результат — повышение эффективности функционирования производства или иного объекта. Все системные решения — улучшение и автоматизация документооборота, обработка на ЭВМ больших массивов информации, составление различных сводок и отчетов, решение оптимизационных задач и т.п. — имеют смысл только в том случае, если в результате существенно повышается производительность труда работающих, улучшается использование материальных, энергетических и трудовых ресурсов, возрастает количество и качество выпускаемой продукции.

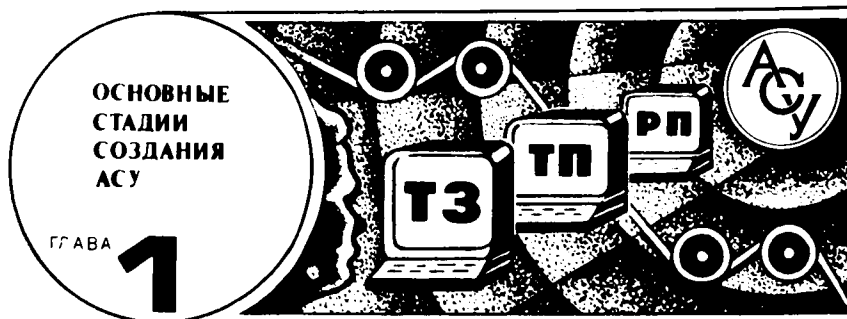
Нельзя забывать, что АСУ является человеко-машинной системой. Очень важно построить хорошую модель, адекватную объекту, создать эффективный алгоритм, написать и отладить быструю программу, не требующую больших вычислительных ресурсов, организовать сбор, хранение и поиск информации и т.д. Но если при этом не будет столь же тщательно учтен человеческий фактор, интересы работающих в системе управления людей, то трудно ожидать успешной эксплуатации такой системы. Во взаимодействии человека и ЭВМ безусловный приоритет следует отдавать человеку. Трудности, если они возникают, должны разрешаться за счет усложнения работы ЭВМ. Нельзя облегчать себе жизнь созданием более простых алгоритмов и программ за счет увеличения объема функций, выполняемых человеком, какими бы легкими они ни казались разработчику.

Разработанные системы должны обладать высокой степенью адаптации к непрерывно меняющимся требованиям пользователя и живучести, т.е. способности работать в условиях отклонений от нормальных ситуаций. Обеспечение таких свойств трудно поддается формализации и, кроме того, требует непрерывного участия инженера-системотехника в процессе анализа функционирования системы и введения необходимой модернизации без перерывов в работе. По существу именно это наряду с наблюдением за тем, насколько успешно выполняет свои функции АСУ, и обеспечением ее работы в соответствии с проектными решениями и является основной обязанностью эксплуатационников-системотехников. Не следует смешивать эти функции с деятельностью людей, эксплуатирующих АСУ

в качестве пользователей — решающих задачи управления на основе созданных программного, информационного, технического и других видов обеспечения АСУ. Таким образом, понятие "эксплуатация АСУ" имеет два четко выраженных аспекта: использование АСУ в качестве инструмента, повышающего эффективность управления, и обеспечение ее работоспособности и развития.

Важнейшим условием успешной работы АСУ является подготовка персонала аппарата управления. Уже в процессе разработки должны осуществляться контакты с сотрудниками организации для правильной постановки задач, разъяснения и обсуждения принимаемых решений, обучения новым методам и формам работы. Сохранение и развитие этих контактов в процессе эксплуатации системы обеспечивает ее своевременную и эффективную адаптацию и модернизацию.

Инженер-системотехник должен обладать разносторонними знаниями, которые он получает при изучении различных курсов учебного плана своей специальности. Осмысление взаимосвязи комплекса специальных курсов, их обобщение и целеустремленное приложение к разработке АСУ, изучение методов проектирования является содержанием предлагаемого вниманию читателя учебника.



Выделение отдельных стадий создания АСУ и определение содержания работ на каждой стадии имеет существенное значение для более четкого планирования, оперативного контроля и управления деятельностью коллектива разработчиков и отдельных исполнителей.

Создание эффективно действующей АСУ немислимо без тесного взаимодействия и сотрудничества разработчиков системы и будущих ее пользователей. На разных стадиях это сотрудничество меняется по содержанию и объему, поэтому для каждой из них важно регламентировать права и обязанности заказчика и разработчика.

Специфика АСУ предопределяет некоторую размытость границ между стадиями, одновременное выполнение работ более ранних и более поздних стадий. В этих условиях регламентация становится особенно необходимой.

Наиболее существенное значение имеет выделение стадий разработки для крупных и не имеющих аналогов систем, где требуется выделение дополнительных стадий и введение различных методов декомпозиции системы для параллельного и взаимосвязанного проектирования отдельных ее частей.

§ 1.1. Регламентация порядка разработки

Разработка новых и развитие действующих АСУ регламентируется общепромышленными руководящими методическими материалами (ОРММ), обязательными в части состава, содержания и порядка выполнения работ для всех организаций, разрабатывающих и внедряющих АСУ.

АСУ является сложной системой, состоящей из взаимосвязанных подсистем функционального и обеспечивающего назначения. Если на предприятии или в производственном объедине-

нии создается специализированная АСУ, например автоматизированного проектирования, и функционирует АСУ предприятием, то должно быть обеспечено их тесное взаимодействие. АСУ предприятием или производственным объединением также должна взаимодействовать с отраслевой АСУ в соответствии с подчиненностью, а также с АСУ тех предприятий и объединений, с которыми существуют производственные или иные связи. Взаимодействие предусматривает прежде всего функциональные связи по решаемым задачам, которые для отраслевых АСУ и АСУ предприятиями реализуются в рамках функциональных подсистем. Задачи, решаемые на уровне предприятия, выдают исходную информацию для решения аналогичных, но обобщенных задач на уровне отрасли. В то же время задачи отраслевого уровня создают управляющую информацию для задач предприятия. В "ручных" системах такие связи реализуются в виде документо-потоков, в наиболее развитых АСУ — по каналам связи между ЭВМ, что требует соответствующих технических средств, лингвистического и программного обеспечения.

АСУ производственным объединением создается, как правило, на базе собственного *вычислительного центра* (ВЦ) с разветвленной сетью терминалов различного типа. В соответствии с ОРММ *разработчиком* АСУ являются специализированные научно-исследовательские и проектные организации. *Заказчиком* таких работ являются производственные объединения и предприятия, для которых создается АСУ. В некоторых случаях предприятие или объединение создает АСУ самостоятельно, используя специальное подразделение ВЦ или вне его. В этом случае оно само получает права, обязанности и несет ответственность как разработчик.

Установлены следующие стадии создания АСУ: *предпроектная*, включающая разработку технико-экономического обоснования (ТЭО) и технического задания (ТЗ) на создание АСУ; *разработки проектов*, включающая разработку технического и рабочего проектов, а для небольших АСУ — единого технорабочего проекта системы; *ввода в эксплуатацию*, включающая проведение монтажных и пусконаладочных работ по технической части системы, завершение мероприятий по подготовке предприятия к внедрению АСУ, опытную эксплуатацию и приемо-сдаточные испытания системы.

Технико-экономическое обоснование является предпроектным документом, подтверждающим экономическую целесообразность и производственную необходимость создания АСУ. На основе сбора и анализа данных, характеризующих возможности повышения объема и качества выпускаемой продукции, снижения материальных, финансовых и трудовых затрат за счет создания АСУ, в ТЭО обосновываются основные решения по функциональным и обеспечивающим частям АСУ, приводится ориентировочный расчет затрат и экономической эффективности создаваемой АСУ. Возможны случаи, когда в процессе подготовки ТЭО выявляется нецелесообразность создания АСУ. Если справедливость такого заключения обоснована и подтверждена компетентными специалистами, дальнейшие работы по созданию АСУ следует прекратить до появления принципиально новых и

(или) более дешевых средств вычислительной техники, расширения объемов производства, усложнения решаемых задач и т.п.

Техническое задание составляют на основании ТЭО. Для больших систем, создаваемых очередями, ТЗ составляется и утверждается на всю систему в целом, с выделением первой очереди системы. При этом допускается формулировать задание на всю систему в обобщенном виде с конкретизацией задач первой очереди, вплоть до составления и утверждения ТЗ на первую очередь, с указанием основных показателей развития системы в целом. На дальнейшее расширение или реконструкцию АСУ составляется и утверждается отдельное техническое задание. Разработку ТЗ на создание АСУ организует предприятие-заказчик с участием организации-разработчика. Фактически в большинстве случаев основная тяжесть разработки ТЗ ложится на сотрудников организации-разработчика, как более квалифицированных и подготовленных в области системного анализа и имеющих соответствующий опыт. Решения, принятые на этапе разработки ТЗ, во многом предопределяют качество и эффективность создаваемой АСУ, что настоятельно требует полноценного участия и взаимодействия наиболее квалифицированных сотрудников как заказчика, так и разработчика. Это должно быть обеспечено во всех случаях, независимо от того, какая организация взяла на себя основную долю разработки ТЗ.

В процессе разработки технического проекта иногда возникает необходимость дополнительно проработать некоторые варианты, не предусмотренные ТЗ, но по предварительным оценкам улучшающие технико-экономические показатели создаваемой АСУ. О такой возможности разработчик сообщает заказчику и в случае необходимости составляется дополнение к техническому заданию, на основании которого разрабатываются новые варианты решения функциональных задач.

В зависимости от конкретных производственно-технических условий, особенностей используемых методов и средств разработки и в соответствии с экономической целесообразностью создание АСУ осуществляется в виде последовательных очередей. На действующих предприятиях создание АСУ должно осуществляться не более чем в две очереди. В первой очереди должно быть предусмотрено регулярное решение задач технической подготовки производства, оперативного управления основным производством, технико-экономического планирования и материально-технического снабжения на уровнях общезаводского и внутрицехового управления. При этом комплекс задач первой очереди должен обеспечивать нормативный срок окупаемости затрат на создание АСУ и нормы загрузки ЭВМ. Для вновь строящихся и реконструируемых предприятий состав и последовательность очередей создания АСУ определяются планами строительства или реконструкции исходя из необходимости обеспечения максимальной эффективности вводимых в действие комплексов задач АСУ. Объем работ первой очереди АСУ устанавливается в соответствии с пусковым комплексом строящихся объектов.

При создании АСУ должны быть обеспечены опережающие разработка

и внедрение классификаторов технико-экономической информации; информационной базы данных; технического и программного обеспечения для подготовки машинных носителей, загрузки и ведения информационной базы; мероприятий по вводу в эксплуатацию зданий и помещений ВЦ, монтажу и эксплуатации комплекса технических средств АСУ.

При разработке АСУ необходимо руководствоваться: действующими законодательными и нормативными актами по вопросам проектирования систем управления промышленными объектами; общепромышленными документами по основным направлениям разработки автоматизированных систем управления; нормативными документами, инструкциями, государственными и отраслевыми стандартами в области создания АСУ; документами по организации и управлению предприятиями соответствующей отрасли промышленности; каталогами типовых прикладных программных средств, вычислительной техники, средств связи и автоматики; установленными нормативами затрат ресурсов на разработку и внедрение АСУ и ценниками для определения сметной стоимости автоматизации.

В процессе разработки и внедрения АСУ создается различная проектная и проектная документация, предусмотренная *государственными стандартами* (см. гл. 9) и оформляемая в соответствии с ними.

Научно-методическое руководство созданием АСУ осуществляет главный конструктор проекта или лицо с теми же функциями, но иным официальным наименованием: главный инженер проекта, руководитель темы и др. Он несет ответственность за качество и сроки выполнения работ по созданию АСУ.

§ 1.2. Предпроектная стадия создания АСУ

Предпроектная стадия включает комплекс научно-исследовательских работ и организационных мероприятий, цель которых — определить целесообразность создания АСУ и, в случае положительного заключения, разработать ТЗ.

Назначение предпроектной стадии — проведение обследования предприятия. Оно направлено на выявление возможностей за счет повышения эффективности управления предприятием на базе использования средств вычислительной техники и современных экономико-математических методов увеличить объем и улучшить качество выпускаемой продукции как при существующих производственных мощностях, так и при их предполагаемом развитии, с одновременным снижением удельных затрат материальных, энергетических и трудовых ресурсов на производство.

Важнейшие условия начала работ по созданию АСУ — заинтересованность руководящего состава предприятия, в первую очередь директора; понимание трудности или невозможности дальнейшего управления предприятием "ручными" методами, в особенности при его развитии; готовность сотрудничать с разработчиками при преодолении неизбежных труд-

ностей в процессе создания АСУ, особенно при освоении новых комплексов задач.

Обследование проводится по специальной программе, представляющей собой перечень вопросов, ответы на которые достаточно полно характеризуют производственно-хозяйственную деятельность предприятия и позволяют определить основные параметры будущей АСУ.

Должна быть проведена оценка уровня подготовленности предприятия к созданию АСУ — соответствие используемых в основном производстве технических средств и технологии современному уровню; наличие налаженного нормативного хозяйства, четкой системы технической подготовки производства, классификаторов и шифраторов и уровня общей культуры производства.

Полученные в результате обследования данные служат основой для разработки рекомендаций по совершенствованию организационной и функциональной структуры существующей системы управления; определения состава подсистем и комплексов задач в создаваемой АСУ и очередности их разработки; составления ТЗ на создание АСУ.

Обследование функциональной структуры заключается в выявлении перечня, содержания и периодичности выполнения функций управления на уровне заводских и внутрицеховых подразделений, по каждому подразделению и отдельным должностным лицам. По каждой функции управления определяют количество работников, участвующих в ее реализации, их образование, должностные оклады, затрачиваемое ими время; наличие и содержание формальных методов решения задач, связанных с выполнением этой функции; применяемые технические средства; должностные инструкции и положения, регламентирующие ход выполнения функции управления должностными лицами.

Обследование материальных потоков позволяет выявить характер и особенности производства, оказывающие влияние на требования к созданию эффективной АСУ. При этом определяется характер производственных операций в каждом цехе и на основных его участках; какие материалы, детали и узлы и в каком количестве получает и сдает каждое подразделение; точки учета движения промежуточной и готовой продукции; средства ее транспортировки, средства и формы складирования; периодичность и сроки сдачи продукции.

Обследование информационных потоков состоит в уточнении применяемой терминологии для обеспечения взаимопонимания персонала разработчика и заказчика; выявлении используемых на предприятии внешних и внутренних документов и составлении схемы документооборота; определении информационных связей предприятия и его подразделений. Схема документооборота составляется на таком уровне детализации, чтобы она была обозримой и в то же время достаточно информативной для определения требований к ее совершенствованию и автоматизации. При необходимости составляется несколько схем документооборота — укрупненная общая для предприятия и дополнительные на уровне цехов и отделов.

Количественные характеристики информационных потоков позволяют определить объемы поступающей, обрабатываемой и выдаваемой информации подразделениями и предприятием в целом, а также изменения этих объемов во времени.

Изучение производства, методов планирования и учета проводится с целью определения требований к последним в условиях функционирования АСУ. Определяется тип основного производства — массовый, крупносерийный, серийный или индивидуальный, а также режим работы по числу дней в неделю и количеству смен. Выявляется номенклатура готовых изделий, трудоемкость, себестоимость и оптовая цена каждого из них, а также количество наименований и число входящих в каждое изделие деталей и узлов. Кроме того, определяется продукция, поставляемая по кооперации и в виде запасных частей. Фиксируется производственная программа в виде плана выпуска на текущий, прошлый и планируемый годы по номенклатуре, запасным частям и поставкам по кооперации, а также на весь товарный выпуск.

Характеристика предприятия включает также описание его структуры, представляемой в виде схемы, на которой указаны все подразделения с указанием административной подчиненности. При необходимости схема структуры сопровождается пояснительной запиской. Для сборочных цехов и участков составляется схема поступления на сборку узлов, деталей и комплектующих изделий из других цехов и со складов, с указанием *откуда, что и в каком количестве* поступает. Определяется загрузка сборочных цехов — темп выпуска и номенклатура изделий, одновременно собираемых на каждом сборочном конвейере. В характеристику предприятия входит также состав работающих в целом по заводу и с разбивкой по цехам. По каждой категории работающих приводится их число, для рабочих — с разбивкой на основное и вспомогательное производство. Приводится состав производственного оборудования — общее количество по цехам с выделением уникального оборудования. В заключительном разделе характеристики предприятия дается перспектива его развития на ближайшие пять лет — ввод новых производственных мощностей, существенные изменения технологий производства, освоение новых изделий.

Важной частью предпроектной стадии создания АСУ является анализ предприятия, направленный на выявление общих тенденций и факторов развития производства; целей и критериев эффективности функционирования и развития производства; факторов, препятствующих и способствующих достижению целей; общих и специфических характеристик развития предприятия, его роли и места в общем комплексе деятельности отрасли и в народном хозяйстве. Наиболее важным и наименее формализуемым является определение целей функционирования предприятия и критериев эффективности управления им. Для этого целесообразно использовать уже сформулированные цели и критерии вышестоящей системы управления, т.е. для предприятия — производственного объединения, а для последнего — отрасли в целом. Если такие цели и критерии ранее

не были сформулированы, следует самим разработчикам сделать, по крайней мере, обоснованную прикидку. Используя экспертные методы построения дерева целей, доходят до уровня исследуемого предприятия. Экспертами, оценивающими предлагаемые разработчиками формулировки, являются обычно сотрудники руководящего состава изучаемой и вышестоящей организации. Далее выявляются факторы, способствующие достижению цели. Фактически ими являются подцели, определяемые как следующий уровень ветвления дерева цели. Для промышленного предприятия, если целью является увеличение объема и повышение качества выпускаемой продукции, такими факторами могут быть совершенствование методов планирования и оперативного управления; улучшение использования оборудования и производственных мощностей; совершенствование производственной структуры; улучшение использования материалов; повышение квалификации персонала и совершенствование баланса рабочей силы и т.д.

Перечисленные факторы не предусматривают радикальных мероприятий, требующих существенных капиталовложений, таких, как полная или частичная реконструкция производства, замена оборудования или введение принципиально новой технологии. Ясно, что такие мероприятия, как проводимые сами по себе, так и в совокупности с перечисленными выше факторами, оказывают существенное влияние на объем и качество продукции. Однако предложения подобного типа, как правило, выходят за рамки компетенции разработчиков АСУ.

Для вновь строящихся и реконструируемых предприятий в последнее время создают автоматизированные технологические комплексы. Благодаря развитию микропроцессорной техники системы управления в таких комплексах встраиваются непосредственно в технологическое оборудование, образуя единое целое. Естественно, что в разработке таких автоматизированных технологических комплексов участвуют на равных как конструкторы и технологи, так и специалисты по управлению. Однако на более высоких уровнях управления разработчики АСУ исходят из уже принятой технологии.

К факторам, препятствующим повышению эффективности производства, относят несбалансированность планов производства с планируемыми финансовыми, трудовыми, материальными и энергетическими ресурсами; недостаточный уровень технологической и трудовой дисциплины на производстве; наличие физически и морально устаревшего оборудования; нарушение обязательств поставщиками; внесение неоправданно частых изменений в планы производства, финансирования и снабжения; низкую финансовую дисциплину потребителей готовой продукции; недостаточный уровень производительности вспомогательных цехов, транспортного и складского хозяйства; низкий общий уровень культуры производственно-хозяйственной деятельности и т.п.

В результате анализа выявляют пути и возможности повышения эффективности производства на основе совершенствования управления, в том

числе его автоматизации. Это позволяет наметить комплекс первоочередных задач управления, включаемых в состав АСУ. Одновременно вырабатывают общие рекомендации по улучшению управления предприятием независимо от применения вычислительной техники. Кроме того, подготавливаются обоснование очередности разработки задач и предварительные предложения по составу технических средств.

Завершающий этап предпроектной стадии – разработка технического задания; он предопределяет идеологию будущей АСУ. Определяется в основном состав и структура разрабатываемой АСУ, т.е. из каких подсистем она состоит, как они связаны между собой, какие комплексы задач и в каком порядке будут разрабатываться, их назначение. При этом не указывается, как эти задачи будут решаться, какие экономико-математические или иные методы будут использованы; не детализируются формы представления входной и выходной информации. Эти и другие аналогичные им вопросы о том, какими средствами и способами будет выполнять свои функции система, решаются на стадии проектирования. Образно говоря, ТЗ должно дать обоснованный ответ на вопрос, что должна делать разрабатываемая АСУ, а ТЭО – на вопрос, зачем она создается. Подробно состав и содержание технико-экономического обоснования и технического задания рассмотрены в гл. 9.

§ 1.3. Стадия разработки проектов

На основе согласованного и утвержденного технического задания разрабатывается *технический проект* (ТП), а на его основе – *рабочий проект* (РП) АСУ. При наличии проверенных и показавших хорошую эффективность проектных решений, которые по своим характеристикам пригодны для применения в разрабатываемой системе, рекомендуется разработка единого технорабочего проекта, объединяющего разработку ТП и РП. Это позволяет существенно сократить сроки разработки АСУ и объем проектной документации.

Технический проект. На этапе разработки технического проекта заказчик обязан провести подготовку к вводу АСУ в эксплуатацию, что включает в себя подготовку информационного и технического обеспечения разрабатываемой АСУ, проведение организационных мероприятий и обучение персонала.

По информационному обеспечению заказчик обязан завершить под методическим руководством разработчика создание или приведение в порядок классификаторов готовых изделий, деталей и узлов, материалов, комплектующих изделий, установленного оборудования, структурных подразделений, персонала, нормативов затрат труда и времени по рабочим операциям, поставщиков и потребителей и др. Вся охватываемая перечисленными классификаторами и другая нормативно-справочная информация должна быть подготовлена для записи на машинные носители, проверена

и откорректирована. Должны быть уточнены исходные данные в соответствии с рекомендациями разработчика по составу и структуре информационной базы системы, а также по составу и структуре данных и организации документооборота, предусмотренного в системе для решения комплексов задач управления.

По техническим средствам заказчик обязан на этапе разработки ТП оформить заказные спецификации; заключить договоры на поставку средств вычислительной техники, передачи данных и другого оборудования и материалов; заключить договоры на выполнение строительно-монтажных и пусконаладочных работ, обеспечить качественное и своевременное их завершение.

Организационные мероприятия заказчика заключаются в составлении и согласовании с разработчиком плана-графика совместных работ; назначении руководителей и исполнителей, ответственных за подготовку информационной базы и реализацию автоматизированных функций управления; рассмотрении на научно-техническом совете, согласовании и утверждении технического проекта; подготовке предприятия к вводу АСУ в эксплуатацию.

Обучение персонала проводится по группам — управленческого персонала методам управления в условиях функционирования АСУ и персонала эксплуатации новых технических средств — их обслуживанию.

В свою очередь разработчик обязан: разработать технический проект в соответствии с утвержденным техническим заданием и включающий в себя экономико-математические модели и алгоритмы обработки данных; уточнить состав применяемых пакетов прикладных программ, порядок их настройки для решения предусмотренных ТЗ комплексов задач, а также алгоритмы тех задач, для которых будут разработаны оригинальные программы; обеспечить первоочередную разработку и утверждение комплекса технических средств, а также заказные спецификации на все виды оборудования; разработать и ввести в эксплуатацию у заказчика программы с соответствующей рабочей документацией на них по вводу и ведению информационной базы системы; принять участие в обучении персонала заказчика правилам эксплуатации технических средств, программного и информационного обеспечения, методам взаимодействия с ЭВМ и в условиях автоматизированного решения задач управления; осуществлять методическое руководство при создании заказчиком классификаторов и нормативно-справочной информации. В проекте должны быть разработаны принципиально новые методы решения задач, а возможно и задачи, не существующие в действующей системе.

Знания, полученные в результате предпроектного обследования действующей системы управления, не являются исчерпывающими, а ТЗ на АСУ обычно содержит лишь главные, а потому достаточно общие требования к разрабатываемой системе. Кроме того, за время предпроектной стадии и организации работ над техническим проектом меняется сама управляемая система. Эти и другие факторы приводят к тому, что изучение системы

продолжается в течение всего процесса ее проектирования. Все выявленные при этом существенные изменения должны быть отражены либо в виде дополнений к техническому заданию, либо непосредственно в техническом проекте, если они не меняют принципиальных положений ТЗ.

Особенно это относится к разработке алгоритмов решения конкретных задач. По мере перехода от принципиальных решений к укрупненным схемам, а затем ко все более детальным шагам алгоритма следует непрерывно сопоставлять получаемые решения с эффектом от их практической реализации. Обычно можно представить эффект от решаемой задачи по разработанному алгоритму, вручную шаг за шагом выполняя все процедуры на специально подобранном контрольном примере небольшого объема. Такое прикидочное решение задачи часто требует дополнительного изучения действующей системы, но более целенаправленного и детального. В результате либо вносятся изменения в первоначальный алгоритм, что повышает его качество, либо меняется постановка задачи введением дополнительных ограничений или уточнением критериев оптимизации.

Вводя изменения в какие-либо задачи, надо внимательно следить, как это сказывается на других связанных с ней задачах. Обеспечение взаимосвязи и согласованности между всеми задачами разрабатываемой АСУ осуществляет главный конструктор проекта, либо, в больших проектах, руководитель группы разработчиков подсистемы в пределах своей компетенции. Принципиальные изменения, требующие дополнения или изменения ТЗ, согласовываются с заказчиком и утверждаются в установленном порядке.

Рабочий проект. Он разрабатывается на основании утвержденного технического проекта и утверждению не подлежит. Законченный рабочий проект содержит всю техническую документацию, необходимую для отладки АСУ в целом или соответствующей ее очереди, подготовки ее к вводу в эксплуатацию, проведения приемо-сдаточных испытаний и последующего нормального ее функционирования.

На этапе создания РП заказчик обязан завершить формирование и организовать ведение информационной базы или банка данных АСУ; ввести в промышленную эксплуатацию комплекс технических средств и вычислительный центр; ввести в повседневную деятельность методы планирования и управления производством в соответствии с принятыми в проекте АСУ решениями; разработать под методическим руководством и с участием разработчика и утвердить должностные инструкции для работников управленческого персонала в условиях функционирования АСУ; подготовить в соответствии с инструкциями разработчика контрольные примеры и организовать поэтапную приемку рабочих программ, проверяя их на контрольных примерах, обеспечив их использование в реальной работе по мере приемки; завершить обучение управленческого персонала и работников ВЦ работе в условиях функционирования АСУ, предоставить разработчику машинное время и необходимые материалы (магнитные диски, ленты и т.п.) для подготовки рабочих программ и их сдачи. Последнее условие

в части информационной базы и банков данных желательно выполнить на этапе технического проектирования.

Разработчик на этом этапе обязан завершить разработку программного обеспечения создаваемой АСУ, в первую очередь отладку и сдачу заказчику рабочих программ информационной базы или банка данных; для пакетов прикладных программ выполнить генерацию необходимых рабочих программ и их стыковку с другими программами; осуществлять методическое руководство и помощь заказчику по формированию и ведению информационной базы и по разработке должностных инструкций управленческому персоналу и работникам ВЦ; завершить разработку рабочей документации по всем видам обеспечения; осуществить поэтапную сдачу заказчику рабочих программ на контрольном примере, документации рабочего проекта с оформлением актов сдачи-приемки АСУ в опытную эксплуатацию. На этапе рабочего проектирования особенно возрастает значение согласования и координации деятельности всех участников разработки.

При подготовке программ часто возникает желание усовершенствовать в той или иной степени отдельные алгоритмы, внести изменения в структуру информационной базы, форму входных или выходных документов и т.п. Внесение таких изменений недопустимо, если даже они несколько улучшают характеристики отдельных задач системы. В противном случае одни изменения будут накладываться на другие, и в лучшем случае срок подготовки программного обеспечения недопустимо растянется, а в худшем — система просто выйдет из-под контроля и уже никто не будет знать, что, когда и почему она делает. Все предложения по совершенствованию и корректировке алгоритмов и программ следует накапливать до полного завершения комплекса программ по одной или нескольким задачам. Когда будет завершена их системная отладка, следует рассмотреть все накопленные предложения, отобрать заслуживающие реализации и однократно внести соответствующие изменения. Исключения из этого правила могут быть сделаны лишь в чрезвычайных случаях, например, если на стадии программирования выяснится полная неработоспособность алгоритма. Подобные случаи подлежат разбору у главного конструктора проекта, который и принимает необходимые решения.

Приемка комплекса задач в опытную эксплуатацию заключается в решении контрольного примера, специально подготовленного заказчиком. Контрольный пример решается в присутствии представителей заказчика и разработчика и при успешном его решении оформляется акт сдачи-приемки. Состав и содержание технического и рабочего проектов приведены в гл. 9.

§ 1.4. Стадия ввода в эксплуатацию

В отличие от технических систем, которые вводятся в эксплуатацию одномоментно, всей системой сразу, для АСУ и ее подсистем ввод в эксплуатацию означает постепенный переход от существующих методов управ-

ления к использованию функциональных комплексов задач или отдельных задач создаваемой системы. Ввод в эксплуатацию той или иной задачи или комплекса задач определяется только степенью их готовности и может быть осуществлен сразу после утверждения технического задания, независимо от степени готовности технического или рабочего проекта. Это означает, что ввод в эксплуатацию всей АСУ или одной из ее очередей совпадает с вводом последнего из установленных техническим заданием комплексов задач.

Если существует пакет прикладных программ, который легко адаптируется к условиям конкретной создаваемой АСУ, нет смысла откладывать его использование до завершения разработки технического и рабочего проектов. В то же время некоторые задачи могут оказаться настолько сложными, что алгоритм их решения может быть получен только в результате специальных научных исследований.

Возможность начала ввода в эксплуатацию комплекса задач определяется наличием технических средств, включая ЭВМ, обеспечивающих нормальное функционирование задач на реальной информации и в заданном режиме; готовностью отлаженных и проверенных рабочих программ с соответствующей документацией; рабочим состоянием информационной базы или ее части, соответствующей данному комплексу задач; обученным персоналом и подготовленным к эксплуатации объектом. Состав и сроки выполнения работ определяются согласованными между заказчиком и разработчиком планами-графиками.

На стадии ввода АСУ в эксплуатацию заказчик обязан завершить организационно-технические мероприятия по подготовке предприятия к внедрению; внести предусмотренные проектом АСУ изменения в организационную структуру предприятия и обеспечить выполнение персоналом предприятия должностных и технологических инструкций; завершить опытную эксплуатацию комплексов задач и прием их в промышленную эксплуатацию; проверить эффективность реализованных проектных решений в условиях промышленной эксплуатации; подготовить по согласованию с разработчиком программу приемо-сдаточных испытаний; организовать работу приемочной комиссии и провести предусмотренные программой испытания.

Разработчик на стадии ввода в эксплуатацию обязан по результатам опытной эксплуатации скорректировать техническую документацию; осуществлять методическое руководство и принимать участие в сдаче задач или комплексов задач в промышленную эксплуатацию; принимать участие в разработке программы приемо-сдаточных испытаний и в работе комиссии по приему АСУ в промышленную эксплуатацию.

Опытная эксплуатация. Начало опытной эксплуатации комплекса задач или задачи определяется приказом заказчика, согласованным с разработчиком. Этим же приказом определяются сроки опытной эксплуатации и состав комиссии по приемке комплекса задач. К приказу прилагается согласованная с разработчиком программа, в которой определяются усло-

вия и число расчетов задачи, порядок проверки технических средств и устранения выявленных при опытной эксплуатации недостатков. Если в процессе опытной эксплуатации у заказчика возникают дополнительные требования, не предусмотренные в техническом задании и техническом проекте, то они не являются основанием для отрицательной оценки результатов опытной эксплуатации. Такие требования могут быть удовлетворены по дополнительному соглашению в согласованные сроки, для чего составляется дополнение к ТЗ.

Промышленная эксплуатация. После приема АСУ в промышленную эксплуатацию ответственность за ее работу несет заказчик. Практически любая функционирующая АСУ требует так называемого сопровождения, под которым понимается внесение изменений, вызванных как стремлением совершенствовать систему, так и неизбежными изменениями в управляемой системе или ее внешней среде. Эти изменения разрабатывают и внедряют либо персонал заказчика, либо разработчики по дополнительному соглашению.

Для более четкого проведения приемо-сдаточных испытаний регламентирован ряд положений.

Установлено, что ответственность за организацию и проведение приема АСУ в промышленную эксплуатацию несет заказчик. Перед этим должен быть завершен прием заказчиком всех комплексов задач, предусмотренных техническим заданием, в промышленную эксплуатацию. Если заказчиком были самостоятельно разработаны и введены в эксплуатацию дополнительные комплексы задач, отдельные задачи и технические средства, не предусмотренные ТЗ, то они могут быть включены в состав принимаемой АСУ только по согласованию с разработчиком и после внесения их в ТЗ в качестве дополнений.

Для приема в промышленную эксплуатацию АСУ предприятием распоряжением министерства или ведомства, которому подчинен заказчик, по согласованию с министерством, которому подчинен разработчик, назначается приемная комиссия. Заказчик предъявляет АСУ приемной комиссии и обязан обеспечить ей нормальные условия работы по принятой программе. Проект программы проведения испытаний и приема АСУ в эксплуатацию готовит заказчик совместно с разработчиком и передает приемной комиссии для рассмотрения и утверждения. Для решения организационных вопросов, возникающих в процессе приема АСУ в эксплуатацию, разработчик приказом руководителя назначает ответственного представителя.

Комиссия в соответствии с программой проверяет функционирование комплексов задач, входящих в принимаемую АСУ, а также условия эксплуатации и режим работы технических средств. Для проверки отдельных подсистем комиссия организует рабочие группы. Кроме проверки функционирования комплексов задач комиссия проверяет представленную ей документацию, расчет экономической эффективности создания АСУ и подготовку персонала, ответственного за функционирование АСУ. По

результатам приема составляется акт, дата подписания которого комиссией считается датой ввода АСУ в эксплуатацию.

Состав и содержание документов, оформляемых на стадии ввода АСУ в эксплуатацию, приведены в гл. 9.

§ 1.5. Неформальные этапы разработки систем

Одной из важнейших специфических особенностей, отличающих АСУ от технических систем любой сложности, является значительно более тесная связь с внешней средой. Обеспеченная ресурсами техническая система может длительное время функционировать без существенных связей с внешней средой, т.е. в достаточной степени автономна. АСУ в этом смысле гораздо ближе к живым объектам, которые просто не могут существовать без непрерывного взаимообмена с внешней средой веществом, энергией и информацией. Во многих случаях внешняя среда влияет на алгоритмы и процедуры принимаемых решений, а часто и на цель и критерии эффективности, заставляя перестраивать или видоизменять внутреннее содержание системы. Это же относится к накладываемым на систему внешней средой ограничениям, изменения которых могут быть настолько радикальными, что требуют перестройки функционирования всей системы. Теснотой связи с внешней средой во многом и определяются требования гибкости и адаптивности АСУ к изменениям, которые должны учитываться с самого начала ее разработки.

Поэтому наряду с официальными стадиями разработки АСУ, рассмотренными выше, выделяют логические этапы — внешнее и внутреннее проектирование или соответственно проектирование на макро- и микроуровнях. В принципе такие этапы существуют и при разработке любой технической системы. Первый из них заключается в определении требований к разрабатываемой системе исходя из ее целесообразного использования в определенных условиях, а второй — в собственно проектировании системы. В АСУ эти этапы четче выделены, имеют гораздо большее значение, а реализация их порой требует специалистов разного профиля.

При внешнем проектировании в максимальной степени применяется методология системного анализа. Локализуется сама система, определяются ее границы; выявляются факторы внешней среды, влияющие на систему или находящиеся под ее влиянием; определяются входные воздействия, на которые она должна реагировать, и связь ее выходов с внешней средой; устанавливается требуемая реакция системы на входные воздействия; определяются цель ее функционирования, критерии эффективности и системные ограничения. Иными словами, это этап выяснения взаимодействия системы с внешней средой, на котором определяется, что и зачем будет делать система и почему она должна действовать так, а не иначе.

Внутреннее проектирование определяет содержание самой системы, оно отвечает на остальные системные вопросы: *как, какими методами,*

способами и средствами будет выполнять система свои функции, *кто, где и когда* будет выполнять необходимые для этого операции и процедуры.

Внешнее и внутреннее проектирование связаны между собой. Может оказаться, что задачи, сформулированные при внешнем проектировании, не могут быть эффективно решены ввиду отсутствия адекватных методов и моделей либо технических средств с необходимыми характеристиками, причем ни те, ни другие не могут быть получены за приемлемое время или стоимость. Поэтому требуется, по крайней мере, одна, а иногда и больше итераций.

Начинать разработку надо всегда с внешнего проектирования, определяя максимальные требования к системе и игнорируя возможные внутренние ограничения, как если бы она обладала идеальными возможностями бесконечной мощности. Затем определяют, можно ли удовлетворить эти требования известными методами и техническими средствами независимо от того, располагает ли разработчик такими возможностями в настоящий момент. При положительном прогнозе оценивают реальность использования этих методов и средств в системе. Если имеются значительные трудности, выясняют, какие наиболее близкие по характеристикам реально допустимые средства могут быть использованы, в какой степени при этом видоизменяются требования к системе, насколько они должны быть снижены и являются ли эти сниженные требования приемлемыми для того, чтобы АСУ была достаточно эффективной. Ясно, что при отрицательном ответе дальнейшая работа над системой не имеет смысла, пока не будут выявлены пути преодоления возникших трудностей. Во всяком случае, нельзя идти от обратного, т.е. определять характеристики системы по тем возможностям, которые есть в распоряжении разработчика, даже если они достаточно велики. Это равносильно тому, когда постановка задачи подгоняется под известный метод решения, например путем линеаризации нелинейных зависимостей, снижения размерности и т.п. Задача будет решена, но полученные результаты практической ценности в большинстве случаев иметь не будут.

Арсенал методов, используемых на этапе внутреннего проектирования, достаточно богат. Вместе с тем практика показывает, что весьма продуктивными являются методы: *единичной нити, большой нагрузки и состязательных или конфликтных ситуаций*, особенно при их совместном использовании.

Метод единичной нити. Он заключается в анализе и последующем синтезе для разрабатываемой системы реакции системы в целом и ее элементов на каждый возможный вид входных воздействий в отдельности. Результат анализа удобно представить в виде цепочки последовательных реакций или комплекса процедур различных подразделений и других элементов системы, обеспечивающих нужную реакцию системы в целом. В отдельных случаях целесообразно проследивать цепочку реакций в обратном направлении — от выхода к входу. Например, анализ хода из это-

вления одного вида готовой продукции полезно начинать с выхода, определяя последовательно, какие узлы поступают на сборку, какие детали нужны для сборки узлов и какие материалы нужны для изготовления деталей. Информационные потоки чаще лучше изучать начиная со входа, выясняя, например, какие внутренние документы порождаются системой при появлении на ее входе одного заказа, что происходит в отдельных подразделениях вплоть до выпуска заказанной продукции и сопровождающей финансовой и технической документации. Метод единичной нити помогает четче структуризовать систему, оценить времена реакций подразделений и системы в целом на различного рода воздействия.

Синтез структуры, поиск возможных решений по отдельным элементам системы с использованием метода единичной нити остается искусством и пока не поддается строгой формализации. Поэтому по возможности для синтеза следует использовать более мощные средства, если они имеются. Однако для анализа, сопоставления возможных вариантов простой и наглядный метод единичной нити оказывается часто весьма полезным. При этом следует помнить, что при построении единичной нити игнорируется поступление в систему многих как однотипных, так и разнотипных сигналов (в дальнейшем термины сигнал и воздействие будут использоваться как синонимы).

Не вызывает особых затруднений обслуживание одного заказа, но если их поступает несколько тысяч в день, то возникает совсем другая и чрезвычайно трудная проблема — большая нагрузка.

Метод большой нагрузки. Как и при исследовании единичной нити, в качестве инструмента используется модель — аналитическая, графоаналитическая или графическая — движения материальных и информационных потоков. Однако если при анализе единичной нити используют в основном логические методы, то при анализе большой нагрузки важное значение имеют методы и модели теории массового обслуживания. Это объясняется тем, что большая нагрузка всегда связана с появлением очереди — либо система не успевает обрабатывать поступающие входные сигналы и они выстраиваются в очередь, либо система или ее элементы простаивают в ожидании входных сигналов, образуя очередь блоков обслуживания. В полностью детерминированной системе работа блоков обслуживания согласована с потоком поступающих требований и проблема очереди не возникает, но в АСУ управляемые процессы стохастичны по своей природе, чем и объясняется неизбежное появление очередей. Если время ожидания и другие характеристики очереди оказываются неудовлетворительными, то можно сократить время ответной реакции элементов, заменив их на более быстросодействующие, или изменить структуру прохождения заказа через систему, т.е. единичную нить. Другим способом является введение параллельных каналов обслуживания, т.е. установка дополнительных элементов системы. Наконец, улучшить характеристики системы можно путем введения буферного, накопительного хранения промежуточных результатов между элементами, чтобы сгладить пиковые нагрузки. На практике обычно применяется сочетание указанных методов.

Таким образом, время отклика системы на одиночное входное воздействие определяется принятым вариантом единичной нити и характеристиками блоков обслуживания, а число обслуживаемых блоков, емкость, расположение и тип промежуточных хранилищ устанавливаются при решении задач большой нагрузки.

На этапах единичной нити и большой нагрузки предполагается, что система работает в некотором предусмотренном разработчиками *нормальном* режиме, отклонения от которого не рассматриваются, как если бы они не существовали. На самом деле всегда существуют случайные изменения, которые нарушают работу системы, понижают ее эффективность, а в некоторых случаях могут вовсе вывести ее из строя. Входные воздействия могут быть непредусмотренного типа, поступать слишком рано или слишком поздно. Элемент системы может либо выйти из строя, либо действовать не оптимальным образом. Персонал системы или ее пользователи могут предпринимать действия, препятствующие нормальной работе, а порой дискредитирующие или выводящие ее из строя.

Метод конфликтных ситуаций. При анализе конфликтных ситуаций основная трудность — в их выявлении. Разработчик системы должен суметь предугадать возможные случаи отклонений от нормального режима работы. Наилучшие результаты дает хотя и не очень изящный, но достаточно надежный путь перебора всех предусмотренных проектом входных воздействий, элементов и реакций системы. Предварительно желательно определить, какие отклонения типичны для входных воздействий. Например, если на вход системы поступает документ, возможны следующие отклонения: поступление документа до или после установленного срока; число копий меньше требуемого; отсутствие необходимых подписей и (или) печатей; ошибки заполнения — некоторые графы пропущены, не сходятся контрольные суммы по строкам или колонкам, записаны непредназначенные данному пользователю позиции. Фактически рассматривают каждую составную часть входного документа и путем логического анализа выявляют возможные существенные отклонения.

Таким же образом определяют возможные отклонения от нормального режима для каждого элемента системы и ее выходов.

Затем необходимо принять решение о поведении системы в случае каждого типа выявленных отклонений. Наиболее простым является игнорирование сигналов с отклонениями, для чего достаточно иметь специальный блок обнаружения. Однако в большинстве случаев это недопустимо. Тогда надо считать каждый тип отклонений особым сигналом и определять реакцию на него системы теми же методами, которые применялись к обычным входным воздействиям.

Внешнее и внутреннее проектирование АСУ осуществляют в виде последовательных итераций, уточняя и детализируя результаты по мере расширения знаний о системе на различных стадиях проектирования.

Сложность и объем задач, подлежащих решению в процессе разработки крупных АСУ, неизмеримо возрастают при создании систем, не имеющих аналогов. Каждая АСУ в известном смысле уникальна, тем не менее для систем определенного уровня и назначения могут быть выявлены аналоги, определены типовые проектные решения, найдены подходящие пакеты прикладных программ. Однако существуют системы управления, уникальные в полном смысле слова, не имеющие аналогов ни в нашей стране, ни в мировой практике. Это, как правило, системы достаточно высокого уровня и специфического назначения, такие, как здравоохранение, агропромышленный комплекс, материально-техническое снабжение в масштабах страны и многие другие. Сюда же можно отнести республиканские АСУ, охватывающие все отрасли, подведомственные Совету Министров союзной или автономной республики.

Для таких систем характерны словесные, обобщенные формулировки целей и критериев, трудно формализуемые постановки специфических задач, сложная разветвленная многоуровневая иерархическая структура с рассредоточенными по большой территории объектами разнохарактерного типа, не говоря уже о многочисленных тесных связях с другими системами такого же уровня и масштаба, чрезвычайно большой размерности решаемых задач и массовом характере обращений к системе со стороны пользователей.

Сложность и объем задач, подлежащих решению в процессе разработки крупных АСУ, обуславливают целесообразность проведения в отдельных случаях дополнительной стадии проектирования — генеральной схемы АСУ*. В ее разработке принимают участие наиболее квалифицированные специалисты по системному анализу и созданию АСУ вместе с компетентными специалистами высокого уровня по создаваемой системе. Такая группа не должна быть многочисленной, порядка 3–5 человек, но она должна быть поддержана коллективом квалифицированных специалистов, быстро обеспечивающих ведущую группу необходимыми сведениями по ее запросам.

Пока не существует формального аппарата, который можно было бы использовать для достаточно строгого построения генеральной схемы, ее приходится формировать на основе логики, здравого смысла, опыта и искусства участников разработки. При этом возможны две одинаково опасные крайности. С одной стороны, стремление максимально формализовать и автоматизировать функции системы, насытить ее техническими средствами для сбора, передачи и обработки информации может привести к тому, что созданная разработчиками модель окажется неадекватной действительности. В лучшем случае это приведет к излишним затратам и созданию малоэффективной АСУ, а в худшем — дискредитирует саму идею исполь-

* Иногда это называют эскизным или аванпроектом.

зования вычислительной техники для повышения эффективности организационного управления в рассматриваемой системе. С другой стороны, желание максимально сблизить модель создаваемой системы с действительностью может привести к тому, что не будут найдены принципиально новые функции и методы управления, постановки задач и способы их решения. В системе будут сохранены все существующие функции, распределение их по уровням, решаемые задачи и процедуры сбора и обработки информации, лишь их выполнение будет соответствующим образом алгоритмизировано и реализовано с использованием ЭВМ. Такой подход неизбежно приведет к нежелательным результатам, как и в предыдущем случае.

Изучение системы начинают с ее структуры. Надо выяснить число уровней иерархии, количество элементов каждого уровня, их функции и назначение в обобщенном виде. После этого переходят к анализу системы в целом. Надо понять и постараться сформулировать основную задачу системы, ее роль и место в народном хозяйстве страны, какие цели перед ней поставлены, как можно оценить эффективность их достижения. Специалисты по АСУ должны постараться сформулировать ответы на эти вопросы для себя, в привычных терминах. Для этого очень важно понять не только как работает система, но и почему она работает именно так, а не иначе. Только после этого сопоставляют свое представление о системе, ее функциях, целях и критериях с тем, какие существуют официальные формулировки этих понятий, а при их отсутствии — как это представляют себе руководящие работники разрабатываемой системы. Это выяснение надо проводить не только с представителями заказчика, входящими в группу разработчиков, но с как можно большим кругом руководителей как в личных беседах, так и на рабочих совещаниях небольшого состава. Это очень важный момент, от которого зависит успех всей разработки. Крайне редко точки зрения на работу системы сразу совпадают во всех деталях. Если даже такое совпадение происходит, оно должно насторожить разработчиков, заставить еще раз проанализировать систему и расширить свои знания о ней.

В более частом случае, когда возникают некоторые расхождения не принципиального характера, надо выяснить их природу. Полезно несколько утрировать свою точку зрения, усилить и обоснованно развить предположения о необходимости введения новых и исключении некоторых существующих функций системы, переформулировании цели и критериев оценки ее деятельности. Не надо навязывать свои предложения, убеждая в их полезности и эффективности, а уметь выслушать и проанализировать возражения и встречные предложения. Сформированные в результате такого взаимодействия характеристики системы служат основой для их дальнейшей детализации. Проводят укрупненно распределение функций по уровням системы, строят дерево целей, осуществляя для каждого уровня процедуры, описанные выше. Эта работа может проводиться уже несколькими группами достаточно квалифицированных разработчиков при общей координации и руководстве руководителями разработки.

В результате разрабатывается генеральная схема АСУ. В ней содержатся структура создаваемой системы, выполняемые системой в целом и на каждом уровне функции, перечни задач, цели и критерии эффективности. Определяется разделение функций и задач между персоналом и вычислительными средствами. Приводится в самом общем виде описание алгоритмов и процедур обработки информации в создаваемой системе. Ориентировочно намечается, какие типы технических средств потребуются для реализации функций на каждом уровне системы и для организации каналов связи между ними. Указывается, какие оптимизационные, аналитические, прогнозные и учетные задачи будут решаться в системе. Проводится разделение разработки системы на очереди, дается оценка сроков их разработки и ввода в эксплуатацию, определяется необходимое число разработчиков по основным группам и финансирование.

Генеральная схема обсуждается у заказчика на совещании с участием наиболее квалифицированных специалистов заказчика и разработчика. После внесения необходимой корректировки по полученным замечаниям генеральная схема утверждается заказчиком и в дальнейшем служит основой для проектирования системы. Вместе с тем в процессе разработки системы следует учитывать, что генеральная схема создавалась на ранних этапах изучения системы и в отдельных деталях может корректироваться.

Составление генеральной схемы не предусмотрено существующими ГОСТами и методиками по созданию АСУ, поэтому ее необязательно оформлять в составе официально утверждаемой документации. Однако по существу такой начальный этап в цепи последовательного уточнения и детализации модели уникальной АСУ крайне важен, а формальное утверждение генеральной схемы заказчиком во избежание последующих недоразумений является просто необходимым.

§ 1.7. Распределение функций между специалистами

В разработке АСУ принимает участие несколько групп разработчиков, каждая из которых объединяет специалистов разного профиля и работает над определенной функциональной частью АСУ. Результаты одних групп служат основой для других, они взаимосвязаны. Поэтому необходимо четко определить функции каждой группы специалистов, точно определить не только содержание результатов каждой группы в отдельности, но и что и в какой форме передается другим участникам работ. Основной группой являются *инженеры-системотехники*, специалисты по проектированию и эксплуатации АСУ. Вместе с ними работают программисты различной квалификации, математики и специалисты по техническим средствам.

На определенных этапах работ задачи различных групп тесно смыкаются. Подавляющее большинство алгоритмов решения задач управления и процедур автоматизированной обработки данных разрабатывают специ-

алисты по проектированию АСУ. Это относится и к тем алгоритмам решения оптимальных задач, которые могут быть решены известными экономико-математическими методами или при незначительной их модификации. Однако при формализации некоторых задач планирования и управления, построении сложных моделей, разработке методов и алгоритмов их оптимизации может возникнуть необходимость в развитии существующих или разработке новых математических методов. Разработка таких методов и алгоритмов — задача *математиков*. При этом следует помнить, что решающим фактором является правильная системная проработка задачи, что не входит в обязанности математика, если он не берет на себя функции системного аналитика. Имеется много печальных примеров, когда прекрасные математики тратили уйму времени и труда на получение практически бесполезных результатов только потому, что задача не была правильно системно подготовлена. Успех может быть достигнут только при самом тесном сотрудничестве этих двух групп.

В меньшей степени сказанное относится к *программистам*. Инженер-системотехник должен уметь программировать, представлять себе при алгоритмизации задачи все тонкости и трудности ее программирования. Современные тенденции создания непроецедурных языков позволяют получать результаты обработки данных непосредственно самому пользователю, без участия программистов. Однако пока программирование некоторых сложных задач традиционно выполняют прикладные программисты.

Специалисты по техническим средствам более обособлены от инженеров-системотехников. Последние формируют требования к техническим средствам, а их монтаж, наладка, обеспечение работоспособности в процессе эксплуатации — задача специалистов по техническим средствам. В их компетенцию также входит конструирование, изготовление и эксплуатация специальных технических устройств, не выпускаемых серийно, если они включены в проект системотехником с соответствующим обоснованием и сформулированными техническими требованиями.

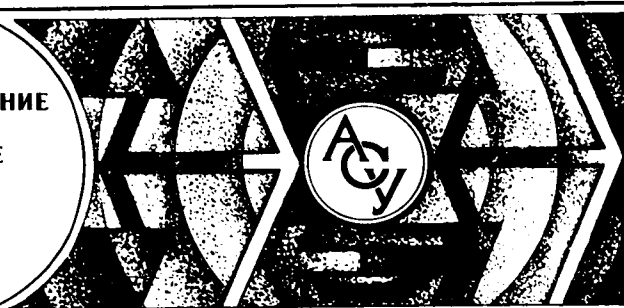
Взаимодействие различных групп специалистов меняется во времени по этапам разработки. Сначала системотехники взаимодействуют со специалистами заказчика. По мере продвижения изучения системы в орбиту работ включаются математики; начинается этап интенсивного внутреннего проектирования, во время которого объем совместных работ со специалистами заказчика существенно сокращается. По мере разработки алгоритмов все больше загружаются программисты. Параллельно ведут свою работу специалисты по техническим средствам. К окончанию программирования усиливается связь между системотехниками, математиками и программистами, идет проверка правильности подготовленных программ. Когда программы получены в окончательном виде и технические средства смонтированы и проверены, начинается ввод системы в эксплуатацию, в процессе которого для устранения выявляемых недостатков работают совместно все группы специалистов.

Важной группой являются *системные программисты*, которые обеспе-

чивают эффективную эксплуатацию операционных систем, систем управления базами данных и другого программного обеспечения ЭВМ общего назначения, повышающего эффективность функционирования АСУ.

При эксплуатации банка данных необходимо иметь лицо или небольшую группу лиц, осуществляющих функции *администратора банка данных*; он следит за строгим соблюдением регламентированных правил его использования.

В крупных системах в группе системотехников могут при необходимости создаваться подгруппы, выполняющие специальные разделы проекта, такие, как создание и ведение нормативно-справочной информации, разработка специализированных информационных языков, методическое руководство работой заказчика по созданию словарей и классификаторов и т.п.



Макроуровень проектирования является скорее искусством, чем основанным на инженерных расчетах процессом. Это объясняется наличием большого числа неформальных связей системы управления с внешней средой. Существенное влияние оказывает слабо регламентируемый человеческий фактор, отсутствие строго сформулированных критериев, трудность выявления большого множества ограничений, выделения из них наиболее существенных.

Вместе с тем произвольное толкование важнейших вопросов идеологии работы системы, субъективизм при их решении недопустимы. Созданные модели, в большей или меньшей степени формализованные, помогают применить методологию системного анализа, облегчают анализ системы на уровне логики и здравого смысла, формализуя результаты изучения существующей системы, подлежащей автоматизации. Поэтому разработчик должен наряду с изучением и использованием излагаемых ниже методов проявлять творческий подход и создавать свои более строгие методы для этой стадии проектирования, которые, возможно, в дальнейшем найдут более широкое применение.

§ 2.1. Методы изучения и анализа существующих систем управления

Создание АСУ начинают с изучения укрупненных характеристик системы, подлежащей автоматизации. Последняя рассматривается как некоторая целостность, обладающая определенными свойствами. Выделение частей системы проводится укрупненно лишь настолько, чтобы было понятно, каким образом и какими средствами она выполняет свои функции. Детализация средств и способов реализации системой своих функций

проводится лишь в той степени, в какой это необходимо для понимания работы системы в целом. Сведения о системе обычно получают из литературных источников и других печатных материалов, непосредственным наблюдением за процессом функционирования системы, в результате бесед и опросов сотрудников. При этом следует помнить о необходимости перекрестной проверки, сопоставления сведений, полученных из разных источников. Выявляемые разногласия являются свидетельством необходимости более тщательного изучения данного фрагмента; недопустимо считать один из источников, хотя бы собственные наблюдения, более достоверным и пренебрегать другими.

В соответствии с методологией системного анализа изучение начинают с выявления глобальной или общей цели исследуемой системы.

Общая цель системы определяется ее назначением. Для промышленного предприятия это производство продукции определенной номенклатуры, для транспортной организации — перемещение грузов заданного характера, для высшего или среднего специального учебного заведения — выпуск специалистов установленного профиля и т.д. Назначение определяет основную функцию системы. Возможно наличие у системы нескольких функций, причем некоторые из них по значению близки к основной. Например, транспортная организация наряду с перевозкой грузов может перевозить людей, как для собственных нужд, так и для других организаций. При изучении системы следует выявить все выполняемые ею функции, расположить их по приоритету и установить взаимосвязи между ними.

Следующим шагом является изучение структуры системы. Для предприятия или организации это легче всего сделать с использованием штатного расписания, где указаны все работающие с разбивкой по подразделениям.

Для формализованного представления результатов изучения системы наиболее удобны системные спецификации, представляющие собой набор документов определенной формы. Первые два документа системных спецификаций содержат общее описание организации и ее структуры. Следующий документ — таблица функций организации — представляет собой прямоугольную матрицу, по строкам которой перечислены выполняемые организацией функции, а по столбцам — ее подразделения и непосредственно подчиненные организации. На пересечении строк и столбцов матрицы отмечают, участвует ли данное подразделение в реализации рассматриваемой функции или нет. Это можно указывать каким-либо знаком — "крестиком", "галочкой" — или приводить дополнительные поясняющие сведения — долю, характер участия, число или процент привлекаемых сотрудников и т.п.

Следующим шагом является изучение самого процесса функционирования предприятия или организации, технологии его деятельности. Для систем материального производства наиболее эффективным путем является изучение движения материальных потоков методом единичной нити; для непроизводственных систем тем же методом выявляют последователь-

ность операций по обработке входных заявок, документов, запросов пользователей и т.п. Наиболее удобной формой представления результатов этого этапа изучения системы является структурно-информационно-временная схема (СИВС), входящая в системные спецификации.

СИВС представляет собой прямоугольную сетку, разбитую на три горизонтальные зоны. Основной частью СИВС является средняя зона, в которой по горизонтальным строкам перечислены все структурные подразделения изучаемой организации. По строкам верхней зоны перечислены вышестоящие организации, с которыми изучаемая система связана при выполнении своих функций, а по строкам нижней зоны — подчиненные организации, а также поставщики исходных материалов и потребители продукции. В левой части схемы выделен столбец для записи наименований всех вышеуказанных организаций и подразделений. В оставшейся части схемы, используя условные обозначения по ГОСТ 19.002–80 и ГОСТ 19.003–80, составляют схему материальных потоков. Слева направо показано движение во времени или по ходу технологического процесса. Строгий масштаб времени не обязателен, просто операция, изображенная на схеме левее, происходит раньше, чем помещенная правее.

Схему рисуют сначала очень укрупненно и для одной единичной нити "вход — выход", по мере необходимости ее детализируют и накладывают на нее другие материальные потоки. В дальнейшем в СИВС отображают аналогичным образом потоки движения информации. Сопоставление информационных потоков с материальными позволяет проанализировать их синхронность и синфазность, выявить излишние процедуры, связанные со сбором, движением и переработкой информации.

Представляя собой простое и наглядное отображение рабочего процесса, СИВС позволяет всем участникам разработки достаточно легко ознакомиться с деятельностью предприятия.

Получив общее представление о деятельности предприятия, движении материальных и информационных потоков, переходят к выявлению и формализации цели и критериев эффективности управления. Следует установить, удовлетворяет ли потребители продукция предприятия по таким параметрам, как количество выпускаемых изделий по всем номенклатурным позициям; их качество по сравнению с лучшими известными образцами; сроки поставки по продолжительности, ритмичности; число срывов; стоимость изделий в сопоставлении с аналогами; обновление номенклатуры и совершенствование изделий и т.п. Надо определить, являются ли удовлетворительными достигнутые значения показателей, характеризующих работу самого предприятия — прибыли, рентабельности, фондоотдачи, выполнения договорных обязательств, себестоимости продукции, материало-, трудо- и энергоемкости изделий, их технологичности, ритмичности производства и др.

В результате изучения предприятия в целом формулируют цели, критерии эффективности функционирования и развития предприятия, существующие ограничения.

Быстрые темпы развития народного хозяйства привели к тому, что по производству ряда важнейших видов продукции наша страна вышла на первое место в мире. В связи с этим растет число наименований продукции, по которым спрос удовлетворяется полностью. Планирование и управление производством при наличии и отсутствии дефицита существенно отличаются по характеру и содержанию применяемых методов.

В условиях значительного превышения спроса над предложением основной целью предприятия является максимизация выпуска продукции, а критерием эффективности — достигнутый прирост по сравнению с некоторым моментом времени, принятым за точку отсчета — тот же период прошлого года, предыдущая пятилетка и т.д. При многономенклатурном производстве, если превышение спроса существует практически по любой позиции номенклатуры, не так существенно, по каким позициям получен больший прирост, а по каким меньший. Используется интегральный показатель — общий прирост производства в так называемом "валовом" выражении, в ценностном исчислении. Максимизация этого показателя осуществляется в условиях ограничений, которыми в основном являются производственные мощности, трудовые и материальные ресурсы, а также экономические факторы — себестоимость продукции и рентабельность производства. Недостатки оценки эффективности производства по "валу" хорошо известны, тем не менее этот показатель оказывается очень живучим.

В настоящее время в качестве одного из важнейших показателей, характеризующих эффективность работы предприятия, установлено выполнение договорных обязательств. При этом целью является полное выполнение договорных обязательств, а критерием эффективности — процент позиций и объемов поставок, по которым не выполнены договорные обязательства.

Так как договоры на поставку заключаются с потребителями по детальной, так называемой специфицированной номенклатуре, выполнение полностью всех договорных обязательств означает, что по всем номенклатурным позициям потребители получили предусмотренные количества продукции. Однако из этого вовсе не следует, что спрос на данную продукцию полностью удовлетворен, так как сами договоры заключаются предприятием-производителем с учетом имеющихся ограничений, что не гарантирует отсутствия оставшегося неудовлетворенным спроса. Система договоров как бы переводит спрос из случайного и превышающего возможности производства в детерминированный и согласованный с этими возможностями. Это позволяет более четко и уверенно планировать и оперативно управлять ходом производства как поставщиков, так и потребителей их продукции, но не дает формальных оснований стремиться к удовлетворению оставшегося за рамками договоров неудовлетворенного спроса. Вместе с тем, если в ходе производства появляется возможность перевыполнить план, вопрос о том, на какие позиции номенклатуры направить дополнительные ресурсы, должен решаться специальной процедурой, вне рамок договорных отношений.

В условиях сбалансированного спроса и производства резко возрастает значение прогностических и вообще вероятностных методов. Основной целью для промышленных предприятий остается развитие производства, однако оно уже выражается не столько в количестве выпускаемой продукции, сколько в непрерывном повышении ее потребительских свойств, обновлении номенклатуры, улучшении качества и других мерах, направленных на удовлетворение непрерывно растущих потребностей. Если в условиях дефицита для нормального хода и развития производства важнейшее значение имеет снабженческая функция, то при достаточном для удовлетворения спроса уровне производства резко возрастает роль и значение функции организации сбыта продукции.

При определении возможностей роста производства в количественном и качественном отношении выявляют факторы, способствующие и препятствующие достижению поставленной цели. Обычно эти факторы тесно связаны между собой. Например, если повышению выпуска продукции препятствует ограниченная производительность участка гальванических покрытий, то повышение производительности этого участка способствует достижению более высокого уровня выпуска продукции. Таким образом, выявление вышеупомянутых факторов по существу сводится к определению "узких мест" производства, что в первом приближении определяется сравнением пропускной способности всех участков в расчете на объем выпуска готовых изделий.

Определение целей и критериев эффективности для предприятия сильно осложняется тем, что такие сложные системы являются, как правило, многоцелевыми и многокритериальными. В современных условиях наиболее плодотворным, если не единственным возможным, является метод экспертных оценок, позволяющий определить набор целей и их приоритеты, а также множество критериев и весовых оценок для них.

Трудности формализованного определения целей и критериев не должны разочаровывать разработчиков. Отсутствие адекватного математического, лингвистического, алгоритмического аппарата не позволяет построить аналитическую модель предприятия в целом и использовать ее для управления. Поэтому управление предприятием обычно опирается на результаты решения многих комплексов задач, гораздо более четких и формализованных. При их постановке следует хорошо представлять себе хотя бы на содержательном уровне общие цели и критерии предприятия, чтобы все частные цели были направлены на достижение общей, а критерии были непротиворечивыми.

Анализ информационных потоков. Для описания потоков информации на макроуровне и совершенствования существующих потоков информации используют графические и матричные методы, а также системные спецификации.

Основными элементами информационного потока являются некоторые объемы информации на носителях и процедуры преобразования информации. Располагая эти элементы в виде вершин графа, дуги которого соот-

ветствуют передаче информации от одного элемента к другому, получают графическую модель в виде ориентированного графа. Наиболее удобно представлять модель в виде СИВС, что позволяет проследить движение информации не только по логико-функциональным связям, но и по организационным. Применяемые при этом условные обозначения позволяют непосредственно на схеме вводить пояснения и количественные характеристики — шифр и наименование документа, частоту его формирования, объем содержащейся в документе и перерабатываемой информации в алфавитно-цифровых знаках. Логический анализ схемы позволяет проследить движение информации от входа к выходу, определить моменты и места образования и использования документов, процедуры преобразования информации, время обработки, наличие дублирования при вводе исходных данных и при их обработке, количество однотипных операций, излишние перемещения данных между подразделениями. При более глубоком анализе СИВС помогает выявить степень использования данных, действительную необходимость передачи их в то или иное подразделение.

При исследовании информационных потоков на основе сетевой модели и стандартного аппарата ее анализа и оптимизации используется иная интерпретация. Под событием понимают определенный документ или машинный носитель, а под работой — процедуры преобразования информации: составление документов, объединение или расчленение документов, другие виды их преобразования и использования. По известным алгоритмам находят критический путь, начальные и конечные возможные и допустимые сроки выполнения работ, резервы времени для работ и событий. Оптимизация процесса переработки информации заключается как в перераспределении ресурсов, используемых для ее преобразования, так и в изменении содержания работ.

Другой разновидностью методов, основанных на использовании теории графов, является графоаналитический метод, заключающийся в построении информационного графа и анализе его матрицы смежности. В качестве компонентов потока информации рассматриваются исходная, промежуточная и выходная информация в виде соответствующих наборов данных. Между компонентами информационного потока устанавливается отношение порядка: нулевой порядок имеют наборы данных исходной информации, а наивысший — выходные результаты. Упорядоченные таким образом компоненты представляют в виде графа, вершины x_i которого отображают компоненты информационного потока, а дуги направление. Вершины соединяют дугами в том случае, если существует информационная связь между компонентами, представленными соответствующими вершинами, причем без промежуточных результатов. Их наличие требует введения дополнительной вершины между двумя рассматриваемыми. Дуги направлены от вершины более низкого порядка к вершине более высокого порядка. Построенный граф называют информационным. Граф может быть более компактно представлен матрицей смежности, которая строится следующим образом. Элемент (i, j) матрицы, находящийся на пересечении i -й строки и j -го

столбца, равен единице, если вершины x_i и x_j соединены дугой, и равен нулю в противном случае. Матрицу смежности возводят во вторую, третью и т.д. степень, пока не получат матрицу, равную нулю по всем элементам. Общее число полученных при этом матриц равно порядку информационного графа.

Матричная модель описанного типа позволяет определить: порядок схемы потока информации; порядок каждой компоненты потока; число компонентов, непосредственно участвующих в формировании каждого результата; число результатов, в формировании которых непосредственно участвует каждая компонента; число путей фиксированной длины, связывающих любые две компоненты потока; число возможных путей между любыми двумя компонентами потока; все результаты, для формирования которых используется каждая компонента; все компоненты, необходимые для формирования каждого результата [1.9].

Информационные потоки могут быть также представлены комплексом графов [1.9], на каждом из которых отображается преобразование информации при расчете отдельного показателя. Ребра графа ориентированы в виде дерева от исходных показателей к результирующим, которые в свою очередь являются исходными для следующего уровня укрупнения. Такая модель удобна, например, для отображения процесса формирования сводок о различных аспектах работы цехов, предприятий, объединений; при формировании плана и т.п. Если исходные показатели используются для формирования нескольких результирующих, возможно сращивание деревьев — наличие у графов общих вершин и пересекающихся дуг.

Для отображения процессов преобразования информации наряду с графовыми используют матричные информационные модели [1.4], которые отражают во взаимосвязанном виде процедуры создания документов, содержащих экономические показатели и другие реквизиты. Существует несколько типов матричных информационных моделей, выбор среди которых определяется в основном содержанием задач анализа. Так, например, для укрупненного анализа связей между подразделениями удобны модели типа "документ на документ", а для изучения процесса создания документа и последующего его использования предпочтительнее модели типа "показатель на показатель", хотя они и являются более громоздкими.

Наиболее полное формализованное описание как изучаемой, так и проектируемой системы может быть представлено системными спецификациями. Они представляют собой комплект документов, назначение которого — представить в компактной и удобной для использования форме сведения о движении информации в системе, начиная с содержания и формы представления входных данных, описания процедур их обработки и контроля, информационных массивов и кончая формами представления выходных данных. Системные спецификации состоят из трех групп бланков, предназначенных соответственно для представления общих сведений о разрабатываемой системе, изучения и анализа информационных потоков и алгоритмов переработки данных, описания результатов проектирования.

Значительные затраты времени и труда на заполнение бланков системных спецификаций требуют различного их применения. Их использование тем более эффективно, чем сложнее разрабатываемая система. Хорошо подготовленные спецификации упрощают передачу результатов работы одних специалистов другим, повышают качество проектирования за счет снижения вероятности пропусков и неточностей в процессе работы и формализации ввода исправлений и дополнений, упорядочивают планирование работ и контроль за их качеством и сроками выполнения. Описанные ниже бланки приведены в качестве примера и могут видоизменяться применительно к конкретным системам.

В системных спецификациях для документов, массивов и процедур обработки информации целесообразно использовать по два вида бланков — укрупненные и детальные описания. Детализация требуется в тех случаях, когда соответствующие документы или массивы будут использоваться в создаваемой системе.

Все бланки имеют идентификатор, помещенный в нижней правой части бланка. Он содержит шифр бланка и его порядковый номер. Нумерация бланков сквозная, по всем видам бланков. Для каждого типа бланков заранее выделяют группу номеров исходя из размерности задачи. Если содержание бланка не уместится на одной странице, используют несколько страниц с порядковой нумерацией в каждом случае. Номер страницы указывается через тире после номера бланка, после чего в скобках записывается общее число страниц этого бланка. Рядом с номером бланка в идентификаторе помещают наименование и шифры проекта и этапа, к которому относится данный бланк. В нижней левой части бланка указывается, кто его заполнил и проверил.

В табл. 2.1 приведен бланк АД-1, который используется для общей характеристики документа. В графе "Приоритет" указывается, после каких или перед какими документами появляется описываемый, какие документы отменяет или заменяет. В графе "Метод подготовки (использования)" указывают, готовится ли документ путем расчета показателей, сбором сведений от источников информации и т.п., а его использование — для принятия управляющих решений, подготовки других документов и пр. В графе "Последовательность" указывается порядок следования сведений в данном документе. Все количественные характеристики, перечень которых может быть при необходимости изменен или дополнен, указываются при малых разбросах значений только в графе "среднее", а при больших разбросах заполняются все три графы "мин", "макс" и "среднее". В графе "Частота" приводится периодичность выдачи или получения документа — ежедневно, раз в месяц и т.п. Степень использования содержащихся в документе данных приводится в процентах от общего количества их в данном документе. В графе "Дублирование" приводятся наименования или шифры документов и информационных массивов, в которых имеются сведения, повторяющиеся содержащиеся в данном документе. Если образец документа прилагается, желательно, чтобы он был заполнен обычным образом. Цель использования указывается кратко по существу: подготовка отчета, расчет заявки и т.п. В последующих графах от "Обработки" до "Ссылка на" проставляется "крестик" или "галочка", если должностное лицо выполняет соответствующую операцию со всем документом или некоторыми данными из него.

Второй бланк анализа документов АД-2 (табл. 2.2) предназначен для детального описания документа. Как указывалось, его заполнение целесообразно для тех доку-

ментов, вероятность сохранения в системе которых достаточно велика. Наименования подразделения и документа должны совпадать с указанными на бланке АД-1. Под специальными требованиями понимается цвет бумаги, особый формат, недопустимость помарок и исправлений и т.п. Уровень и шаблон элемента документа определяют аналогично принятому в КОБОЛе описанию данных. Высший уровень 01 присваивается тем элементам документа, которые не являются составной частью никакого

Т а б л и ц а 2.1

Наименование подразделения		Шифр подразделения							
Вход/выход		Шифр документа							
Наименование документа	Кем, когда утвержден	Число копий							
Источник (откуда поступает)									
Имеет отношение к документам									
Кем готовится (используется)		Приоритет							
Метод подготовки (использования)		Последовательность							
Характеристики	мин	макс	среднее						
Время обработки									
Объем (в знаках)									
Частота									
Степень использования	Дублирование		Образец прилагается						
Движение документа									
№	Должность лица	Цель использования	Степень использования	Обработка	Хранение	Контроль	Передача	Ссылка на	Примечания
Дополнительные сведения									
Источник	Заполнил	Дата	Проект (тема)		Шифр		АД-1		
	Проверил	Дата	Этап (раздел)		Шифр		NN		

Таблица 2.2

Наименование подразделения					Шифр		
Наименование документа					Шифр		
Специальные требования							
Длина				Ширина			
Уровень	Наименование элемента	Число символов	Шаблон	Частота	Использование	Дублирование	Примечания
Дополнительные сведения							

Источник	Заполнил	Дата	Проект (тема)	Шифр	АД-2
	Проверил	Дата			
			Этап (раздел)	Шифр	NN

другого элемента, например шапка документа и его текстовая часть. Уровень 02 имеют элементы, непосредственно входящие составной частью в элемент уровня 01, уровень 03 — входящие в уровень 02, и т.д.

Например, описание сведений о сотруднике имеет следующий вид:

- 01 сведения о сотруднике
 - 02 личные данные
 - 03 фамилия
 - 03 имя
 - 03 отчество
 - 02 сведения о рождении
 - 03 место рождения
 - 04 город, поселок, деревня
 - 04 область, район
 - 03 дата рождения
 - 04 число
 - 04 месяц
 - 04 год

Шаблон состоит из условных знаков: X — любой символ; А — буква; 9 — цифра; П — пробел. Число символов указывается повторением условных знаков или записью в скобках; например, четырехзначное обозначение года имеет шаблон 9999, или 9(4). Если число символов переменное, шаблон приводится для максимально возможного их числа; например, дата типа "01 мая 1987" имеет шаблон 9(2) А(8) 9(4). В графе "Частота" указывается средняя частота заполнения данного элемента в документах в долях единицы или процентах. В графах "Использование" и "Дублирование" указывают шифры документов или информационных массивов, в которых используется или повторяется информация, содержащаяся в данном элементе документа.

Для описания и анализа информационных массивов также используют два вида

бланков – для укрупненного описания бланк АМ-1 и для детального – АМ-2 (табл. 2.3 и 2.4). В графах "Последовательность записей" и "Последовательность элементов" указывается индекс, по которому располагаются последовательные записи в массиве или элементы в записи. Графа "Имеет отношение к" должна содержать перечень шифров бланков, в которых описаны данные и процедуры их обработки, необходимые как для формирования, так и для использования описываемого массива. Заполнение остальных граф аналогично описанному выше.

Таблица 2.3

Наименование подразделения			Шифр	
Наименование массива	Носитель		Шифр	Число копий
Последовательность записей в массиве	Последовательность элементов в записях			
Метод подготовки				
Место хранения				
Характеристики массива	мин	макс	среднее	
Объем в знаках				
Количество записей				
Частота обращения к массиву				
Время поиска одной записи				
Имеет отношение к (№ бланков спецификаций)				
Использование	Дублирование			
Кем обновляется				
Частота обновления	мин	макс	среднее	
Объем разового обновления	мин	макс	среднее	

Данные, используемые для обновления

№	Наименование вводимых данных	Ссылка на АД	Ссылка на АП	Примечания

Текущий массив

Время хранения						
Условия передачи в архив						
№	Должность лица	Цель использования	Частота	Объем	Ссылка на АП	Примечания

Архивный массив

Время хранения

Причины хранения

№	Должность лица	Цель использования	Частота	Объем	Ссылка на	Примечания

Дополнительные сведения

Источник	Заполнил	Дата	Проект (тема)	Шифр	АМ-1
	Проверил	Дата	Этап (раздел)	Шифр	№

Т а б л и ц а 2.4

Наименование подразделения				Шифр			
Наименование массива				Шифр			
Специальные требования							
Физические размеры							
Уровень	Наименование записей и элементов	Число символов	Шаблон	Частота	Использование	Дублирование	Примечание

Дополнительные сведения

Источник	Заполнил	Дата	Проект (тема)	Шифр	АМ-2
	Проверил	Дата	Этап (раздел)	Шифр	№

Для записи сведений об алгоритмах решаемых задач обработки данных укрупненно используют бланк АП-1 (табл. 2.5) и подробно – бланк АП-2 (не приводится). На бланке АП-1 указывается, что является исходной информацией для решения задачи или процедуры обработки данных; используемые при выполнении процедуры массивы; получаемая выходная информация. В центральной части приводится схема алгоритма или СИВС решения задачи, если она не приводилась раньше в виде самостоятельного бланка. В соответствии с этой схемой составляется бланк АД-2, на котором детально описываются отдельные операции типа: ввод, сортировка, выборка, сравнение, вычисление, контроль, передача, вывод. Для каждой операции указывается ее смысловое содержание, например сортировка по видам продукции, вычисление среднего значения и т.п.; приводятся сведения о частоте выполнения операции, ее продолжительности, трудозатратах и др.

Таблица 2.5

Наименование процедуры (задачи)					Шифр				
Наименование подразделения (подразделений)					Шифр				
Условия и время выполнения					Кем выполняется				
Способ выполнения					Применяемое оборудование				
Продолжительность выполнения					мин	макс		ср/абс.	
№	Вход	Объем	Частота	Блок-схема (ОСИВС)	№	Выход	Объем	Частота	Примечания

Дополнительные сведения

Источник	Заполнил	Дата
	Проверил	Дата

Проект (тема)	Шифр	АП-1
Этап (раздел)	Шифр	№

§ 2.2. Структуризация системы

Важным этапом системного анализа является структуризация системы — локализация ее границ и выделение структурных составных частей. Выделенную по определенному признаку часть автоматизированной системы управления называют подсистемой АСУ. Совокупность действий, направленную на достижение определенной цели, называют функцией АСУ. Выполнение автоматизированной системой управления функций, осуществляемое на действующем объекте управления и обеспечивающее достижение заданных целей, называют функционированием АСУ (ГОСТ 24.003—84).

Структура системы управления отражает строение и внутреннюю форму организации, относительно устойчивые взаимоотношения и взаимосвязи элементов системы. В каждой АСУ, как и любой сложной системе, можно выделить большое число элементов, свойств, связей между элементами. Охватить их одновременно в рамках одного понятия структуры не представляется возможным. В процессе создания и в ходе функционирования АСУ выделяют некоторые аспекты внутреннего строения системы управления, различая в соответствии с этим виды структуры системы: организационную, функциональную, комплекса технических средств, сбора и передачи информации и др. Организационная структура системы управ-

ления определяет наличие подразделений разного уровня — отделов, подразделов, цехов, участков и др., и их взаимное административное подчинение. Функциональной структурой АСУ называют структуру, элементами которой являются подсистемы, функции автоматизированной системы управления или их части, а связи между элементами — потоки информации, циркулирующей между ними при функционировании АСУ. Структурой комплекса технических средств АСУ называют структуру, элементами которой являются устройства комплекса технических средств АСУ, а связи между элементами отображают информационный обмен (ГОСТ 24003—84).

В системах организационного управления функциональная и организационная структуры часто во многом совпадают. Это объясняется стремлением создать для систематической и квалифицированной реализации определенной функции управления постоянный коллектив людей, работающих под единым руководством. При этом основываются на том интуитивно ясном и многократно проверенном на практике соображении, что такое объединение в одном подразделении наиболее близких по характеру задач и тесно взаимодействующих исполнителей уменьшает объем передаваемой между подразделениями информации, дублирование функций, время на согласование решений и т.д.

Структура систем организационного управления в народном хозяйстве строится по отраслевому или территориальному признаку. Отраслевой принцип позволяет осуществлять единую техническую и экономическую политику в каждой отрасли. Он применяется в тех случаях, когда речь идет о сложных, специфических видах производства. Если ставится задача комплексного использования материальных ресурсов в некотором районе страны, управление строится по территориальному принципу. По этому же принципу построены органы государственного административно-территориального управления.

Число уровней системы существенно влияет на эффективность управления, хотя эта зависимость не может быть в настоящее время выражена аналитически. Чем меньше число уровней, тем меньше задержек и ошибок в передаче информации от нижних уровней к верхним и обратно, тем выше оперативность принятия решений, скорость и точность реакции на возникающие возмущения. С увеличением числа уровней растет общее количество элементов системы, что приводит к большей их обособленности и необходимости усиления горизонтальных связей, которые, как правило, слабее вертикальных. С другой стороны, чем меньше число уровней, тем большая нагрузка приходится на каждый элемент системы, тем труднее осуществлять управление, с чем и связана тенденция к росту числа уровней системы и элементов каждого уровня. Таким образом, в системе организационного управления существуют факторы, определяющие стремление как к увеличению, так и уменьшению числа уровней.

В процессе функционирования организационных систем управления довольно часто наблюдается временное, оперативное перераспределение задач между уровнями системы, получившее название "делегирование пол-

номочий". Оно заключается в том, что в определенной обстановке решение части задач управления передается с некоторого уровня на более низкий. При этом подсистеме более низкого уровня передаются как полномочия вышестоящей подсистемы в пределах решения данных задач, так и ответственность за правильность, своевременность и качество их решения и реализации.

В сложных, необычных ситуациях происходит и обратный процесс — подсистема более высокого уровня, как говорят, "берет управление на себя", т.е. решает задачи и выполняет функции, обычно входящие в компетенцию управляемой подсистемы. Благодаря тому что подсистема более высокого уровня располагает большими полномочиями и ресурсами, решение задач оказывается более эффективным. Однако при этом подсистема верхнего уровня в какой-то мере выполняет несвойственные ей функции, что неизбежно ограничивает возможности по решению задач управления своего уровня. Поэтому после нормализации ситуации решение этих задач вновь передается подчиненной подсистеме.

В структуре организационных АСУ принято выделять подсистемы по функциональному признаку. Это позволяет четко выделять комплексы задач в подсистемах в соответствии с определенной функцией управления и в то же время в большинстве случаев совпадает с существующей организационной структурой предприятия или отрасли, для которых создается АСУ.

Для АСУ предприятий, производственных объединений и отраслей согласно ОРММ [1.5] выделяют подсистемы по следующим функциям управления: перспективным планированием и развитием; технико-экономическим планированием; основным производством; технической подготовкой производства; материально-техническим снабжением; реализацией и сбытом продукции; бухгалтерским учетом; финансами; кадрами; капитальным строительством; научно-исследовательскими и опытно-конструкторскими работами и др.

В ряде случаев возникает необходимость в выделении функциональных подсистем, не предусмотренных рекомендуемым перечнем. Такая необходимость возникает в случаях выявления дополнительной реально выполняемой долговременно действующей функции управления. Так, например, при большом объеме работ по кооперации со смежниками возможно окажется целесообразным выделить в самостоятельный отдел в организационной структуре и в самостоятельную подсистему АСУ функции управления работами по кооперации. Обоснованием для такого выделения служит наличие большого количества специфических задач данного содержания, требующих для их решения получения и переработки больших объемов информации, а также существенное влияние качества решения этих задач на общую эффективность функционирования предприятия, объединения или отрасли в целом.

Кроме функциональных в АСУ выделяют обеспечивающие подсистемы, которые необходимы для успешной работы функциональных подсистем. Различают следующие виды обеспечения АСУ.

Техническое обеспечение — комплекс технических средств, применяемых для функционирования АСУ.

Математическое и программное обеспечение — совокупность математических методов, моделей, алгоритмов и программ обработки информации, используемых при создании и функционировании АСУ.

Информационное обеспечение — совокупность реализованных решений по объемам, размещению и формам организации информации, циркулирующей в АСУ при ее функционировании. Оно включает нормативно-справочную информацию, необходимые классификаторы технико-экономической информации, унифицированные документы, массивы данных, используемые при решении задач АСУ.

Организационное обеспечение — совокупность документов, регламентирующих деятельность персонала АСУ в условиях ее функционирования.

Лингвистическое обеспечение — совокупность языковых средств для формализации естественного языка, построения и сочетания информационных единиц при общении персонала АСУ, в условиях ее функционирования, со средствами вычислительной техники.

Правовое обеспечение — совокупность правовых норм, регламентирующих правоотношения при функционировании АСУ и юридический статус результатов ее функционирования (ГОСТ 24.003—84).

Комплекс технических средств и информационное обеспечение являются общими для всех задач, решаемых в АСУ. В связи со спецификой методов разработки, ввода в действие и дальнейшей эксплуатации их принято выделять в самостоятельные подсистемы. Специфика технических средств определяется необходимостью систематического обслуживания, проверки исправности, профилактики, различных видов ремонта. Эта специфическая область работы АСУ выполняется специалистами по электронике и вычислительной технике и не входит в круг обязанностей специалистов по созданию и эксплуатации АСУ. Однако проектирование этого комплекса, обеспечивающего функционирование АСУ, является прямой обязанностью разработчиков системы.

Остальные виды обеспечения используются применительно к конкретным задачам и, как правило, их в самостоятельные подсистемы не выделяют.

§ 2.3. **Выбор задач, подлежащих автоматизации, и их постановка**

Выбор задач. На основе проведенного изучения и анализа предприятия определяют подлежащие автоматизации функции управления с учетом рекомендуемого перечня функциональных подсистем, выявленных факторов, способствующих и препятствующих повышению эффективности деятельности предприятия, степени и характера подготовленности объекта к переходу на использование экономико-математических методов и вычис-

лительной техники в управлении. При этом наиболее важными являются два обстоятельства: влияние выделенных функций управления на конечный результат, т.е. на повышение эффективности функционирования управляемой системы, и обеспечение автоматизации выделенной функции таким образом, чтобы комплекс задач, ее реализующих, не прерывался отдельными задачами, решаемыми в ином режиме.

Это общее положение уточняется в каждом конкретном случае применительно к основному производственному процессу — движению материальных потоков и последовательно выполняемым операциям технологического процесса.

Известно, что основными этапами управления являются: долгосрочное планирование на год и больше; краткосрочное планирование на квартал, месяц, декаду или неделю, обеспечивающее реализацию годового плана; текущее планирование на день, смену и оперативное управление, направленные на реализацию краткосрочных планов в условиях случайных возмущений; наконец, непосредственное управление технологическим процессом — "ручное" или автоматическое.

Каждый этап в отдельности или несколько этапов охвачены обратной связью, по которой передается информация о результатах реализации принятых решений, используемая в качестве исходной для принятия последующих решений.

Выбор подлежащих автоматизации функций осуществляется в соответствии со схемой уровней управления либо "сверху вниз", либо "снизу вверх". В первом случае важно определить, с какого из верхних уровней следует начинать автоматизацию функций. Если проведенный анализ показал, что долгосрочное планирование осуществляется плохо формализуемыми методами, четко сформулировать цель и критерии эффективности не удастся, методы автоматизации расчета плана не просматриваются, то, следовательно, ни система, ни разработчики не готовы к автоматизации этой функции и ее следует пока оставить без изменения, принимая рассчитанный "вручную" план заданным. Таким же образом рассматривается возможность автоматизации краткосрочного планирования при известном долгосрочном плане; текущего планирования при известном краткосрочном плане и оперативного управления.

Когда некоторый уровень управления представляется возможным автоматизировать, следует определить, целесообразно ли это делать, возрастет ли эффективность управляемого производства настолько, что это окупит затраты на автоматизацию. Необходимые расчеты являются оценочными, прикидочными, здесь допустимо использовать экспертные оценки на основе учета факторов, которые при автоматизации будут способствовать повышению эффективности производства.

Выделив некоторый уровень как подлежащий автоматизации, рассматривают аналогичным образом последующие, вплоть до непосредственного управления технологическим процессом.

Возможно, что, начиная с некоторого уровня, автоматизация

управления вновь окажется пока нецелесообразной. В таком случае вполне допустимо выбрать два или три уровня, непосредственно связанные друг с другом. Однако крайне нежелателен выбор уровней, непосредственно не связанных между собой, например долгосрочное планирование и оперативное управление.

Рассмотрев возможность автоматизации управления основным производством, переходят к вспомогательному производству и другим функциональным подсистемам, определяя возможность и целесообразность автоматизации тех же уровней управления по другим функциям.

Описанный подход может быть применен и к определению последовательности во время создания системы по очередям, для определения состава очередей.

При подходе "снизу вверх" сначала рассматривают нижний уровень — непосредственное управление технологическим процессом, и определяют возможность и целесообразность его автоматизации, а затем переходят к оперативному управлению и т.д. В остальном подход аналогичен описанному выше.

Возможен принципиально иной подход, когда рассматривается в качестве самостоятельного объекта одна или несколько функциональных подсистем, например управление капитальным строительством, материально-техническим снабжением, бухгалтерский учет и к ним применяется один из описанных выше подходов. Таким же образом можно рассматривать комплексы задач внутри функциональных подсистем: составление сетевых графиков строительства, снабжение изделиями определенного профиля, расчет зарплаты и др. Подход является довольно распространенным и дает известный экономический эффект, однако он заведомо ниже, чем мог бы быть при включении данного комплекса задач в подсистему. Кроме того, в этом случае совершенно необходимо предусмотреть возможность последующей стыковки выделенного комплекса задач с другими, иначе его придется практически разрабатывать заново. Избежать серьезной доработки удастся лишь очень опытным разработчикам и только при условии предварительной проработки всей подсистемы в целом. Поэтому такая последовательность автоматизации отдельных функций должна иметь достаточно весомое обоснование.

В полученном тем или иным способом перечне функций, подлежащих автоматизации, их располагают по приоритету, обычно в порядке убывания экономической эффективности их автоматизации. Из этого перечня выбирают первые функции в том объеме, который определяется ограничениями на выделенные ресурсы — количество разработчиков, технические средства, директивные сроки и т.п.

В большинстве случаев в состав очередей включают не функции целиком, а отдельные комплексы задач. Такой подход при ограниченных ресурсах вполне оправдан, однако необходимо помнить, что выделенные комплексы задач должны составлять единую систему, а не просто набор независимых задач. Это удастся сделать, если вначале составить укрупнен-

ную схему будущей системы в разрезе очередей. При этом для каждой очереди должны быть определены взаимосвязи комплексов задач — последовательность решения и выдачи результатов, использования одними комплексами выходной информации других, обработка данных на общих технических средствах, единая информационная база и т.п. Одновременно определяют внешние связи подсистемы по входной и выходной информации в сопоставлении со входами и выходами комплексов задач. Дополнительно проводят анализ воздействия выделенных комплексов задач на управляемую систему, определяя целостность, системность этих воздействий.

При выборе комплексов задач управления оценивают возможность существенного улучшения методов управления по сравнению с применяемыми в "ручной" системе, введения более четкого контроля за основными параметрами хода производства, снижения трудоемкости сбора и обработки информации, повышения оперативности и достоверности используемой при управлении информации.

Постановка задачи. Одной из серьезных трудностей, стоящих перед разработчиками АСУ, является определение перечня задач, реализующих некоторую функцию управления, и их формальная постановка. Трудность заключается в том, что надо сначала понять, как эта функция реализуется в "ручной" системе, какие для этого решают задачи и какие методы используют, а затем постараться выявить иные задачи и найти более эффективные методы. Психологические свойства человека таковы, что чем привычнее становится существующая система, тем труднее представить себе иные способы достижения конечного результата. Однако поиск новых задач и методов совершенно необходим, иначе будет создана система, автоматизирующая все недостатки "ручной".

Как и при выявлении подлежащих автоматизации функций, решению этой проблемы помогает концепция материальных и информационных потоков. Дискретизируя во времени и в пространстве движение материальных потоков, для каждой точки, в которой возможны существенные изменения состояния этих потоков, определяют их чувствительность к управлению и необходимые управляющие воздействия исходя из набора технологических операций, выполняемых до и после этого состояния. Получив набор управляющих воздействий, определяют возможность их группирования по функциям управления. Таким образом выявляют, *по каким функциям управления или подсистемам, какие должны выдаваться управляющие воздействия, в какие моменты времени и в каких точках производственного процесса.* Далее определяют, *какими методами, путем решения каких задач могут быть получены необходимые управляющие воздействия, желательно оптимальные.*

В дальнейшем, сопоставляя определенные на содержательном уровне новые задачи с аналогичными решаемыми в "ручной" системе, оценивают преимущества и недостатки предлагаемых задач и методов относительно существующих. Здесь важно избежать как переоценки, так и недооценки замечаний квалифицированных представителей заказчика, поэтому р про-

цессе обсуждения нужно стремиться к выяснению истинных причин, по которым заказчик не принимает те или иные предложения разработчика.

После уточнения перечня и содержательной постановки задач по каждой функции переходят к формальным постановкам отдельных задач. Суть ее — в формальной записи целевой функции и ограничений для оптимизационных задач или в записи формул и уравнений для задач прямого счета, или в любом ином виде формальной записи содержания задачи.

Формальная постановка задачи сопровождается изложением ее организационно-экономической сущности (описанием задачи на содержательном уровне), с указанием круга объектов, для которых она предназначена; обоснованием необходимости ее решения; описанием исходных данных и получаемых результатов. Определяются связи данной задачи с другими; требования к решению задачи: периодичность решения, предельные сроки получения результатов, необходимая точность расчета и другие специфические условия.

Для каждой задачи должен быть определен метод решения, позволяющий получить требуемые результаты наиболее эффективным путем — с наименьшими затратами ресурсов ЭВМ, простотой ввода данных, если это требуется делать вручную, наглядностью получаемых результатов, удобством их использования. Далее необходимо составить алгоритм расчета в виде схемы с пояснениями: порядок и последовательность ввода исходной информации, ее источник (массив данных или ввод с пульта); порядок расчетов, выдача данных на дисплей или печать, или запись на машинный носитель. Следует предусмотреть методы контроля достоверности входной и выходной информации, а при необходимости — меры защиты от несанкционированного доступа.

Для проверки задачи, охватывающей ее постановку, алгоритмизацию, программирование и проведение расчетов на ЭВМ, подготавливается контрольный пример. Он включает исходные данные, которые могут быть как реальными, так и специально подобранными. Объем их может быть меньшим или равным тем объемам, которые реально ожидаются в системе. Подбор исходных данных не является отладочным тестом для программ, он предназначен для проверки правильности решения данной задачи, с точки зрения пользователя, т.е. реализации некоторой функции управления. Поэтому должен быть предусмотрен метод проверки правильности получаемых результатов. Результаты постановки задач целесообразно представить в виде системных спецификаций.

§ 2.4. Общие характеристики системы

Основной характеристикой системы управления является ее назначение по классу управляемых объектов. Вместе с тем существует ряд параметров, характеризующих свойства системы в целом. К ним относятся: время реакции на входной сигнал, пропускная способность, коэффициент готовности, локализованность, количество пользователей и их удаленность,

средства доступа и общения с пользователями и др. Все эти параметры определяются или задаются при проектировании системы на макроуровне.

Время реакции определяется от момента поступления входного сигнала до появления соответствующего выходного результата. Эта характеристика особенно важна для систем, работающих в реальном масштабе времени. Следует оговориться, что фактически все без исключения системы управления работают в реальном масштабе времени: управляющие воздействия выдаются с таким расчетом, чтобы они могли быть своевременно реализованы. Это относится ко всем системам управления, включая системы перспективного планирования: ясно, что план должен быть сформирован и выдан до начала планируемого периода, а не позже. Однако время выдачи плана не очень критично, минуты, часы и даже дни ничего не решают. Поэтому принято считать, что режимом реального времени является такой, когда для обеспечения необходимой с точки зрения технологического процесса скорости реакции системы управления требуется принимать специальные меры, а задержка выдачи результатов приводит к снижению скорости и сбоям управляемого процесса.

Требования к скорости реализации могут быть различными для единичного входного сигнала и поступающей одновременно пачке сигналов. В большинстве случаев должны быть предусмотрены два режима — ускоренный для обработки единичных сигналов и обычный для пачки. Кроме того, возможны случаи, когда сигналы поступают по одному в случайные моменты времени, но имеют различный приоритет в отношении скорости обработки. Наличие таких или подобных им режимов должно быть выявлено на предпроектной стадии и учтено при проектировании.

Пропускная способность определяется по количеству сигналов, которые могут быть обработаны в единицу времени. При случайном характере поступления сигналов пропускная способность определяет наличие и длину очередей. В системах массового обслуживания, когда число пользователей велико, а время ожидания ими обслуживания критично (при длительном ожидании пользователи зачастую покидают систему необслуженными), систему рассчитывают на максимальную, пиковую нагрузку. Такие пики кратковременны, поэтому в остальное время система недогружена. В подобных случаях рассматривают возможность загрузки системы задачами, не требующими быстрой реакции, допускающими задержку получения результатов. Такие задачи часто называют фоновыми, они как бы образуют фон, на котором происходит решение задач, требующих быстрого решения, с переменной интенсивностью. Если сигналы поступают в систему с переменной интенсивностью, но не требуют немедленной реакции, следует рассмотреть возможность их накопления в некоторой промежуточной емкости — буфере, где они хранятся в ожидании обслуживания.

Коэффициент готовности системы определяется как вероятность того, что она окажется работоспособной в произвольно выбранный момент времени в установившемся (стационарном) режиме эксплуатации. Иными словами, это доля или процент времени, в течение которого систе-

ма фактически работоспособна в стационарном режиме, относительно общего времени, когда она должна быть работоспособна. Подавляющее большинство нарушений работоспособности в АСУ связано со сбоями или выходом из строя различных элементов технических средств; в меньшей степени, но заметно, сказываются недостатки в программном обеспечении.

Время восстановления обуславливает максимальную и среднюю продолжительность нерабочего состояния системы и распределение перерывов в работе по времени. Пользователю не безразлично, происходит ли в системе один перерыв в сутки продолжительностью 30 мин или один перерыв на 2 мин каждый час, хотя общее время неработоспособности в обоих случаях одинаково.

Каким бы высоким ни был коэффициент готовности, разработчик должен решить ряд вопросов, связанных с возможными перерывами работы, и в первую очередь определить, что должно происходить в это время в системе. Помимо обычных мер по вводу в действие резерва, если он экономически оправдан, следует так разрабатывать систему, чтобы обеспечить возможность автономной работы в течение некоторого времени всех элементов системы, кроме, естественно, вышедших из строя.

Восстановление нормального режима не должно требовать от пользователя каких-либо специальных мер и операций, он должен просто продолжать прерванную работу. Это означает, что разработчик должен предусмотреть возможность автоматического запоминания состояния системы в момент, непосредственно предшествующий аварийной ситуации, и полного восстановления этого состояния после ее ликвидации. Все операции, выполненные в автономном режиме, должны быть впоследствии обработаны так, как если бы они выполнялись в нормальном режиме. Если провести полное восстановление системы в автоматическом режиме по каким-либо причинам невозможно, система должна подсказывать пользователю, какие операции ему необходимо дополнительно выполнить и с какого места начать повторение операций прерванной задачи. В любом случае недопустима потеря какой-то части введенной в систему информации.

Как указывалось выше, система должна быть ориентирована на удовлетворение интересов пользователя. Недопустимо вводить какую-либо регламентацию поведения пользователя исходя из интересов или удобства разработчиков. Упрощая себе процесс разработки за счет предъявления излишних требований к пользователям, разработчики рискуют создать нежизнеспособную систему, к которой пользователи просто не будут обращаться. Сказанное в полной мере относится к нарушениям нормальной работы системы. Если они серьезно влияют на ритм работы пользователя, требуют от него существенной дополнительной работы, то есть большая вероятность отказа от такой системы.

Значительную трудность для разработчика представляет анализ возможных конфликтных ситуаций. Многие разработчики избегают эти трудности самым простым способом — игнорируя их. Предписания и инструкции, регламентирующие нормальный режим работы, не гарантируют отсутствия

конфликтов между пользователями и системой. Такие конфликты возникают при случайных или преднамеренных действиях пользователя, нарушающих установленные правила. Так, например, может быть утерян выданный ЭВМ документ, не введены в систему вовсе или введены в искаженном виде некоторые исходные данные для расчетов, не использованы в процессе управления полученные на ЭВМ результаты и т.п. Выявленные возможные конфликтные ситуации служат основой для принятия разработчиком решения — как должна функционировать система в этих условиях. Лучше всего, если система немедленно реагирует на эти ситуации, вводя необходимые дополнительные операции, предотвращающие нежелательные последствия. Более простым способом является фиксация в памяти системы происшедших нарушений с последующей выдачей их для анализа и принятия мер, направленных на исключение подобных нарушений в дальнейшей работе.

§ 2.5. Техничко-экономическая эффективность системы

Основным источником экономического эффекта от создания АСУ является улучшение экономических показателей управляемой системы — предприятия или организации, достигаемое за счет повышения качества управления.

На экономические показатели предприятия или производственного объединения оказывает существенное влияние множество факторов, среди которых можно выделить совершенствование технологии производства, модернизацию оборудования, изменение номенклатуры продукции, культуру производства, квалификацию работающих, организацию производства, общественные и социальные мероприятия и многие другие. Поэтому выделить в чистом виде зависимость экономических показателей предприятия от ввода в эксплуатацию АСУ практически невозможно. Существующие методики предполагают сравнение показателей работы предприятия до и после внедрения АСУ при прочих примерно равных условиях. Кроме того, результаты производственно-хозяйственной деятельности в значительной степени зависят не только от решения отдельных комплексов задач, но и связей между ними. Неподготовленность или неудачное решение одной задачи может привести к неудовлетворительным результатам по нескольким комплексам задач.

Во многих случаях источник экономического эффекта лежит за пределами системы. Например, повышение качества продукции, ритмичности ее выпуска улучшает показатели работы потребителей, но практически не сказывается на экономических показателях производителя.

Эти и другие факторы, определяющие специфику создания и функционирования АСУ по сравнению с техническими системами, вносят известную неопределенность в расчет экономических показателей, характеризующих технико-экономическую эффективность создания и внедрения АСУ.

Основным показателем экономической целесообразности создания АСУ предприятиями и производственными объединениями является годовой экономический эффект, выражаемый в виде годового прироста прибыли (годовой экономии). Отношение годового прироста прибыли к затратам на создание АСУ определяет экономическую эффективность ее создания.

Годовой прирост прибыли, тыс. руб.,

$$\Pi^A = [(A_2 - A_1)/A_1]\Pi_1 + [(C_1 - C_2)/100]A_2,$$

где A_1, A_2 – годовой объем реализуемой продукции, соответственно до внедрения АСУ и на год, следующий за годом ввода системы в эксплуатацию, тыс. руб.; C_1, C_2 – затраты на 1 руб. реализуемой продукции соответственно до и после внедрения АСУ, коп.; Π_1 – прибыль от реализации продукции до внедрения АСУ, тыс. руб.

Первое слагаемое этой формулы определяет годовой прирост прибыли за счет роста объема реализуемой продукции, а второе – за счет снижения издержек производства.

К сумме прибыли прибавляют также сумму снижения непроизводительных потерь в виде штрафов, неустоек, если эти расходы, не входящие в себестоимость продукции, снижены за счет внедрения АСУ. Если повышение качества продукции не отражено в виде надбавок к действующим ценам, дополнительная прибыль за счет этого фактора, образующаяся у потребителей, также может быть включена в сумму годового прироста прибыли при условии подтверждения этого факта в установленном порядке предприятиями-потребителями.

Расчетный коэффициент эффективности капитальных вложений на создание АСУ

$$E_p = (\Pi^A / K_K^A) \geq E_{\text{нвт}}^A.$$

где K_K^A – капитальные вложения на создание АСУ, определяемые по приросту основных фондов, связанному с созданием АСУ, тыс. руб.; $E_{\text{нвт}}^A$ – нормативный коэффициент эффективности капитальных вложений на создание АСУ, утверждаемый Госпланом СССР для каждой отрасли.

Срок окупаемости капитальных вложений, годы,

$$T = K_K^A / \Pi^A.$$

Если в АСУ обеспечивается решение социальных и других специальных задач, по которым не может быть подсчитан экономический эффект по реализуемой продукции, то коэффициент эффективности капитальных вложений по согласованию с Госпланом СССР допускается ниже нормативного. Однако в целом по министерству нормативный коэффициент должен соответствовать утвержденному для данной отрасли.

При расчете экономической эффективности подсистемы или комплекса задач доля общих затрат, связанных с проектированием, созданием и внедрением АСУ, рассчитывается по коэффициенту загрузки, определяемому как отношение фактических затрат времени работы ЭВМ при функционировании данной подсистемы или решении комплекса задач к полному полезному фонду времени работы ЭВМ за год.

Предполагается, что создание АСУ на предприятиях и в производственных объединениях дает основной эффект за счет увеличения объема выпускаемой продукции на существующих производственных мощностях путем повышения производительности труда и лучшего использования оборудования. Прирост продукции ΔA^A обеспечивает годовой объем производства $A_2 = A_1 + \Delta A^A$ и коэффициент роста производства продукции $\gamma = A_2/A_1$.

В связи с внедрением АСУ снижается себестоимость продукции. Состав статей себестоимости, способы их расчета и методы калькулирования определяются специальными нормативными документами.

При расчетах себестоимости затраты делят на условно-постоянные и условно-переменные.

К условно-постоянным относятся расходы, величина которых практически не зависит от объема выпуска продукции: около 30% затрат на содержание оборудования и рабочих мест, 20% — на текущий ремонт и 40% — на амортизацию производственно-

го оборудования и транспортных средств.

Условно-переменные называются затраты, зависящие от объема выпускаемой продукции. Их экономия в тыс. руб. за год $\Delta C_{\text{пр}}^{\text{п}} = \Delta C_{\text{м}}^{\text{п}} - \Delta C_{\text{т}}^{\text{п}}$ — от снижения материальных затрат; $\Delta C_{\text{т}}^{\text{п}}$ — снижение затрат на топливо и эр-гию на технологические цели; $\Delta C_{\text{м}}^{\text{п}}$ — экономия фонда заработной платы производ-ственных рабочих; $\Delta C_{\text{др}}^{\text{п}}$ — уменьшения потерь от брака; $\Delta C_{\text{н}}^{\text{п}}$ — снижения непроиз-водительных затрат; $\Delta C_{\text{зп}}^{\text{п}}$ — экономия затрат на подготовку и освоение производ-ства; $\Delta C_{\text{об.пр}}^{\text{п}}$ — снижение затрат на содержание и эксплуатацию оборудования. Ско-номия по составляющим условно-переменных затрат определяют прямым счетом и исключают из соответствующих статей калькуляции себестоимости.

По ряду пунктов себестоимости экономия определяется косвенно. Условно-переменная часть этих затрат, тыс. руб.,

$$C_{\text{об.пр}}^{\text{п}} = C_{\text{об.пр}} + C_{\text{об.пр}}^{\text{п}}$$

где $C_{\text{об.пр}}^{\text{п}}$ и $C_{\text{об.пр}}$ — условно-переменная часть затрат на содержание и эксплуатацию производственного оборудования соответственно в условных функционированиях АСУ и до ее внедрения; $C_{\text{об.пр}}^{\text{п}}$ — условно-постоянные затраты на содержание и эксплуата-цию оборудования после внедрения АСУ. Переменная часть этой статьи составляет примерно 85% общих затрат.

Так как в их составе изменяется только условно-переменная часть, в среднем прирост на общезаводском уровне составляет 0,3, а на цеховом — 0,4 от прироста объема производства.

Цеховые затраты, после внедрения АСУ, тыс. руб.,

$$C_{\text{ц}}^{\text{п}} = C_{\text{ц}}^{\text{п}} [1 + (\gamma - 1) D] - \Delta C_{\text{н}}^{\text{п}} - \Delta C_{\text{др.ц}}^{\text{п}}$$

где $C_{\text{ц}}^{\text{п}}$ — цеховые затраты до внедрения АСУ; D — коэффициент зависимости их прироста от прироста объема производства; $\Delta C_{\text{н}}^{\text{п}}$ — экономия цехового фонда зарплат; $\Delta C_{\text{др.ц}}^{\text{п}}$ — снижение непроизводительных цеховых затрат.

Общезаводские затраты в условиях функционирования АСУ, тыс. руб.,

$$C_{\text{з}}^{\text{п}} = C_{\text{з}}^{\text{п}} [1 + (\gamma - 1) D] - \Delta C_{\text{з}}^{\text{п}} - \Delta C_{\text{др.з}}^{\text{п}}$$

где составляющие аналогичны указанным выше для цеховых расходов. Непроизводительные затраты $C_{\text{н}}^{\text{п}}$ возрастают пропорционально росту произ-

водства.

Затраты на эксплуатацию АСУ и содержание вычислительного центра опре-деляют по следующим статьям: основная и подотчетная заработная плата perso-нала вычислительного центра с отчислениями на социальное страхование исходя из расчета численности персонала и утвержденной схемы должностных окладов; затраты на производственное потребление энергии; амортизация оборудования и линий связи; используемые материалы; текущий ремонт технических средств; на-

себестоимость продукции после внедрения АСУ, тыс. руб.,

$$C_{\text{п}}^{\text{п}} = C_{\text{м}}^{\text{п}} + C_{\text{т}}^{\text{п}} + C_{\text{н}}^{\text{п}} + C_{\text{об}}^{\text{п}} + C_{\text{эксп}}^{\text{п}} + C_{\text{мш}}^{\text{п}} + C_{\text{н}}^{\text{п}} + C_{\text{др}}^{\text{п}}$$

где $C_{\text{мш}}^{\text{п}}$ — прочие производственные расходы.

Общую экономию от снижения себестоимости определяют следующим образом. Находят затраты на 1 руб. продукции до и после внедрения АСУ C_1 и C_2 соответствен-

но: $C_1 = C/A_1$ и $C_2 = C^A/A_2$, где C и C^A – себестоимость продукции до и после создания АСУ. Тогда общая экономия

$$(C_1 - C_2)A_2/100.$$

Затраты на создание АСУ K_K^A определяют как сумму следующих видов затрат: $K_{п}^A$ – предпроизводственные затраты; $K_{об}^A$ – затраты на оборудование и строительно-монтажные работы для АСУ; $\Delta O_{об}^A$ – изменение величины оборотных средств (разность объемов оборотных средств до и после внедрения АСУ); $K_{л}^A$ – остаточная стоимость ликвидируемого оборудования, зданий, сооружений, использование которых невозможно, за вычетом $K_{в}^A$, используемых в АСУ или на других участках.

Предпроизводственные затраты на отдельную подсистему или комплекс задач определяются в соответствии с договором на разработку АСУ или по действующей сметной стоимости научно-исследовательских и опытно-конструкторских работ.

Остаточную стоимость высвобождаемого и ликвидируемого оборудования, зданий, сооружений определяют как разность

$$K_{в}^A - K_{л}^A = K_{п.ст}^A (1 - H_a T_{экс}),$$

где $K_{п.ст}^A$ – первоначальная стоимость; $T_{экс}$ – длительность эксплуатации действующего производственного оборудования, зданий, сооружений; H_a – годовая норма амортизации на полное восстановление, тыс. руб./год.

По данным о снижении себестоимости и дополнительных капитальных вложениях определяют основные показатели эффективности АСУ: годовой прирост прибыли или годовую экономию $П^A$, эффективность затрат E_p и срок окупаемости T .

§ 2.6. Научно-технический уровень АСУ

Специфика создания и функционирования АСУ затрудняет оценку ее качества. Для одной и той же управляемой системы могут быть созданы самые различные АСУ, внешне вполне удовлетворительно выполняющие свои функции, но на самом деле качественно отличающиеся друг от друга. Тем более затруднительно сопоставлять системы для разных объектов.

Для интегральной оценки в виде некоторого обобщенного количественного показателя степени соответствия качества АСУ поставленным при ее создании задачам, сравнения уровня разработки различных по назначению систем, сопоставления с тенденциями и уровнем научно-технического прогресса введен *показатель научно-технического уровня (НТУ)* АСУ предприятиями и производственными объединениями. Оценка НТУ проводится на стадиях планирования, разработки, внедрения и функционирования АСУ.

На *стадии планирования* определяют плановый НТУ исходя из анализа тенденций изменения частных показателей НТУ, отражающих ранее решавшиеся задачи управления и предполагаемые для решения в планируемом периоде.

На *стадии разработки* проектов оценка НТУ используется для уточнения проектного уровня системы, сопоставления выполненных проектов с заданным проектным уровнем, выбора методов проектирования, обеспечивающих достижение заданного НТУ, определения необходимых для повышения НТУ научно-исследовательских работ. Задают значение НТУ

исходя из тенденций развития отрасли народного хозяйства под влиянием научно-технического прогресса.

На *стадии внедрения и функционирования* оценка достигнутого *НТУ* позволяет определить направление развития системы и соответствие *НТУ* проектным условиям.

В соответствии со стадиями создания АСУ различают потенциальный, прогнозируемый, проектный и достигнутый *НТУ* АСУ.

Потенциальный НТУ — это наивысший из возможных, он используется для оценки перспектив развития АСУ.

Прогнозируемый НТУ получают для различных вариантов АСУ; он позволяет выбрать наилучший из возможных вариантов в условиях ограничений по времени разработки и выделенным ресурсам.

Проектный НТУ определяется конкретными условиями создания и функционирования АСУ. Он не должен быть меньше некоторого уровня, минимально необходимого для возмещения затрат на создание и эксплуатацию системы.

Достигнутый НТУ определяют по фактическим значениям показателей работы АСУ в реальных условиях.

При выборе показателей для определения *НТУ* предъявляют следующие требования: каждый показатель должен характеризовать элемент или совокупность элементов, от которых зависит *НТУ* АСУ, а совокупность показателей — АСУ в целом; показатель должен содержать количественную оценку; число показателей должно быть ограничено, чтобы сбор значений и их обработка были достаточно простыми и нетрудоемкими; допустимо для значений показателя применять безразмерную шкалу.

Схема образования показателя оценки *НТУ* АСУ приведена на рис. 2.1.

Оценка *НТУ* для АСУ в целом определяется согласно [1.9] по формуле

$$Y_A = K_{Эс} Y_Э + Y_С + K_{За} Y_{За} + K_{Опт} Y_{Опт},$$

где $K_{Эс} + 0,1 K_{За} + 0,1 K_{Опт} = 1$; $K_{Эс} = 0,4$; $K_{За} = K_{Опт} = 3$; $Y_Э$ — экономический уровень; $Y_С$ — системотехнический уровень; $Y_{За}$ — уровень охвата автоматизацией задач управления; $Y_{Опт}$ — уровень организации производства и труда.

Экономический уровень

$$Y_Э = (T_H / T_H)^{1/3},$$

где T_H и T_H — сроки окупаемости соответственно нормативный и для рассматриваемой системы.

Системотехнический уровень

$$Y_С = \sum P_j Y_j, \quad j = \overline{1,3},$$

где P_1, P_2, P_3 — весовые коэффициенты важности показателей: Y_1 — организационного, Y_2 — технического и Y_3 — математического обеспечения. Значения P_1, P_2, P_3 в зависимости от класса предприятия приведены в табл. 2.6.

В свою очередь

$$Y_1 = Y_{\Pi} \sum P_{1i} Y_{1i}; \quad i = \overline{1,4}.$$

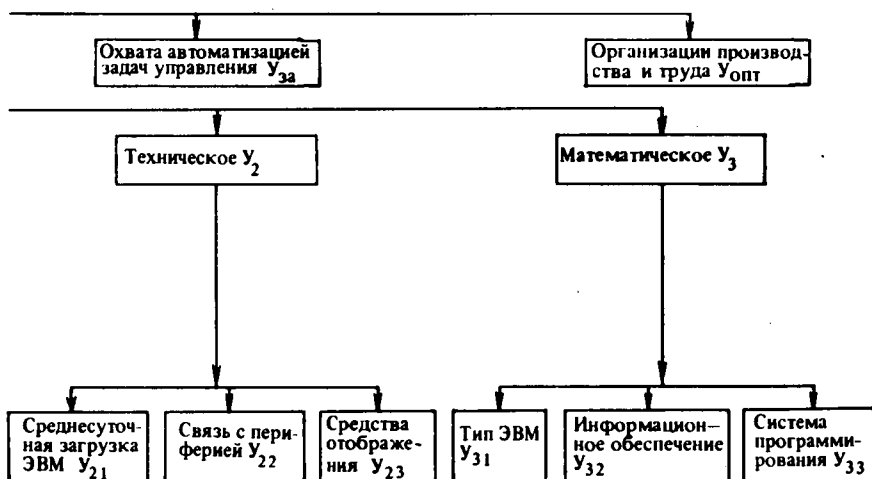


Рис. 2.1

Значение U_{Π} определяется в зависимости от принятой методологии проектирования: при автоматизированном проектировании $U_{\Pi} = 1,0$; на базе типовых проектных решений $U_{\Pi} = 0,8$; в соответствии с руководящими методическими материалами прототипов $U_{\Pi} = 0,7$ и при оригинальном проектировании $U_{\Pi} = 0,6$. Значения U_{11} определяются подготовкой исходных данных. При индивидуальной подготовке $U_{11} = 6$; при централизованной $U_{11} = 10$ и при комбинированной $U_{11} = 8$. В зависимости от использования данных показатель U_{12} принимает значения: при регламентации использования документа $U_{12} = 5$ и при регламентации использования показателя $U_{12} = 10$.

Т а б л и ц а 2.6

Число работающих (класс предприятия)	Тип производства	Виды обеспечения		
		организационное	техническое	математическое
		P_1	P_2	P_3
До 2000	Индивидуальное	0,6	0,3	0,1
	Серийное	0,5	0,3	0,2
	Массовое	0,4	0,3	0,3
2000–8000	Индивидуальное	0,7	0,2	0,1
	Серийное	0,5	0,2	0,3
	Массовое	0,3	0,2	0,5
Свыше 8000	Индивидуальное	0,7	0,2	0,1
	Серийное	0,4	0,2	0,4
	Массовое	0,1	0,2	0,7



По устойчивости к нарушениям различают системы, функционирующие при отказе элемента АСУ с нарушением — $Y_{13} = 5$; без нарушения за счет резервирования функций персоналом $Y_{13} = 8$ и за счет резервирования средств автоматизации — $Y_{13} = 10$. Если доля оптимизационных задач в общем их числе составляет до 5%, то $Y_{14} = 3$; от 5 до 20% — $Y_{14} = 8$ и свыше 20% — $Y_{14} = 10$.

Показатель уровня технического обеспечения

$$Y_2 = Y_{\Pi} \sum P_{2i} Y_{2i}; \quad i = \overline{1,3}.$$

Значения P_{2i} равны: $P_{21} = 0,5$; $P_{22} = 0,3$; $P_{23} = 0,2$. При среднесуточной загрузке ЭВМ до 8 ч $Y_{21} = 3$; до 14 ч — $Y_{21} = 5$; до 18 ч — $Y_{21} = 8$, свыше 18 ч — $Y_{21} = 10$. При неавтоматической связи с вычислительным центром $Y_{22} = 2$; при полуавтоматической $Y_{22} = 6$ и при автоматической $Y_{22} = 10$. Если в системе предусмотрена сигнализация при возникновении отклонений, то $Y_{23} = 3$; при регламентных средствах отображения $Y_{23} = 7$; при ответно-запросных устройствах $Y_{23} = 10$.

Показатель уровня математического обеспечения

$$Y_3 = Y_{\Pi} \sum P_{3i} Y_{3i}; \quad i = \overline{1,3}.$$

Значения P_{3i} равны: $P_{31} = 0,3$; $P_{32} = 0,5$; $P_{33} = 0,2$. Для машин серии ЕС и других этого класса $Y_{31} = 8$. При локальном решении задач информационного обеспечения $Y_{32} = 3$; при едином нормативном хозяйстве $Y_{32} = 5$ и с единой информационной базой $Y_{32} = 10$. Если система программирования математического обеспечения относится к алгоритмическим языкам, $Y_{33} = 5$, а для операционных систем $Y_{33} = 10$.

Уровень охвата автоматизацией задач управления определяют по формуле

$$Y_{3a} = N_a / N_{\text{ст}},$$

где N_a — количество задач, решаемых автоматизированным способом; $N_{\text{ст}}$ — число задач, которые возможно автоматизировать для данного класса предприятия. Для более точных расчетов может быть дополнительно введена оценка сложности задач, учитывающая состав планируемых и отчетных показателей в задаче и их важность для выполнения определенной функции управления.

Показатель уровня охвата автоматизацией задач управления может быть определен по формуле

$$y_{\text{за}} = \frac{0,1 \sum_{i=1}^n P_{pi} + 0,6 \sum_{i=1}^m P_{zi} + 0,3 \sum_{i=1}^k P_{mi}}{P_0} + \frac{0,2 \sum_{i=1}^l O_{mi} + 0,3 \sum_{i=1}^p O_{ci} + 0,5 \sum_{i=1}^s O_{ri}}{O_0},$$

где P_p – количество плановых показателей в выходных документах, полученных на основе алгоритма, отображающего традиционные методы планирования; P_z – количество плановых показателей в выходных документах, полученных в результате использования методов оптимизации; P_m – количество плановых показателей в выходных документах, полученных при использовании алгоритма, предусматривающего выдачу нескольких вариантов планов и выбор одного из них управленческим персоналом в зависимости от конкретной производственной ситуации; P_0, O_0 – общее количество соответственно плановых и отчетных показателей предприятия, определяемое в результате анализа документооборота предприятия на стадии проектирования; n, m, k – количество задач, решаемых в АСУ для получения соответственно показателей P_p, P_z, P_m ; O_m – количество отчетных показателей, полученных в выходных формах документов с месячной периодичностью; O_r – количество отчетных показателей, полученных в выходных формах документов с периодичностью более суток; l, p, s – количество задач, решаемых в АСУ для получения соответственно показателей O_m, O_e, O_r .

Уровень организации производства и труда по предприятию в целом

$$y_{\text{отп}} = \frac{C_1 y_{\text{итр}} + (C_2 y_{\text{оф}} + C_3 y_{\text{об}}) E_n}{C_1 + (C_2 + C_3) E_n} y_k,$$

где C_1 – зарплата производственных рабочих, руб.; C_2 – средняя стоимость основных фондов, руб.; C_3 – средняя фактическая величина нормируемых оборотных средств, руб.; $y_{\text{итр}}$ – уровень использования трудовых ресурсов; $y_{\text{оф}}$ – уровень использования основных фондов; $y_{\text{об}}$ – уровень использования оборотных средств; y_k – уровень качества продукции; E_n – нормативный коэффициент эффективности капитальных вложений.

Схема получения входящих в $y_{\text{отп}}$ показателей приведена на рис. 2.2.

Уровень использования трудовых ресурсов

$$y_{\text{итр}} = K_1 K_2 K_3,$$

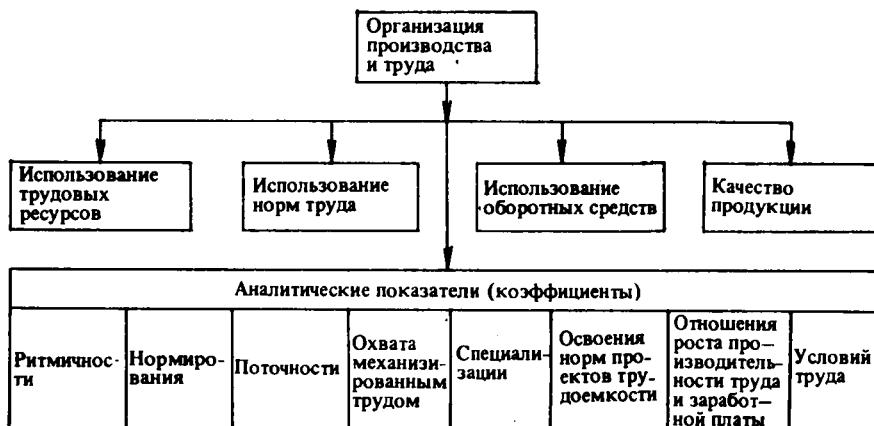


Рис. 2.2

где K_1 , K_2 и K_3 — коэффициенты соответственно использования рабочего времени, интенсивности труда и использования квалификации рабочих, определяемые по формулам:

$$K_1 = 1 - (\Sigma \Pi / \Phi_p),$$

где Π — количество рабочих; Π — потери рабочего времени одним рабочим; Φ_p — общий фонд рабочего времени за вычетом потерь;

$$K_2 = \Sigma S_{ср} / \Sigma S_{ср.пр},$$

где $S_{ср}$ — средний процент нормы выработки; $S_{ср.пр}$ — средний прогрессивный процент выполнения плана;

$$K_3 = \Sigma h / \Sigma h_1,$$

где h — тарифная ставка выполнения работ; h_1 — тарифная ставка рабочих.

Уровень использования основных фондов

$$U_{оф} = K_4 K_5;$$

где

$K_4 = \Sigma \frac{CL}{n_o} / \Sigma \frac{C_i}{n_o}$; $K_5 = N_{общ} / \Sigma 0,8 N_e T_p L$, C — стоимость установленного оборудования; C_i — стоимость оборудования на балансе; L — показатель использования оборудования по времени; 0,8 — КПД оборудования; $N_{общ}$ — общий расход электроэнергии; N_e — мощность главного привода; T_p — располагаемый фонд времени работы оборудования; n_o — число единиц оборудования.

Уровень использования оборотных средств

$$U_{об} = K_{оф} / K_{опл};$$

где $K_{опл} = \Pi / C_o$, $K_{оф}$ и $K_{опл}$ — фактическая и плановая оборачиваемость оборотных средств; Π — выручка от реализации (по оптовым ценам); C_o — средний остаток оборотных средств по плану.

Уровень качества продукции

$$U_k = 1 - \Sigma V_i / C_{п}, \quad i = \overline{1,4},$$

где V_1 — стоимость забракованной продукции; V_2 — затраты по реализации; V_3 — затраты на гарантийный ремонт; V_4 — прочие затраты, связанные с выпуском продукции ненадлежащего качества; $C_{п}$ — себестоимость за планируемый период.

Аналитические показатели уровня организации производства и труда представляют собой следующие коэффициенты:

коэффициент ритмичности

$$K_6 = 1 - \Sigma D / \Pi;$$

где D — невыполнение плана за день или декаду; Π — продукция по плану;

коэффициент нормирования труда

$$K_7 = (P_c / P_o) K_n,$$

где P_c — число рабочих-сдельщиков; P_o — общее число рабочих; K_n — коэффициент напряженности норм;

коэффициент поточности

$$K_8 = \Pi_1 / P_o,$$

где Π_1 — число рабочих, занятых на поточных линиях;

коэффициент охвата механизированным трудом

$$K_9 = B_1 / P_o,$$

где B_1 — число рабочих, занятых механизированным трудом;

коэффициент специализации

$$K_{10} = \sum_{n_{\text{и}}} t_i d / T,$$

где t_i – трудоемкость i -го изделия; d – удельный вес изделия в общей трудоемкости;
 T – общая трудоемкость в нормо-часах; $n_{\text{и}}$ – число видов изделий;
коэффициент освоения норм проектов трудоемкости

$$K_{11} = \sum_{n_i} t_i N_i q / T,$$

где t_i – трудоемкость i -го изделия; N_i – годовая программа i -го изделия; q – коэффициент освоения проектной трудоемкости; T – трудоемкость на программу; n_i – число изделий;

коэффициент отношения роста производительности труда и заработной платы

$$K_{12} = K_{\text{рпт}} / K_{\text{рз}},$$

где $K_{\text{рпт}}$ – коэффициент роста производительности труда за анализируемый период;
 $K_{\text{рз}}$ – коэффициент роста заработной платы;
коэффициент условий труда

$$K_{\text{ут}} = \sum m a / \sum m,$$

где m – количество рабочих обследованных подразделений; a – коэффициент соответствия показателей условий труда нормативным.



Развитие АСУ пошло по пути создания функциональных подсистем, аналогично функциональным подразделениям административно-организационного управления. Этим определяется и структура системы и состав решаемых в подсистемах задач. Такой традиционный подход сохранен и в данной книге.

В перспективе можно ожидать перехода к принципиально новому подходу, когда в основу построения подсистем будет положена структура технологического процесса, для которого создается система управления.

По мере движения материального потока будут последовательно решаться задачи управления его движением, необходимой технологической обработки, обеспечения энергией, транспортными операциями и т.д.

Возможность использования на каждом рабочем месте встроенных или локальных вычислительных средств с объединением их в локальную сеть позволит создать гибкую управляющую систему, легко перестраиваемую в производстве.

§ 3.1. Общие положения

Проектирование АСУ должно основываться на системной методологии. *Системный подход* в широком смысле означает, что прежде всего должны быть сформулированы цели, критерии эффективности и ограничения для системы в целом. Структуризация системы и дальнейшее ее проектирование должны охватывать как управляющую, так и управляемую части системы. Взаимосвязанное проектирование этих двух основных частей системы в наилучшей степени соответствует установленным целям и критериям. Однако для таких сложных систем, какими являются предприятия и производственные объединения, указанный подход крайне затруднен и имеются лишь отдельные попытки осуществить такое взаимосвязанное проектирование.

Не существует пока и отработанных методов построения модели функционирования предприятия, а на уровне отрасли не удастся определить в количественных терминах цель и критерии эффективности.

В более узком смысле под *системным проектированием* понимают проведение системного анализа объекта управления и на этой основе проектирование АСУ. Если только предприятие не требует коренной реконструкции по его техническому и технологическому уровню, то в основном считаются заданными как технология, так и существующая структура системы управления. Констатация такого положения не означает согласия с тем, что оно является правильным и допустимым. Приходится признать, что существующий в этой области разрыв между теорией и практикой, чрезвычайная сложность объекта управления, большинства решаемых задач вынуждают мириться с существующим подходом к созданию АСУ.

Наиболее распространенная постановка общей задачи — обеспечение производства продукции в заданном объеме и ассортименте при минимизации совокупных затрат на ее производство. Максимизация производства отдельных наиболее важных видов производимой продукции применяется в тех случаях, когда ресурсы ограничены, продукция заведомо эффективна и дефицитна, а производство ее рентабельно и прибыльно. Если спрос на продукцию существенно зависит от принятых или перспективных цен, а объем и ассортимент продукции заранее не заданы и должны быть определены в процессе решения задачи, то обычно стремятся максимизировать прибыль.

Указанные постановки применимы главным образом к задачам планирования. Их декомпозиция в виде дерева целей по функциональным подсистемам может быть выполнена лишь экспертными методами. Поэтому в большинстве случаев проектировщики проводят оптимизацию по отдельным задачам в функциональных подсистемах, предполагая, что это улучшает работу системы в целом. Если разработка ведется по нескольким направлениям — одновременно разрабатываются задачи функциональных подсистем, информационного и технического обеспечения, то важно уметь сохранить взаимосвязь всех частей разработки, помнить при решении конкретных задач, особенно касающихся обеспечивающих подсистем, о системе в целом.

Выбор решений по техническим средствам во многом определяет возможности решения функциональных задач и, наоборот, принятые функциональные задачи в свою очередь предъявляют требования к комплексу технических средств. Сказанное относится в полной мере к информационному и другим видам обеспечения.

Для ускорения и снижения затрат на проектирование в настоящее время разработаны *пакеты прикладных программ* (ППП). Построение ППП по модульному принципу позволяет набирать из них программное обеспечение АСУ для конкретного предприятия. При этом разработчики ППП создают определенную *функциональную избыточность*, что позволяет выбирать из модулей с одинаковым функциональным назначением тот,

который наиболее близок к условиям разрабатываемой системы. Независимость модулей позволяет использовать их в любых комбинациях. Вместе с тем разработчики системы должны помнить о необходимости учета ее индивидуальных особенностей. К ним относятся не только внешние по отношению к системе факторы, но и такие условия, как требования к частоте и точности решения задач, наличие и доступность исходной информации и т.д.

Другим уровнем типизации являются *типовые проектные решения* (ТПР). Их создание определяется возможностью декомпозиции системы управления на отдельные элементы, которые для различных систем могут быть одинаковыми. Выделение таких элементов, пригодных для многократного использования, и привело к концепции типовых проектных решений. В практике создания АСУ ТПР разрабатывались для различных уровней, вплоть до создания соответствующих программ. Однако на этом уровне они оказались не вполне удобными и вместо них употребляются пакеты прикладных программ. Применение ТПР может быть полезным при постановке задач и разработке математического обеспечения. Область их использования определяется, с одной стороны, степенью типизации задач, а с другой — близостью рассматриваемой задачи к тому классу, на который ориентировано данное ТПР.

Все ТПР разделяют на классы — *Задача, Техника, Персонал*.

Класс ТПР *Задача* предназначен для решения задач управления и, в свою очередь, разбивается на подклассы, каждый из которых ориентирован на определенную функциональную подсистему. В задаче выделяются ее организационно-экономическая сущность, алгоритм решения, исходные и выходные данные.

Класс ТПР *Техника* определяет состав, размещение и порядок использования комплекса технических средств.

Класс ТПР *Персонал* регламентирует действия персонала в условиях функционирования АСУ, определяет права, обязанности и ответственность лиц, работающих в АСУ.

Одним из основных условий создания высокоэффективной АСУ является ориентация ее на пользователя. При функционировании АСУ, решении задач управления действует большое количество ограничений, которые надо учесть при ее разработке. В процессе самого проектирования также существует много ограничений. Это приводит к тому, что в поисках лучшего пути, за который часто принимают наиболее простой, быстрый и дешевый, разработчики сознательно или подсознательно перекладывают часть возникающих проблем на пользователя. Этот путь чреват пагубными последствиями. Пользователи, в свою очередь руководимые стремлением минимизировать объем своей работы, не выполняют инструкции разработчика и игнорируют систему, которая не облегчает, а усложняет им жизнь. Следует при этом учитывать основную особенность АСУ — задачи управления могут решаться и решались до ее создания "вручную", без использования ЭВМ. В принципе их можно решать "вручную" и при наличии

АСУ. Весь вопрос только в качестве, эффективности решений. Именно поэтому встречаются случаи, когда АСУ работает сама по себе, без какого-либо влияния на управляемый объект, а управление объектом осуществляется фактически без нее. Недопустимость создания подобных систем очевидна.

АСУ является инструментом управления, в котором решающую роль играет человек. Кроме таких задач, которые регламентированы руководящими материалами, многие задачи, причем не только справочного характера, но и оптимизационные, в определенной степени зависят от лица, которое их использует в процессе принятия решения. Наиболее индивидуальны и субъективны требования в отношении частоты и формы представления справочной и отчетной информации. Но даже и по основным функциям управления достаточно часто назначение нового руководителя влечет за собой изменение форм и методов управления и соответствующей постановки задач. В этих условиях от опыта разработчика зависит многое. Он должен решить, надо ли создавать оригинальные алгоритмы и программы, можно ли модифицировать уже используемые или найти готовые ППП. Практика создания и функционирования многих АСУ в нашей стране позволила выработать и регламентировать не только набор функциональных подсистем, но и выделить задачи наиболее общего характера применительно к АСУ определенных классов и назначения. Ниже описаны некоторые из таких подсистем.

§ 3.2. Технико-экономическое планирование

Технико-экономическое планирование — основа планового управления производством. Отсюда возникают задающие воздействия и исходные данные другим функциональным подсистемам. Выделение задач технико-экономического планирования в функциональную подсистему АСУ позволяет ускорить разработку планов; повысить их экономическую и научно-техническую обоснованность; обеспечить возможность быстрой коррекции планов при возникающих отклонениях в ходе их выполнения, должной последовательности и непрерывности планирования; установить методическое единство технико-экономического и календарного планирования.

К числу первоочередных задач технико-экономического планирования относят расчет производственной программы; нормативной себестоимости товарного выпуска продукции; нормативных калькуляций на деталь, узел, готовое изделие; нормативной трудоемкости производственной программы; плановой численности производственных рабочих; планового фонда заработной платы производственных рабочих.

Расчет производственной программы. Как правило, расчет производственной программы совмещают с ее оптимизацией; при этом обычно используют один из двух критериев: достижение экстремальных значений одного или нескольких технико-экономических показателей, таких, как максимизация объема производства, прибыли, фондоотдачи, при заданных

ресурсах и ограничениях или минимизация использования ресурсов при ограничениях на технико-экономические показатели.

Например, в качестве критериев возможно использовать выражения

$$\sum_{j=1}^n X_j \rightarrow \max \quad \text{или} \quad \sum_{j=1}^n c_{lj} X_j \rightarrow \min ,$$

а ограничения могут иметь следующий вид:
по оборудованию

$$\sum_{j=1}^n a_{ij} X_j \leq A_i; \quad i = \overline{1, m_1};$$

по профессиям

$$\sum_{j=1}^n b_{kj} X_j \leq B_k; \quad k = \overline{1, m_2};$$

по ресурсам

$$\sum_{j=1}^n c_{lj} X_j \leq C_l; \quad l = \overline{1, m_3};$$

количество изделий j -го вида определено заданиями и ограничено только снизу или сверху:

$$X_j \leq X_j \leq \overline{X_j}; \quad j = \overline{1, n}.$$

Здесь a_{ij} – норма затрат рабочего времени оборудования i -го типа на выпуск единицы изделия j -го вида; X_j – планируемое количество изделий j -го вида; A_i – фонды времени работы оборудования i -го типа; m_1 – количество единиц оборудования i -го типа; b_{kj} – норма трудоемкости для k -й группы профессий при выпуске единицы изделия j -го вида; B_k – фонды рабочего времени по k -й группе профессий; m_2 – количество групп профессий; c_{lj} – норма затрат ресурсов l -й группы на выпуск единицы изделия j -го вида; C_l – фонды ресурсов l -й группы; m_3 – количество групп ресурсов; $\underline{X_j}, \overline{X_j}$ – соответственно нижняя и верхняя границы производственной программы по j -му виду изделий. Для решения задачи обычно используют аппарат линейного программирования.

В большинстве случаев производственная программа до окончательного утверждения многократно согласовывается и корректируется, в том числе по неформальным соображениям. В связи с этим решения, полученные на ЭВМ, могут быть использованы как рекомендательные или исходные для дальнейшей обработки.

Расчет нормативной себестоимости. *Себестоимость продукции* является одним из важнейших экономических показателей, характеризующим эффективность производственно-хозяйственной деятельности предприятия. В основе определения себестоимости лежит величина затрат на производство продукции; на нее непосредственно влияют такие факторы, как производительность труда, расход материальных и трудовых ресурсов, объем производства и его изменения. Себестоимость товарной продукции, рассчитанную на основании нормативных калькуляций, называют *норматив-*

ной. Как правило, расчет нормативной себестоимости товарного выпуска выполняют ежеквартально. Расчет ведется на одно изделие и на весь товарный выпуск. Исходными данными являются сведения о плановом количестве данного типа изделий, стоимости основных материалов за вычетом отходов, стоимости возвратных отходов, покупных и комплектующих изделий, основной и дополнительной заработной плате, отчислениях на соцстрах, расходах на содержание и эксплуатацию оборудования, освоение и подготовку производства, прочих производственных и непроизводственных расходах. Большая часть исходных данных получается в результате расчета *нормативных калькуляций* (см. ниже).

Нормативная себестоимость продукции по предприятию в целом определяется на основании сведений о плане выпуска изделий по заводу. Сначала определяется нормативная себестоимость по прямым затратам как сумма стоимости материалов и комплектующих изделий и основной заработной платы, приходящихся в соответствии с нормативной калькуляцией на одно готовое изделие. Затем, суммируя дополнительные нормативные затраты, определяют полную нормативную себестоимость. Умножая нормативную себестоимость каждого изделия на плановое количество выпуска этого изделия и затем суммируя по всем изделиям, получают *нормативную себестоимость товарного выпуска*.

Расчет нормативной калькуляции. Сумму производственных затрат на изготовление деталей, узлов и готового изделия, рассчитанную на основе действующих норм расхода материальных и трудовых ресурсов, называют *нормативной калькуляцией*.

Сначала определяют нормативную калькуляцию по *прямым затратам* на деталь на основании следующих исходных данных: нормативы расхода материалов и комплектующих изделий на одну деталь; нормативы отходов; пооперационные трудовые нормативы, цены и расценки.

Нормативная себестоимость по прямым затратам на одну деталь i -го типа определяется формулой

$$A_i = \sum_j a_{ij} S_j + \sum_k a_{ik} S_k + \sum_l a_{il} S_l,$$

где a_{ij} — норма расхода материалов j -го вида на одну деталь i -го типа; S_j — стоимость единицы материала j -го вида; a_{ik} — норма расхода комплектующих и покупных изделий k -го вида на одну деталь i -го типа; S_k — стоимость комплектующих и покупных изделий k -го вида; a_{il} — штучно-калькуляционное время на l -ю операцию по детали i -го типа; S_l — штучно-калькуляционная расценка на l -ю операцию.

При этом

$$a_{il} = a'_{il} + a''_{il}/r_i,$$

где a'_{il} — "штучное" время — технологическое время на l -ю операцию; a''_{il} — подготовительно-заключительное время на l -ю операцию; r_i — размер партии деталей i -го типа. Значение S_l определяется, аналогично a_{il} , на основе расценок на рабочие и подготовительно-заключительные операции.

На основе материальных и трудовых затрат определяют *косвенные затраты*. Для этого используют специальный справочник распределения накладных расходов по калькуляционным статьям, в котором содержатся данные о дополнительной заработной плате, отчислениях на соцстрах, транспортно-заготовительных расходах, расходах по содержанию и эксплуатации оборудования, цеховых расходах и проценте косвенных затрат. Суммируя прямые и косвенные затраты, получают нормативную калькуляцию на деталь.

Используя данные о вхождении деталей в узлы и узлов в готовые изделия, по структуре изделия определяют нормативную калькуляцию на узел и на готовое изделие как произведение нормативной калькуляции детали на число деталей в узле (или узлов в готовом изделии) плюс стоимость сборки и дополнительных расходов на транспорт, хранение и т.п.

Расчет нормативной трудоемкости производственной программы. Содержанием задачи является *расчет затрат рабочего времени по нормам на выполнение производственной программы*. Одновременно можно вести расчет *нормативного фонда прямой заработной платы*. Расчет ведется по основным цехам, а суммирование результатов дает *нормативную трудоемкость производственной программы по предприятию*. Исходными данными являются сведения о производственной программе — количество готовых изделий по наименованиям, пооперационные нормы времени и расценки на детали, узлы, изделия. Расчет заключается в раздетализовке изделий согласно заданной структуре изделия, определении норм затрат рабочего времени и расценок по деталям и дальнейшей свертки этих данных до уровня готовых изделий с учетом трудовых и денежных затрат на сборку, транспортные и другие промежуточные операции. В результате расчета выдаются сведения о затратах нормированного времени (в нормо-часах) на единицу изделия и на всю производственную программу на год с разбивкой по кварталам. Аналогично выдаются данные о фонде прямой заработной платы на производственную программу по нормам на изделие и на всю программу на год с разбивкой по кварталам.

Другие показатели технико-экономического планирования. Расчет плановой численности производственных рабочих позволяет получить значение этого показателя как по предприятию в целом, так и по цехам с разбивкой по профессиям. Это позволяет планировать дополнительную потребность в рабочих различной специальности, подготовку и повышение квалификации рабочих, проводить экономический анализ показателей по труду и заработной плате. Кроме информации о нормативной трудоемкости производственной программы исходными данными являются: задание по снижению трудоемкости, плановые проценты выполнения норм выработки по цехам и участкам, плановые проценты потерь рабочего времени, плановые затраты на прочую продукцию, план выпуска прочей продукции, отклонения по трудоемкости производственной программы, фонд полезного рабочего времени одного рабочего. В результате расчета выдаются сведения о плановой численности производственных рабочих, а также плановые

данные по перечисленным выше показателям на год по предприятию, цехам и участкам.

Наличие сведений о численности рабочих по профессиям позволяет рассчитать плановый фонд прямой зарплаты производственных рабочих и на этой основе — плановый фонд заработной платы производственных рабочих. Для этого используется информация о плановых процентах премии; доплатах за работу в ночное время, неосвобожденным бригадирам, за обучение учеников и пр.; дополнительной заработной плате за очередные и учебные отпуска, за выполнение государственных обязанностей и др.; о повышении удельного веса технически обоснованных норм.

Кроме перечисленных выше расчетов могут быть определены плановая и фактическая стоимости основных фондов; плановые и фактические суммы амортизационных отчислений; плановые и фактические суммы платы за фонды; сводные сведения о планируемых премиях по фонду зарплаты, доплатах различных видов, дополнительной зарплате и другие на год по предприятию, цехам и участкам.

§ 3.3. Календарное планирование

Межцеховое календарное планирование. В массовом производстве межцеховое календарное планирование представляет собой в основном централизованное подетальное планирование. Это означает, что на основе плана производства готовой продукции, содержащего полный перечень выпускаемых предприятием готовых изделий, к которым относят и выпускаемые предприятием запасные части, узлы и детали, с указанием количества и сроков выпуска, составляется развернутый подетальный план для отдельных цехов и предприятия в целом. При этом система плановых расчетов является комплексной, охватывает все цехи основного производства, увязывая по срокам планы запуска-выпуска заготовительных, механических и сборочных цехов. Критерием оценки при составлении планов для сборочных цехов является выполнение заказов в директивные сроки, а для остальных — равномерность загрузки оборудования с учетом необходимости его профилактических осмотров и плановых ремонтов.

Межцеховое календарное планирование должно определять *потребность* в выпуске готовых изделий в соответствии с годовым планом с учетом имеющихся в наличии запасов таких изделий, а также находящихся в производстве к моменту планирования. Различают *общую*, или *полную*, потребность и *чистую*, определяемую как разность между общей потребностью и запасами. Кроме того, должны быть определены: общие и чистые потребности в компонентах — узлах и деталях; размеры запускаемых в производство партий; сдвиг во времени выпускаемых или запускаемых компонентов с учетом необходимого опережения; реакция на рост или уменьшение объема заказов или прогнозов изменения потребности в определенных видах готовых изделий.

Определение чистой потребности в готовой продукции и потребностей в компонентах относится к технико-экономическому планированию и входит составной частью в техпромфинплан предприятия; определение размеров и сроков запуска партий деталей — к задачам собственно календарного планирования.

Рассмотрим комплекс задач межцехового календарного планирования, как это реализовано в соответствующем пакете прикладных программ [2.2].

Первая задача комплекса планирование потребностей состоит из двух частей. В первой части на основании плана производства или выходных данных подсистемы прогнозирования спроса на готовую продукцию, если она имеется, определяется чистая потребность в готовых изделиях (включая запасные части и детали). Входными данными являются общие потребности, которые задаются в виде перечня готовых изделий, их шифров, требуемого количества по каждому из них и сроков исполнения. Эти потребности могут быть при необходимости увеличены на величину ожидаемых потерь. Сравнивая их с имеющимися запасами, определяют чистую потребность. Дополнительно рассматриваются оформленные заказы по отношению к общим потребностям по периодам времени. Выявленные несоответствия в дальнейшем устраняют путем соответствующих изменений в сроках исполнения заказов.

Выходом является чистая потребность в готовых изделиях, определяемая как требуемые количества и календарные сроки завершения изготовления изделий по каждому наименованию. Эти сведения имеют самостоятельное значение, а также являются входными данными для второй части задачи планирования потребностей, которая служит для определения чистой потребности в компонентах.

Для каждого изделия, выпускаемого предприятием, в память ЭВМ вводят нормативно-справочную информацию о структуре изделия в виде поуровневой входимости компонентов в изделия, с указанием наименований и шифров узлов и деталей, а также количества, необходимого для выпуска одного изделия.

Сначала с помощью такого массива осуществляют первый шаг разузлования, определяя для каждого изделия наиболее крупные компоненты, из которых оно состоит. Зная чистую потребность в изделиях, определяют общую потребность в компонентах первого уровня. Аналогично предыдущему, путем сравнения с имеющимися запасами определяют чистые потребности в этих компонентах. Затем выполняют следующие шаги разузлования, определяя для каждого последующего уровня дерева структуры изделия соответствующие компоненты и таким же образом определяя чистые потребности в них. Процесс продолжается до полного разузлования дерева структуры изделия, т.е. до получения сведений по каждой отдельной детали. Выходом являются чистые потребности по каждому наименованию и шифру детали с указанием количества и требуемых сроков завершения их изготовления.



Рис. 3.1

В межцеховом календарном планировании решаются еще четыре независимые друг от друга задачи — планирование заказов, опережение, изменение потребностей, диалоговое планирование.

Задача планирования заказов предназначена для определения размеров партий деталей, т.е. разбиения чистых потребностей на отдельные партии такого размера, который соответствует принятой политике заказов для данного изделия. Возможны две политики — определение партий фиксированного, заранее заданного и хранящегося в памяти ЭВМ объема и переменного, вычисляемого объема.

В случае, когда плановые заказы на изготовление деталей имеют фиксированный объем, планирование заказов осуществляется следующим образом (рис. 3.1). Каждый раз, когда на определенный плановый период выявляется чистая потребность в деталях данного типа, оформляется плановый заказ. Если чистая потребность не удовлетворена, эта процедура повторяется. Когда чистая потребность удовлетворена, проверяется, остался ли резерв деталей. Величину этого резерва надо запомнить до следующего цикла определения чистой потребности.

При вычисляемом объеме партии заказываемых деталей важным является стремление оптимизировать размер партии. В общем случае это определяется сопоставлением издержек на хранение и на запуск партии деталей, т.е. издержек, связанных с наладкой оборудования и подготовкой к выпуску партии. При малых значениях чистых потребностей они накапливаются для выпуска партии оптимального размера. Возможные дополнительные издержки, связанные с задержкой удовлетворения чистой потребности, учитываются при определении оптимального размера партии. Если чистые потребности велики, оформляется несколько заказов на партии деталей, размер каждой из которых является оптимальным.

Во многих случаях, кроме политики заказов, размер партии определяется с учетом различных ограничений. Такие ограничения могут иметь временный характер; округление объема заказа и даты его завершения могут быть наложены на минимальное и (или) максимальное количество деталей в одной партии. Как политика заказов, так и ограничения могут быть определены для каждого вида деталей.

Определение размеров партий проводится по уровням структуры изделия.

Задача опережения предназначена для смещения во времени чистых потребностей и соответственно плановых заказов, чтобы правильно установить дату начала выполнения заказов. Это позволяет учесть время для изготовления деталей и узлов или для их закупки и получения. Длительность цикла изготовления или закупки узлов и деталей заранее вводят в память ЭВМ. Смещение заказов осуществляется по уровням структуры изделия, причем на каждом уровне — до перехода к аналогичным расчетам на следующем, более низком уровне. Одновременно со сдвигом начала заказа во времени проверяется, приходится ли полученная дата на расчетный или на более ранний период, когда завершение заказа в требуемый срок уже невозможно. В последнем случае выдается аварийное сообщение о несоответствии.

Задачу изменения потребностей решают в тех случаях, когда по внешним причинам надо изменить некоторые заказы или позиции плана без пересчета календарного плана в целом.

Входными данными являются: шифр изделия; новое значение требуемого количества; новая дата требуемого срока исполнения; номер заказа, если он уже существует; шифр изменения, указывающий его характер. Изменения могут быть как в сторону увеличения, так и уменьшения потребности. Вводить их следует с наивысшего для данного изменения уровня; для готовых изделий — с верхнего уровня структуры изделия.

Если изменение относится к подготовленным, но еще не запущенным в производство заказам, то соответствующие заказы корректируют. Необходимость изменения заказа, исполнение которого уже начато, выдается в виде аварийного сообщения о несоответствии.

Задача диалогового планирования позволяет прерывать и корректировать работу подсистемы на любом уровне планирования. Она облегчает просмотр и модификацию планируемых заказов, позволяет ответственному лицу регулировать и вводить в систему измененные значения количества изделий и даты их выпуска. Изменения, вносимые на определенном уровне, должны вводиться до перехода к следующему уровню.

Диалоговое планирование и изменение потребностей можно использовать совместно. Однако диалоговое планирование имеет более высокий приоритет. Система не может знать, на основе каких соображений был откорректирован тот или иной заказ. Поэтому если изменение потребностей требует пересмотра заказов, уже откорректированных в режиме диалогового планирования, то планируемые заказы не изменяются, о чем выдается соответствующее сообщение.

Важной задачей межцехового планирования является расчет мощностей. Он является основой для построения календарных планов и заключается в долгосрочном определении загрузки оборудования ресурсами. Предстоящие работы сопоставляются с располагаемыми ресурсами, в том числе рабочей силой, что позволяет так установить начальные сроки выполнения работ, чтобы обеспечить равномерную загрузку оборудования и выявить периоды времени, когда существует его перегрузка или недогрузка.

Задача планирования ресурсов получает входные данные о шифрах изделий, требуемых количествах и сроках их получения. Постоянные данные содержат информацию о располагаемой мощности каждого участка и рабочих мест, которую условно задают в единицах времени, обычно в часах. Различают *максимальный* и *нормальный* уровни ресурсов. *Максимальный уровень* определяется исходя из полной загрузки — трехсменной работы, семидневной недели и т.п. *Нормальный уровень* соответствует обычно принятому режиму работы.

Располагаемое время уменьшается на величину уже запланированной загрузки и резерва, необходимого для выполнения возможных срочных внеплановых работ. Может быть учтена эффективность работы (выполнение норм).

В память ЭВМ для каждого изделия вводят сведения по рабочим местам о технологическом времени работы оборудования, временах наладки, транспортировки и ожидания. По этим данным и данным о сроке завершения изготовления изделия определяется дата начала выполнения заказа без учета загрузки оборудования, а также загрузка этим заказом каждого участка и цеха. Получаемые загрузки вычитаются из располагаемого времени. Если возникают недогрузки, то начало выполнения заказа сдвигается на более раннюю дату, предпочтительно в наименее загруженные периоды времени. При перегрузке участка рассматривается возможность передачи работы на какой-либо иной, менее загруженный участок, если они взаимозаменяемы.

Чрезмерное удлинение цикла выполнения заказа нежелательно, так как при этом возрастает объем незавершенного производства. Если объем работ по всем заказам превышает производственные возможности, нормальное функционирование предприятия нарушается со всеми вытекающими из этого последствиями. Планирование мощностей позволяет предвидеть долговременный эффект от включения заказов в план и принимать соответствующие меры.

Внутрицеховое календарное планирование. Основной целью внутрицехового календарного планирования является обеспечение требуемых сроков выполнения заказов цехом в целом и его участками. Критерий эффективности — минимизация задержек фактических сроков выполнения заказов относительно запланированных с учетом приоритетов заказов, фиксируемых в виде весов задержек. Кроме того, учитывается необходимость улучшения других показателей, оценивающих работу цеха, таких,

как степень использования трудовых и материальных ресурсов, себестоимость продукции, ее качество и т.п.

Постоянная информация, необходимая для решения задач внутрицехового календарного планирования, содержит сведения об участках и рабочих местах цеха, а также о заказах.

Под *рабочим местом* понимается некоторое оборудование, на котором выполняется одна или несколько однотипных технологических операций. Термин "рабочее место" отождествляется чаще со станком, чем с рабочим, так как для выполнения операций на одном рабочем месте могут потребоваться несколько рабочих.

Под *участком*, или *рабочим участком*, понимается некоторая совокупность из одного или более рабочих мест. Обычно рабочие места участка выполняют одинаковые или близкие по технологии операции.

Сведения о рабочих местах и участках содержат идентификатор рабочего места (участка); информацию о номинальном числе рабочих мест, в том числе по сменам (для участков); эффективность использования, т.е. коэффициент выполнения норм для конкретного рабочего места; среднее время транспортировки изделий до любого другого участка в цехе (если это время не включено в продолжительность операций).

Постоянные сведения о каждом возможном виде заказа содержат: идентификатор вида заказа; постоянное значение приоритета; список необходимых для выполнения заказа технологических операций (идентификаторы) в порядке их выполнения (технологическая карта); время подготовки к выполнению каждой операции (транспортировки после предыдущей операции, наладки станка или другого оборудования и т.п.); нормативное время выполнения операции.

Исходной переменной информацией для решения задач внутрицехового календарного планирования являются следующие сведения: перечень заказов, подлежащих выполнению; дополнительный коэффициент приоритетности каждого заказа; требуемый срок завершения; наиболее ранний возможный срок начала выполнения; количество изделий в заказе (при нескольких видах изделий — по каждому из них); перечень выполняемых заказов; текущее состояние каждой операции по каждому выполняемому заказу; наличие необходимых ресурсов — материалов, комплектующих изделий, рабочих соответствующей квалификации и т.п. Как постоянная, так и большая часть переменной информации содержится в информационной базе системы управления.

Основной задачей внутрицехового календарного планирования является составление сменного задания: для каждого участка и рабочего места указывается, какие заказы и в какой последовательности должны выполняться в течение смены. Если объем заказа превышает возможности его выполнения в течение смены, задание выдается пооперационно.

Задача решается в такой последовательности.

1. Для каждого рабочего места выстраивается очередь, состоящая из операций всех заказов, выполняемых на данном рабочем месте. Очередь

выстраивается в порядке приоритетов, скорректированных по поступившей информации. Если операция может быть выполнена на различных рабочих местах, ее включают в наиболее короткую очередь.

2. На каждое рабочее место назначается первая операция. Ею является либо та, которая не закончена выполнением в предыдущей смене, либо первая из очереди.

3. Определяется время завершения назначенной операции. Если это время меньше конца планируемого периода, назначается следующая операция из очереди, в противном случае назначенная операция на данное рабочее место фиксируется.

4. Для каждой назначенной операции определяется ранний возможный срок ее выполнения как сумма времени ее завершения и времени транспортировки соответствующего изделия на следующее рабочее место.

5. Для назначенной операции проверяется, обеспечено ли ее выполнение ресурсами. Если ресурсы не обеспечены полностью, выдается аварийное сообщение для принятия мер. Следует отметить, что проверка обеспеченности ресурсами проводится ранее для заказа в целом, поэтому пооперационная проверка является контрольной, окончательно подтверждающей подготовленность к выполнению операции.

6. Пункты 3, 4, 5 повторяются до тех пор, пока на всех рабочих местах время завершения операций будет равно или больше времени окончания планируемого периода.

7. При назначении каждой операции следует проверить, выполняется ли условие завершения ее к назначенному сроку. Если этот срок не удовлетворяется, проводится поиск возможности выполнения ее в срок на каком-либо ином рабочем месте, с учетом приоритетов. Если такой возможности нет, выдается аварийное сообщение для принятия мер.

8. Если на каком-либо рабочем месте очереди нет, следует проверить, нельзя ли назначить операции из очередей других рабочих мест. Если таких операций нет, выбирается новый заказ из очереди заказов, в котором есть операция, подлежащая выполнению на данном рабочем месте.

В результате такого моделирования пооперационного выполнения заказов ежедневно выдаются сменные задания на 3—5 дней вперед. Для первой смены из этого списка задание подлежит выполнению, а для остальных является ориентировочным, позволяющим подготовиться к их выполнению. Задания корректируются по мере поступления информации о фактическом ходе их выполнения, изменениях приоритетов и т.п.

Внутрицеховое календарное планирование включает также задачи [2.2]: планирования, определяющего последовательность запуска; оценивания времени завершения; управления инструментами.

Задача планирования определяет последовательность запуска. Кроме определения сменных заданий, в которых указывается порядок поступления заказов на участки и последовательность их выполнения на рабочих местах, выдаются результаты анализа загрузки участков. Указывается число заказов, подлежащих выполнению на каждом участке по рабочим мес-

там, число часов наладки оборудования и рабочего времени. Эти сведения могут быть обобщены для разных уровней руководства. Выдаются сведения о заказах в очереди на начало периода, поступление которых ожидается, и подлежащих завершению в конце периода.

При определении последовательности запуска заказов в производство исполнение многих заказов выходит за пределы того интервала времени, для которого выдаются задания.

Задача оценивания времени завершения операции выдает для таких заказов прогностическую оценку времени их окончания без подробного расписания их выполнения.

Сообщения о состоянии заказов содержат сведения о каждом выполняемом заказе. Идентификатором заказа является номер и наименование изделия. Указываются объем заказа (количество изделий), дата начала — планируемая и фактическая, текущий и последующие рабочие участки, какие операции осталось выполнить и как будет использовано время от момента выдачи сообщения до даты завершения выполнения заказа. По оставшимся операциям, нормам времени на обработку, транспортировку и рассчитанным временам ожидания можно получить оценку даты завершения заказа, сравнением которой с требуемым сроком определяется число дней опоздания или опережения. Место заказа или операции в очереди определяется правилом предпочтения. Приоритет может определяться многими факторами: степенью важности заказа, временем до срока завершения, числом и временем выполнения операций, стоимостью заказа и др.

Примером правила предпочтения является "резерв времени на оставшиеся операции". Разница между требуемым сроком выполнения и текущей датой уменьшается на суммарное время, необходимое для выполнения оставшихся операций (включая подготовительные операции, транспортировку и ожидание). Результат деления полученного числа на количество оставшихся операций дает резерв времени. Операции занимают место в очереди в порядке возрастающих резервов времени.

Задача управления инструментами осуществляет планирование, учет и управление инструментальным хозяйством. В процессе ее решения обрабатываются запросы на инструмент, выдаются сообщения о наличии и использовании инструмента, определяется очередность его получения. Она также позволяет определить: типы инструмента, по которым имеется дефицит; заказы, выполнение которых не обеспечено инструментом; дефицитность различных типов инструмента и нехватку складских помещений.

Кроме описанных выше в комплексе задач календарного планирования могут включаться задачи, обеспечивающие руководство информацией, повышающей эффективность управления. Информация о загрузке участка, например, по каждому участку содержит сведения по списку рабочих мест о наличии рабочего персонала на смену, ресурсе времени в часах, ожидаемой загрузке; сведения об итоговом времени наладки и обработки заказов, оценках сроков их поступления на обработку в сопоставлении с ресур-

сами на обработку; об объеме свободных ресурсов или их нехватке; об анализе опаздывающих заказов, в том числе о количестве операций и общем времени ожидания, которые могут быть сгруппированы по возможным сверхурочным работам.

§ 3.4. Управление технической подготовкой производства

Многие задачи по управлению основным производством связаны с наличием сведений о технической подготовке производства. На каждое вновь осваиваемое производством готовое изделие эти сведения рассчитывают и сохраняют в памяти ЭВМ. На каждое новое изделие должны вводиться в память следующие данные: полный перечень всех деталей и узлов с условиями их входимости, т.е. дерево структуры изделия; специфицированные по типоразмерам нормы расхода материалов на каждую деталь; перечень рабочих операций на изготовление каждой детали, сборку узлов и изделия; перечень подготовительно-заключительных, транспортных и других операций на изготовление каждой детали, сборку узлов и изделия; нормативный размер партии деталей (если размер партии не вычисляется каждый раз перед ее запуском); штучное время (время на обработку одной детали) по каждой рабочей операции; подготовительно-заключительное, транспортное и другое дополнительное время (по партии деталей) для каждой рабочей операции; расценки штучного и подготовительно-заключительного времени по каждой операции.

По мере совершенствования технологии, изменения используемых материалов все хранящиеся в памяти сведения должны систематически обновляться. Различия между хранимыми сведениями и фактически используемыми материалами и технологией совершенно недопустимы.

Комплекс программ для управления технической подготовкой производства ориентирован на решение задач, выполняющих следующие расчеты: специфицированных норм расхода материалов на изделие; нормативной трудоемкости и нормативной заработной платы на деталь и на изделие; циклов и трудоемкости обработки деталей по цехам; расценок обработки деталей с выдачей нормативно-расценочных карт [2.2].

Расчет специфицированных норм расхода материалов на изделие. Решение задачи позволяет определить нормы расхода специфицированных по типоразмерам материалов на выпускаемые предприятием готовые изделия, а также долю каждого типоразмера материала в общем весе изделия, вес заготовок с последующим анализом эффективности использования материалов.

Наличие специфицированных норм расхода материалов на готовое изделие позволяет определить специфицированную потребность в материалах и составлять заявки на них в укрупненных группах; подготавливать исходные данные для расчета отпускных цен; рассчитывать сводные нормы расхода материалов.

Расчет заключается в извлечении из памяти данных о специфицированных нормах расхода материалов на все детали и узлы, входящие в данное изделие в соответствии с деревом структуры изделия, и суммировании их по типоразмерам материалов:

$$N_i = \sum_d n_{id} k_d,$$

где N_i — норма расхода i -го типоразмера материала на изделие; n_{id} — норма расхода i -го типоразмера материала на деталь d ; k_d — количество деталей типа d в изделии. В дальнейшем их можно суммировать по всем планируемым изделиям, получая сводную специфицированную по типоразмерам материалов потребность для предприятия в целом.

Для составления заявки группируют специфицированные нормы расхода материалов по каждому изделию по заявляемым укрупненным группам и затем суммируют по всему плановому выпуску.

Расчет нормативной трудоемкости и нормативной заработной платы на деталь. Подetailная *нормативная трудоемкость и заработная плата* определяются при калькулировании стоимости отдельных деталей для внутризаводского хозрасчета или при товарном отпуске деталей в виде готовой продукции, в качестве запасных частей. Кроме того, эти данные нужны для последующего определения общей нормативной трудоемкости и расхода заработной платы на изделие в целом.

Расчет ведется по формулам:

$$T_d = \sum_{i=1}^n \left(T_{i1} + \frac{T_{i2}}{P_d} \right); \quad P_d = \sum_{i=1}^n \left(P_{i1} + \frac{P_{i2}}{K_d} \right),$$

где T_d — подetailная нормативная трудоемкость; T_{i1} — штучное время на i -ю операция, суммируемое по всем n рабочим и подготовительно-заключительным операциям; T_{i2} — подготовительно-заключительное время на i -ю операцию (на партию деталей); P_d — нормативный размер партии деталей; P_{i1} — расценка штучного времени на i -ю операцию; P_{i2} — расценка подготовительно-заключительного времени на i -ю операцию; K_d — количество операций при изготовлении данной детали.

Путем дальнейшего суммирования можно определить нормативную трудоемкость и нормативную заработную плату для выпуска деталей по каждому цеху и по предприятию в целом.

Расчет нормативной трудоемкости и нормативной заработной платы на изделие. При определении пропускной способности производственных подразделений, цехов и участков, производственных мощностей, составлении планов по труду и заработной плате, планировании необходимой численности и состава работающих и в других целях необходимо знать нормативную трудоемкость и нормативную заработную плату на одно изделие.

Нормативная трудоемкость на изделие определяется следующим выражением:

$$T_i = \sum_j T_{ji} d_j + \sum_k T_k d_k,$$

где T_i — нормативная трудоемкость на изделие; T_{ji} — нормативная трудоемкость на изготовление детали j -го типа; d_j — количество деталей j -го ти-

па, необходимых для изготовления одного i -го изделия (с учетом нормируемого брака); T_k — суммарное время на сборочные операции узла k -го типа (включая транспортные и другие дополнительные операции); d_k — количество узлов k -го типа в одном изделии.

Нормативная заработная плата на изделие определяется по аналогичному выражению, где вместо трудоемкости используются нормативная заработная плата на изготовление детали и расценки на сборочные операции по сборке узлов. Суммирование может осуществляться по тем деталям и узлам, которые изготавливаются и собираются в данном цехе, или по предприятию в целом. В результате расчета формируются, запоминаются и выдаются для использования сведения о нормативной трудоемкости и нормативной заработной плате на изделие по цеху или по предприятию.

Расчет циклов и трудоемкости обработки деталей по цехам. Циклы обработки партии деталей необходимы для календарного планирования производства, с тем чтобы обеспечить планомерный и ритмичный процесс сборки и выпуска готовых изделий.

Цикл обработки деталей рассчитывается по формуле

$$T_y = \frac{T_{дП}}{K_{вн}} + T_{ср} d_{оп},$$

где $K_{вн}$ — средний коэффициент выполнения норм; $T_{ср}$ — среднее время задержки и транспортировки деталей между операциями; $d_{оп}$ — количество операций.

Трудоемкость обработки деталей по цехам заключается в суммировании подетальной нормативной трудоемкости по всем деталям, изготавливаемым в данном цехе.

Расчет расценок обработки деталей и выдача нормативно-расценочных карт. В некоторых случаях затруднительно вводить одновременно с началом производства нового изделия расценки штучного и подготовительно-заключительного времени по всем операциям этого изделия.

Если известно нормативное время обработки, штучное и подготовительно-заключительное по каждой рабочей операции, то расценки на обработку деталей по операциям технологического процесса рассчитывают на основании установленных тарифных коэффициентов в соответствии с тарифной сеткой и разрядом работы.

Эта процедура может также использоваться при необходимости изменения расценок. Расчет заключается в вычислении соответствующей расценки путем умножения нормативного времени, предусмотренного для данной операции, на коэффициент, зависящий от тарифной сетки и разряда работ. Таблица значений коэффициентов находится в памяти ЭВМ и для расчета надо ввести идентификаторы деталей, тарифные сетки и разряды работ для каждой детали. В результате решения задачи выдается *нормативно-расчетная карта*, в которой содержатся сведения о нормах времени и расценках на все или некоторые (по запросу) операции обработки детали.

Управление основным производством должно обеспечить выпуск за планово-учетный период в сборочных цехах такого набора изделий, который определен соответствующим комплектом заказов. За этот же период в цехах, выпускающих детали и узлы, должны быть изготовлены комплекты деталей для выпускаемых в сборочных цехах изделий, а в заготовительных цехах — восполнен расход заготовок.

Набор работ, а следовательно, и ритм для каждого планово-учетного периода в серийном производстве различны, так как они определяются переменным составом заказов. В отличие от серийного массовое производство характеризуется единым ритмом работы всех производственных звеньев, который обусловлен ритмом выпуска готовой продукции. Здесь основной задачей является согласование ритмов всех поточных линий и конвейеров, для чего требуется поддержание на расчетном уровне заделов по всем деталям и узлам; набор работ для каждого планово-учетного периода практически одинаков.

Изготовление в серийном производстве деталей партиями, оптимальный размер которых чаще всего не совпадает с потребностями сборочных цехов, предопределяет необходимость расчета плана производства деталей для каждого планово-учетного периода. Решение такой задачи требует предварительного расчета общей потребности в деталях данного типа, т.е. нужных для сборки заданного числа и типов выпускаемых изделий. При этом определяют "чистую" потребность, равную общей потребности за вычетом наличных производственных запасов, а также деталей, находящихся в производстве.

Различные партии деталей отличаются как по трудоемкости их изготовления, так и по количеству необходимых для этого ресурсов производственных мощностей. Поэтому в серийном производстве кроме управления производством деталей и заготовок необходимо также управление рабочими местами.

Календарный план нельзя использовать для непосредственного управления производственным процессом, так как он не учитывает и не может учесть возможные случайные отклонения в располагаемых ресурсах, загрузку производственных мощностей и ряд других факторов.

Для непосредственного управления производственным процессом используется сменно-суточное планирование, т.е. составление оперативного сменного задания. Оно формируется на базе календарного плана и учитывает оперативные сведения об обеспечении намечаемых работ заготовками, инструментами, материалами, трудовыми ресурсами и т.п. При этом важнейшими являются два вопроса: что включать в оперативный план, если некоторые работы из числа предписываемых календарным планом невозможно выполнить из-за необеспеченности ресурсами, и как конкретно распределять работы между рабочими местами? Следует также помнить о работах, по различным причинам не включенных в предыдущие сменные

задания, и включать их с соответствующими приоритетами, как только будут ликвидированы имевшиеся ограничения на ресурсы.

Управление производством деталей. Комплекс задач управления производством деталей включает решение задач межцехового календарного планирования.

В результате решения этих задач находят распределение по плановым периодам времени запуска и выпуска партий каждого типа деталей, по каждому цеху и по предприятию в целом на весь горизонт планирования с разбивкой по периодам. Одновременно в ходе решения задач определяют сроки начала и окончания сборки узлов любого промежуточного уровня, вплоть до готового изделия.

Расчет плана производства и графика запуска-выпуска деталей по цеху и участку для каждого наименования детали определяет показатели объема производственного плана на календарный период в штуках и нормо-часах. Для этого используется план по заказам; по нему определяется план по запуску как сумма всех плановых партий, запускаемых в данном месяце, выраженная в штуках деталей. Дата выпуска определяется путем прибавления к дате запуска нормативного цикла обработки данного количества деталей. В план по выпуску включаются партии деталей, которые были начаты и не могли быть закончены в прошлом плановом периоде, и исключаются те партии деталей, которые не могут быть закончены в текущем плановом периоде. Плановая трудоемкость по запуску или выпуску определяется умножением нормативной трудоемкости изготовления одной детали на количество деталей по плану запуска или выпуска.

Результаты расчета выдаются на производственные участки в виде задания на месяц по каждой детали в штуках и нормо-часах с указанием количества деталей к запуску и к выпуску на каждую календарную дату.

Контроль состояния комплектации выпускаемых изделий. Расчет используется для регулирования хода производства. Исходными данными являются сведения об общих потребностях деталей по периодам планирования, о наличии готовых деталей, о находящихся в исполнении заказах и о величине страхового запаса, обеспечивающего нормальную работу при случайных сбоях на предшествующих по технологии участках. Потребность на текущий период определяется как сумма общих потребностей с начала планового месяца, увеличенная на значение страхового запаса. Недостаток определяется как разность между общей потребностью и наличием готовых и находящихся в производстве деталей. Руководителю участка сообщаются только те позиции, по которым есть недостаток. По этим деталям должны быть приняты меры для ликвидации дефицита.

Расчет фактического расхода деталей на товарный выпуск осуществляется для учета расхода деталей и контроля за правильностью хранимой в памяти ЭВМ информации. По вводимым в ЭВМ данным о фактическом выпуске готовых изделий и по дереву структуры изделия рассчитывается величина фактического расхода деталей на товарный выпуск. Аналогично определяется фактический расход заготовок и комплектующих изделий.

Вычитая полученные значения расхода из данных о наличии, получают новые значения наличия деталей. Эти сведения передаются производственной службе предприятия для анализа. Если разность между наличием и расходом на выпуск отрицательна, это свидетельствует об ошибке в учете деталей, которую следует немедленно выявить и устранить.

Расчет оперативно-производственных нормативов выполняется для определения величин месячных партий запуска по каждому типу детали. Заранее определяется диапазон, в который должна укладываться величина полного цикла обработки месячной партии. Этот диапазон принимается в пределах от 2 до 20 рабочих дней.

Длительность производственного цикла определяется выражением

$$\Pi = \frac{T_d \Pi_d}{k} + p(n - 1),$$

где T_d — нормативная трудоемкость изготовления одной детали; Π_d — размер партии детали; k — средний коэффициент выполнения норм; p — среднее время пролеживания и транспортировки между рабочими операциями; n — количество рабочих операций при изготовлении детали. Если величина рассчитанного цикла вышла за верхнюю границу принятого диапазона, то первоначальная величина партии, которая определяется путем деления годового плана на 12, уменьшается, например, в 2, 4, 8 и т.д. раз, пока величина цикла не войдет в принятый диапазон. Если же величина цикла меньше нижней границы диапазона, он увеличивается от 2 до 6 раз (не более), пока не войдет в диапазон.

Управление производством заготовок. Комплекс задач управления производством заготовок содержит задачи: расчет графика производства заготовок, расчет сменно-суточных заданий на производство заготовок, планирование заказов на производство заготовок, а также задачу учета выполненных операций резки и расхода металла.

Расчет графика производства заготовок. Исходными данными для решения этой задачи являются сведения о потребностях деталей по периодам планирования на товарный выпуск изделий, требуемых для изготовления деталей заготовок, пооперационных трудовых нормативах. Даты выпуска плановых партий заготовок определяются по датам запуска соответствующих деталей, с учетом нормируемого времени на пролеживание и транспортировку. План производства заготовок определяется как объединение всех плановых партий заготовок, которые должны быть выпущены в данном плановом периоде. Распределение партий по календарным датам дает график производства заготовок. Для удобства и более полной характеристики заготовки одновременно выдается наименование материала, из которого она выполняется. Эти сведения вводятся в память ЭВМ при запуске в производство нового типа изделия.

Расчет сменно-суточных заданий на производство заготовок. Его особенностью является то обстоятельство, что различные партии заготовок могут выполняться из одного материала. Такие партии в ряде случаев це-

лесообразно объединять, выполняя их в едином производственном процессе. Кроме того, как и при производстве деталей, необходимо учитывать действительный фонд времени, который может быть использован на каждом рабочем месте.

Сменно-суточное задание выдается на каждое рабочее место. В нем указаны номера и размеры партий заготовок, марка, типоразмер и количество требуемого для изготовления материала, сведения о его наличии. Одновременно выдается наряд, в котором указаны типы заготовок, которые нужно изготовить из материала определенной марки и типоразмера, их количество (номер и размер партии), норма расхода материала на одну заготовку, а также нормы времени и расценки на операцию. В случае возможной замены исходного материала указывают материальные и трудовые нормативы при изготовлении заготовок из иного материала, используемого при замене.

Планирование заказов на производство заготовок. В качестве исходных данных используют сведения об общих потребностях в заготовках данного типа на планируемый период, наличии готовых и находящихся в производстве заготовок, рабочем цикле их производства. Расчет заключается в определении чистых потребностей, сдвига во времени от даты использования заготовок на более раннее время в соответствии с полным циклом времени их производства, а также в определении размера партии по принятой политике заказа (нормируемый или вычисляемый объем партии).

Управление рабочими местами. Комплекс задач управления рабочими местами реализует учет выполнения рабочих операций, оперативный учет выработки рабочих и начисление сдельной заработной платы. Расчет плановых сроков выполнения операций, выдача сменных заданий, входящие в состав задач управления рабочими местами, реализуются программами из комплекса программ для управления цехом.

Учет выполнения рабочих операций. Пооперационный учет производства по каждому участку за сутки осуществляется путем обработки сведений о выполненных за сутки технологических рабочих операциях и операциях по передаче деталей из одного подразделения в другое. Эти сведения вводятся в систему из сменного рапорта мастера или начальника участка.

Выходными данными являються справки о сдаче готовых деталей, поступающие в производственно-диспетчерский отдел для подетального планирования и регулирования хода производства, а также выдаваемые по запросу справки о состоянии движения производства по партиям деталей. В последнем случае сообщается: номер партии деталей; общее количество операций; количество завершенных операций; наименование или номер выполняемой операции и оборудования, на котором она выполняется; количество деталей, обработанных на текущей операции; плановые даты запуска и выпуска партии деталей.

Оперативный учет выработки рабочих и начисление сдельной заработной платы. Наличие в наряде сведений о выполняемой рабочими работе

позволяет вместе с учетом выполнения обработки деталей начислять заработную плату. Нормы времени, расценки и другие данные, необходимые для этого, хранятся в памяти ЭВМ.

Сумма заработной платы определяется по формуле

$$S = \frac{R_1 k d + R_2}{E} \frac{P}{100},$$

где S — заработная плата за обработку деталей на данной операции; R_1 — расценка штучного времени; k — расчетный коэффициент; d — количество обработанных деталей; R_2 — расценка подготовительно-заключительного времени (если обработана не вся партия, то указывается соответствующая часть R_2); E — единица нормирования; P — процент оплаты.

Сводка выработки и начисленной (без удержаний) заработной платы выдается ежедневно, чтобы рабочие знали объем выполненной ими работы и ее оценку. В сводке указывается для каждого рабочего перечень выполненных им за прошлый день операций, выработка в нормо-часах и заработная плата за день и нарастающим итогом с начала месяца.

Кроме информирования рабочих сводка позволяет анализировать выработку рабочих-сдельщиков и начисление им заработной платы, контролировать расход установленного по плану фонда заработной платы, выполнение норм выработки и производительности труда.

§ 3.6. Управление материально-техническим снабжением

Основной задачей органов материально-технического снабжения является обеспечение производства необходимыми материалами, комплектующими и другими изделиями и пр. при условии соблюдения режима экономии в затратах материальных ресурсов.

Все задачи материально-технического снабжения можно функционально разделить на три группы: определение потребности в материалах и комплектующих изделиях на производственную программу в укрупненной и специфицированной номенклатуре, разработка плана снабжения — составление заявок на фонды и распределение выделенных фондов; оперативный учет и управление движением материальных ресурсов; бухгалтерский учет материальных ресурсов.

Подсистема материально-технического снабжения функционально тесно связана с подсистемами планирования и оперативного управления производством и с подсистемой бухгалтерского учета.

Определение потребности и разработки плана снабжения. Производство имеет дело с конкретным материалом, характеристики которого включают марку материала, его типоразмер. Полный перечень таких исходных материалов называют *специфицированной номенклатурой*.

В масштабах народного хозяйства такой перечень насчитывает многие миллионы позиций, что исключает возможность их использования при планировании производства и распределения продукции на уровне Гос-

плана СССР и Госснаба СССР. Поэтому предприятие составляет заявки на обоснование и получение фондов на материальные ресурсы в укрупненной, сводной номенклатуре. Ряд специфицированных номенклатур объединяется в группы. Так, например, группа "лист медный" содержит свыше 300 специфицированных позиций такого типа: "лист медный холоднокатаный, твердый, марки М2, ГОСТ 495-70, размером 1,2х600х2000 мм".

Если число типов выпускаемых изделий сравнительно невелико и по каждому из них известны дерево структуры изделия, исходные материалы на изготовление деталей и необходимые комплектующие изделия, то нетрудно составить *специфицированную потребность на изделие*, суммируя количество одинаковых материалов по всем деталям. Умножая на количество изделий по плану и суммируя по всем изделиям, получают *специфицированную потребность на плановое производство* готовых изделий. Сводя позиции специфицированной номенклатуры в установленные укрупненные группы, получают *потребность в материалах в сводной номенклатуре* для заявки в вышестоящие снабженческие органы.

При стабильном производстве описанную операцию можно выполнить однократно, введя в память ЭВМ сводные нормы расхода материалов на изделие и обращаясь к ним в периоды составления заявок. Если число типов выпускаемых изделий велико, группируют близкие по характеристикам изделия и выделяют в каждой группе типовой представитель, для которого определяют сводные нормы расхода материалов. Все остальные типы пересчитываются через пересчетные коэффициенты. Наконец, при чрезвычайно большой номенклатуре производства, например, запасных частей, определяют усредненные сводные нормы расхода каждой группы материалов на 1 тыс. или на 1 млн. руб. производства.

Если известны сводные нормы расхода материалов на изделие, заявки на фонды определяют следующим образом.

Потребность на программу выпуска в k -м виде материала на основное производство

$$P_{\text{коп}} = \sum_{j=1}^m \Pi_j b_{kj},$$

где Π_j – годовой план выпуска j -го типа изделия; b_{kj} – норма расхода k -го вида материала на j -е изделие.

Общая потребность в материалах k -го вида на непредвиденные работы

$$P_{k \text{ нр}} = \sum_{j=1}^m \Pi_{j \text{ нр}} b_{kj},$$

где $\Pi_{j \text{ нр}}$ – количество изделий j -го типа по годовому плану на группу изделий "непредвиденные расходы".

Общая потребность в материалах и комплектующих изделиях k -го вида на НИР, инструменты и оснастку

$$P'_k = P_{k \text{ нир}} + P_{k \text{ инс}},$$

где $P_{k\text{нир}}$ – расход k -го вида материала на НИР; $P_{k\text{инс}}$ – расход k -го вида материала на инструмент и оснастку.

Расчет потребности в материалах на изменение незавершенного производства (НЗП) основывается на данных об объеме производства, нормах длительности производственного цикла изготовления отдельных деталей и узлов и нормах расхода материалов на различных стадиях производства. Изменение соотношения между НЗП и товарной продукцией в процентах

$$L_{\text{нз.п}} = \frac{B_{\text{нз.п}}^{\text{к}} - B_{\text{нз.п}}^{\text{н}}}{C_{\text{тов}}} 100,$$

где $B_{\text{нз.п}}^{\text{к}}$, $B_{\text{нз.п}}^{\text{н}}$ – объем НЗП на конец и начало планируемого года; $C_{\text{тов}}$ – стоимость товарной продукции в планируемом году. Тогда потребность в материалах (комплектующих) k -го вида на изменение НЗП

$$P_{k\text{нз.п}} = P_k L_{\text{нз.п}},$$

где P_k – потребность в материалах k -го вида на годовую программу.

Суммарная потребность в материалах k -го вида для расхода на все нужды основного и вспомогательного производства

$$P_{k\text{сум}} = P_{k\text{оп}} + P_k' + P_{k\text{нз.п}} + P_{k\text{нр}}.$$

Переходящий запас материалов k -го вида на начало года

$$P_{k\text{пер.зап}} = P_{k\text{сум}} H_{\text{пер.зап}} / T_{\text{раб.год}},$$

где $H_{\text{пер.зап}}$ – норматив переходящего запаса по данной номенклатуре материала (на этапе предварительных расчетов принимается $H_{\text{пер.зап}} = 30$ дн); $T_{\text{раб.год}}$ – число рабочих дней в году.

Общая потребность в материалах k -го вида на год

$$P_{k\text{общ}} = P_{k\text{сум}} + P_{k\text{пер.зап}}.$$

Ожидаемый остаток на начало планируемого года

$$O_{k0} = \Phi_{k0} + O_{k1} - P_{k2-4} - P_{k\text{н}}; \quad \Phi_{k0} = \Phi_{k\text{год}} - \Phi_{k1},$$

где $\Phi_{k\text{год}}$ – утвержденный фонд текущего года; Φ_{k0} – нереализованная часть годового фонда; Φ_{k1} – фонд, реализованный в первом квартале; O_{k1} – остаток на конец первого квартала (в цехах и складах); P_{k2-4} – потребность в k -м виде материала на второй, третий и четвертый кварталы текущего года; $P_{k\text{н}}$ – ожидаемая реализация неликвидов и сверхнормативов.

Величина запрашиваемого фонда на материалы

$$\Phi_{k3} = P_{k\text{общ}} - O_{k0}.$$

По составленным заявкам, если они признаны правильными, осуществляется выделение фондов. По дефицитным позициям величина выделенных фондов оказывается меньше заявленной потребности, что требует корректировки производственных планов или поисков дополнительных источников ресурсов.

На самом деле ситуация гораздо сложнее, чем описано выше. Заявки начинают собирать в апреле – мае года, предшествующего планируемому. В это время планы производства еще не известны, причем они принципиально не могут быть известны. Чтобы спрогнозировать план производства изделий, надо знать потребность в них, которая выявляется путем сбора

заявок, по необходимости проводимого до составления планов производства. По этой и другим причинам потребность в материалах в сводной номенклатуре является лишь первым приближением к плану снабжения предприятия.

Выделенные фонды требуют специфицирования, так как поставщики по нарядам-заказам или договорам поставляют уже конкретные марки и типоразмеры исходных материалов для производства.

Для учета и планирования хода материально-технического снабжения составляется *план снабжения*, который состоит из двух сбалансированных частей — *плана поставок* и *плана распределения*.

Исходными данными для формирования плана поставок являются сведения, содержащиеся в фондовых извещениях, нарядах на поставку продукции и договорах о поставках. В память ЭВМ вводят данные о поставках, потребителях и материальных ценностях.

Данные о поставщиках содержат следующие сведения: шифр и наименование поставщика, его грузовой адрес (откуда поставляется продукция), наименование отделения Госбанка или Стройбанка, расчетный счет, номенклатурные группы, по которым он поставляет продукцию.

Данные о потребителях содержат аналогичные сведения, если они являются самостоятельными хозрасчетными предприятиями. Если же они являются подразделениями предприятия, то указывается наименование подразделения и его адрес (в случае территориального размещения не по адресу предприятия).

Данные о материальных ценностях могут быть укрупненными до группы и специфицированными. В общие характеристики на группы материальных ценностей включаются: номенклатурный номер, наименование, ГОСТ или договор (если поставка выполняется по договорным условиям), технические условия (ТУ), марка, номер по преysкуранту.

Специфицированные данные дополнительно содержат сведения о номенклатурном номере, форме изделия, размерах в трех измерениях, единицах измерения, сорте, цене по преysкуранту с указанием страницы и позиции преysкуранта, условиях хранения, наличии наценки, размере страхового запаса, дополнениях к наименованию.

По мере поступления оформленных договоров и нарядов в ЭВМ вводятся общие сведения о договоре (наряде): наименование и шифр поставщика, его грузовой адрес, номер и дата заключения договора, общая сумма поставок, расчетный счет в Госбанке или Стройбанке. По каждой позиции договора вводятся: наименование материала, единица измерения, общее количество поставки с разбивкой по кварталам. Ввод сведений о поставщике является контрольным, эти данные сопоставляются ЭВМ с ранее введенными данными, и в случае расхождений выдается сообщение об ошибке.

Совокупность всех записей, введенных по договорам и нарядам, образует *план поставок*. В течение года по различным причинам в него вводятся изменения, что требует процедур корректировки хранящихся в памяти данных.

Когда известны поставщики и количество поставляемой ими продукции с распределением поставок по кварталам, составляют план распределения, в котором указывают, кому предназначается поставляемая продукция.

Исходными данными для составления плана распределения являются: заявки потребностей, отчеты, отражающие приход, расход и остаток материальных ценностей за предыдущий год, сведения о выделенных для поставки материалах. Для каждой строки плана распределения, сформированного по потребителям, вводятся: номер заявки потребителя, наименование материала, заявленное количество на год с разбивкой по кварталам, источник поставки (номер договора или наряда), выделенное данному потребителю количество на год с разбивкой по кварталам.

Общее количество поставок от всех поставщиков по одному специфицированному виду материалов должно быть равно сумме количеств этого материала, выделенных потребителям. Выделение некоторого количества в резерв оформляется как распределение фиктивному потребителю.

Совокупность записей для всех потребителей по всей специфицированной номенклатуре образует *план распределения*. Как и план поставок, в течение года план распределения может изменяться и корректироваться.

Оперативный учет и управление движением материальных ресурсов. Движение материальных ресурсов, которые часто именуют материальными ценностями, начинается с отправки их поставщиком по железной дороге, авиационным, автомобильным или другим транспортом (рис. 3.2). Одновременно с отправкой груза оформляются и отправляются в банк — счет для оплаты и получателю — сопроводительные документы, в которых указываются: полное наименование каждой позиции груза, количество, номер направленного в банк для оплаты счета, цена единицы продукции и общая стоимость. Банк после обработки пересылает счет получателю.

Счета поставщиков, обработанные банком, поступают в бухгалтерию и оттуда передаются в отдел снабжения на акцептирование. Под акцептированием понимается согласие на оплату счета, в результате чего соответствующая сумма банком снимается со счета получателя и зачисляется на счет поставщика. Организация имеет право акцептировать или неакцептировать весь счет или его отдельные позиции в течение трех суток.

Получив счет, сотрудник отдела снабжения вызывает соответствующие данные из плана поставки и проверяет, предусмотрено ли получение указанных в счете материалов планом поставки. Если такая поставка предусмотрена, счет акцептируется, в противном случае выясняется причина расходжения. Содержание акцептированного счета вводится в ЭВМ. Дальнейшие процедуры зависят от того, какая из двух возможных ситуаций реализуется. Счет может поступить либо до фактического прихода груза с сопроводительными документами, либо после этого.

Если соответствующий груз еще не поступил, акцептованный счет запоминается, а сведения о грузе заносятся в специальную рубрику "грузы в пути". В ней кроме наименования и количества груза указывается, на какой склад его следует направить после поступления.

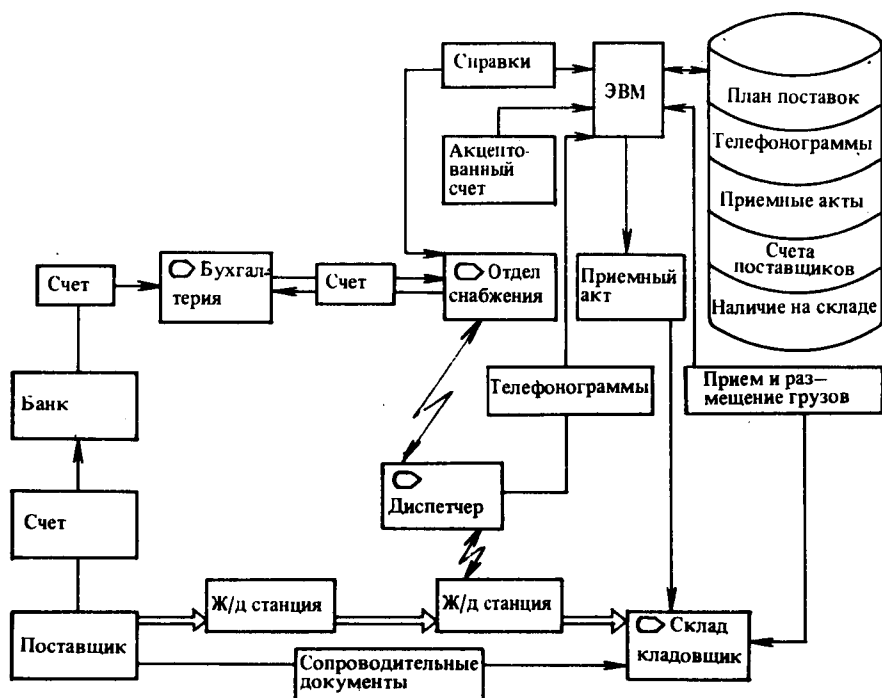


Рис. 3.2

О приходе груза представитель железнодорожной станции сообщает отделу снабжения или диспетчерской службе этого отдела, если она существует, телефонограммой. В ней содержатся следующие данные: дата, номер счета поставщика, адрес станции отправления, наименование поставщика, номер вагона или контейнера, номер накладной, вес груза. Диспетчер по номеру счета определяет, что счет уже акцептован и узнает, на какой склад следует вывести поступивший груз. Содержание телефонограммы также вводится в память ЭВМ.

На складах может использоваться система с постоянным и переменным закреплением мест хранения. В первом случае каждый сектор склада, стеллаж или ячейка закреплены за конкретным видом хранимых материальных ценностей независимо от их фактического наличия. Во втором случае ячейки после освобождения обезличиваются и в них можно поместить любые иные материальные ценности.

Поступивший на склад груз принимается по сопроводительным документам с контролем их соответствия счетам поставщиков. Далее проверяется соответствие фактического наличия груза сопроводительным документам. Несоответствие фиксируется по специальным правилам особым актом.

Кладовщик вводит в ЭВМ сведения о прибывшем грузе, указывая поставщика, его грузовой адрес, номер и дату выписки накладной, наименование (номенклатурный номер) и количество материальных ценностей. Данные о грузе обрабатываются ЭВМ, которая определяет адреса мест хранения по списку фиксированных мест хранения или выбирая из списка свободных ячеек, если поступил груз, которого нет на складе.

Кладовщик получает указание о размещении груза в заданную ячейку. При наличии автоматических штабелеров они по команде ЭВМ осуществляют загрузку в нужную ячейку. Если по каким-либо причинам ячейка оказалась занятой или кладовщик считает выбор ячейки неудачным, он размещает груз в любой ячейке, но во всех случаях сообщает ЭВМ о местонахождении принятого и размещенного груза.

Получив подтверждение (квитирование) о приеме груза, ЭВМ через установленный на складе терминал распечатывает приемный акт, содержащий следующие сведения: номер и дату составления, номер склада, фамилию кладовщика, наименование поставщика, номер и дату накладной, номер и дату счета. Далее по строкам акта указывается наименование, сорт, размер принимаемого груза, единица измерения, полученное количество, цена единицы и суммарная стоимость. Приемный акт является документом, устанавливающим материальную ответственность кладовщика. Поэтому кладовщик подписывает приемный акт и пересылает два подписанных экземпляра в бухгалтерию. Один экземпляр остается на складе для возможных последующих контрольных проверок.

Если груз с сопроводительными документами поступает раньше счета, процедуры в принципе сохраняются, за небольшими изменениями. Так как в памяти ЭВМ нет необходимого счета, ввод информации о пришедших грузах кладовщик осуществляет построчно с сопроводительных документов. Сведения о поступивших, но не оплаченных грузах хранятся в памяти до прихода соответствующего счета, после чего происходит их сопоставление для выявления ошибок и несоответствий. Ежедневно для контроля выдается ведомость принятых грузов и приходных документов за день. На этом процесс приема грузов заканчивается.

Сведения о поступивших за прошедший день грузах выдаются в виде сводных ведомостей принятых грузов и приходных документов. Эти сведения используются для контроля, а также для начала процесса распределения и выдачи имеющихся на складах материальных ценностей.

Распределение прибывших грузов может проводиться в автоматическом или "ручном" режимах. Автоматический режим используют в тех случаях, когда распределяемые количества полностью соответствуют плану распределения или должны быть распределены в равных пропорциях. "Ручное" распределение позволяет учитывать неформальные приоритеты потребителей.

Потребителям рассылаются извещения о том, какие грузы по плану распределения могут быть ими получены. При желании потребителя получить соответствующие грузы выдаются требования на получение груза.

Сотрудник отдела снабжения может аннулировать требование в случае длительной задержки получения грузов со склада или неплатежеспособности потребителя.

При предъявлении требования на склад кладовщик проверяет с помощью ЭВМ его правильность и получает указание адреса, по которому расположен этот груз на складе. При наличии автоматического штабелера ЭВМ выдает на него команду и груз подается штабелером к месту выдачи, в противном случае это делает кладовщик по указанному ЭВМ адресу. Выдав груз потребителю, кладовщик вводит в ЭВМ и проставляет на требовании фактически выданное количество, которое может отличаться от затребованного. Кладовщик и получатель груза подписывают требование, подтверждая факт передачи груза. Два экземпляра подписанных требований передаются в бухгалтерию. Для ускорения оформления платежных документов, если пересылка требований занимает длительное время, дубликат требования с заполненной графой "отпущено" и фамилиями подписавших сразу печатается на терминале, установленном в бухгалтерии.

Ежедневно на ЭВМ формируются по информации о выданных грузах счета-фактуры, один экземпляр которых вместе с копией требования отправляется потребителю. Используя данные счетов-фактур, рассортированные по потребителям, ЭВМ формирует платежные требования и сопроводительные реестры к ним, передаваемые в отделения Госбанка или Стройбанка для проведения необходимых банковских операций. Ускорение подготовки и передачи для оплаты платежных документов существенно влияет на время оборота оборотных средств, что является одним из основных финансовых показателей, характеризующих эффективность системы материально-технического снабжения.

Учитывая важность сохранности материальных ценностей, при разработке автоматизированной системы особое внимание должно уделяться защите данных от несанкционированного доступа, их резервированию, контролю достоверности и ведению архивов, по которым могут быть восстановлены все детали работы системы за прошлое время.

Бухгалтерский учет материальных ресурсов. Комплекс задач бухгалтерского учета материальных ресурсов (ценностей) предназначен для автоматизации учета движения материалов на складах предприятия, контроля за их сохранностью, расчета сумм по балансовым счетам, калькуляции фактической себестоимости производства в части материальных затрат, анализа производственной деятельности предприятия в части использования материальных ресурсов. Задачи можно функционально разделить на три группы: учета движения материалов на складах, расчетов с поставщиками материалов и расхода материалов на производство.

Учет движения материалов на складах кроме учета поступления и расходования материальных ценностей и контроля за их сохранностью включает в себя получение документов аналитического и синтетического бухгалтерского учета, содержащих как итоговые суммы по каждому балансовому счету, так и распределение этих сумм по группам материалов, видам опера-

ций и пр. Балансовый счет определяет, на какие цели израсходованы материалы — на основное производство, ремонтные работы, хозяйственные нужды и т.п.

На основании приходных актов и требований комплекс программ готовит и выдает оборотную ведомость движения материальных ценностей, т.е. какие материалы и в каких количествах поступили, а какие отпущены со склада. Оборотная ведомость выдается по складу, группам материалов, балансовым счетам.

Учет расчетов с поставщиками материалов заключается в следующем: по оплаченным счетам поставщиков и приходным ордерам подготавливаются и выдаются ведомости или справки на терминалы по задолженности поставщикам, по материалам в пути (счета получены, а материалы — нет), по неотфактурованным поставкам (материалы пришли, а счета — нет), по расчетам с поставщиками и др. Все программы выдачи различных ведомостей реализуют сортировку и суммирование по различным признакам уже введенных в ЭВМ данных.

Учет расхода материалов на производство осуществляется по вводимым в ЭВМ данным, содержащимся в требованиях, по которым выдаются материалы с цеховых складов, и в лимитно-заборных книжках, заменяющих в некоторых случаях требования.

Такой учет осуществляется для распределения затрат на материалы по различным видам производств, производственным подразделениям и конкретным заказам, для калькуляции фактической себестоимости. Введенные в ЭВМ данные сортируются и суммируются по балансовым счетам, статьям затрат и номерам заказов по цехам и по предприятию в целом. В результате выдаются ведомости распределения материальных ценностей по шифрам производственных затрат и группам материалов, по балансовым счетам, по расходу и балансовым счетам непроизводственных подразделений.

§ 3.7. Бухгалтерский учет

Бухгалтерский учет предназначен для обеспечения своевременного и полного учета поступающих в соответствующую организацию денежных средств, товарно-материальных ценностей и основных средств, а также отражения их движения; достоверного учета издержек производства и обращения, исполнения смет расходов, реализации продукции, выполнения строительно-монтажных и других работ, составления калькуляций себестоимости продукции, работ и услуг; точного учета и экономического анализа результатов хозяйственно-финансовой деятельности организации; правильного начисления и своевременного перечисления платежей и т.д. В табл. 3.1 приведена выдержка из плана счетов бухгалтерского учета производственно-хозяйственной деятельности предприятий, строек и хозяйственных организаций.

Таблица 3.1

№ счета	Наименование счета	№ счета	Наименование счета
	1. Основные средства (фонды)	63	Расчеты по претензиям
01	Основные средства (фонды)	64	Расчеты с покупателями и заказчиками
02	Износ основных средств (фондов)	70	Расчеты с рабочими и служащими
		73	Расчеты с бюджетом
03	Капитальный ремонт		7. Отвлеченные средства
	2. Производственные затраты	80	Отвлеченные средства за счет прибыли
05	Сырье и материалы	82	Использование кредитов банка за счет прибыли фондов
06	Топливо		8. Недостачи и потери
07	Строительные материалы и оборудование к установке	84	Недостачи и потери от порчи ценностей
08	Запасные части		9. Фонды и резервы
12	Малоценные и быстроизнашивающиеся предметы	85	Уставный фонд
	3. Затраты на производство	86	Амортизационный фонд
20	Основное производство	87	Фонды экономического стимулирования и специального назначения
21	Полуфабрикаты собственного производства		10. Кредиты банков и финансирование
24	Расходы по содержанию и эксплуатации оборудования	90	Краткосрочные ссуды банка
25	Общехозяйственные расходы	92	Долгосрочные кредиты банка
26	Общезаводские расходы	93	Финансирование капитальных вложений
27	Накладные расходы		11. Финансовые результаты
28	Брак в производстве		Доходы будущих периодов
	4. Готовая продукция, товары и реализация	98	Прибыли и убытки
40	Готовая продукция	99	Забалансовые счета
41	Товары	001	Арендованные основные средства (фонды)
44	Издержки обращения		Товарно-материальные ценности, принятые на ответственное хранение
46	Реализация	004	Оборудование, принятое для монтажа
	5. Денежные средства		
50	Касса	005	
51	Расчетный счет		
	6. Расчеты		
60	Расчеты с поставщиками и подрядчиками		

Основные средства социалистических предприятий включают в себя средства труда. По определению К. Маркса, "... средство труда есть вещь или комплекс вещей, которые человек помещает между собой и предметом труда и которые служат для него в качестве проводника его воздействий на этот предмет. Он пользуется механическими, физическими, химическими свойствами вещей для того, чтобы в соответствии со своей целью применить их как орудия воздействия на другие вещи. ... В более широком смысле к средствам процесса труда относятся все материальные условия, необходимые вообще для того, чтобы процесс мог совершаться". (Маркс. К. Капитал. Т. I. 1969. С. 190, 191.)

Особенность основных средств (фондов) заключается в том, что в процессе производства они сохраняют свою форму, а их стоимость передается на вновь создаваемый продукт по мере износа.

Основные средства (фонды) в отраслях народного хозяйства по *вещественно-натуральному характеру и назначению* подразделяют на следующие виды: здания; сооружения; передаточные устройства (электросети, трубопроводы); машины и оборудование; транспортные средства; инструмент; производственный инвентарь и принадлежности; хозяйственный инвентарь; рабочий и продуктивный скот; многолетние насаждения; капитальные затраты по улучшению земель (без сооружений); прочие. По *функциональному назначению* их подразделяют на производственные и непроизводственные.

Различают первоначальную, остаточную и восстановительную стоимость основных средств.

Первоначальная стоимость складывается из затрат на проектно-изыскательские, строительные и монтажные работы; на приобретение, доставку и монтаж оборудования в соответствии со сметной стоимостью.

Остаточная стоимость представляет собой остаток от вычета из первоначальной стоимости начисленного износа.

Восстановительная стоимость определяется как сумма затрат, необходимых в современных условиях для приобретения или строительства данного объекта.

Производственные затраты включают в себя расходы на сырье и основные материалы, являющиеся продуктами сельского хозяйства или добывающей промышленности; вспомогательные материалы, способствующие процессу производства (смазка, краски и т.п.); топливо; тара; запасные части; молодняк животных и животные на откорме; малоценные и быстроизнашивающиеся предметы, находящиеся на складе или в эксплуатации.

Для правильного указания в документах по учету движения материалов применяют систематизированный перечень установленных наименований материалов и их единиц измерения, используемых на предприятии, называемый *номенклатурой*. Каждому наименованию, марке, сорту и размеру материалов присваивается *номенклатурный номер* (шифр), который в обязательном порядке указывается во всех документах по учету движения материалов. Полное наименование материалов часто бывает многословным, что неудобно для повседневного употребления. Введение номенклатурных номеров позволяет избежать или уменьшить число ошибок, связанных с произвольным сокращением таких наименований, перестановкой слов, заменой их кратким жаргонным названием. В современных АСУ, работающих в диалоговом режиме, номенклатурные номера имеют вспомогательную роль.

Материалы учитывают в затратах производства по их фактической себестоимости. Но в текущем учете при систематической обработке данных о движении материалов их фактическая себестоимость может быть еще неизвестна. В этом случае используют оптовые цены или плановую себестоимость, которые называют *учетными ценами*.

Затраты на производство группируют по элементам затрат и статьям калькуляции. *Элементами затрат* называют экономически однородные затраты независимо от того, где на производстве и на какие виды изделий они расходуются. Элементами затрат на производство являются: сырье и основные материалы, в том числе покупные изделия и полуфабрикаты; вспомогательные материалы; топливо; энергия; заработная плата; отчисления на социальное страхование; амортизация основных фондов; прочие расходы.

Для определения себестоимости изделий на каждый их вид составляют калькуляции по статьям затрат. Типовая группировка содержит следующие калькуляционные статьи затрат: сырье и материалы; возвратные отходы (вычитаются); топливо и энергия на технологические цели; основная и дополнительная заработная плата производственных рабочих; отчисления на социальное страхование; расходы на подготовку и освоение производства, на содержание и эксплуатацию оборудования; общезаводские, цеховые и прочие производственные расходы, а также потери от брака и внепроизводственные расходы.

Совокупность затрат по всем перечисленным калькуляционным статьям, кроме внепроизводственных расходов, составляет *производственную себестоимость*, а вместе с ними — *полную себестоимость*.

Готовая продукция представляет собой полностью законченные изделия. Она отражается в балансе по фактической себестоимости, а в текущем учете — по учетным ценам.

Денежные средства, предназначенные для расчетов с поставщиками и потребителями, содержатся на расчетном счете в Государственном банке. Каждое предприятие или организация имеет только один расчетный счет.

Остальные разделы плана счетов бухгалтерского учета понятны по их наименованиям из табл. 3.1. Остановимся лишь на расчетах с рабочими и служащими.

Исходными данными для бухгалтерского учета труда и заработной платы являются сведения о выработке и начислении заработной платы рабочим; данные табельного учета и листки нетрудоспособности; персональные данные по всему составу работающих; распределение сумм между членами бригады при бригадной оплате труда; сведения о премиях и доплатах и различного рода удержаниях.

Расчеты по заработной плате выполняются два раза в месяц. За первую половину месяца оформляется только платежная ведомость на выплату аванса. Размер аванса устанавливается коллективным договором предприятия. За вторую половину месяца проводятся полные расчеты, выдается платежная ведомость и различные сводные документы по начислению и распределению заработной платы. Все виды разовых оплат, автоматизация которых нецелесообразна, рассчитывают вручную и вводят в ЭВМ в виде готовых сумм по ведомости на разовые оплаты.

В память ЭВМ вводят условно-постоянные данные, корректируемые по мере необходимости, как правило, перед очередным циклом расчетов.

Наименование показателя	Диапазон изменения
Номер участка	000–999
Табельный номер	00001–99999
Фамилия, имя, отчество	А (20)
Признак удержания взносов	0–5
Шифр профессии	00001–99999
Система оплаты	1–9
Категория	00–99
Тарифная сетка	00–15
Разряд	1–6
Оклад или часовая тарифная ставка повременщика	000,01–999,99
Тарифная ставка сдельщика	00,001–99,999
Плановый аванс	000,01–999,99
Признак выплаты аванса	1–9
Коэффициент оплаты ночных часов	1–9
Шифр налогоплательщика	1–5
Персональная надбавка	000,01–999,99

Сведения о работающих, приведенные в табл. 3.2, вводятся с карточек личного состава работающих, заполняемых в отделе кадров, и дополняются данными для лицевых счетов, ведущихся бухгалтерией.

Для ведения учета использования рабочего времени каждому работающему присваивается табельный номер и его явка на работу фиксируется в таблице учета рабочего времени. В таблице указывается число часов работы по числам месяца и всего за месяц. По каждому полному или неполному дню неявок указывается причина – отпуск очередной, по беременности и родам, без оплаты; болезнь; командировка; прогул; работа в подшефной организации и т.д.

Доплаты к заработной плате могут быть определены в виде некоторых сумм или в процентах от оплаты. К первым относятся персональные надбавки, оплата за ночные смены и праздничные дни и др. Ко второй группе относятся доплата неосвобожденным бригадирам за руководство бригадой, шоферам за классность, оплата ученикам и пр.

После выдачи аванса заработной платы и закрытия платежной ведомости в ЭВМ вводят сведения о лицах, по тем или иным причинам не получившим аванс, чтобы при окончательном расчете заработной платы за месяц аванс не был удержан (если аванс не переводится на депоненты).

Основным документом, по которому начисляется заработная плата рабочим, является наряд. Для работающих повременно таким документом является табель учета рабочего времени. Кроме того, для разовых расчетов вносятся различные листки на доплаты за непредусмотренные технологическим процессом работы с указанием причин и виновников доплат; листки о простое не по вине рабочего; листки нетрудоспособности. В последнем случае оплата проводится в зависимости от вида пособия: по об-

щим заболеваниям; по профессиональным заболеваниям; по беременности и родам; по травмам бытовым и производственным.

Из заработной платы удерживаются налоги в установленных размерах, с соответствующими льготами инвалидам и участникам Великой Отечественной войны и некоторым другим категориям граждан. Кроме того, из заработной платы удерживают суммы по исполнительным листам, за приобретенные в кредит товары, в случаях недостач и т.п.

Основными выходными документами являются платежные ведомости на выдачу заработной платы за первую и вторую половину месяца. Кроме того, выдаются различные сводные отчеты, содержание которых определяется конкретными условиями предприятия.

При проектировании подсистемы бухгалтерского учета и заработной платы следует учитывать основную ее особенность, заключающуюся в том, что на ЭВМ ведут расчеты финансовых и денежных средств. В связи с этим возникает ряд более жестких требований, чем в других подсистемах АСУ.

Прежде всего это относится к необходимости обеспечения повышенной точности и достоверности выходных данных. Для этого помимо повышенной надежности технического и программного обеспечения надо свести к минимуму участие человека в процессе обработки данных, полностью исключить составление документов от руки. На ЭВМ следует выдавать в виде распечаток только такие внутрисистемные документы, которые связаны с физической передачей денежных сумм или материальных ценностей от одного лица другому, что необходимо сопровождать юридическим подтверждением этого факта личными подписями. Тем не менее полностью избежать ручного ввода данных невозможно, так как ряд документов поступает извне системы, а для их автоматического считывания пока не существует массовых надежных технических устройств. Поэтому правильность ввода контролируют несколькими методами одновременно.

Повышенного внимания требует алгоритмизация расчетов. Так, например, нельзя округлять результаты, отбрасывая дробные доли; следует строго следить, чтобы при переводе средств с одной статьи бухгалтерского баланса на другую не изменялись ни каждая переводимая сумма, ни общий их итог.

Манипулирование финансовыми и денежными средствами с помощью ЭВМ требует повышенных мер защиты от несанкционированного доступа, аналогичных системам повышенной секретности. При этом нельзя существенно усложнять работу пользователя в санкционированном режиме и в то же время следует обеспечить защиту от весьма квалифицированных лиц, в том числе персонала вычислительного центра.

Работа системы должна происходить в реальном времени. Одним из показателей успешности работы финансовой службы и бухгалтерии является ускорение оборачиваемости оборотных средств. Недопустимо, чтобы обработка документов на ЭВМ прямо или косвенно задерживала этот процесс.

Особое внимание должно быть уделено ведению протоколов работы

системы и ее архивов. Для проверки возможных злоупотреблений или устранения недоразумений могут потребоваться достаточно детальные сведения за прошлые периоды.

В системе неизбежно возникает время от времени необходимость корректировки хранимых данных. Этот режим должен быть тщательно продуман и регламентирован, чтобы исключить возможность его использования в иных целях.

§ 3.8. Управление сбытом продукции

В современных условиях хозяйственной деятельности предприятий, когда одним из основных критериев является объем реализованной продукции и выполнение обязательств по договорам с потребителями как по объемам, так и по срокам и номенклатуре поставок, важное значение имеет эффективная деятельность сбытовых подразделений предприятий и объединений.

Функции отдела сбыта заключаются в следующем. Отдел получает планы выпуска и реализации товарной продукции. Согласно этим планам, а также заключенным договорам и разнарядкам составляются планы отгрузки готовых изделий собственного производства, покупных изделий, поставляемых в комплекте с изделиями своего производства, другой товарной продукции. Получая сведения о готовности продукции, наличии ее на складе, данные о сортности, отдел сбыта составляет график отгрузки, набирая в случае необходимости комплекты изделий для одного потребителя или набирая в один грузовой адрес продукцию по весу до нормы загрузки железнодорожного вагона. Одновременно оформляются необходимые документы и счета потребителям. В ходе этой работы ведется оперативный учет и прогнозирование выполнения планов реализации и прибыли, планов отгрузки и др.

Основой функционирования подсистемы сбыта является разработка оперативных календарных графиков отгрузки, обеспечивающих своевременное поступление сумм за реализованную продукцию. Исходными данными для решения комплекса задач управления сбытом являются следующие сведения: наименования, грузовые адреса, расчетные счета и другие реквизиты потребителей; план поставок на год с разбивкой по кварталам; данные, содержащиеся в договорах и нарядах, о том, когда, кому и что должно быть поставлено; текущие сведения о выполнении плана производства; справочники по ценам и др.

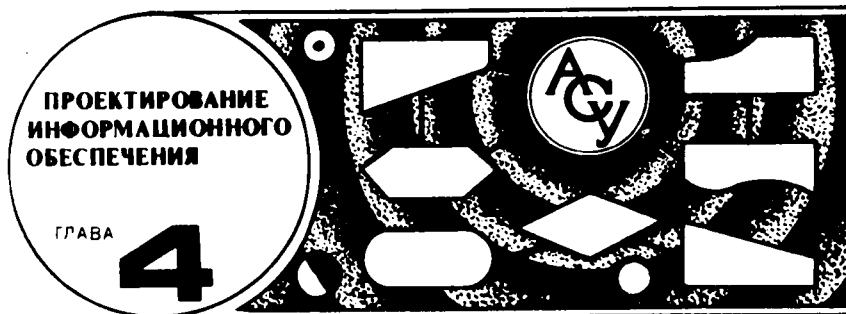
Важной функцией отдела сбыта является слежение за своевременным выполнением договорных обязательств. Для этого ведутся и непрерывно обновляются сведения о договорах и поставках. В первом случае данные группируются по договорам, во втором — по видам продукции. По каждому договору сообщается его номер и дата; шифр изделий с указанием планового и фактически отгруженного количества каждого изделия на год с

разбивкой по кварталам. По фактически отгруженным изделиям вводятся данные, подтверждающие их оплату получателем. По каждому виду продукции сообщается его шифр, наименование, остаток за прошлые кварталы; планируемое количество на текущий квартал; количество изделий, фактически отгруженных за текущий квартал, в том числе в счет прошлых и будущих кварталов; остаток наличия и обязательств на текущий квартал.

Подсистема управления сбытом продукции тесно связана с другими подсистемами АСУ в соответствии с функциональными и логическими связями отдела сбыта с другими подразделениями предприятия. В первую очередь это связь с подсистемой оперативного управления основным производством, позволяющая прогнозировать выполнение плана отгрузки по мере хода изготовления соответствующих изделий; связь с отделами технического контроля для определения сорта продукции и выполнения особых условий (экспортное или северное исполнение и т.п.); с подсистемой управления транспортом для реализации отгрузки; подсистемой бухгалтерского учета в части отчетности о движении готовой продукции и оформления расчетов с потребителями и т.п.

Управление сбытом при массовом производстве имеет свои особенности, которые заключаются в следующем. На склад готовой продукции поступает большое количество изделий немногочисленных типов. В этом случае конкретизация количества, содержания и адресов отгрузки готовой продукции должна осуществляться практически ежедневно. Для этого следует иметь упорядоченные сведения о наличии изделий на складе на текущий день по типам. Это позволяет выяснить возможность удовлетворения требований на отгрузку каждого типа изделий; распределению подлежит наличие за вычетом нормативного остатка; разовая поставка должна удовлетворять установленным транзитным нормам, т.е. регламентированному объему партии; ассортимент поставки должен соответствовать заказам; при замене одних изделий другими должна соблюдаться их взаимозаменяемость или равноценность для потребителей и др. Кроме того, могут быть установлены ограничения на общее количество отгружаемых изделий за день, а также ограничения на количество платежных требований. В ряде случаев необходимо распределять отгружаемую продукцию по таре, для чего в памяти ЭВМ должны храниться по шифрам изделий и видов тары таблицы входимости, определяющие возможность помещения определенного количества готовых изделий или их наборов в каждый из типов тары.

Сбыт готовых изделий, реализация готовой продукции потребителям являются завершением производственного процесса, и качественное решение задач управления сбытом во многом определяет оценку деятельности предприятия в целом.



Не будет преувеличением сказать, что в значительном числе случаев низкая эффективность действующих АСУ объясняется неудовлетворительной проработкой в проекте вопросов информационного обеспечения. Это относится к своевременной и достоверной исходной информации, хорошо организованной технологии ее получения и ввода в систему. Следует предусматривать адекватные методы контроля вводимых данных.

Существенное значение имеют методы предохранения системы от возможных потерь информации, защиты ее от несанкционированного доступа.

Наиболее важным является решение вопросов использования информации для управления — кому, в каком объеме, с какой точностью и степенью детализации должна выдаваться информация для принятия решения.

В ответственных случаях работоспособность системы, не говоря уже об ее эффективности, определяется тем, насколько внимательно проработаны в проекте вопросы ведения протоколов взаимодействия персонала с системой, хранения в архивах достаточно детальных сведений для последующего анализа деятельности как отдельных сотрудников, так и системы в целом.

§ 4.1. Основные понятия и определения

Информационное обеспечение АСУ — совокупность реализованных решений по объектам, размещению и формам организации информации, циркулирующей в автоматизированной системе управления при ее функционировании. Оно включает в себя нормативно-справочную информацию; необходимые классификаторы технико-экономической информации; массивы данных, необходимых для решения задач; унифицированные

документы, используемые в АСУ (ГОСТ 24.003—84). Основное назначение информационного обеспечения — своевременно выдавать системе управления, в частности, лицам, принимающим решения, достоверную информацию, необходимую и достаточную для принятия оптимальных или близких к ним управленческих решений.

Система автоматизированного управления имеет дело не с самим объектом, а с информацией о нем. Поэтому основной функцией информационного обеспечения является создание и ведение *динамической информационной модели* объекта, которая в каждый момент времени содержит данные, соответствующие фактическим значениям параметров с минимально допустимой задержкой во времени. Эти данные должны выдаваться любому пользователю, которому они необходимы для принятия решений. Соответствие данных фактическим значениям параметров с заданной точностью является важной характеристикой информационного обеспечения. Для его реализации применяют специальные методы контроля и обеспечения достоверности.

Нормативно-справочная информация заимствуется из нормативных документов и справочников. Ее содержание определяется теми постоянными или условно-постоянными данными, т.е. не изменяющимися, по крайней мере, в течение трех-четырех циклов работы, которые являются исходными для решения задач. Под *данными* в общем случае понимают любые сведения, являющиеся объектом обработки в АСУ. Вводимые в систему и хранимые в ней данные представляют собой отдельные значения параметров объектов или процессов в виде слов, буквенных обозначений, числовых величин, таблиц, графиков или в иной форме. Некоторую совокупность или набор данных, используемых для определенной цели, называют *информацией*.

Классификаторы технико-экономической информации служат для унификации применяемых в АСУ наименований и обозначений с целью их однозначного определения. Каждая позиция классификатора содержит кроме словесного наименования цифровой код, структура которого позволяет более четко определить иерархические уровни и группировки классифицируемой информации.

Массивы данных содержат в упорядоченном и систематизированном виде все необходимые для функционирования АСУ данные. Это позволяет быстро находить или вводить в систему нужные данные, осуществлять перекрестные ссылки и т.п. Совокупность упорядоченной информации, используемой при функционировании АСУ, называют ее *информационной базой*.

Унифицированные документы представляют собой набор форм: организационно-распределительной информации в соответствии со стандартами; установленной отчетности, направляемой в вышестоящие организации; подготовленных разработчиками для внутрисистемного пользования.

При проектировании информационного обеспечения существенное значение имеет *время доступа* — интервал времени от момента поступления

в систему запроса до момента представления соответствующей информации пользователю. Помимо общего стремления к сокращению непроезводительной затраты времени на ожидание ответа в отдельных случаях этот параметр приобретает особое значение. Так, если после некоторого критического срока уже невозможно успеть принять и реализовать решение, ответ теряет смысл. Обычно существует некоторый предел, до которого время доступа можно снижать без каких-либо осложнений и специально принимаемых мер. Дальнейшее уменьшение времени доступа если и возможно, то требует дополнительных ресурсов и затрат. Здесь надо искать компромисс между затратами на снижение времени доступа и потерями от того, если это не будет сделано, т.е. необходима постановка и решение оптимизационной задачи.

Для систем реального времени с удаленными терминалами используется несколько иная характеристика — время ответа, под которым понимают интервал времени от момента окончания формирования запроса на терминале до момента индикации на нем первого символа ответа. Это время складывается из времени передачи по каналу запроса и ответа, времени доступа к информационным массивам и процессорного времени, затрачиваемого на ответ, с учетом обработки запросов других пользователей.

В режиме коллективного пользования важное значение имеют психологические особенности человека, участвующего в диалоге. По данным наблюдений психологов, в разговоре двух лиц время ожидания ответа составляет не более 2 с. При большей задержке ответа второй участник замечает это и начинает тяготиться беседой. При диалоге человека с ЭВМ задержка ответа ЭВМ свыше 15 с исключает диалоговое взаимодействие, пользователь недоволен ожиданием и делает попытки переключиться на другую деятельность; задержка от 4 до 15 с психологически затрудняет принятие решений в режиме диалога; от 2 до 4 с — обычно приемлема, но может оказаться чрезмерной в ситуациях, требующих высокой концентрации внимания пользователя; от долей секунды до 2 с — наиболее эффективное время для пользователя, особенно если ему надо помнить информацию, содержащуюся в предыдущих ответах; ответ за доли секунды, мгновенный с точки зрения пользователя, в некоторых ситуациях оказывает подавляющее действие и должен быть искусственно задержан.

Важное значение имеет способ хранения данных. Объем хранимых данных в современных системах чрезвычайно велик, он составляет многие миллионы алфавитно-цифровых знаков. Система динамична, данные непрерывно обновляются, а используются относительно небольшие, но самые различные части хранимых данных, причем в разных комбинациях. Управлять всем этим потоком достаточно сложно. Благодаря введению формализации обновление данных, перевод с одного уровня памяти на другой и физическая перезапись на разные носители могут быть запрограммированы и эти функции возложены на ЭВМ. В наибольшей степени эта тенденция нашла отражение в базах и банках данных, где пользователь полностью освобожден от заботы о физическом размещении данных в памяти ЭВМ.

Форма представления данных должна обеспечить наибольшее удобство их последующего использования. В пакетном режиме работы ЭВМ наиболее распространенной формой вывода данных были распечатки, содержащие алфавитно-цифровую информацию в виде отдельных строк или таблиц, реже — графический вывод в виде графиков или чертежей. Основным недостатком такого способа был вывод значительно большего объема данных, с тем чтобы пользователь сам выбрал из них нужные в каждом конкретном случае. Неудобство пользования объемными распечатками приводит во многих случаях к тому, что их вообще не используют. Ввод данных в пакетном режиме, обычно требующий перфорации большого объема данных и получения распечаток, занимал длительное время, что также не способствовало успеху системы.

Современные возможности ввода и вывода информации из ЭВМ, работающей в режиме коллективного пользования, позволяют строить системы информационного обеспечения таким образом, чтобы в наибольшей степени индивидуализировать общение пользователя с ЭВМ. Должен быть обеспечен ввод с клавиатуры запроса в таком виде, который учитывает степень подготовленности пользователя и в максимальной степени конкретизирует запрос. Ответ, лаконичный и конкретный, выводится в этом режиме на дисплей немедленно по получении запроса. Вместе с тем диалоговый режим коллективного пользования требует от проектировщика учета многих особенностей. Необходимо регламентировать доступ участников системы к хранимым данным. Даже в справочном режиме не всем пользователям разрешается получение любых данных о функционировании АСУ. Тем более ограничен доступ к введению изменений в массивы.

Наличие неподготовленных пользователей требует особого внимания при выборе метода построения диалога, учета психологии поведения человека при общении с ЭВМ. В хорошо разработанной системе должна быть налажена регистрация действий пользователей в виде архивных протоколов, позволяющих анализировать работу системы с пользователями.

Большое значение имеют вопросы надежности и, в частности, способность системы к самовосстановлению после случайных выходов из строя. Должна быть предусмотрена возможность сохранения важнейших функций системы при частичных выходах из строя аппаратуры, при перегрузках и других нарушениях режима. Необработанные задания также требуют регламентации — обработка некоторых теряет смысл, другие, наоборот, должны быть обработаны с наивысшим приоритетом сразу после восстановления системы. Восстановление системы, т.е. переход к выполнению функций в полном объеме, не должно требовать специальных действий пользователей — сотрудников АСУ.

Важный этап проектирования информационного обеспечения — анализ информационных потоков, проводимый на стадии предпроектного обследования, его результаты служат основой разработки модели системы управления.

Различают *макро- и микроуровень* исследования потоков информации [1.4]. Анализ информационных потоков на макроуровне позволяет понять общую схему работы объектов управления, провести совершенствование документооборота. Анализ на микроуровне обеспечивает выявление элементов информационного отображения объекта управления, взаимосвязей между ними, структуры и динамики информационных потоков.

Начальным этапом анализа информационных потоков является составление программы исследования. Намечается примерный перечень вопросов, на которые необходимо дать ответ в процессе изучения форм документации и недокументированных сообщений. В этот перечень входят [1.9]: назначение документа; количество его экземпляров; наименования обязательных реквизитов и показателей, а также указание должностных лиц, которые их заполняют; значимость показателей документов; частота формирования документов и разработки показателей.

Выделяют два метода анализа входящих и исходящих документов: *инвентаризации* и *типических групп*. Первый метод позволяет получить полную информацию обо всех документах и, таким образом, о потоках информации в существующих системах управления. Однако этот метод очень трудоемок и поэтому применяется достаточно редко. Второй метод используется значительно чаще, поскольку предусматривает регистрацию не каждого документа, а лишь представителей однотипных групп. Метод типических групп наиболее эффективен при обследовании систематизированных массовых и повторяющихся документов.

Множество документов, связанных с системой управления, можно разделить на следующие группы: 1) официальные положения и инструкции, регламентирующие функции подразделений и определяющие сроки и процедуры обработки информации и принятия решений; 2) входные документы, возникающие вне системы; 3) систематически обновляемые записи в виде картотек или книг, используемые в процессе работы; 4) промежуточные документы, получаемые и (или) используемые в процессе обработки данных; 5) выходные документы.

Рассмотрим более подробно некоторые из используемых в настоящее время методов анализа информационных потоков.

Графический метод. Он применяется для описания потоков информации главным образом на макроуровне. Отношения между элементами потока, в основном документов, изображают в виде описанной ранее графической структурно-информационно-временной схемы, на которой даются краткие пояснения, описывающие движение информации и материальных потоков [1.4].

Методы описания потоков информации с использованием теории графов. В настоящее время его используют наиболее широко. Выделяют методы на основе использования сетевой модели, графоаналитический метод и метод с использованием графов типа "дерево".

Метод с использованием сетевой модели базируется на использовании традиционных методов анализа. При этом под событием понимается определенный документ, являющийся конечным событием, если он представляет собой результат выполнения какой-либо работы, или начальным, если будет использоваться в дальнейшем ходе выполнения работ. Под работой понимается определенная задача или функция, выполняемая сотрудниками органа управления.

Графоаналитический метод основан на анализе матрицы смежности информационного графа. Исходными для анализа информационных потоков являются данные о парных отношениях между наборами информационных элементов, формализуемые в виде матрицы смежности. Под информационными элементами понимают различные типы входных, промежуточных и выходных данных.

Под матрицей смежности B понимают квадратную бинарную матрицу, проиндексированную по обеим осям множеством информационных элементов $D = \{d_1, d_2, \dots, d_S\}$, где S — число этих элементов.

В позиции (i, j) матрицы смежности записывают 1, если между информационными элементами d_i и d_j существует отношение R_0 , такое, что для получения значения информационного элемента d_j необходимо обращение непосредственно к элементу d_i . Наличие такого отношения между d_i и d_j обозначают в виде $d_i R_0 d_j$, а отсутствие — $\neg d_i R_0 d_j$, чему соответствует запись 0 в позиции (i, j) матрицы B . Для простоты дальнейших преобразований условно принимают, что каждый информационный элемент недостижим из самого себя:

$$d_i \neg R_0 d_i; \quad i = \overline{1, S}.$$

Матрице B ставится в соответствие граф информационных взаимосвязей $G(D, R_0)$. Множеством вершин графа $G(D, R_0)$ является множество D информационных элементов, а каждая дуга (d_i, d_j) соответствует условию $d_i R_0 d_j$, т.е. записи 1 в позиции (i, j) матрицы B . Структура графа $G(D, R_0)$ вследствие неупорядоченности сложна для восприятия и анализа. Составленная на основе первичного представления разработчика об информационных элементах и их взаимосвязях, она не гарантирована от возможных неточностей и ошибок. Для формального выделения входных, промежуточных и выходных данных, определения последовательности процедур их обработки, анализа и уточнения взаимосвязей на основе графа $G(D, R_0)$ строят матрицу достижимости.

Матрицей достижимости M называют квадратную бинарную матрицу, проиндексированную по обеим осям множеством информационных элементов D , аналогично матрице смежности B . Запись 1 в каждой позиции (i, j) матрицы достижимости M соответствует наличию для упорядоченной пары информационных элементов (d_i, d_j) смыслового отношения достижи-

мости R . Элемент d_j достижим из элемента d_i , т.е. выполняется условие $d_i R d_j$, если на графе $G(D, R_0)$ существует направленный путь от вершины d_i к вершине d_j , или если в процессе получения значения элемента d_j используется значение элемента d_i . Если $d_i \bar{R} d_j$, то отношение достижимости между элементами d_i и d_j отсутствует и в позиции (i, j) матрицы M записывают 0. Отметим, что отношение достижимости транзитивно, т.е. если $d_i R d_k$ и $d_k R d_j$, то $d_i R d_j$; $i, k, j = 1, S$.

Записи 1 в j -м столбце матрицы M соответствуют информационным элементам, которые необходимы для получения значения элемента d_i , соответствующего рассматриваемому столбцу, и которые образуют множество элементов предшествования $A(d_i)$ для этого элемента. Записи 1 в i -й строке матрицы M соответствуют всем элементам, достижимым из рассматриваемого элемента d_j и образующим множество достижимости $R(d_j)$ этого элемента. Информационные элементы, строки которых в матрице M не содержат единицу (нулевые строки), являются выходными информационными элементами, а информационные элементы, соответствующие нулевым столбцам матрицы M , являются входными. Это условие может служить проверкой правильности заполнения матриц B и M , если наборы входных и выходных информационных элементов известны. Информационные элементы, не имеющие нулевых строки или столбца, являются промежуточными.

Полученный на основе матрицы M граф информационных взаимосвязей может быть структуризован по уровням с использованием итерационной процедуры; это позволяет выделить основные этапы обработки данных, их последовательность и циклы обработки на каждом уровне.

Исходной информацией для описанного выше и других методов аналогичного назначения являются перечни входных и выходных элементов с указанием связей между ними. Получение этой информации является сложной слабо формализуемой задачей, основанной на изучении информационных потоков. Общая методика заключается в анализе этих потоков от выходов к входам. Исходя из функций системы и ее цели, определяют множество материальных выходов и для каждого из них — набор независимых параметров, полностью характеризующих данный выход по всей совокупности задач управления. Сопоставляя наборы параметров, характеризующих выходы, с перечнем информационных элементов, выделенных из множества исходных данных для решения задач, определяют, содержится ли данный параметр в обоих списках. Положительный результат является подтверждением необходимости включения данного информационного элемента в состав информационного обеспечения, а отсутствие совпадения требует более тщательного анализа.

Дополнительный анализ либо выявит ошибочный пропуск данного элемента в одном из списков, либо станет ясно, что данный параметр нет необходимости включать в состав информационного обеспечения. Накладывая результаты анализа одного параметра на другие и исключая дублирование, получают полный набор выходных информационных элементов.

Аналогичным методом получают набор входных информационных элементов. Парные взаимосвязи между информационными элементами выявляют также двумя способами для возможности перекрестной проверки — движением вдоль информационного потока, используя методику единичной нити, и по постановкам задач, рассматривая аналитические зависимости между параметрами. В обоих случаях следует иметь в виду возможность наличия промежуточных информационных элементов.

Метод с использованием графов типа "дерево" используют для описания системы потоков информации. Строится граф взаимосвязи показателей (типа "дерево") и так называемые графы расчетов, описывающие преобразование информации в процессе формирования отдельных показателей. При построении дерева взаимосвязи показателей ребра ориентируют с учетом иерархии от исходных к результирующим, что позволяет строить графы с более высокой степенью укрупнения.

Полученный комплекс графов отражает процесс движения и преобразования информации в системе и может быть использован для анализа эффективности этого процесса. Применение метода целесообразно, когда имеется результирующий (главный) показатель на каждом уровне [1.9]. Обработку результатов изучения информационных потоков и анализа документооборота в ряде случаев удобно проводить с помощью матричных информационных моделей [1.4].

Для анализа различных информационных потоков с целью их увязки используют метод реквизитов. Основным элементом сообщения, которое несет определенную смысловую нагрузку, является показатель, состоящий из одного или нескольких наименований реквизитов. Значения в документах обычно группируются по названиям реквизитов, и для анализа документооборота удобно использовать только наименования реквизитов. При этом для облегчения изучения документооборота создают специальные картотеки реквизитов с использованием карт с краевой перфорацией. На картах каждому реквизиту ставится в соответствие некоторый шифр; эта процедура составляет первый этап метода. Затем составляется таблица реквизитов промежуточной, хранимой и выходной информации. После составления таблицы в картотеку вносят дополнительные сведения о реквизитах, которые в дальнейшем используются для количественной оценки информации. Данный метод позволяет выявить идентичные реквизиты, дублирование документов, упорядочить потоки информации, рационально скомпоновать показатели, унифицировать реквизиты.

§ 4.3. Проектирование форм входных данных и выходных результатов

Решение задач управления предусматривает интенсивный обмен информацией между всеми участниками системы — как между людьми, так и между людьми и ЭВМ. Возможны два вида обмена информацией — документированный и не документированный.

Документированный — это обмен документами, главным образом на бумаге, подготовленными и заполненными людьми, либо ЭВМ в виде распечаток. Во втором случае это непосредственный или телефонный разговор, общение с ЭВМ через клавиатуру дисплея и вывод сообщений на его экран. Возможен также обмен информацией между ЭВМ по каналам связи, что может быть при необходимости документировано.

Недокументированный обмен информацией носит оперативный характер. Он может быть использован для: немедленного получения сведений от лица, являющегося их источником; устной передачи распоряжений; подтверждения выполнения полученных приказов или команд; сообщений о ситуациях, требующих безотлагательного вмешательства и т.п. Его преимуществом является скорость получения информации, отсутствие ограничений на содержание запросов и ответов, возможность введения любых уточнений и детализации. Основным недостатком недокументированных сообщений является отсутствие правовой и юридической ответственности источника информации за достоверность и точность выдаваемых сведений, невозможность последующей проверки содержания ранее переданного сообщения. Поэтому в случаях, связанных с материальной, финансовой или иной повышенной ответственностью, используют только документированные сообщения.

В производственных системах используют главным образом организационно-распорядительную, производственно-техническую и проектно-конструкторскую документацию, из них в АСУ применяют большей частью первые две группы, которые будут рассмотрены ниже. Проектно-конструкторская документация регламентирована группой Государственных стандартов на Единую систему конструкторской документации (ЕСКД). В наименьшей степени регламентирована организационно-распорядительная документация, поскольку смысловое содержание официальных писем, приказов и распоряжений трудно поддается формализации и обычно излагается в произвольной форме.

Проектирование документов. Для унификации организационно-распорядительной документации, применяемой в учреждениях, организациях и предприятиях, и более эффективного использования ЭВМ при ее подготовке и использовании, введены Государственные стандарты ГОСТ 6.38—72 и ГОСТ 6.39—72. В системе организационно-распорядительной документации выделены три группы документов: *организационная*, в которую входят уставы, инструкции, правила; *распорядительная* — постановления, распоряжения, приказы, решения; *справочно-информационная* — письма, справки, протоколы, акты. Виды документов могут подразделяться на отдельные разновидности. Каждый документ оформляется на бланке — листе бумаги с воспроизведенными на нем реквизитами, содержащими постоянную информацию. Реквизитом называют информационный элемент документа. Совокупность реквизитов определяет видовую принадлежность документа.

Применяют два вида бланков — для писем и для других документов. В бланке для писем содержится реквизит "индекс предприятия связи, почтовый и телеграфный адрес, номера телетайпа и телефона, номер счета в банке", а также постоянная часть реквизита "ссылка на индекс и дату входящего документа", имеющая вид "на № ... от ...". На бланках для других документов эти площади должны оставаться свободными для заполнения реквизитами "наименование вида документа" и "место составления или издания". Бланки могут иметь два формата — А4 (210х297 мм) и А5 (148х210 мм). Допускается два варианта расположения реквизитов — с угловым и продольным штампом. Бланки с угловым штампом предпочтительны, так как позволяют справа от штампа на свободной площади располагать реквизит "адресат", а количество документов, в которых такой реквизит требуется, составляет около 85% всей организационно-распорядительной документации. Бланки с продольным штампом, располагаемым вдоль верхней части листа, применяют в случаях, когда ввиду большого количества печатных знаков в наименованиях организации и вышестоящих организаций, которым она подчинена, эти реквизиты не могут быть размещены на площади, отводимой для углового штампа.

Разработку бланков начинают с определения перечня реквизитов, которые затем распределяют по площади макета бланка в виде отводимых для них площадей. В штампе бланка первым располагают наименование министерства или ведомства, под ним — наименование главного управления или управления, ниже — наименование организации. Допускается в штампе фирменного бланка промышленных предприятий и торговых фирм наименование организации выполнять рисованным шрифтом. Рекомендуется дату — число, месяц и год — оформлять в виде трех пар арабских цифр, разделенных точками, например 25.06.86.

В оформляемых на бланках документах обязателен заголовок к тексту. Его, как правило, начинают с предлога "О" (о чем документ, например "О разработке АСУ"). При необходимости разделения текста на части — разделы, подразделы — их нумеруют арабскими цифрами. Номер подраздела должен состоять из номера раздела и подраздела, разделенных запятой.

Документ адресуют в организацию и (или) ее подразделение. При адресовании документа лицу, не представляющему организацию, допускается указывать только фамилию, без указания должности. В составе реквизита "адресат" можно указывать почтовый адрес. При адресовании частному лицу сначала указывают почтовый адрес.

Некоторые документы (в основном внутренние документы в одной организации) допускается оформлять не на бланке. В этом случае наименование организации включают в наименование должности в реквизите "подпись". Если документ должны подписывать несколько должностных лиц, их подписи располагают одну под другой в последовательности, соответствующей занимаемым должностям. Для лиц равной по значению должности их подписи располагают на одном уровне. При визировании документа личная подпись должна сопровождаться датой.

В исходящих документах обязательно указывают фамилию лица, подготовившего текст документа, и номер его служебного телефона. В сопроводительных письмах этот реквизит не указывают.

Производственно-технические документы представляют собой, как правило, бланк установленной формы, который заполняется служебной информацией, т. е. данными, являющимися значениями переменных документа.

В документе различают следующие части (ГОСТ 6.10.2-75 и ГОСТ 2.105-79): *заголовочную часть*, содержащую наименование вида документа, краткое его содержание и периодичность выпуска, а также наименование раздела сводного документа, к которому относится данный документ; *головку*, состоящую из заголовков граф и подграф (столбцов); *боковик*, в котором приведены наименования строк; *содержательную часть*, представляющую собой двухмерную таблицу, заполненную данными полностью или частично; *оформляющую часть* — текстовые комментарии и подписи в конце таблицы.

Документ может состоять из нескольких разделов с промежуточными заголов-

ками. В документе может быть одна или несколько итоговых переменных, которые могут относиться как к отдельным разделам, так и к документу в целом.

При ручной подготовке формы документа составляют его макет на листе бумаги, предварительно определив содержание заголовочной части, головки и боковика. Основная задача — так составить документ, чтобы его было одинаково удобно заполнять и использовать. Это определяется расположением столбцов и строк. Распространенной ошибкой является выделение для значений переменных недостаточного места; следует выделять такую ширину столбца или свободного места для заполнения, чтобы оно было достаточным для записи максимально возможного по числу символов значения соответствующей переменной плюс два пробела — по одному спереди и сзади. При записи от руки на один символ выделяют примерно 3 мм. Общим ограничением является стандартная ширина предполагаемого формата. Допускается расположение формы документа по длине формата. Если суммарная ширина всех граф больше ограничения на ширину или длину формата, то надо либо пересмотреть перечень граф и исключить менее важные, либо продлить форму на второй лист, что менее удобно при использовании. Во всяком случае недопустимо хотя бы незначительно уменьшать ширину граф из-за того, что они не помещаются на листе.

В современных системах имеется программное обеспечение, позволяющее автоматизировать создание формы документа по минимальной информации, сообщаемой пользователем. В наиболее развитых системах такого типа пользователь вводит информацию, содержащую наименование (заголовочную часть) документа, наименование граф с указанием требуемой ширины, в той последовательности, в которой они должны быть расположены в документе. Программа оптимизирует форму документа по заданному критерию и допускает корректировку спроектированной формы по желанию пользователя в режиме диалога с ЭВМ.

Во многих случаях документы можно формировать путем заполнения заранее подготовленной типографским способом формы, что существенно упрощает и ускоряет выдачу готовых документов.

При создании АСУ следует стремиться к максимальному сокращению числа документов в системе и полному исключению их ручного заполнения. Необходимость каждого типа документа должна быть доказана требованиями финансовой или юридической ответственности, например подтверждения факта передачи материальных ценностей от одних ответственных лиц другим и т.п. Все без исключения документы такого типа должны готовиться на ЭВМ по данным, вводимым непосредственно ответственным лицом; при этом необходимо обеспечить соблюдение правил защиты от несанкционированного доступа и ведение на ЭВМ архивных протоколов действий пользователя.

Проектирование диалога "человек—ЭВМ". Недокументированный обмен информацией в виде диалога между человеком и ЭВМ может и должен быть в известной степени формализован, по крайней мере, до создания реальной возможности ведения свободного диалога на естественном языке в форме речевого общения. Последняя задача поставлена в японском проекте создания ЭВМ пятого поколения. В современных развитых системах человек вводит сообщения в ЭВМ с помощью клавиатуры, а ЭВМ выдает их через экран дисплея.

Язык, на котором ЭВМ общается с пользователем, должен быть ясным, легким для понимания и предельно вежливым. Проектирование языка общения с ЭВМ, называемого входным или языком пользователя, заключается в определении словарного состава и синтаксиса. Чем меньше словарь и проще синтаксис, тем легче и быстрее осваивается язык, но тем меньше его возможности.

В простейшем случае пользователь оперирует сообщениями типа *да*, *нет*, а синтаксис практически отсутствует. Научиться общению с ЭВМ на таком языке самый неподготовленный и неквалифицированный человек может за несколько минут. Однако содержание диалога в этом случае жестко определено рамками, навязанными ЭВМ, т.е. заложенными в нее разраб-отчиками системы.

Наиболее развитые языки общения максимально приближены к естественному. Разрешение трудностей такого общения возложено главным образом на машину, что облегчает диалог неподготовленному пользователю. В этом случае при вводе сообщения допускаются не искажающие смысла орфографические ошибки, замена одних слов другими и перестановка их в предложении в рамках принятого синтаксиса; входные сообщения могут варьироваться в определенных пределах и ЭВМ должна уметь их идентифицировать. Главная задача проектировщика такого развитого языка — исключение возможности неоднозначного определения смысла сообщения. Особого внимания требуют *омоимы* — слова одинакового написания, но с различным смыслом; *синонимы* — различные слова с одинаковым смыслом; *паронимы* — близкие по звучанию и написанию, но имеющие не одинаковый смысл. Обязанность помнить все введенные в язык ограничения и запреты на отдельные слова или конструкции предложений возлагается в меньшей степени на пользователя и в большей — на ЭВМ. При попытке ввести недопустимое в данном языке сообщение ЭВМ должна различить это, указать пользователю на допущенную им ошибку и, кроме того, подсказать возможные варианты. Вообще помощь в виде подсказки должна предоставляться пользователю по его просьбе на любом этапе диалогового общения с ЭВМ. В процессе диалога ЭВМ не должна требовать от пользователя выполнения таких процедур, которые не являются обязательными или которые она может выполнить сама.

При проектировании диалога следует иметь в виду, что ЭВМ должна быстро реагировать на обращения, в том числе и в тех случаях, когда она не может выполнить то, что от нее просят. При этом от ЭВМ требуется обоснование причины отказа и указание, что следует делать пользователю в сложившейся ситуации.

Диалог должен проектироваться с учетом возможности обращения к ЭВМ пользователей с различной степенью подготовленности. Для хорошо подготовленного пользователя важна скорость получения желаемых результатов, для менее подготовленного большее значение имеет хорошее взаимопонимание. Один и тот же пользователь в пределах одного диалога может в одних вопросах быть более подготовленным, а в других — менее. Это

требует соответствующей гибкости от разработчика, создания системы ведения диалога, адаптирующейся к конкретному пользователю. Определив уровень подготовленности пользователя, ЭВМ должна дальше работать с ним на этом уровне, не утомляя излишними подробностями и разъяснениями и в то же время не требуя постоянного обращения за подсказкой, что вызывает психологический дискомфорт. Повышение уровня компьютеризации большей части населения, предусмотренное при развитии среднего образования, позволит постепенно повышать уровень представлений о возможностях неподготовленного пользователя.

Пользователь может прервать свою работу в любой момент и на любом месте. При этом не следует устанавливать какие-либо специальные правила и процедуры окончания или прерывания сеанса диалога, без выполнения которых система заблокируется или выйдет из строя. Более того, в процессе диалога пользователь должен иметь возможность в любой момент вернуться назад и продолжать диалог с нужного места, не обязательно с начала. Он не обязан заботиться и о том, как его общение с диалоговой системой сказывается и какие может иметь последствия для других работ. В необходимых случаях должны быть обеспечены защита информации от несанкционированного доступа или требования секретности.

Кроме языка общения с пользователем для повышения эффективности диалога в ходе проектирования системы разрабатывают сценарий диалога и дисплейные форматы.

Сценарий диалога — это форма представления логической структуры диалога, определяющей последовательность сообщений, идущих от пользователя к ЭВМ и обратно. Структуру диалога можно представить также в виде схемы, аналогично схемам алгоритмов.

Диалог может вестись с подсказками или без подсказок человеку со стороны ЭВМ, со свободным или форматированным вводом и выводом сообщений.

Большое число возможных типов диалога, рассчитанных на работу с подсказкой, можно разделить на два основных типа — выбор одного из заранее подготовленных ответов по так называемому "меню" и ввод ответов на вопросы в форме "заполни пустые места".

При обращении к системе в режиме "меню" пользователь получает перечень возможных его ответов. Выбор осуществляется установкой маркера у соответствующего режима или набором номера режима на клавиатуре. Если перечень возможных режимов длинный, осуществляется "перелистывание страниц" путем вывода на экран частей перечня либо построчное перемещение позиций перечня снизу вверх на экране. Другим решением является выбор по дереву — сначала выводятся наименования нескольких крупных частей перечня, по выбранной части выводятся более мелкие группы, и так до тех пор, пока не будет выбран окончательный конкретный ответ. При поиске по дереву следует учитывать, что, с одной стороны, желательно иметь ограниченное число уровней выбора, обычно не более трех обращений, а с другой — список каждого уровня сам по себе не должен быть длинным.

При использовании режима "заполни пустые места" пользователь получает на экране некоторую форму, пустые позиции которой он должен заполнить. При разработке таких форм следует обеспечить ясное и однозначное представление пользователю, чего от него ждут, избегать возможных ошибок заполнения. Существует большое количество модификаций этих способов — "множественное меню", когда пользователь может одновременно указать несколько позиций перечня; вывод в качестве образца заполненной формы, в которой пользователь должен заменить указанные ему позиции своими данными; использование сочетания этих двух основных методов и многие другие.

Языки проектирования. Их используют для формализации и повышения эффективности процесса проектирования, в том числе для его автоматизации. Важным требованием является вывод результатов проектирования в удобной для дальнейшей работы и корректировки форме.

Для этой цели созданы специальные языки документирования. Их основным назначением является удовлетворение требований стандартов к проектной документации, оптимизация процессов ее подготовки и выпуска, автоматизация корректировки.

Языки проектирования должны обладать определенной универсальностью, а их словарь — содержать термины, понятные и общепотребительные для пользователей систем. Одним из наиболее развитых представителей таких языков являются языковые средства АРИУС [2.27].

Языковые средства АРИУС ориентированы на автоматизированное проектирование систем обработки данных. При этом обеспечивается формализованное описание: выходных документов и сообщений, ориентированных на использование в задачах управления; входных форматированных сообщений, поступающих с пультов ручного ввода; локально-функциональных связей между переменными. При этом используется аппарат функциональных отношений, таблиц решений и неэлементарных процедур.

Функциональные связи между переменными подразделяются следующим образом: элементарные случаи определения табличных переменных; связи, выражаемые арифметическими формулами; связи в форме таблиц решений; вычислительные процедуры, не реализуемые средствами АРИУС.

Совокупность локально-функциональных описаний переменных некоторого документа табличной формы составляет его микроописание. Макроописание характеризует документ в целом — его шифр, заголовочную часть и головку, определяющую форму представления документа на терминале.

В алфавите языковых средств АРИУС 63 символа: 31 буква, 10 цифр, 7 знаков отношений, знаки конъюнкции и дизъюнкции, 5 разделителей и 8 специальных символов, включая пробел.

Отдельные табличные переменные описываются семью атрибутами: локальный код, дескриптор, формат, количество строк, формула определяемости, функционеры и вид. Из них обязательными являются локальный код, формат и формула определяемости. Количество строк указывается только для ключевых переменных.

Локальный код — фактически двухзначный порядковый номер этой переменной в документе. Совместно с шифром документа обеспечивает идентификацию переменной в потоке данных.

Дескриптор — наименование столбца таблицы, строки или отдельной позиции.

Формат значений переменных записывают в обозначениях языка ФОРТРАН.

Формула определения – арифметическое выражение, записанное для табличных переменных с использованием цифровых констант и символов элементарных функций.

Функционер определяется как пользователь, работающий с данной переменной.

Вид определяет смысловое содержание переменной – плановый или учетный показатель и т.д.

Проектное описание системы обработки данных, составленное разработчиком с помощью языковых средств АРИУС, представляет собой форматированный текст (Ф-текст) на бланках стандартной формы, называемых *терминальными спецификациями*. Совокупность значений атрибутов для некоторого документа представляет собой Ф-текст описания этого документа. Значениями атрибутов могут быть как числа, так и отдельные слова или выражения языка.

Имеется четыре формы терминальных спецификаций: карта анализа и постановки задач (КАПЗ), форма представления информации (ФПИ), таблица решений (ТР) и таблица неэлементарных процедур (ТНП). Основными формами терминальных спецификаций являются КАПЗ и ФПИ. Они служат для описания выходов проектируемой системы, их связей с пользователями, а также для описания структуры внутренних рабочих массивов и функциональных связей между переменными. В терминальных спецификациях ТР и ТНП описываются процедуры принятия решений при наличии альтернативных путей обработки данных и неэлементарные процедуры обработки.

§ 4.4. Информационные языки и классификаторы

Информационные языки. Одним из условий эффективного функционирования АСУ практически любого типа и назначения является рациональная организация общения пользователей с системой. В качестве средства общения используются *информационные языки*, представляющие собой совокупность словаря и правил записи и передачи в ЭВМ сообщений об объектах, ситуациях и запросах, позволяющих использовать формализованные процедуры их обработки.

Разработка информационных языков преследует следующие основные цели: включение в функции ЭВМ большого числа арифметических, логических и прочих операций, связанных с поиском, обработкой, редактированием и выдачей информации в форме, удобной для использования; упрощение и повышение эффективности общения с ЭВМ пользователей различной степени подготовленности; унификацию языка системы и реализацию принципов интегрированной обработки информации; упрощение и сокращение числа типов используемых форм планово-экономической и управленческой документации за счет внутримашинного хранения данных.

По мере развития вычислительной техники повышается интеллектуальный уровень и упрощается общение с ЭВМ. Развитие процесса взаимодействия человека и машины, имеющего первостепенную важность для облегчения и повышения эффективности работы с ЭВМ, шло в направлении создания языков высокого уровня, оперирующих понятиями, близкими к естественному человеческому языку. Вначале появились так называемые процедурные языки, операторы которых воспринимались ЭВМ как указания о выполнении тех или иных процедур. Следующий шаг – создание непроецедурных языков, оперирующих более близкими к естественному языку

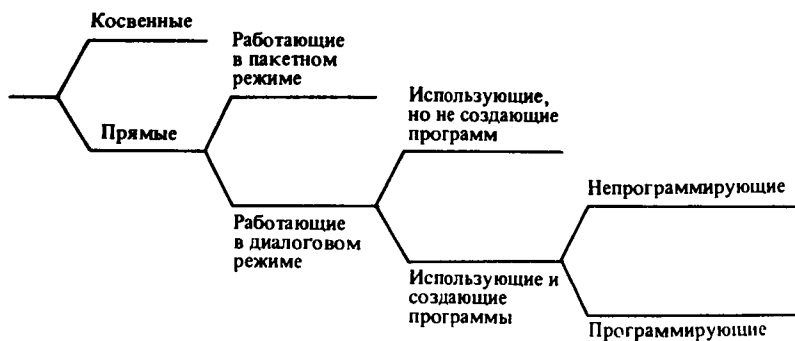


Рис. 4.1

понятиями. Ожидается, что интеллектуальный уровень ЭВМ пятого поколения позволит использовать в диалоговом общении с ними просто естественный язык, обеспечить ввод и вывод информации в форме человеческой речи, графических данных, изображений и других средств обычного общения между людьми. Пока таких возможностей нет, основным средством общения пользователей с ЭВМ являются специальные информационные языки.

При разработке информационных языков АСУ необходимо учитывать уровень квалификации и подготовленности пользователей, которые будут с ней взаимодействовать при эксплуатации системы. В настоящее время большинство методов и средств обработки данных и информационных языков ориентированы на пользователей, являющихся программистами, т.е. имеющих достаточную квалификацию и профессиональную подготовку. Однако широкие масштабы внедрения вычислительной техники в различных сферах производства и управления обуславливают необходимость разработки информационных языков, ориентированных на пользователей, как правило, не имеющих высокой квалификации в области программирования, так называемых конечных пользователей.

Можно выделить пять категорий конечных пользователей (рис. 4.1). В первую категорию входят *косвенные пользователи*, которые не используют непосредственно вычислительную технику, а обращаются к ней через посредников. К таким пользователям относятся руководящие работники предприятий, организаций и т.п., получающие необходимую им информацию с помощью других людей, чаще всего подчиненных сотрудников организации. Во вторую категорию входят *прямые пользователи*, непосредственно взаимодействующие с ЭВМ в пакетном или диалоговом режиме. Среди прямых пользователей, которые работают в диалоговом режиме, можно выделить третью категорию пользователей — *использующих* в своей деятельности возможности АСУ, уже заложенные в виде готовых программ обработки данных, и таких, которые, кроме того, *могут сами разрабатывать* задачи и программы. Последний тип пользователей, в свою очередь включает пользователей, *не использующих традиционное программи-*

рование, а работающих с заложенными в систему возможностями генерации прикладных программ, и *программирующих пользователей* (категории четвертая и пятая соответственно). В настоящее время число программирующих конечных пользователей достаточно велико, но по существующим оценкам они составляют меньшинство, поэтому при разработке АСУ следует ориентироваться в основном на непрограммирующих пользователей.

Информационные языки можно подразделить на следующие классы [1.9]: информационные языки с простыми функциями, информационно-поисковые языки, информационно-справочные языки, информационные языки автоматизированных систем обработки данных (АСОД), информационно-логические языки АСУ.

Информационные языки с простыми функциями служат для реализации простейших операций загрузки, хранения и выдачи текстовой информации из ЭВМ.

Информационно-поисковые языки предназначены для организации эффективного поиска, обновления и выборки хранимой в памяти ЭВМ информации.

Информационно-справочные языки выполняют те же функции, что и информационно-поисковые, но обладают более широкими возможностями: языки данного типа позволяют осуществлять несложные операции группировки, сортировки, агрегации показателей и т.п. для формирования ответов по запросам пользователей. На практике языки данного класса применяют в сочетании с языками формулирования информационных запросов к системе, называемыми *языками запросов*.

Информационные языки АСОД предназначены для ввода и размещения данных в специализированных массивах, используемых для решения определенных комплексов функциональных задач.

Информационно-логические языки осуществляют отдельные логические операции типа логических выводов.

Проблемы построения и использования информационных языков приобрели важное значение в связи с ростом потока обрабатываемой информации с помощью ЭВМ.

Наиболее важным отличием информационных языков от естественных является их однозначность — каждому смысловому значению соответствует одна и только одна последовательность слов и символов и наоборот.

Использование естественного языка в АСУ позволяет исключить индексирование и кодирование информации вне ЭВМ, снять ограничения на содержание запросов. Однако применение естественного языка при вводе информации и ее обработке увеличивает время обработки и требует исправления ошибок, появляющихся из-за неоднозначности и нечеткости языковых выражений.

В информационном языке используют средства, позволяющие с помощью специальных индексов определить группировки естественного языка, имеющие одинаковый смысл. Сведения о существовании этих группи-

ровок позволяют увеличить семантические возможности информационного языка и повысить полноту выдачи информации по запросу.

В отличие от естественных информационных языки значительно ограничивают возможности перефразирования. Это определяется не только устранением синонимов, но и использованием строгих правил синтаксиса, определяющих расположение слов в записи. Для информационных языков фонетическое выражение знаков не обязательно, поскольку эти языки ориентированы на письменную форму выражения, состоящую из букв, цифр и некоторых специальных знаков.

Большинство используемых информационных языков принадлежит к *равномерным кодам*, у которых число символов во всех кодовых комбинациях одинаково. Свойство равномерности кода вытекает из природы информационного языка, требующей удобной и компактной формы записи информации, что особенно важно при машинной обработке. Информационные языки отличаются от естественных тем, что при передаче информации в них стремятся ограничить избыточность.

В информационных языках важное значение имеют имена (понятия), так как они в основном предназначены для передачи существенных сведений о предметах и явлениях. Поэтому в ряде языков отсутствуют связи и весь аппарат выражения отношений между понятиями. Это так называемые языки без грамматики. В языках с грамматикой последняя выполняет подчиненную словарю роль и имеет иной характер по сравнению с грамматикой естественных языков.

Отметим некоторые основные требования, предъявляемые к информационным языкам при их разработке и использовании. Поисковые образы обычно предназначены для того, чтобы в какой-то момент процедуры поиска их читал человек, ведущий поиск. Поэтому для их составления нужно использовать словарь и синтаксис, которые были бы понятны лицу, ведущему поиск. Значит, информационный язык должен быть близок к естественному или легко изучаем. Форма, словарь и синтаксис информационного языка должны быть понятны человеку, достаточно просто и однозначно воспринимаемы ЭВМ. Это предполагает большую строгость в языке, чем это необходимо при ручной обработке. Используемый в АСУ информационный язык должен быть ориентирован на основную единицу экономической информации — показатель, для записи которого обычно используется простой информационный язык с позиционным синтаксисом.

Основные компоненты информационных языков — словарный состав и синтаксис. *Словарный состав* информационного языка — совокупность слов, используемых в данном языке. В информационном языке "слово" аналогично понятиям "слово", "словосочетание", "фраза" в естественных языках. *Синтаксис* языка — комплекс правил построения из элементов словарного состава фразеологических (синтаксических) единиц с такими значениями, которые нельзя выразить с помощью отдельных слов основного словарного состава.

Под *словарем* информационного языка понимают списки, которые

определяют словарный состав языка, а также написание, употребление и значение слов. Под определение словаря подходят термины "тезаурус", "словарь терминов", "гlossарий" и др. Каждый из них можно рассматривать как специальный словарь языка, определяющий значения слов, а также связей между ними. Словарь является органической частью информационного языка.

Для поиска некоторого документа или факта в информационно-поисковой системе используют *поисковое предписание* или *поисковый образ*, представляющий собой текст на информационно-поисковом языке, являющийся результатом перевода запроса с естественного языка. Поисковый образ должен однозначно соответствовать запрашиваемому документу или факту. В естественном языке для этого используют *ключевые слова*, отображающие определенные части смыслового содержания запроса. В информационном языке в составе поискового образа для описания смыслового содержания запроса используют *дескрипторы*. Каждый дескриптор является единицей информационного языка и ставится в однозначное соответствие группе ключевых слов. Синтаксическую единицу, составленную из дескрипторов информационного языка, называют *термином*. Группа терминов, связанных вместе по определенным правилам, образует *фразу*.

В настоящее время в различных странах разработано и используется несколько тысяч информационных языков. В сложных иерархических системах управления, а также при взаимодействии систем различного уровня и назначения возникает проблема информационного обмена между системами или их подсистемами. Совокупность языковых средств, обеспечивающих решение этой задачи, называют *языковой структурой*. Рациональный выбор языковой структуры, обеспечивает обмен информацией с минимумом затрат на перекодирование информации. Можно выделить следующие основные варианты языковых структур: с языковыми приоритетами; с независимыми равноправными языками в подсистемах; с языком-посредником; с единым языком.

В структуре с *языковыми приоритетами* каждая подсистема имеет свой язык, однако при обмене информацией между подсистемами разного уровня всегда используется язык подсистемы более высокого уровня. Структура с языковыми приоритетами часто используется в системах с иерархической организацией подчинения. Преимущество структуры — малое число трансляторов и словарей.

В структуре с *независимыми языками* количество сообщений в каждой отдельной подсистеме меньше, чем во всей системе. Это позволяет сократить среднюю длину сообщений, упростить запись и поиск информации в отдельной подсистеме, сокращая затраты внутри подсистемы на обработку сообщений. Однако каждая подсистема вынуждена хранить большое число трансляторов и словарей и затрачивает значительные ресурсы на перевод информации с языка абонента на свой внутренний язык. Эти затраты тем выше, чем интенсивнее обмен между системами. Обычно число языков в такой структуре равно числу подсистем, каждый язык используется внутри

одной подсистемы. Возможны два способа обмена информацией между подсистемами: на языке той системы, в которую обращаются с запросами, и на языке подсистемы, которая обращается с запросом.

В структуре с *языком-посредником* в каждой подсистеме может использоваться свой внутренний язык, а при обмене информацией между подсистемами используется язык-посредник. В сети взаимодействующих АСУ роль языка-посредника может выполнять система общесоюзных классификаторов. Язык-посредник должен включать в себя все понятия, используемые при обмене информацией, и обеспечивать однозначность всех первичных сообщений.

В структуре с *единым языком* для обмена информацией между подсистемами не требуется перевод с одного языка на другой и отпадает необходимость в организации трансляторов. Однако на практике создание единого информационного языка для иерархической АСУ практически невозможно, так как это приводит к необходимости иметь громоздкие словари, а следовательно, к большей длине сообщений и увеличению затрат на обработку этой информации. В то же время для более простых систем, включающих в себя несколько АСУ на одном уровне иерархии, применение единого языка может оказаться выгодным.

В последние годы широкое распространение получила концепция *банков данных*. Банк рассматривают как единую многоцелевую информационную базу, обеспечивающую исходными данными решение различных по назначению и содержанию задач в режиме коллективного доступа к хранимой информации. В связи с широким развитием банков данных возникла необходимость разработки специальных языков, с помощью которых осуществляется взаимодействие с банком данных. В этот набор входят языки описания данных, языки запросов и языки управления базами данных.

Язык описания данных предназначен для записи схем баз данных на трех уровнях. Эти уровни позволяют определить общую логическую структуру базы данных, не зависящую от прикладных программ и способов ее физической реализации в памяти системы, а также определить ее отображение в структуре данных прикладных программ и в структуре памяти. Язык описания данных должен включать в себя средства определения механизмов защиты и секретности, процедуры ввода связанных изменений и вычисления виртуальных элементов данных.

Язык запросов служит для связи пользователей с банком данных. Эти языки и их интерпретаторы могут быть написаны пользователями, чтобы наиболее полно определить специфику приложения. Язык запросов транслируется в базисный язык манипулирования данными. Интерпретатор этого языка инициирует непосредственное обращение к банку данных. Кроме того, операторы языка манипулирования данными могут быть включены в языки программирования высокого уровня.

Программное обеспечение банка данных — система управления базами данных — содержит компиляторы и интерпретаторы языка описания дан-

ных, языка манипулирования данными и языка запросов, манипулятор банка данных и набор сервисных программ манипулятора и администратора банка данных.

Классификаторы информации. Для упорядочения и систематизации описания предметов и понятий их распределяют по некоторым признакам на группы. Процесс такого упорядоченного распределения называют *классификацией*. Учитываемые при этом признаки сходства или различия объектов называют *основаниями классификации*. Совокупность правил распределения оснований классификации образует *систему классификации*. В результате применения некоторой системы классификации создается *классификатор* — систематизированный свод наименований и обозначений или шифров группировок.

Система классификации должна обеспечивать: достаточную *емкость*, чтобы вместить все множество классифицируемых объектов; *гибкость*, позволяющую расширять это множество без нарушения структуры классификации; *глубину*, позволяющую учитывать дополнительные признаки различия; *возможность сопряжения* с другими классификациями; *простоту* ручного и автоматизированного ведения классификатора.

Различают иерархическую и фасетную системы классификации. В иерархической системе между группировками устанавливается отношение подчинения. Сначала по одному признаку выделяют крупные группировки, затем внутри них по другому признаку более мелкие и т.д. При фасетной системе весь набор признаков формируется в параллельные фасеты. Исходное множество объектов делится на ряд независимых группировок по различным комбинациям числовых характеристик признаков, взятых из соответствующих фасетов.

Каждому объекту в классификаторе присваивается шифр в соответствии с принятой системой кодирования. Цель кодирования — представить идентификатор объекта в более компактной и удобной форме, чем его полное наименование. Наиболее важными свойствами системы кодирования являются: *полнота*, позволяющая охватывать все множество объектов; *унифицированность*, обеспечивающая единство шифров всех объектов; *однозначность*, сохраняющая уникальность идентификатора, представленного шифром; *дешифруемость*, позволяющая обрабатывать шифры независимо от их семантики и затем отыскивать соответствующий объект; *избыточность и гибкость*, дающие возможность расширения изменений в наборе шифров без нарушения принятой структуры классификации. При выборе системы кодирования желательно иметь шифры минимально возможной длины, в то же время сохраняя их информативность, т.е. возможность достаточно простого сопоставления шифра с полным наименованием объекта. Рекомендуется для кодирования использовать десятичные цифры и буквы русского алфавита, что облегчает обработку шифров.

Существуют четыре основные системы кодирования технико-экономической информации, из которых первые две называют классификационными, так как они основаны на заранее осуществленной классификации, остальные две — регистрационными.

Последовательная система кодирования соответствует иерархической системе классификации. Шифр каждой нижестоящей группировки образуется путем добавления элементов к шифру вышестоящей. Такая система обладает хорошей информативностью, но шифры имеют большую длину и сложную структуру, в связи с чем ее используют в системах с небольшими изменениями шифров, например в классификаторах отраслей народного хозяйства, органов государственного управления.

Параллельная система кодирования, называемая также разрядной или позиционной, соответствует фасетной системе классификации. В отличие от последовательной системы, здесь нет зависимости признака, записанного в одних разрядах шифра, от других. Например, шифр документа 37200407 означает, что документ формируется в производственном отделе 37, подотделе 20, содержит плановую информацию 04, номер документа в подотделе 07.

Порядковая система кодирования — наиболее простая, объектам присваиваются последовательные номера. Для удобства используют равномерный код, добавляя слева нули в разрядах, дополняющих шифр до нужного числа разрядов — 007, 027, 342 и т.д. Шифры порядковой системы имеют малую длину, но не содержат никаких сведений об объекте. Ее используют при малом числе объектов и одном классификационном признаке.

Серийно-порядковая система кодирования отличается от порядковой тем, что при наличии двух и более классификационных признаков объекты делят на группы, каждой из которых выделяется серия номеров, присваиваемых аналогично порядковой системе — одной группе присваивают номера от 001 до 200, второй — от 201 до 350 и т.д. Другой вариант этой же системы — выделение нескольких разрядов для одного признака, а остальные — для последовательного присвоения номеров внутри выделенного признака, например 25001 ... 25850; 32001 ... 32640 и т.д.

Разработка классификаторов для конкретной АСУ должна осуществляться на основе единых классификаторов, утвержденных как обязательные для отрасли или страны в целом. Если это по обоснованным причинам неприемлемо, разрабатывают локальные классификаторы со средствами перевода их в отраслевые и общесоюзные.

В нашей стране создается *Единая система классификации и кодирования технико-экономической информации* (ЕСККИ), состоящая из трех частей — научно-технические и методические материалы; общесоюзные классификаторы; автоматизированная система ведения общесоюзных классификаторов.

Общесоюзный классификатор технико-экономической информации. Цель создания общесоюзных классификаторов технико-экономической информации (ОК ТЭИ) — научно обоснованная классификация и унификация обозначений технико-экономической информации для обеспечения ее обработки с применением средств вычислительной техники при планировании, учете и управлении народным хозяйством. Система содержит около 30 классификаторов по различным видам информации. Основными из них

являются общесоюзные классификаторы: промышленной и сельскохозяйственной продукции; управленческой документации; технико-экономических показателей; обозначений объектов административно-территориального деления СССР и союзных республик; обозначений органов государственного управления СССР и союзных республик; месторождения полезных ископаемых; стран мира и территорий; министерств, ведомств и других потребителей материально-технических ресурсов; единиц измерений, использованных в АСУ и ряд других. Структурная схема ОК ТЭИ приведена на рис. 4.2.

Для примера приведено построение систем классификации, принятых в Общесоюзном классификаторе промышленной и сельскохозяйственной продукции (ОКП) и Общесоюзном классификаторе управленческой документации (ОКУД).

Пример 4.1. Система классификации и кодирования промышленной и сельскохозяйственной продукции основывается на иерархических классификационных группировках. В соответствии с принятой системой классификации используется единая по разрядности система цифрового кодирования продукции. Код конкретной продукции представляется десятью цифровыми знаками следующим образом:

Высшие классификационные группировки (ВКГ ОКП)						Дальнейшая конкретизация видовой группировки			
Класс	Под-класс	Группа	Под-группа	Вид		Внутривидовые группировки			
1	1	1	1	1	1	1	1	1	1

В ОКП первые ступени классификации составляют высшие классификационные группировки с единым по длине кодовым обозначением в шесть разрядов, содержащие пять иерархических уровней. По признакам отраслевой принадлежности вся продукция распределена по 98 классам (первые два разряда). Классы делятся на подклассы, подклассы на группы и т.д. Например, код 43 2126 0512 обозначает:

43 — класс — "Приборы и средства автоматизации специального назначения";

43 2 — подкласс — "Приборы, аппараты, оборудование и посуда из стекла, кварца и фарфора";

43 21 — группа — "Приборы и аппараты из стекла, кварца и фарфора";

43 212 — подгруппа — "Приборы для измерения влажности и температуры";

43 2126 — вид — "Термометры специальные";

43 2126 05 — типовой представитель — "Термометры для установок, определяющих качество нефтепродуктов СП-28—СП-34";

43 2126 051 — типовой представитель — "Термометры для установок, определяющих качество нефтепродуктов СП-28 — СП-34 обыкновенного исполнения";

43 2126 0512 — типовой представитель — "Термометр для определения температуры топлива в топливной магистрали СП-29".

Пример 4.2. Объектами классификации в ОКУД являются унифицированные формы документов, входящие в унифицированные системы документации (УСД). Классификация ведется по иерархическому принципу и состоит из трех уровней: класс форм; подкласс форм; регистрационный номер формы.

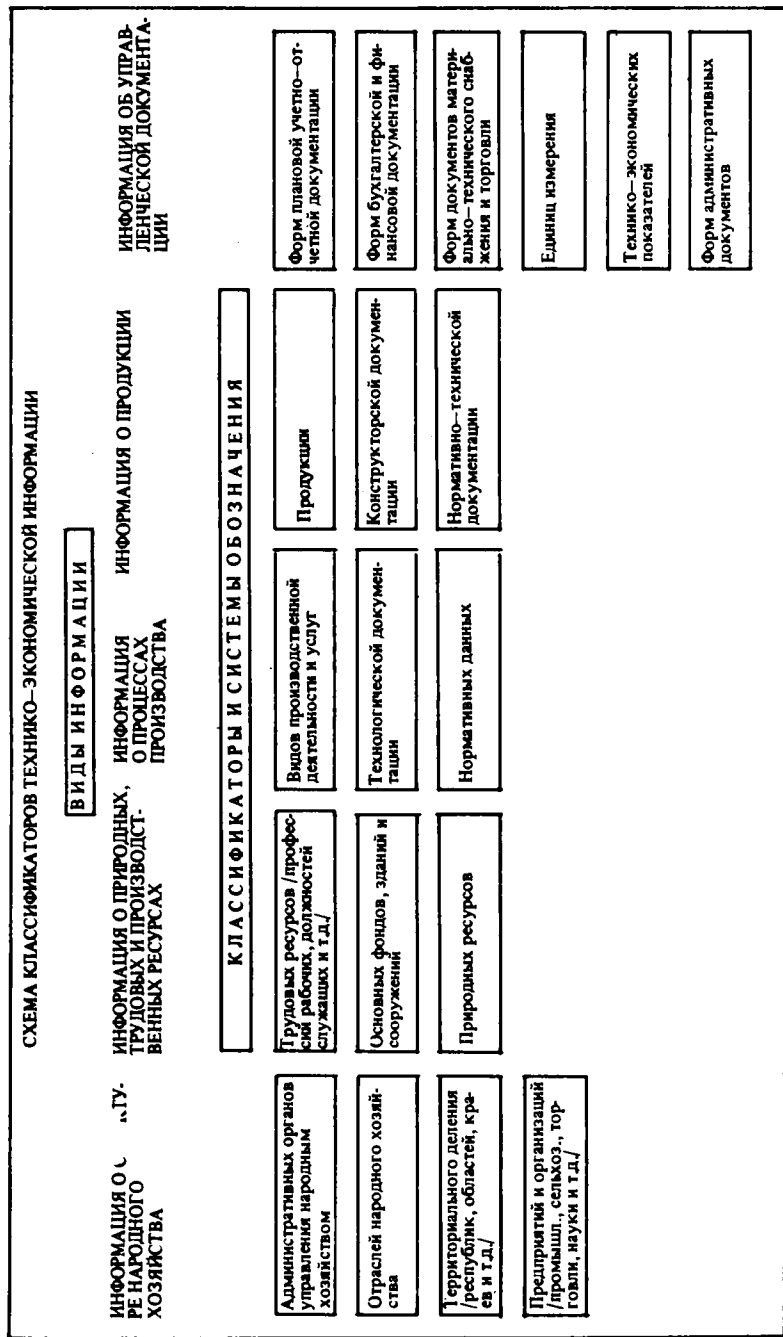


Рис. 4.2

В соответствии с УСД выделены 13 классов форм. Код формы представляется восемью десятичными знаками следующим образом:

Класс форм		Подкласс форм		Регистрационный номер формы			Контрольное число
1	1	1	1	1	1	1	1

Для унифицированной системы отчетно-статистической документации (класс форм 06) приняты коды дополнительных признаков форм, содержащие четыре десятичных разряда:

Коды дополнительных признаков			
Способ представления	Периодичность		Срок хранения
1	1	1	1

Например, код

06 06 271 1 1 08 3 обозначает:

06 – форма относится к отчетно-статистической документации;

06 06 – форма содержит документацию по статистике промышленности;

06 06 271 – форма представляет собой отчет о работе тепловой электростанции;
1 – контрольное число кода (поразрядная сумма цифр, записанная в двоичной системе);

1 08 3 – обозначает, что форма представляется почтой, ежемесячно и хранится пять лет.

Отраслевые классификаторы. Они строятся на базе ОК ТЭИ. По способу использования в АСУ их можно разделить на две группы: классификаторы, используемые на предприятиях (организациях) при подготовке информации, передаваемой на ГВЦ для решения задач; классификаторы, используемые непосредственно на ГВЦ при формировании баз данных для простановки признаков и нормативно-справочной информации.

По видам информации классификаторы могут быть разделены на четыре группы: информация о географическом положении, о структуре народного хозяйства и отрасли; информация о природных ресурсах; информация о производственных ресурсах; данные об управленческой информации.

§ 4.5.

Организация баз данных и выбор систем управления базами данных

Широкое использование систем переработки информации в различных сферах управленческой деятельности, для решения сложных экономических задач, проведения научных исследований обуславливают повышенные требования к эффективности организации данных в условиях коллективного пользования.

Традиционный способ организации данных в виде массивов, ориентированных на конкретные задачи, отличается статичностью, жесткой привязкой

к соответствующему программному обеспечению, приводит к дублированию при накоплении и проверке данных, вызывает значительный перерасход памяти на их хранение, частую реорганизацию данных при изменении задач и т.п.

Одним из главных путей преодоления указанных недостатков является создание баз данных, обеспечивающих поддержание в системе динамической информационной модели сложных управляемых объектов и процессов и коллективный доступ к ней.

Организация баз данных. *База данных* (БД) [2.10] определяется как совокупность взаимосвязанных данных, характеризующихся: возможностью использования для большого количества приложений; возможностью быстрого получения и модификации необходимой информации; минимальной избыточностью информации; независимостью прикладных программ; общим управляемым способом поиска.

Возможность использования базы данных для многих прикладных программ пользователя упрощает реализацию комплексных запросов, снижает избыточность и повышает эффективность использования информации в системах обработки данных. Минимальная избыточность и возможность быстрой модификации позволяют поддерживать данные на одинаковом уровне обновления. Независимость данных и использующих их прикладных программ является основным свойством базы данных. Независимость данных подразумевает, что изменение данных не приводит к изменению прикладных программ.

Традиционной формой организации баз данных, обеспечивающей такую независимость, является трехуровневая структура: логическая структура данных прикладного программиста (подсхема); общая логическая структура данных (схема); физическая структура данных.

Схемы и подсхемы базы данных часто изображают в виде диаграмм. На рис. 4.3 приведена общая схема логической структуры базы данных и подсхемы двух прикладных программистов, которые имеют различные представления о данных. Сплошные линии обозначают связи на схеме. Простые связи обозначаются одной стрелкой, связи "один ко многим" – двойной стрелкой. Штриховые линии отображают перекрестные ссылки. Наличие перекрестных ссылок позволяет избежать повторения записей ПОСТАВЩИК и СПЕЦИФИКАЦИИ – ПАРТИИ – ТОВАРА в каждой записи СТАТЬЯ ЗАКУПКИ.

Создание базы данных не является единовременным процессом, оно растягивается на весь период ее существования. Трехуровневая организация обеспечивает возможность быстрого изменения структуры базы данных в условиях ведения и модификации систем управления и наращивания задач пользователей, а также в условиях совершенствования и наращивания аппаратных средств. Трехуровневая организация обеспечивает взаимную независимость изменений общей логической структуры базы данных и прикладных программ (логическая независимость данных) и возможность изменения физического расположения и организации данных без измене-

Общая логическая схема

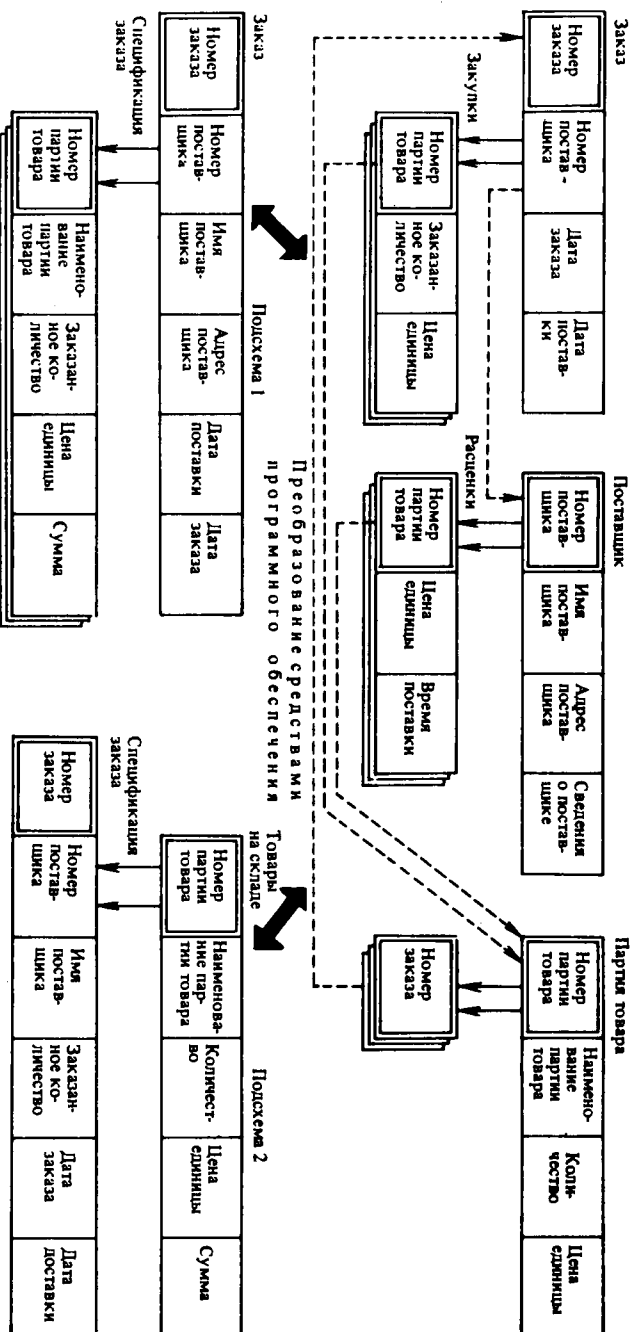


Рис. 4.3

ния общей логической структуры данных и структур данных прикладных программистов (физическая независимость).

Системы управления базами данных. Использование *систем управления базами данных* (СУБД) позволяет исключить из прикладных программ описание данных и детальное программирование управления данными. Описания заменяются ссылками на общую логическую структуру данных, а программирование управления — командами манипулирования данными, которые выполняет универсальное программное обеспечение.

Основной функцией СУБД наряду с обновлением данных является обработка запросов пользователей по поиску и передаче данных прикладным программам. Например, последовательность основных действий, реализуемых СУБД в процессе считывания записи, состоит из следующих операций: прикладная программа выдает запрос СУБД на чтение записи, в котором содержится значение ключа сегмента или записи; СУБД осуществляет в подсхеме прикладной программы поиск описания данных, на которые выдан запрос; СУБД с помощью общей логической схемы данных определяет, какого типа логические данные необходимы; СУБД по описанию физической организации данных определяет, какую физическую запись требуется считать; СУБД выдает операционной системе команду чтения требуемой записи; операционная система считывает данные в системные буферы; СУБД на основании сравнения схемы и подсхемы выделяет информацию, запрошенную прикладной программой; СУБД передает данные из системных буферов в рабочую область прикладной программы.

Действия, которые осуществляет СУБД при обновлении данных, аналогичны операциям считывания. СУБД будет осуществлять необходимые преобразования данных в системных буферах, обратные тем преобразованиям, которые были сделаны при считывании данных. Затем система управления базами данных выдает операционной системе команду ЗАПИСАТЬ.

Общая архитектура системы управления базами данных приведена на рис. 4.4. Она присуща всем СУБД, которые различаются ограничениями и возможностями по выполнению соответствующих функций.

Процесс сравнения и выбора таких систем для конкретного применения сводится к сопоставлению их возможностей с конкретными требованиями и ограничениями пользователя.

Для работы с системой управления базой данных необходимо несколько языков: языки программирования, языки описания схем и подсхем, языки описания физических данных. Прикладной программист использует *языки программирования* для написания программ (КОБОЛ, ФОРТРАН, ПЛ/1, АССЕМБЛЕР) и средства для описания данных — *язык описания подсхем*. Язык описания подсхем может представлять собой средство описания данных в языке программирования, средство, обеспечиваемое СУБД, а также независимый язык описания данных. Многие СУБД используют средства описания данных языков программирования [2.23]. Для прикладного программиста СУБД должна обеспечить средства передачи команд и интерпретации сообщений, выдаваемых системой. Интерфейс между

прикладной программой и СУБД — язык манипулирования данными — встраивается в язык программирования. Запись запрашивается на языке манипулирования данными и считывается в рабочую область прикладной программы; аналогично при включении записи в базу данных прикладная программа помещает ее в рабочую область и выдает команду на языке манипулирования данными. Типичными командами языка манипулирования данными являются: открыть файл или набор записей; закрыть файл или набор записей; определить местонахождение и считать указанный эк-

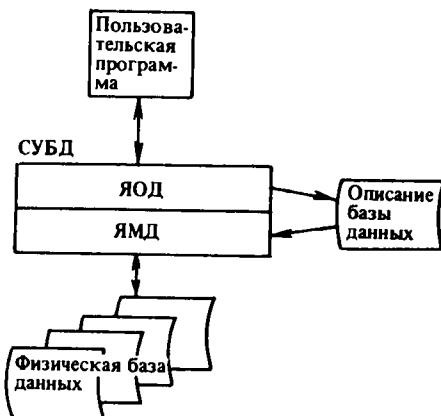


Рис. 4.4

земпляр записи; передать содержимое указанных элементов данных из определенного экземпляра записи; заменить значение определенных элементов указанного экземпляра записи величинами из рабочей области программы; вставить в набор записей запись из рабочей области; удалить определенный экземпляр записи из последовательности записей; запомнить новый экземпляр записи в базе данных; удалить определенный экземпляр записи из базы данных; переупорядочить записи в группе в убывающей или возрастающей последовательности по указанному ключу.

Системный программист использует язык *описания общей логической схемы данных*. Этим языком может быть расширение языка программирования, средство СУБД или независимый язык. Для описания размещения данных системный программист использует некоторую форму языка описания физических данных. Этот язык определяет размещение данных на физических устройствах, управление буферизацией, способы адресации и поиска.

При возникновении в системе управления новых задач, связанных с переработкой больших объемов информации, возникает проблема выбора способа организации данных, обеспечивающих ее решение. При этом необходимо рассмотреть возможные способы организации данных и эффективность их применения для решения поставленной задачи: организация данных в виде файлов; использование существующей базы данных (без изменений общей логической схемы данных); разработка новой логической схемы базы данных с использованием универсальной СУБД; разработка базы данных и специальной СУБД.

Любой необходимый для оценки и выбора способа организации базы данных анализ должен начинаться с тщательного изучения потребностей пользователя. На основе изучения пользовательских требований следует составить полный список подлежащих хранению данных с указанием их характеристик и взаимосвязей, список взаимодействующих с базой процес-

сов и пользователей с указанием их приоритетов и заданием параметров доступа к данным, сформулировать требования к времени и стоимости обращения к базе данных. Кроме того, необходимо проанализировать имеющиеся аппаратные средства (размер основной памяти процессора, конфигурацию операционной системы с учетом средств передачи данных, доступных ресурсов и возможностей расширения внешней памяти), а также численность и квалификацию системных и прикладных программистов.

Если возникает необходимость организации данных для решения ряда прикладных задач при отсутствии функционирующей базы данных, следует оценить целесообразность использования концепции базы данных для решения поставленных задач. Основой для использования базы данных являются следующие факторы: значительное перекрытие массивов по данным, дублирование накопления и проверки данных, необходимость решения проблемы непротиворечивости элементов; значительное число потребителей информации; значительное число типов запросов; относительное постоянство номенклатуры запросов. Вместе с тем при принятии решения об организации базы данных следует помнить, что ее создание может привести к снижению эффективности отдельных прикладных программ и запросов, так как цель организации БД — повышение эффективности всей системы.

При возникновении новых задач в системе управления с функционирующей базой данных следует оценить возможность использования общей структуры данных для описания *предметных областей*, т.е. совокупности объектов и процессов новых пользователей на языке подсхем функционирующей базы данных или, если существующая общая логическая схема нуждается в модификации, оценить затраты на внесение необходимых изменений и влияние этих изменений на общие характеристики системы с точки зрения удовлетворения потребностей всех пользователей системы.

В случае принятия решения о разработке новой базы данных возникают задачи выбора универсальной или разработки специализированной СУБД, а также проектирования ее общей логической и физической структуры, причем выбор СУБД, как правило, является основой для проектирования и организации логической и физической структур базы данных.

Существующие СУБД обеспечивают три основных подхода в управлении данными: иерархический, сетевой и реляционный (рис. 4.5) [2.10, 2.23]. Иерархический подход основан на представлении иерархии объектов. *Иерархические взаимосвязи* непосредственно поддерживаются в физических конструкциях СУБД. Иерархические взаимосвязи являются частным случаем *сетевых взаимосвязей*. Например, поставщик может поставлять несколько видов товаров, а каждый вид товара может иметь несколько поставщиков. *Реляционные системы* не вводят различия между объектами и взаимосвязями. Сетевые и иерархические взаимосвязи могут быть представлены в виде двухмерных таблиц, называемых *отношениями* и обладающих следующими свойствами: каждый элемент таблицы представляет собой один элемент данных (повторяющиеся группы отсутствуют); элементы столбца имеют одинаковую природу, столбцам однозначно присвоены

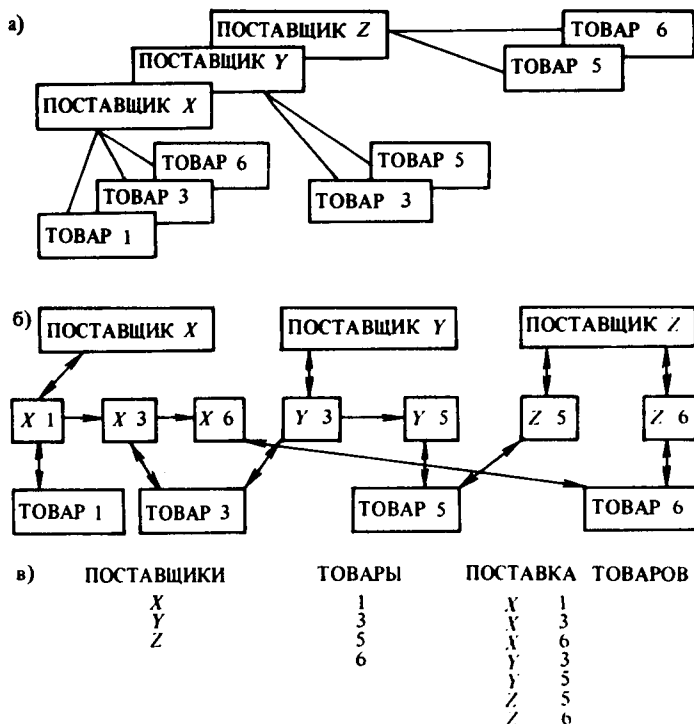


Рис. 4.5

имена; в таблице нет двух одинаковых строк; строки и столбцы могут просматриваться в любом порядке вне зависимости от их информационного содержания. База данных, построенная с помощью отношений, называется *реляционной* и в идеале обладает следующими преимуществами: возможностью использования неподготовленными пользователями; простотой системы защиты (для каждого отношения задается правомерность доступа); независимостью данных; возможностью построения простого языка манипулирования данными с помощью алгебры отношений [2.18].

При выборе универсальной СУБД для реализации конкретной совокупности прикладных программ на основе использования базы данных следует оценить предоставляемый пользователю язык описания данных, язык манипулирования данными и средства поддержания физической базы данных. В качестве характеристик, определяющих язык описания данных, обычно выделяют: наглядность, простоту изучения, степень независимости данных, процедуры защиты от несанкционированного доступа, элементы описания (типы данных, размер и имя и др.), поддерживаемые взаимосвязи (иерархические, сетевые, реляционные). Среди характеристик языков манипулирования данными следует выделить: предоставляемые средства доступа (в физической последовательности, по значению элемента данных, по ключу

чу), совместимость с базовыми (высокоуровневыми) языками программирования, простоту изучения и использования, независимость данных, возможности и средства одновременного использования базы несколькими прикладными программами.

Выбор СУБД. Он осуществляется с учетом потребностей пользователя одним из следующих методов [2.23]: анализа возможностей, экспериментальной проверки, имитации и моделирования.

Метод анализа возможностей основан на балльной оценке приведенных выше характеристик СУБД с точки зрения требований пользователя. Каждая характеристика изучается с двух позиций — присутствует ли она в предлагаемой СУБД и каково ее качество. Качество ранжируется по стандартной шкале. Коэффициент ранжирования умножается на выделенный для данной составляющей вес, и взвешенные по каждой составляющей баллы суммируются.

Метод экспериментальной проверки состоит в создании определенной прикладной среды и получении с ее помощью эксплуатационных характеристик заданной программно-аппаратной системы. Для экспериментальной проверки необходимо спроектировать и загрузить типовую базу данных; затем с использованием языка манипулирования данными СУБД промоделировать требования по обработке существующих и ожидаемых прикладных программ и выполнить экспериментальную проверку рассматриваемых СУБД.

В *методе имитации и моделирования* работы СУБД применяют математические выражения, определяющие зависимость одного из параметров от других. Например, время обращения можно представить в виде функции от числа обращений к диску, количества передаваемой информации и процессорного времени формирования отклика на запрос. Так как перечисленные параметры зависят от способа хранения данных и способа доступа к ним, для различных СУБД требуются разные модели. Если эти модели разработаны, их можно использовать для оценки времени и стоимости обработки при использовании различных СУБД, задавая разнообразные условия (изменяя размеры базы данных, методы доступа, коэффициенты блокирования и т.п.).

Неквалифицированный проектировщик может наложить ограничения конкретной СУБД уже на ранней стадии проектирования базы данных. При этом пользовательские требования искусственно задаются иерархическими и сетевыми структурами определенной СУБД без рассмотрения других возможных проектных решений. Такой подход может привести к уменьшению эффективности системы.

Ниже приведены краткие характеристики некоторых универсальных СУБД.

СУБД ИНЭС ориентирована на решение информационно-поисковых задач главным образом с использованием диалога. В ней обеспечиваются возможности быстрого обращения к базе для получения данных справочного характера и эффективного просмотра больших объемов данных при составлении сводок и при формировании исходных массивов для решения экономических задач.

В системе допускается обращение к данным из прикладных программ пользователя, написанных на языке АССЕМБЛЕР, ПЛ/1, КОБОЛ, АЛГОЛ-60, ФОРТРАН-4.

Используются специальные входные языки (язык ввода экономических показателей, язык ввода документов) и язык запросов, которые удовлетворяют основным требованиям, предъявляемым к языкам описания данных и языкам манипулирования данными.

Для работы с данными, имеющими иерархическую структуру, служит специальный метод доступа. СУБД ИНЭС имеет систему формирования выходных сообщений и систему визуализации сообщений, которые позволяют пользователю задавать структуру документа и его реквизиты, осуществлять поиск, изменение и корректировку данных и вывод их на дисплей.

СУБД КВАНТ-М представляет собой систему реального времени, предназначенную для работы на мини-ЭВМ и используемую для решения задач в информационно-поисковых и справочных системах (фактографических, библиографических, резервирования заказов и т.п.).

Пользовательские программы могут быть написаны на языках КОБОЛ, ФОРТРАН, БЕЙСИК-2 и обращаются к базе данных с помощью САМ-интерфейса.

СУБД КВАНТ-М поддерживает базу данных, состоящую из набора массивов (файлов). Записи массива имеют одинаковую структуру и уникальный последовательный номер (ISN). Записи состоят из полей, которые являются минимальной единицей данных в базе. Поле может быть объявлено ключом. Для описания данных в файлах создается схема, содержащая имена полей записей, их тип и признак, указывающий, является ли поле ключом. Для пользователей создается одна или несколько подсхем, к которым они имеют доступ.

Физическая структура данных использует инвертированные списки значений ключей и обеспечивает независимость адресации от физического расположения данных. Система обеспечивает следующие типы доступа: последовательный по ISN; в логической последовательности; по запросу.

Языком манипулирования данными является язык КВАНТ СКРИПТ-М. Это англоподобный диалоговый язык, предназначенный для эффективного поиска и выделения записей в базе данных и вывода их на дисплей.

В последнее время организован выпуск быстродействующих персональных ЭВМ с большой оперативной и внешней памятью, с возможностью их объединения в локальную вычислительную сеть. Для этих ЭВМ появились и продолжают появляться универсальные СУБД, предоставляющие пользователю средства и удобства иногда более богатые, чем у мини-ЭВМ и больших ЭВМ. Этот процесс еще не установился, в связи с чем не имеет смысла приводить в учебнике характеристики этих СУБД.

Специализированные базы данных. Процесс проектирования специализированной базы данных включает: логическое проектирование, физическое проектирование, разработку специализированной СУБД.

Логическое проектирование предусматривает анализ требований, моделирование данных прикладных программ и их интеграцию, разработку логической схемы. Анализ требований пользователей проводится с применением стандартных методов системного анализа: документирования, обследования, имитированных отчетов. В результате анализа требований получают систематизированные наборы данных и спецификации обработки с использованием стандартных определений данных, минимизирующих противоречия в разработанных спецификациях.

Модель данных предназначена для отображения пользовательской среды. Так как база данных создается для многих пользователей, этап интеграции призван разрешить противоречия между требованиями пользователей в процессе моделирования.

На этапе создания схемы модели пользователей необходимо объединить в одну, из которой можно выделять все альтернативные представления. Задача синтеза оптимальной логической структуры базы данных состоит в определении оптимальной в смысле принятого критерия логической структуры, обеспечивающей реализацию совокупности запросов всех пользователей системы. Среди используемых критериев синтеза логической структуры базы данных можно выделить минимум стоимости хранения, обновления и передачи информации за определенный период, минимум суммарных информационных потоков, минимальный объем хранимой информации, минимум стоимости обновления информации, максимум быстродействия, максимум надежности. Если спецификации требований и модели данных не зависят от СУБД, то схема базы данных представляет собой ее описание на языке описания данных. В дополнение к схеме разрабатываются описания подмножеств базы данных (подсхемы) пользователей. Проектирование физической базы данных включает физическое представление данных, выбор и документирование методов доступа, размещение данных.

Основываясь на логической схеме, проектировщик физической организации должен определить представление каждого элемента данных, записей и массивов. Для каждого физического массива необходимо установить его размер. При определении физического представления данных необходимо учитывать следующие факторы: экономию памяти, минимизацию избыточности (использование ссылок или указателей), способ обработки (последовательная, произвольная) и адресации, коэффициент активности массива и записей (определяет выбор устройств с прямым или последовательным доступом), независимость данных, время отклика на запрос (определяет организацию данных и тип запоминающего устройства).

Специфика задач, решаемых на основе разрабатываемой базы данных, требования пользователя и логическая схема базы данных являются основой для выбора *способа адресации и организации поиска данных*. Среди способов адресации различают: последовательное сканирование, блочный поиск, двоичный поиск, индексно-последовательный способ, прямую адресацию, перемешивание и различные их комбинации.

Последовательное сканирование связано с проверкой ключа каждой записи. Способ может быть эффективен только при пакетной обработке последовательных массивов на магнитной ленте.

Блочный поиск используется в случаях упорядоченных по ключу последовательных массивов. Записи группируются в блоки и каждой блок проверяется, пока не будет найден нужный блок, записи которого считаются.

При *двоичном поиске* поисковая область каждый раз делится пополам, и ключ полученной записи сравнивается с поисковым ключом. Двоичный

поиск обычно не пригоден для устройств с прямым доступом и применяется при поиске индексов массивов.

Индексно-последовательный способ адресации использует индексы. Индекс верхнего уровня указывает местоположение индекса нижнего уровня, который указывает место расположения блока записей. Блок записей либо сканируется, либо в нем проводится двоичный или блочный поиск. Метод предусматривает упорядоченность записей по ключу. В случае произвольного массива применяется индексно-произвольный способ. Однако при этом требуется значительно больший по размерам индекс, так как он должен содержать по одному элементу для каждой записи массива, а не для блока записей.

Прямая адресация предусматривает некоторое преобразование ключа в адрес. Существует много методов преобразования ключа в адрес в массиве. Наиболее простой способ состоит в указании во входном сообщении относительного мониторингового адреса записи. В некоторых приложениях адрес вычисляется на основе идентификаторов объектов.

Способ *перемешивания* состоит в том, что ключ элемента данных преобразуется в квазислучайное число, которое используется для определения места расположения записи.

При проектировании физической структуры базы данных необходимо определить и документировать способ доступа к каждому типу записей, определить записи, доступ к которым проводится непосредственно по ключам, и записи, доступ к которым осуществляется с помощью указателей из других записей или индексов. Каждому массиву в зависимости от метода доступа необходимо выделить место на физических устройствах (магнитных дисках, лентах). При этом данные размещаются таким образом, чтобы обеспечить приоритет часто используемым данным или максимизировать степень близости хранимых данных.

Наиболее широко используемым является индексно-последовательный способ адресации. При этом можно использовать два метода доступа — ISAM и VSAM. В методе доступа ISAM записи группируются так, чтобы они могли располагаться на отдельных дорожках цилиндров модуля дисков, а одна дорожка на каждом цилиндре отводится для индексов, указывающих на записи, расположенные на данном цилиндре. При необходимости внесения новых данных они помещаются в область переполнения (в дорожку индексов включаются также указатели на области переполнения). Метод доступа VSAM аналогичен методу ISAM, однако метод VSAM не зависит от типа оборудования и не оперирует такими категориями, как дорожки и цилиндры. Вместо цилиндров, разделенных на дорожки, используются управляемые области, в свою очередь подразделяемые на управляемые интервалы. В методе VSAM для одной управляемой области имеется один набор указателей (индекс).

Проектирование специализированной СУБД предусматривает разработку языка описания данных, языка манипулирования данными и средства поддержания физической базы данных. Основные требования, которым

должны удовлетворять языки описания данных и манипулирования данными, были определены при рассмотрении вопроса выбора универсальной СУБД. Наиболее распространенным языком описания данных программиста (подсхем) является раздел данных КОБОЛа, для описания схем и физической структуры базы данных в современных СУБД, как правило, разрабатываются свои собственные языки описания данных. Ассоциацией по языкам систем обработки данных (CODASYL) предложен язык описания данных, который используется как для логического описания данных, так и для описания их физической организации [2.14].

Распределенные базы данных. В связи с созданием и развитием в настоящее время ряда АСУ на базе сетей ЭВМ актуальным является проектирование распределенных баз данных (РБД). *Распределенная база данных* представляет собой систему информационно-взаимосвязанных и определенным образом взаимодействующих локальных баз данных (ЛБД), имеющих свое информационное содержание и структуру. По существу РБД представляет собой рассредоточенную систему памяти, хранящей все данные, необходимые соответствующей АСУ. Особенность ее в том, что фрагменты сформированной логической структуры размещаются в территориально удаленных базах данных. Физическая реализация связанности РБД осуществляется организацией информационных потоков внутри ЛБД и между ними по каналам связи.

Основной проблемой при создании РБД является размещение данных; это определяет такие характеристики РБД, как объемы хранимых и обновляемых данных, интенсивность потоков информации и надежность систем.

Проектирование РБД может проходить в условиях: а) создание АСУ только начинается и стоит задача выбора оптимальной структуры РБД и размещения отдельных ЛБД; б) существует определенное количество ЛБД и ВД и стоит задача размещения дополнительного числа ЛБД и оптимального изменения структуры связей в системе; в) формирование географии и структуры системы закончено и стоит задача оптимального перераспределения массивов и изменения топологии связей. Наиболее характерными задачами при проектировании РБД являются определение структуры РБД, определение топологии связей, выбор стратегии поиска и обновления информации, выбор системы управления РБД.

Различают *централизованные, децентрализованные и комбинированные* структуры РБД. Наиболее широкое распространение получили комбинированные РБД, для которых характерно наличие *центральной базы данных*, хранящей общесистемную информацию о размещении массивов в РБД. Число ЛБД на каждом уровне иерархии определяется ограничениями на объемы хранимой информации и ограничениями на стоимость создания ЛБД. Размещение ЛБД зависит от размещения потребителей и источников информации.

Выбор топологии сети ЛБД определяется характером их информационных взаимосвязей, направлением и интенсивностью информационных потоков, необходимой надежностью и достоверностью передачи информации.

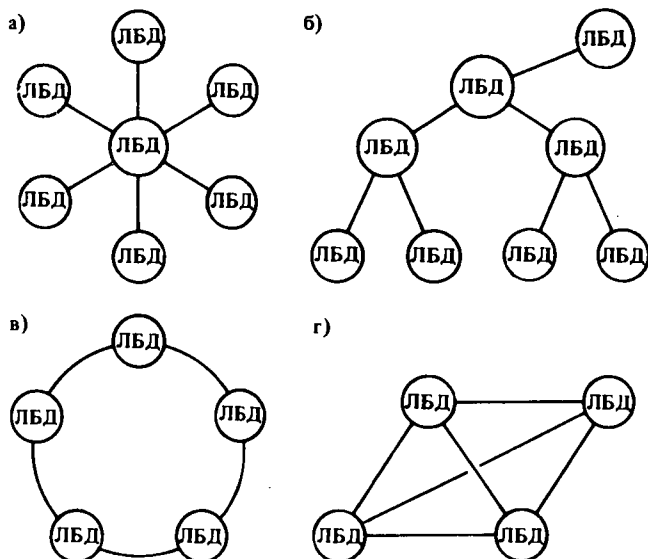


Рис. 4.6

Обычно пользователи прикрепляются к одной ЛБД и через эту ЛБД связываются с остальными базами данных в РБД. Различают следующие типы структур связей ЛБД в РБД: *радиальную, радиально-узловую, кольцевую, каждый с каждым, комбинированную* (рис. 4.6, а – г). Наиболее надежной, с быстрым поиском информации является система со структурой "каждый с каждым". Информационные связи такого типа характерны для объектов, подчиненных друг другу только функционально.

Стратегия поиска влияет на количество и размещение структурной и потоки запросной информации. При отсутствии информации, затребованной пользователем в ближайшей ЛБД, могут быть предложены следующие стратегии поиска: 1) по структурной информации о размещении данных в РБД происходит поиск необходимой ЛБД и обращение к этой ЛБД; 2) осуществляется поиск в ЛБД более высокого ранга; если необходимая информация отсутствует, анализируется структурная информация о содержании всех подчиненных ЛБД; если необходимая информация отсутствует, переходят к ЛБД более высокого уровня иерархии; 3) осуществляется обращение в управляющую ЛБД, где хранится структурная информация о всех ЛБД; 4) осуществляется опрос всех ЛБД либо параллельно, либо последовательно.

Стратегия 1 обеспечивает минимальные объемы запросной информации, но в каждой ЛБД необходимо хранить структурную информацию о размещении массивов в РБД.

Стратегия 2 характерна для иерархических систем, в которых преобладают информационные потоки сверху вниз.

Стратегия 3 минимизирует структурную информацию.

Стратегия 4 отличается большими потоками запросной информации.

Функционирование РБД предполагает наличие в ней потоков обновления информации. Среди стратегий обновления можно выделить следующие: обновление всех дублируемых массивов по всем ЛБД выполняет источник информации; источник обновляет информацию только в ближайшей ЛБД, все остальные дублируемые массивы обновляются по инициативе этой ЛБД; обновление дублируемых массивов проводится по алгоритму (например, минимизирующему суммарные потоки обновления). Стратегия обновления должна обеспечивать заданную надежность, достоверность и быстродействие РБД. Разработка и внедрение эффективных систем управления РБД находятся в настоящее время на начальном этапе. Основным критерий при разработке системы управления РБД — минимальная трудоемкость создания и внедрения ее математического обеспечения. Задача может решаться доработкой и подстройкой существующих СУБД или созданием эффективных специальных систем управления РБД.

Системы управления РБД, создаваемые на базе СУБД, дополнительно должны содержать входной анализатор, обработчик внешних изменений, обработчик внешних запросов, выходной анализатор, транспортную станцию [2.26]. Входной анализатор осуществляет прием сообщений от транспортной станции и их анализ. Транспортная станция представляет собой комплекс программ, обеспечивающий информационный обмен сообщениями между ЛБД в РБД. Обработчик внешних изменений преобразует поступающие от ЛБД сообщения об изменениях в форму запросов, обрабатываемых СУБД, функционирующей в данной ЛБД. Обработчик внешних запросов преобразует внешние запросы на формирование выходных массивов с целью обработки их соответствующей СУБД и последующей передачи адресату. Выходной анализатор на основе анализа информации об изменениях в ЛБД определяет адресатов, преобразовывает информацию к виду, принятому для обмена между ЛБД, и передает эту информацию транспортной станции.

Применение методов построения систем управления РБД с использованием универсальных СУБД позволяет сократить трудоемкость создания РБД и сохранить разработанное математическое обеспечение прикладных программ.

§ 4.6. Достоверность информационного обеспечения

Одним из основных условий эффективного функционирования АСУ является обеспечение требуемого уровня достоверности информации, на основе которой принимаются управленческие решения.

Под *достоверностью информации* в АСУ понимают некоторую функцию вероятности ошибки, т.е. события, заключающегося в том, что реальная информация в системе о некотором параметре не совпадает в пределах заданной точности с истинным значением.

Необходимая достоверность достигается использованием различных методов, реализация которых требует введения в *системы обработки дан-*

ных (СОД) информационной, временной или структурной избыточности. Достоверность при обработке данных достигается путем контроля и выявления ошибок в исходных и выводимых данных, их локализации и исправления. Условие повышения достоверности — снижение доли ошибок до допустимого уровня. В конкретных АСУ требуемая достоверность должна устанавливаться с учетом нежелательных последствий, к которым может привести возникшая ошибка, и тех затрат, которые необходимы для ее предотвращения.

Важным этапом выбора и разработки методов и механизмов обеспечения достоверности информации является *анализ* процессов ее переработки. В ходе анализа изучают структуру обработки данных, строят модели возникновения ошибок и их взаимодействия, рассчитывают вероятности возникновения, обнаружения и исправления ошибок для различных вариантов структур обработки данных и использования механизмов обеспечения требуемого уровня достоверности.

Синтез механизмов обеспечения достоверности осуществляют при техническом и рабочем проектировании АСУ. На этапе синтеза создают технологическую структуру обработки информации, включая выбор методов и механизмов обеспечения достоверности. В качестве критерия используют один из показателей эффективности: максимизация достоверности информации, минимизация затрат на разработку и эксплуатацию систем повышения достоверности и некоторые другие.

Ошибки возникают на любом этапе функционирования АСУ. Их можно разделить на ошибки персонала АСУ и ошибки, вызванные неисправностью технических средств системы.

Ошибки персонала АСУ обусловлены психофизиологическими возможностями человека, объективными причинами (несовершенством моделей представления информации, недостаточной квалификацией персонала, несовершенством технических средств и т.п.) и субъективными причинами (небрежностью, безответственностью некоторых пользователей, преднамеренным искажением информации, плохой организацией труда и т.п.).

Ошибки, вызванные неисправностью технических средств системы, — это ошибки, связанные с неисправностью оборудования, несоответствием его техническим нормам, нарушением необходимых условий работы технических средств и хранения машинных носителей информации, с физическим износом элементов и узлов технических средств, различного рода помехами и т.д.

Иногда выделяют ошибки во входных данных, т.е. ошибки, которые возникают во внешней среде, вне рассматриваемой системы и поступают в нее в составе исходных данных.

Методы повышения достоверности обработки данных. Рассмотрим особенности процессов обработки данных в АСУ и взаимосвязь с этими процессами механизмов возникновения ошибок, контроля обрабатываемых данных и обнаружения ошибок.

При решении задач АСУ осуществляется преобразование $x \rightarrow y$, где

$x = (x_1, x_2, \dots, x_I)$ – множество входных переменных, $y = (y_1, y_2, \dots, y_J)$ – множество выходных переменных. Преобразование π ставит в соответствие каждой n -й комбинации значений входных переменных $\bar{x}_n = (x_1, x_2, \dots, x_i, \dots, x_I)_n$ определенную совокупность значений выходных переменных $\bar{y}_n = (y_1, y_2, \dots, y_j, \dots, y_J)_n$. Таким образом, $y_n = \pi(\bar{x}_n)$ и $\pi: D_1 \times D_2 \times \dots \times D_I \rightarrow F_1 \times F_2 \times \dots \times F_J$, где D_i – область определения входной переменной x_i , $i = 1, I$, а F_j – область определения выходной переменной y_j , $j = 1, J$.

Пусть z – множество промежуточных переменных. Тогда функционирование АСУ представляется в виде ориентированного графа технологии обработки данных, множество вершин V которого соответствует множеству входных, промежуточных и выходных переменных, т.е. $V = X \cup Y \cup Z$, а множество дуг E соответствует множеству процедур обработки этих переменных. Переменные и процедуры будем называть элементами графа $I(\pi)$.

Процедуры обработки реализуются персоналом и техническими средствами, надежность которых не является абсолютной. Пусть R – множество процессов, которые привели к появлению недостоверных исходных данных или вносящих помехи в нормальное функционирование АСУ и приводящих тем самым к появлению недостоверных данных. В этом случае АСУ осуществляет преобразование $(\pi, R): D_1 \times D_2 \times \dots \times D_I \rightarrow F'_1 \times F'_2 \times \dots \times F'_J$, где F'_j – область возможных значений переменной y_j . Очевидно, что F'_j может содержать запрещенные (недостоверные) значения переменной y_j . Для задания множества запрещенных значений $\hat{F} = \hat{F}_1 \times \hat{F}_2 \times \dots \times \hat{F}_J$ и выявления факта принадлежности конкретных значений выходных переменных к этому множеству используют механизмы контроля. При решении задач АСУ с механизмом контроля M^k осуществляется преобразование $M = (\pi, R, M^k): D_1 \times D_2 \times \dots \times D_I \rightarrow (F_1 \cup \hat{F}_1) \times (F_2 \cup \hat{F}_2) \times \dots \times (F_J \cup \hat{F}_J)$, причем для каждого $\bar{x}_n$ устанавливается: $M(\bar{x}_n) \in F$ или $M(\bar{x}_n) \in \hat{F}$. Механизмы контроля используют на различных этапах процесса обработки информации с целью обеспечения ее достоверности.

Методы контроля при обработке информации в АСУ классифицируют по различным параметрам: по количеству основных операций, охватываемых контролем – единичный (одна операция), групповой (группа последовательных операций), комплексный (контролируется, например, процесс сбора данных); по частоте контроля – непрерывный, циклический, периодический, разовый, выборочный, по отклонениям; по времени контроля – до выполнения основных операций, одновременно с ними, в промежутках между основными операциями, после них; по виду оборудования контроля – встроенный, контроль с помощью дополнительных технических средств, безаппаратурный; по уровню автоматизации – "ручной", автоматизированный, автоматический.

Различают системные (организационные), программные и аппаратные методы контроля достоверности [1.4].

Системные методы включают: оптимизацию структуры обработки; поддержание характеристик оборудования в заданных пределах; повыше-

ние культуры обработки; обучение и стимулирование персонала; создание оптимального числа копий и (или) предыстории программ, исходных и текущих данных; определение оптимальной величины пакетов данных и скорости первичной обработки, процедур доступа к массивам и др.

Программные методы повышения достоверности обработки информации состоят в том, что при составлении процедур обработки данных в них предусматривают дополнительные операции, имеющие математическую или логическую связь с алгоритмом обработки данных. Сравнение результатов этих дополнительных операций с результатами обработки данных дает возможность установить с определенной вероятностью наличие или отсутствие ошибок. На основании этого сравнения, как правило, появляется возможность исправить обнаруженную ошибку.

Можно выделить следующие программные методы контроля: счетные — двойной или обратный счет, использование контрольных сумм, контроль по формату, балансовые методы и др.; математические — способ подстановки, проверка с помощью дополнительных связей, проверка предельных значений, статистический прогноз; использующие избыточность информации — контрольных цифр, сравнение данных из различных источников, сравнение с внешними данными, контроль с использованием принципа обратной связи; логические — смысловые проверки, контроль по отклонениям, контроль по заданной последовательности записей, метод шаблонов, контроль за временем решения задач ЭВМ, экспертная оценка получаемых данных; прочие — контрольных испытаний, промежуточных точек и новых начал, комбинированные.

Аппаратные методы контроля и обнаружения ошибок могут выполнять практически те же функции, что и программные. Аппаратными методами обнаруживают ошибки ближе к месту их возникновения и недоступные для программных методов (например, перемежающиеся ошибки).

Следует отметить, что методы контроля достоверности позволяют выявлять ошибки, возникающие и из-за неправильной работы программ. Однако локализация и исправление ошибок в программах требует разработки и использования специальных методов тестирования и отладки, рассмотрение которых выходит за рамки настоящей книги.

Все перечисленные методы контроля обработки данных базируются на использовании определенной *избыточности*. Можно выделить структурную, временную и информационную избыточность, иногда выделяют также прагматическую избыточность. Соответственно различают методы контроля со структурной, временной и информационной избыточностью.

Структурная избыточность требует введения в состав АСУ дополнительных элементов (резервирование информационных массивов и программных модулей, реализация одних и тех же функций различными процедурами, схемный контроль в технических средствах АСУ и т.д.).

Временная избыточность связана с возможностью неоднократного повторения определенного контролируемого этапа обработки данных. Обычно этап обработки повторяют неоднократно и результаты обработки

сравнивают между собой с целью контроля достоверности, или после обработки контроль достоверности осуществляют каким-либо методом, и в случае обнаружения ошибки производят исправления и повторную обработку.

Информационная избыточность может быть естественной и искусственной. *Естественная* информационная избыточность отражает объективно существующие связи между элементами СОД, наличие которых позволяет судить о достоверности информации. *Искусственная* информационная избыточность характеризуется введением дополнительных информационных разрядов для используемых данных и дополнительных операций в процедуры обработки данных, имеющих математическую или логическую связь с алгоритмом обработки данных. На основании анализа результатов дополнительных операций и процедур обработки данных, а также дополнительных информационных разрядов выявляется наличие или отсутствие ошибок определенного типа, а также возможности их исправления.

Прагматическая избыточность — наименее исследованный тип избыточности. Под величиной прагматической избыточности в сообщении относительно некоторого получателя понимают процент элементов сообщения (символов, разрядов, слов, предложений и т.п.), исключение которых не изменит отклика получателя на это сообщение. Примером прагматической избыточности является повторение материала при обучении, неоднократное повторение, разъяснение приказов и распоряжений и т.д.

Для решения задач анализа систем повышения достоверности обработки и защиты данных в АСУ используют понятия графа технологии обработки данных, индикаторного графа и графа ошибок.

Особенности различных структур обработки данных отражаются на графе технологии $I(\pi) = G(V, E)$, введенном выше. Следует отметить, что основными структурными особенностями $I(\pi)$ является наличие обратных связей и модульность программного и информационного обеспечения (характеристика вершин и дуг графа $I(\pi)$).

Для упрощения анализа систем контроля и выбора соответствующих методов контроля вводится понятие *индикаторного графа* $J(\pi)$. Вершинами данного графа являются индикаторы состояний $j(w_i)$ соответствующих информационных элементов (вершин графа технологий $I(\pi)$). При этом индикатор $j(w_i) = 1$, если в информационном элементе содержится ошибка, и $j(w_i) = 0$ в противном случае. Дуги индикаторного графа, соответствующие процедурам, отображают взаимодействие и распространение ошибок.

Ошибку в выходной переменной определим как ошибку результата. Она появляется вследствие возникновения ошибок в одном или нескольких промежуточных элементах данных. Таким образом, ошибки результата связаны с событиями ошибок в вершинах индикаторного графа логическими функциями *И* и *ИЛИ*, аргументами которых будут индикаторы промежуточных событий ошибки.

Одна из основных задач анализа системы контроля состоит в определении вероятностей ошибок выходных переменных. При этом необходимо

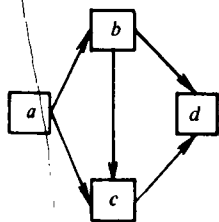


Рис. 4.7

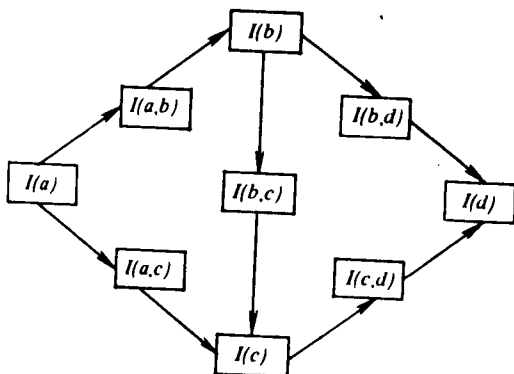


Рис. 4.8

установить зависимости между этими вероятностями и состояниями вершин индикаторного графа (т.е. вероятностями появления ошибок в промежуточных данных). Простейшим видом указанной зависимости является, например, зависимость типа "последовательное соединение", когда появление хотя бы одной ошибки в промежуточных данных (частная ошибка) приводит к появлению ошибки в выходных данных (ошибка результата).

Графическим представлением такого взаимодействия является *граф ошибок* $\epsilon(\pi)$, вершинами которого являются индикаторы входных и выходных переменных указанных выше логических функций, а дуги графа отображают причинно-следственные взаимодействия между индикаторами ошибок — частных, промежуточных и результата. Отметим, что вершина ошибки результата не содержит выходящих дуг.

Рассмотрим информационный граф технологии $I(\pi)$ и соответствующий ему индикаторный граф $J(\pi)$ (рис. 4.7 и 4.8). Пусть ошибка результата состоит в том, что нет связующего пути между вершинами a и d графа $I(\pi)$ (определены соответствующие комбинации частных ошибок индикаторного графа). Соответствующий граф ошибок $\epsilon(\pi)$ представлен на рис. 4.9. На основании графа $\epsilon(\pi)$ определяется возможность ошибки результата в зависимости от наступления частных ошибок.

Определение ошибок результата, их ранжирование и анализ причинно-следственных связей графа $\epsilon(\pi)$ дают возможность найти частные ошибки, вызывающие появление ошибок результата, а также анализа и последующего выбора соответствующих методов контроля. Для формальной постановки задач анализа и синтеза систем повышения достоверности введены понятия минимального разреза и минимального пути графа.

Разрезом назовем множество частных ошибок, совместное появление которых влечет появление ошибки результата. Разрез является *минимальным*, если никакое его собственное подмножество не является разрезом. Если произошла ошибка результата, то произошли частные ошибки, образующие *минимальный разрез*.

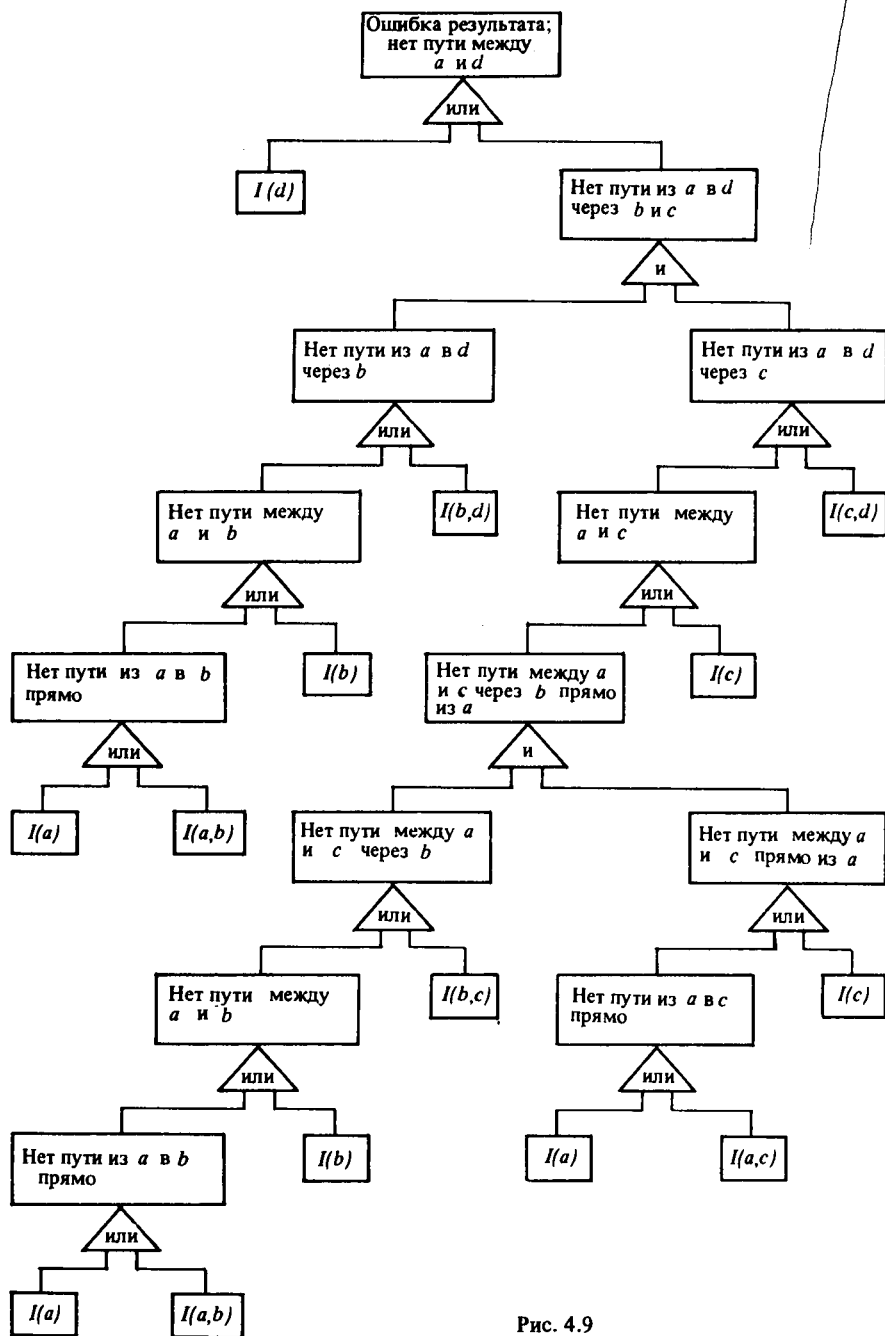


Рис. 4.9

Путь есть множество частных ошибок, совместное неоявление которых влечет неоявление ошибки результата. Путь является *минимальным*, если никакое его собственное подмножество не является путем. Если ошибка результата не появилась, то не появились совместно и все частные ошибки, образующие *минимальный путь*.

Пусть $k = \{k_1, \dots, k_l, \dots, k_k\}$ — множество минимальных разрезов, $p = \{p_1, \dots, p_r, \dots, p_s\}$ — множество минимальных путей, $I(y)$ — индикатор главного события, $I(w_i)$ — индикатор базисного события w_i графа $\epsilon(\pi)$. Если граф ошибок $\epsilon(\pi)$ является деревом, то справедливы следующие отношения:

$$I(y) = 1 - \prod_{e=1}^k [1 - \prod_{w_i \in k_e} I(w_i)],$$

$$I(y) = \prod_{r=1}^s \{1 - \prod_{w_i \in p_r} [1 - I(w_i)]\},$$

которые определяют зависимости главного события ошибки от базисных событий через множества минимальных разрезов и путей.

Под *уровнем ошибки результата* $L(y)$ понимают значение вероятности ее появления. Уровень ошибки является основной характеристикой используемых механизмов систем контроля.

Каждому механизму контроля обработки данных ставится в соответствие вектор системных характеристик, к которым относят вероятности обнаружения и исправления ошибок, стоимость разработки и внедрения методов контроля ошибок, эксплуатационные затраты, время обработки и внедрения, необходимую память, машинное время и т.д. Назовем механизм контроля *эффективным*, если $L(y) \leq \epsilon$, где ϵ — допустимый уровень ошибки результата.

Общая задача анализа системы контроля состоит в определении уровня ошибки результата в зависимости от вероятностей $p(w_i)$ частных ошибок и сравнении его с допустимым.

Для произвольных древовидных структур графа $\epsilon(\pi)$ в общем случае точное определение значения $L(y)$ затруднительно. При независимых базисных событиях ошибки можно определить точную нижнюю и верхнюю границы уровня ошибок $L(y)$ [2.16]:

$$\prod_{r=1}^s \{1 - \prod_{w_i \in p_r} [1 - p(w_i)]\} \leq L(y) \leq 1 - \prod_{w_i \in k_l} p(w_i),$$

где $p(w_i)$ — вероятность появления ошибки в элементе w_i . Для повышения эффективности контроля необходимо уменьшать вероятности появления частных ошибок, что достигается использованием более совершенных методов контроля, введением различных вариантов избыточности в структуру АСУ.

Для случая, когда любая частная ошибка ведет к появлению ошибки результата, получены аналитические выражения для нахождения уровня ошибки $L(y)$ и других вероятностных характеристик [2.16].

При решении практических задач построение графа ошибок для всей системы обработки данных может оказаться затруднительным из-за значительной размерности. В этом случае графы ошибок строят для отдельных задач (подзадач) и с их использованием проводят анализ взаимодействия ошибок.

Рассмотрим практический пример разработки графа ошибок при разработке подсистемы "Выпуск справок", входящей в состав АСУ "Обмен". Требуется при заданном информационном и программном обеспечении выбрать методы контроля информационных элементов и процедур обработки данных, которые обеспечивают минимальное приращение вероятности ошибки результата (наличие ошибок в выходных документах подсистемы). На рис. 4.10 изображена модульная блок-схема подсистемы "Выпуск справок". Процедуры обработки данных, информационные элементы и содержание массивов приведены соответственно в табл. 4.1–4.3.

На основании анализа различных вариантов возникновения ошибок, ведущих к появлению неверных выходных данных, построен граф ошибок (рис. 4.11). Описание событий, используемых в графе ошибок, приведено в табл. 4.4 и 4.5.

Одним из основных методов повышения достоверности при обработке информации является применение методов контроля с использованием

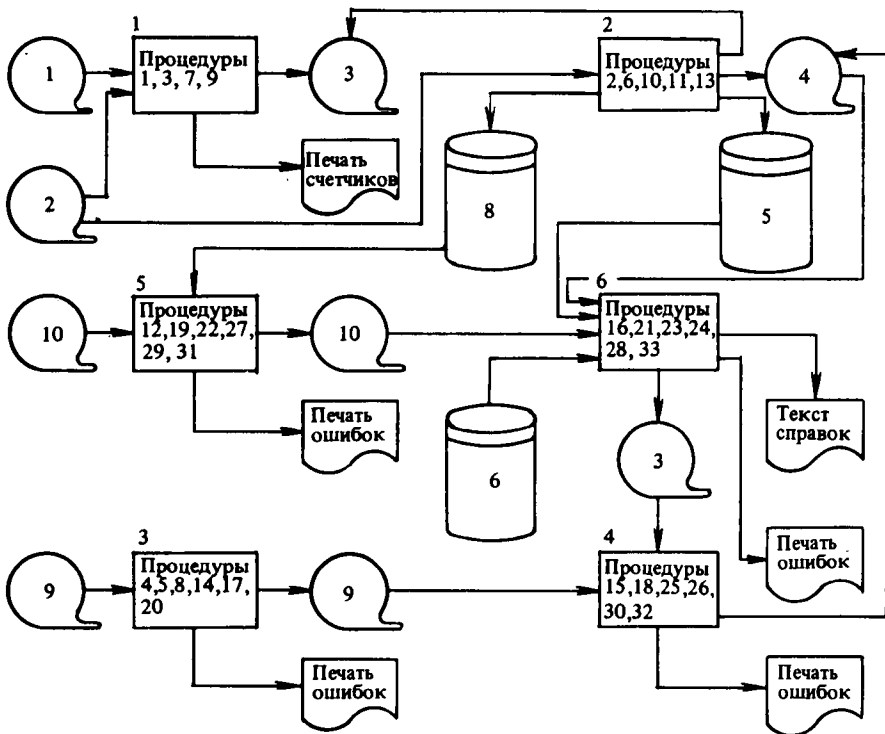


Рис. 4.10

принципа обратной связи. При этом данные, содержащие обнаруженные ошибки, направляются на исправление, после чего вновь повторяется процедура обработки. Методы контроля, использующие принцип обратной связи, можно разделить на два класса. К первому относятся методы, ориентированные на решение задач, в которых длины контролируемых участков могут быть произвольными величинами и являются, как правило, управляемыми параметрами. Эти методы используются в задачах обеспечения надежности информационных баз, определения оптимального интервала между контрольными точками в программе и др. Ко второму классу относят методы, используемые при решении задач, в которых длины контролируемых участков известны, так же как и структура обратных связей между участками, охваченными контролем.

Каждый контролируемый участок характеризуется вектором характеристик – объемом обрабатываемых данных, требуемой вероятностью на своевременность обработки этого объема данных, законами возникновения ошибок (независимые ошибки, Марковская модель возникновения ошибок и др.), имеющимися временными или стоимостными ресурсами, требованиями к достоверности обработки информации, набором методов контроля, использование которых возможно на этом участке.

Таблица 4.1

Код процедуры	Процедура
1	Кодирование района
2	Формирование группы прямого обмена
3	Исключение служебной записи
4	Ввод и контроль изменений
5	Сортировка изменений
6	Формирование предложений
.....	
31	Накопление индивидуальных запросов
32	Печать ошибок предложений
33	Печать текста справок

Таблица 4.2

Код элемента	Элементы
1	Номер клиента
2	Название района
3	Номер зоны поиска спроса
.....	
27	Текст справок
28	Признак ошибки
29	Дата ввода

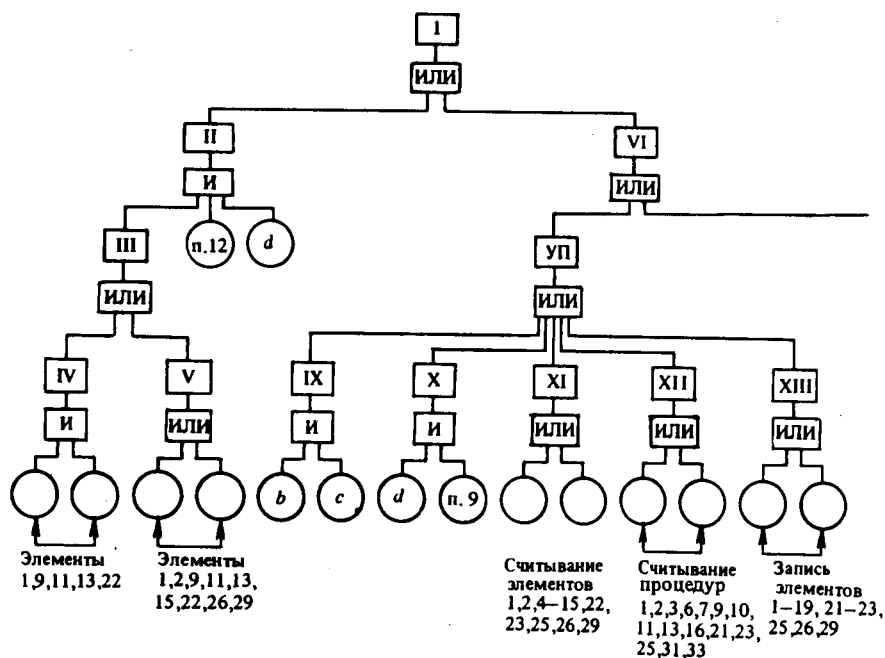
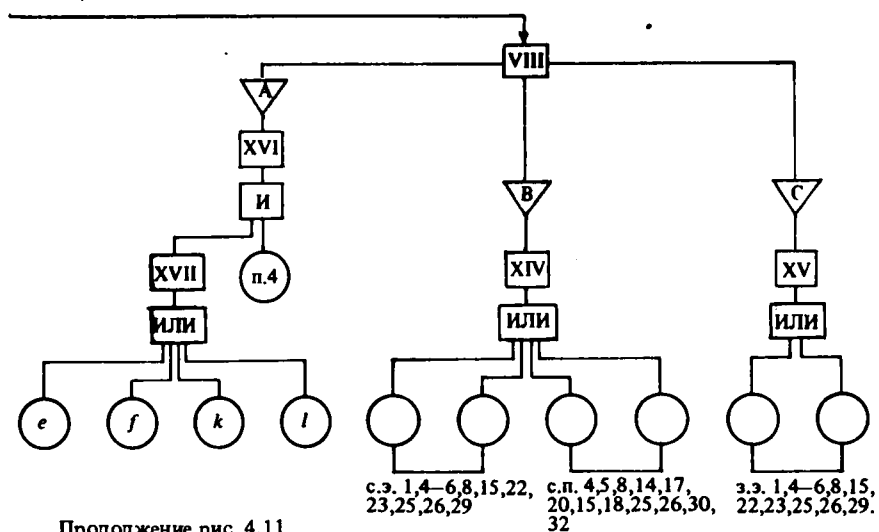


Рис. 4.11

Таблица 4.3

Номер массива	Массив	Содержание массива	Тип внешнего запоминающего устройства
1	ежедневных поступлений	1,2,5-15,22,23,25,26,29	МЛ
2	ежедневных поступлений с закодированным районом	1,4,5-15,22,23,25,26,29	МЛ
3	картотеки	1,4-6,8,15,22,23,25,26,29	МЛ
4	предложений	1,4,8,10,12,14,16,18,21,23,25,29	МЛ
5	спросов	1,3,9,11,13,15,17,22,26,29	МД
6	соответствия запросов и предложений	1,17,18	МД
7	текста справок	1,27,28	АЦПУ
8	постоянных данных	30	МД



Продолжение рис. 4.11

Продолжение табл. 4.3

Номер массива	Массив	Содержание массива	Тип внешнего запоминающего устройства
9	изменений	1,2,5,6,19,31	МЛ
10	индивидуальных запросов	1,19,20,22,24	МЛ

Т а б л и ц а 4.4

Код события	Описание промежуточных событий
I	Клиент получает недостаточную информацию (ошибка результата)
II	Ошибки при формировании запроса
III	Ошибки при формировании запроса допущены клиентом
IV	Ошибки при формировании запроса допущены клиентом, числящимся в картотеке
V	Ошибки при формировании запроса допущены клиентом, который не числится в картотеке
VI	Ошибки при формировании вариантов обмена

Код события	Описание промежуточных событий
VII	Ошибки при обработке новых поступлений, относящихся к группе обмена данного запроса
VIII	Ошибки при внесении изменений, относящихся к группе обмена данного запроса
IX	Ошибки клиента при формировании поступлений, проходящие в систему
X	Ошибки инспектора при формировании поступлений, проходящие в систему
XI, XII	Ошибки при считывании информационных элементов и процедур
XIII	Ошибки при записи информационных элементов
XIV	Ошибки при считывании информационных элементов и процедур на этапе обработки изменений
XV	Ошибки записи информационных элементов на этапе обработки изменений
XVI	Ошибки при внесении изменений, проходящие в систему на этапе ввода

Т а б л и ц а 4.5

Обозначение ошибки	Описание частных ошибок
<i>a</i>	Инспектор не обнаружил ошибок клиента, допущенных при формировании запроса
<i>b</i>	Ошибки в объявлении, допущенные клиентом
<i>c</i>	Инспектор не обнаружил ошибок клиента в объявлении
<i>d</i>	Ошибки в объявлении, допущенные инспектором
<i>e</i>	Ошибки в макете совершения обмена
<i>f</i>	Ошибки в макете продления срока учета
<i>k</i>	Ошибки в макете бронирования и снятия бронирования
<i>l</i>	Ошибки в макете изменения предложения
<i>N.n</i>	Ошибки в процедуре
с.э.	Ошибки считывания информационного элемента
з.э.	Ошибки записи информационного элемента
с.п.	Ошибки, возникающие при считывании процедуры

Для анализа систем контроля данного класса используется понятие "стандартная" схема обработки данных (рис. 4.12). Цикл обработки данных распадается на непосредственно обработку, контроль и исправление ошибочных данных. На некоторых этапах обработки операции контроля и исправления недостоверных данных могут отсутствовать или могут осуществляться для группы из нескольких этапов, на каждом из которых в свою очередь осуществляется локальный контроль и исправление ошибок. После исправления ошибочных данных они вновь обрабатываются с последующим контролем и исправлением ошибок. Контроль и исправление могут повторяться случайное число раз.

Производными "стандартной" схемы обработки данных являются последовательная схема обработки, последовательная схема с общей обратной связью, циклическая и последовательно-циклическая схема, сеть обработки данных.



Рис. 4.12

Стандартная схема обработки данных. Процесс функционирования стандартной схемы обработки представлен на рис. 4.13. Процесс обработки единицы входных данных является *процессом Бернулли*, в котором q — есть вероятность ошибки при обработке единичного объема данных, а вероятность правильной обработки единичного объема есть $p = 1 - q$. Процесс контроля также является *бернуллиевским*, где f — есть вероятность обнаружения ошибки в единичном объеме данных и $l = 1 - f$ — есть вероятность пропуска ошибки. Предполагается, что вероятность принятия правильно обработанного единичного объема данных за ошибку равна нулю; обнаруженные ошибки исправляются с вероятностью единицы.

Под *единицей объема данных* в зависимости от задачи понимается символ, запись, сообщение, массив, документ и т.д., N — общий объем данных. Число циклов обработки единицы данных является случайным и попытка обработки считается *успешной*, если в фазе обработки не произошло ошибки или же ошибка произошла, но не была обнаружена в фазе контроля. Попытка считается *неуспешной*, если в фазе обработки произошла ошибка, которая была обнаружена.

Обозначим через ξ_k число попыток, затрачиваемых в фазе обработки на n -ю единицу данных, $n = 1, N$. Тогда число попыток $T(N)$, затрачиваемых в фазе обработки на N единиц данных, равно сумме случайных величин

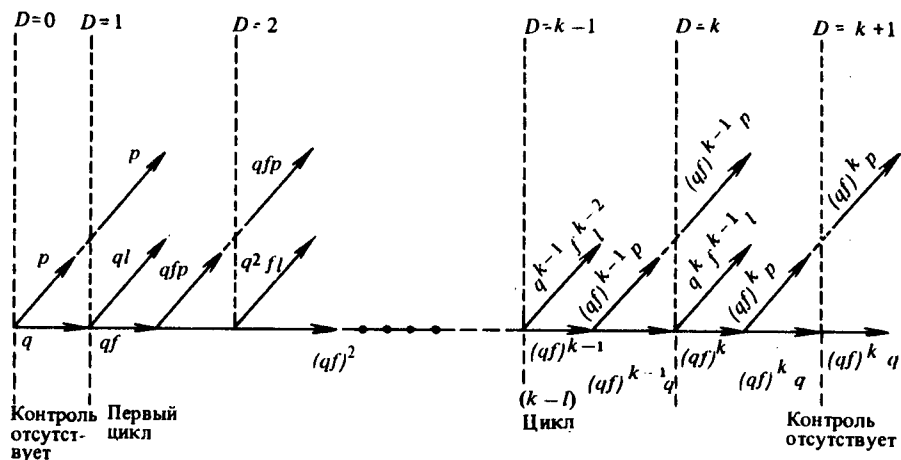


Рис. 4.13

$\xi_k, k = \overline{1, N}$. Если на реализацию попытки в фазе обработки требуется единичное время, то $T(N)$ — есть время, затраченное на обработку N единиц данных.

Основной задачей анализа рассматриваемой схемы является нахождение закона распределения $\Phi(N, x) = p \{ T(N) \leq x \}$ случайной величины $T(N)$. Задачи, связанные с определением вероятностных характеристик времени, которое тратится на обработку данных объема N в фазах обнаружения и исправления, решаются аналогично.

Рассмотрим вначале случай, когда ошибки при обработке возникают независимо. Процесс обработки единицы данных может завершиться успехом на $(k + 1)$ -й попытке, если данные не содержат ошибок с вероятностью $(qf)^k p$ либо содержат необнаруженные ошибки с вероятностью $(qf)^k q(1 - f)$. В противном случае процесс обработки данные единичного объема не заканчивается и они должны быть обработаны, по крайней мере, еще один раз. Вероятность этого события равна $(qf)^{k+1}$. В этом случае величины ξ_k имеют геометрическое распределение

$$p \{ \xi_k = i \} = (qf)^{i-1} (p + q^e),$$

где $i \geq 1$. Вероятность того, что за время i данные единичного объема будут обработаны без ошибок, определяется как

$$p_1(i) = \sum_{j=1}^i p(qf)^j.$$

Представив $p_1(i)$ в виде $p_1(i) = p \rho_1 \rho_2$, где $\rho_1 = (1 - qf)^{-1}$ и $\rho_2 = 1 - (qf)^i$, получим, что при $i \rightarrow \infty$ вероятность безошибочной обработки данных единичного объема равна $p \rho_1$ и $\lim_{i \rightarrow \infty} p_1(i) = 1$ только при $f = 1$.

На рис. 4.14 показаны непрерывные аналоги $p_1(i)$ для различных значений p, f, i . Вероятность того, что за время $(i - 1)$ произошла обработка $(N - 1)$ -й единицы данных и было $(i - N)$ неудачных попыток обработки, равна $C_{i-1}^{N-1} (p + ql)^{N-1} (qf)^{i-N}$. Вероятность удачной попытки равна $p + ql$, поэтому закон распределения времени обработки данных объемом N есть

$$p \{ T(N) = i \} = \begin{cases} 0 & \text{при } i < N, \\ C_{i-1}^{N-1} (p + ql)^N (qf)^{i-N}, & i \geq N. \end{cases}$$

Математическое ожидание и дисперсия случайной величины $T(N)$, распределенной по отрицательному биномиальному закону, определяется соответственно в виде

$$M[T(N)] = N(1 - qf)^{-1},$$

$$D[T(N)] = N_{qf} (1 - qf)^{-2}.$$

В табл. 4.6 приведены вероятности отсутствия ошибок, $M[T(N)]$ и $D[T(N)]$ при $N = 1; 10$ и $i = 11; 110$. При $q \rightarrow 0$ $M[T(N)]$ уменьшается и в

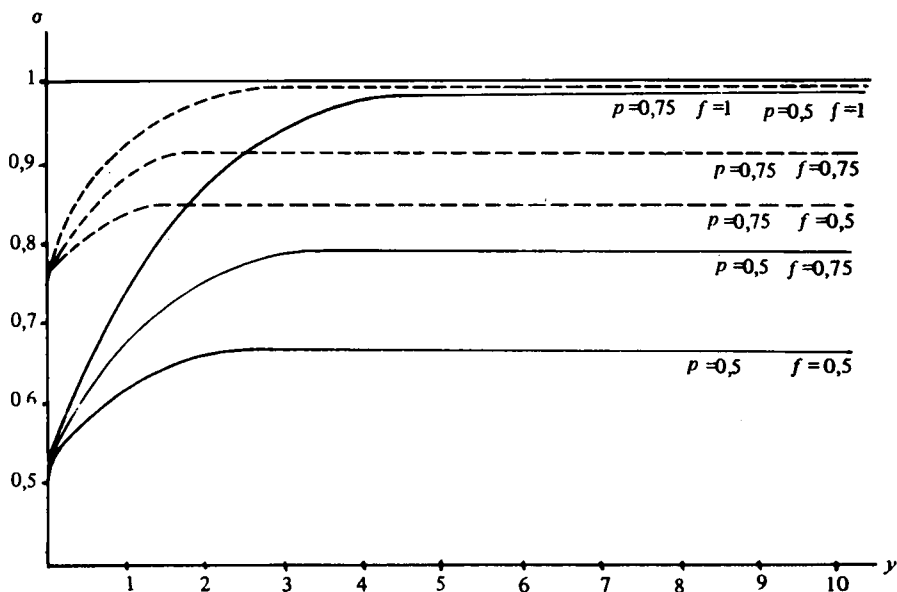


Рис. 4.14

пределе равно N . Использование хороших методов контроля ($f \rightarrow 1$) увеличивает $M[T(N)]$. Если N велико, то на основании центральной предельной теоремы величина $T(N)$, как сумма независимых одинаково распределенных случайных величин ξ_k , распределена по нормальному закону:

$$\Phi(N, x) = p \{ T(N) \leq x \} \approx \Phi^* \left(\frac{(1 - qf)x - N}{\sqrt{Nqf}} \right),$$

где $\Phi^*(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{t^2}{2}} dt$.

Таблица 4.6

p	f	$p_1(10)$	$M_1[T(N)]$	$D_1[I(N)]$	$P_{100}(10)$	$E_{100}[T(N)]$	$D_{100}[T(N)]$
0,5	0,5	0,66	0,33	0,44	$4,93 \cdot 10^{-23}$	33,33	44,44
0,5	0,75	0,79	0,6	0,96	$2,42 \cdot 10^{-21}$	60	96
0,5	1	0,99	1	2	$4,0 \cdot 10^{-20}$	100	200
0,75	0,5	0,85	0,14	0,16	$3,53 \cdot 10^{-8}$	14,28	16,32
0,75	0,75	0,92	0,23	0,28	$1,34 \cdot 10^{-6}$	23,07	28,4
0,75	1	1	0,33	0,44	$2 \cdot 10^{-5}$	33,33	44,44
0,8	0,8	0,95	0,19	0,22	$2 \cdot 10^{-4}$	19,04	22,67
0,9	0,9	0,98	0,09	0,1086	0,197	9,89	10,86

Полученные результаты являются основой решения задач о выделении необходимых временных и стоимостных ресурсов на обработку данных заданного объема.

Для того чтобы с доверительной вероятностью, не меньшей α , были обработаны данные объема N , выделяемое с этой целью время $\bar{T}$ должно быть определено из условия

$$\Phi^* \left[\frac{(1 - qf) \bar{T} - N}{\sqrt{Nqf}} \right] \geq \alpha.$$

Пусть t_α такое значение квантили нормального закона распределения,

что $\Phi^*(t_\alpha) = \alpha$. Так как $\Phi^*(z)$ — монотонно возрастающая функция, условие реализуемости обработки данных объема N запишется в виде

$$\bar{T} \geq (N + t_\alpha \sqrt{Nqf}) (1 - qf)^{-1}.$$

Соответствующее условие для стоимостного ресурса $\bar{c}$ записывается аналогично. Эти соотношения, устанавливая взаимосвязь между характеристиками методов контроля и обработки информации и требуемыми временными и стоимостными ресурсами, являются основой для решения задач выбора оптимальных методов контроля в СОД.

Представление результатов анализа механизмов контроля обработки данных с целью обеспечения достоверности информации в виде совокупности графовых моделей позволяет определить характеристики и исследовать практически любые возможные структуры обработки данных, учитывать различные случаи возникновения и взаимодействия ошибок.

Задачи синтеза механизмов контроля обработки информации для повышения ее достоверности возникают на этапе технического проектирования АСУ, а совершенствования этих механизмов — на этапе промышленной эксплуатации и модернизации. Эти задачи состоят в выборе наиболее рациональных методов контроля достоверности для элементов программного и информационного обеспечения. При этом выбор оптимальной совокупности указанных методов определяется требованиями к достоверности информации, законами возникновения и взаимодействия ошибок, ресурсами на разработку, внедрение и эксплуатацию механизмов контроля достоверности, возможностью использования конкретных методов контроля достоверности и защиты. При постановке и решении этих задач также используют введенные ранее понятия графа ошибок, индикаторного графа, "стандартной" схемы обработки и т.д.

Задача синтеза оптимальной системы повышения достоверности состоит в выборе методов контроля достоверности для частных ошибок, при использовании которых уровень ошибок $L(p)$ минимален и при ограничениях на временные, стоимостные и другие виды ресурсов, выделяемые с целью обеспечения достоверности. Так как определение аналитического вида функции $L(p)$ для произвольных древовидных структур $\epsilon(\pi)$ не представляется возможным, то в качестве целевой функции задачи синтеза системы повышения достоверности рассматривается снижение уровня ошибок на $\Delta L(p)$ при использовании методов контроля достоверности, которое необходимо максимизировать по абсолютной величине.

Пусть при $i = \overline{1, n}$; $j = \overline{1, m_i}$ переменная $x_{ij} = 1$, если для частной ошибки w_i используется j -й метод контроля достоверности, и $x_{ij} = 0$ — в противном случае. Пусть также Δ_{ij} — уменьшение вероятности частной ошибки w_i , вызванное использованием j -го метода контроля достоверности, τ_{ij} и c_{ij} — соответствующие временные и стоимостные затраты для j -го метода и события w_i ; T и C — ограничения на временные и стоимостные затраты. Тогда задача синтеза оптимальной системы контроля достоверности записывается в виде:

$$\min \left\{ \sum_{i=1}^n x_i \sum_{j=1}^{m_i} \Delta_{ij} x_{ij} \right\}$$

при ограничениях: на структуру контроля

$$\sum_{j=1}^{m_i} x_{ij} \leq 1, \quad i = \overline{1, n};$$

на временные затраты, связанные с контролем,

$$\sum_{i=1}^n \sum_{j=1}^{m_i} \tau_{ij} x_{ij} \leq T;$$

на стоимостные затраты, связанные с контролем,

$$\sum_{i=1}^n \sum_{j=1}^{m_i} c_{ij} x_{ij} \leq C.$$

К числу ограничений может быть отнесен также и объем памяти вычислительного комплекса, выделяемый для целей контроля достоверности.

Одним из методов повышения достоверности при обработке информации является применение в базисных вершинах методов контроля с использованием принципа обратной связи. Задача оптимизации структуры обработки информации состоит при этом в выборе такой структуры обработки (т.е. определении узлов обработки, этапов контроля и исправления обнаруженных ошибок, выборе методов обнаружения и исправления ошибок), которая обеспечивает максимум достоверности обрабатываемой информации при заданных ограничениях на время и материальные затраты. Для синтеза систем контроля данного класса, так же как и для их анализа, используются модели, в которых для каждой базисной вершины графа применяется понятие "стандартной" схемы обработки данных [2.16].

Резервирование информационных массивов, основанное на информационной избыточности, является эффективным методом повышения достоверности информации в АСУ и уменьшения вероятности потерь от ее разрушения. При этом под разрушением информационного массива понимается событие, приводящее к невозможности его дальнейшего использования вследствие появления ошибочных данных или выхода из строя физического носителя информации.

В настоящее время в СОД применяют три стратегии резервирования информационных массивов [2.16].

Стратегия I. Используется некоторое число копий информационных массивов. Если основной массив разрушился, то используется первая

его копия, если она разрушилась, то используется следующая копия и т.д.

Стратегия II. Используются особенности организации обновления массивов текущих данных, которые заключаются в том, что в качестве копий текущего массива служат его предыстории (предыдущие массивы и массивы изменений). Если текущий массив разрушился, то он восстанавливается программой обновления из предыдущего массива и массива изменений. Если и этот массив разрушился, то его можно восстановить из предыдущей предыстории и т.д.

Стратегия III. Смешанная стратегия, т.е. для текущего массива создаются его копии и хранится заданное число предысторий. Использование и восстановление массивов происходит аналогично предыдущим стратегиям. Причем вначале используются копии, а в случаях их разрушения массив восстанавливается из предыстории.

Развитием указанных стратегий резервирования являются методы, используемые в архивах машинных носителей, предназначенных для обеспечения длительного и надежного хранения информационных массивов и программных модулей [2.16]. В архиве хранится основной массив (оригинал) и K ($K = 1, 2, 3, \dots$) уровней дубликатов, из которых создаются копии пользователей для непосредственного использования на ЭВМ. Выделены две возможные стратегии организации работы архива машинных носителей.

Стратегия А-1. Копии пользователей получают из дубликата уровня K . Если дубликат разрушается, то его восстанавливают из дубликата уровня ($K - 1$) или при $K = 1$ из оригинала, после чего вновь делается попытка получения копии. При этом уже полученные копии в качестве дубликатов использоваться не могут.

Стратегия А-2. Копии пользователей получают из дубликата уровня K , но при его разрушении дубликат может быть получен из любой ранее созданных копий пользователей.

Основными характеристиками перечисленных стратегий резервирования являются вероятность успешного обновления (использования) информационного массива ρ ; вероятность разрушения основного массива и его копий и (или) предысторий $(1 - \rho)$; среднее время решения задачи (обновления) при условии успешного ее решения $E[T^y]$; среднее время до разрушения массива и его копий $E[T^p]$; среднее время решения задачи (обновления) вне зависимости от того, успешно она решена или нет $E[T]$, т.е. среднее время доступа к ЭВМ при использовании заданной стратегии резервирования; коэффициент готовности $COД K_T$; средние эксплуатационные затраты системы F в фиксированном интервале времени функционирования системы.

Основные результаты сравнения эффективности использования перечисленных стратегий резервирования с точки зрения экстремизации вероятностных, временных и стоимостных критериев сводятся к следующим.

1. Наибольшую вероятность успешного обновления массива обеспечивает стратегия I резервирования.

2. Если вероятность разрушения массива при обновлении мала — $q \rightarrow 0$, то наименьшее время доступа к ЭВМ обеспечивает стратегия II резервирования.

3. Если вероятность разрушения велика — $q \rightarrow 1/2$, и время получения копии массива превышает время обновления — $\tau > \theta$, то наименьшее время доступа к ЭВМ обеспечивает стратегия III, а при $\theta > \tau$ — стратегия II.

4. Наибольшую величину коэффициента готовности при $q \rightarrow 0$ обеспечивает стратегия II, а при $q \rightarrow 1/2$, $\theta > \tau$ — стратегия I резервирования.

5. При $q \rightarrow 1/2$ и $\tau < \theta$ минимальные эксплуатационные затраты обеспечивает стратегия I, а при $q \rightarrow 0$ — стратегия II резервирования.

6. Наибольшую вероятность неразрушения оригинала в архиве машинных носителей обеспечивает стратегия А-2.

На этапах отладки и опытной эксплуатации АСУ, а также при работе с массивами информации большого объема и содержащих информацию, разрушение которой наносит существенный ущерб системе (например, приводит к ее отказу), целесообразно использовать метод восстановительного резервирования. Под восстановительным резервированием понимают создание и хранение одной или нескольких копий и (или) предысторий массива (в зависимости от принятой стратегии резервирования), которые предназначены только для воссоздания разрушенной текущей версии. При этом к массивам, хранящимся в цикле восстановительного резервирования, применяется только операция копирования, что повышает сохранность информации.

Задача определения оптимальных стратегий резервирования существенно усложняется, когда для решения функциональных задач в СОД используется несколько информационных массивов, имеющих различные характеристики.

Существенно усложняются задачи резервирования массивов в вычислительных сетях. Использование различных стратегий резервирования в сетях ЭВМ имеет ряд особенностей, обусловленных спецификой вычислительных систем. В этом случае при анализе стратегий резервирования и выборе оптимальной стратегии необходимо учитывать топологию сети, надежность и стоимость использования каналов связи и ЭВМ сети, задержку сообщений и т.п.

Территориальная распределенность сетей ЭВМ позволяет выделить следующие основные варианты использования стратегий резервирования: централизованное, децентрализованное и динамическое хранение резерва.

При *централизованном* хранении резерва задача выбора оптимальной стратегии решается традиционными методами с учетом надежности, стоимости использования и временных характеристик узлов и каналов связи сети.

При *децентрализованном* хранении используются следующие дисциплины обработки поступающих запросов к массивам: адресация запроса в ближайший согласно некоторому критерию узел с резервом требуемого модуля или массива; одновременная адресация запроса в N узлов с резер-

вом; последовательная передача запроса по узлам с резервом, принадлежащим пути длины N ; из узла, в котором получен запрос, поочередно опрашиваются N узлов с резервом.

Возможно *динамическое* хранение резерва, при котором в некоторые моменты времени происходит перемещение резерва по узлам сети. Очередное местоположение резерва определяется согласно некоторой процедуре (случайной, фиксированной или аддитивной). При анализе и расчете оптимальных стратегий в этом случае необходимо учитывать возможность разрушения резерва при его перемещении и стоимостные затраты на реализацию этого перемещения.

Для решения задач оптимального резервирования в сетях используют эвристические методы. При этом значительные трудности вызывает большая размерность решаемых задач, что приводит к необходимости разработки специальных методов ее снижения.

§ 4.7. Защита информации

На современном этапе развития АСУ все большее значение приобретает защита информации, что связано, главным образом, с увеличением объемов данных, к которым одновременно обращается для решения различных задач большое число пользователей. Это приводит к повышению *уязвимости информации*, под которой понимают возможность ее несанкционированного использования, искажения или уничтожения в результате доступа пользователей, не обладающих специальными полномочиями, к конфиденциальным сведениям. Для уменьшения вероятности несанкционированного использования информации разрабатывают специальные механизмы ее защиты.

Серийно выпускаемая вычислительная техника имеет некоторые встроенные аппаратные и программные средства защиты. Так, например, многие ЭВМ оснащают специальными аппаратными средствами, позволяющими изолировать пользователей друг от друга и от операционной системы, что обеспечивает возможность эффективного контроля операций по выборке и посылке на хранение данных и уменьшает возможность несанкционированного доступа. Предусматривают специальные средства центрального процессора для обеспечения безопасности информации, в том числе программно-читаемые часы для регистрации времени свершения тех или иных событий; программно-читаемый закрепленный в памяти аппаратными средствами код идентификации, различный для каждого центрального процессора; инструкции по очистке блока памяти внесением нулей; использование только специфицированных кодов операций и т.п.

Чтобы не допустить искажения или потери информации, предусмотрена *система защиты памяти* при записи и считывании информации. Наиболее важной является защита при записи, однако в отдельных случаях необходимо знать, какая программа сделала попытку считать данные

или исполнить команду из запрещенной для нее зоны. Попытка нарушить защиту памяти вызывает прерывание программы-нарушителя. Обычно используют несколько способов организации защиты памяти. Часто встречается защита двумя регистрами, называемыми граничными или *регистрами защиты* и содержащими номера нижнего и верхнего граничных блоков сегмента. При появлении команды записи по некоторому адресу он последовательно сравнивается с граничными регистрами. Если требуемый адрес находится за пределами сегмента, указанного нижним и верхним регистрами, то возникает прерывание и после установления его причины управление передается специальной программе, обрабатывающей нарушения защиты памяти. Установку значений указанных регистров при работе ЭВМ в режиме разделения времени проводит специальная управляющая программа-супервизор. Аналогично функционирует система защиты с тремя регистрами. Третий регистр устанавливает, распространяется ли защита на внутреннюю область, определяемую верхним и нижним регистрами (состояние 0), или внешнюю (состояние 1).

В настоящее время развивается тенденция хранить программы в оперативной памяти не целиком, а составленными из отдельных сегментов. Для защиты этих сегментов можно использовать несколько пар регистров защиты, подобных описанным выше. Однако с ростом числа регистров растет число выполняемых логических операций, что усложняет схемную реализацию механизма защиты и может вызвать существенное снижение быстродействия ЭВМ.

Достаточно широко используют метод *защиты оперативной памяти по ключу*. Память разбивают на блоки, каждому из которых ставится в соответствие некоторый ключ. Условия выполнения программы запоминаются в слове памяти, называемом *словом состояния программы*. Супервизор помещает в это слово *ключ защиты программы*. Кроме того, в это же слово записывают первые позиции ключей каждого блока памяти, используемых текущей программой. При появлении адреса из некоторого блока памяти сравнивается ключ защиты в слове состояния программы и первые разряды ключа этого блока памяти. Несовпадение свидетельствует о попытке нарушения защиты и вызывает прерывание.

В системах, работающих в режиме разделения времени, функции защиты возлагают на трансляторы, что накладывает определенные ограничения на языки программирования, так как трансляторы должны до исполнения программ осуществить контроль. Этот метод часто используют для работы с языками, близкими к естественным. В этом случае символическим идентификаторам соответствует определенный адрес памяти (ячейка памяти, например, получившая обозначение А, всегда будет находиться по адресу А' для первого пользователя, по адресу А'' для второго и т.д., что упрощает контроль при трансляции).

Несмотря на широкое использование серийных средств защиты данных в ЭВМ, они не обеспечивают надежное перекрытие всех потенциально возможных каналов несанкционированного использования или утечки ин-

формации, что диктует необходимость разработки механизмов защиты при проектировании АСУ. Одна из основных их характеристик — вероятность "взлома", в результате которого будет нарушена безопасность информации, т.е. произойдет либо несанкционированное использование, либо разрушение (искажение или стирание) отдельных элементов или всех данных. Выбор механизмов защиты определяется особенностями рассматриваемой СОД, критериями синтезируемой системы защиты, используемыми методами защиты, имеющимися ресурсами и т.д. Существующие методы и механизмы защиты включают в себя процедурные, программные и аппаратные способы организации защиты.

Процедурные методы защиты обеспечивают доступ к данным только тем пользователям, которые имеют соответствующие разрешения. Например, для систем с разделением времени процедурные методы защиты должны обеспечивать контроль доступа со стороны пользователей, использующих терминалы ввода — вывода. Идентификация осуществляется путем проверки паролей терминалов, кодов пользователей, шифров работ и посылки обратного сигнала об их законности. Реализация процедурных методов защиты обеспечивается установлением паролей для терминалов, грифов секретности данных, созданием организационных и физических ограничений (сейфы, вахтеры, охрана и т.д.), а повышение их эффективности достигается путем соответствующего обучения и повышения уровня ответственности персонала. Ответственность за нарушение безопасности данных при этом возлагается на группу лиц, в обязанности которых входят: управление доступом к данным; учет попыток несанкционированного доступа к защищенным данным; регистрация лиц, имеющих копии данных ограниченного использования; анализ функционирования системы защиты и повышение качества ее работы; анализ последствий, вызванных "взломом" системы защиты. Процедурные методы защиты используют в основном на этапах первичной обработки данных и выдачи результатов обработки пользователям.

Программные и аппаратные методы защиты используют в основном на этапе обработки данных на ЭВМ. Они обеспечивают: обслуживание только "законных" пользователей; доступ к объектам защиты в соответствии с установленными правами и правилами; возможность изменения (модификации) правил взаимодействия между пользователями и объектами защиты; возможность получения информации о безопасности объектов защиты. Некоторые программные методы обеспечивают возможность анализа функционирования системы защиты с целью повышения качества ее работы и анализа последствий, вызванных "взломом".

Структура и характеристики аппаратных и программных методов защиты зависят от характеристик процессора и структуры памяти. С точки зрения взаимодействия пользователей в системе защита обеспечивает одну из следующих альтернатив: взаимную изоляцию данных пользователей и наличие библиотеки общего пользования; некоторые данные могут использоваться и (или) изменяться рядом пользователей, что контролируется механизмом защиты; удовлетворение специфических требований

160

пользователей в доступе к объектам защиты, что обеспечивается программируемыми механизмами защиты.

Упрощенным представлением организации защиты является следующая модель: "Каждый объект защиты окружен непроницаемым барьером с единственным проходом, у которого поставлен охранник".

Функцией охранника является проверка полномочий пользователей, которая осуществляется следующим образом: полномочия пользователя удостоверяются, если совпадает пароль, которым располагает охранник, и пароль, сообщенный пользователем.

Рассмотрим некоторые методы защиты для режима прямого доступа.

1. Помещения, где находятся терминалы, запирают. Доступ к ресурсам вычислительного комплекса имеют пользователи, которые обладают ключами.

2. Доступ к терминалам разрешается в случае предъявления соответствующего пропуска вахтеру.

3. Каждый терминал имеет список пользователей, которым разрешен доступ с этого терминала к ресурсам вычислительного комплекса. Пользователь сообщает системе свой код (пароль), который сверяется со списком. Если пароли совпадают, доступ разрешается.

4. Для каждого пользователя, код которого помещен в список, указывается временной интервал, в течение которого он имеет право обращаться к системе. В этом случае для доступа необходимо знать не только код пользователя, но и разрешенный ему временной интервал.

5. Пользователь начинает работу с терминалом, вводя свое имя "открытым текстом". По каналу связи имя передается в ЭВМ, где в защищенной таблице отыскивается соответствующий засекреченный ключ шифра. Этот ключ передается шифровальному устройству, входящему в состав ЭВМ, после чего на терминал посылается подтверждение о готовности. Тем временем пользователь вводит этот же ключ в свой терминал. Если ключи совпадают, то сеанс связи продолжается.

6. Если пароль пользователя не может быть сохранен в тайне, он должен быть только однократным. В этом случае пользователь ведет список паролей, вычеркивая из него уже использованные. В других случаях применяется контроль срока действия или общего числа применений конкретного пароля.

7. Список паролей пользователей в системе должен быть надежно защищен. Например, все оригиналы паролей при вводе подвергаются преобразованию *H*. Таким образом, в системе хранится список не оригиналов, а отображений паролей пользователей. Преобразование *H* должно быть плохо обрабатываемым, т.е. возможность найти значение пароля по его отображению должна быть минимальной. Пользователь, обращаясь к системе, предъявляет свой пароль, который преобразуется и сверяется со списком отображений. Совпадение свидетельствует о законности обращения пользователя. Засекреченным должен быть только пароль, а список отображений может и не быть засекреченным, поскольку практически невозможно получить пароль по его отображению.

8. Набор данных может быть защищен установлением для него списка пользователей, имеющих право доступа к этому набору.

9. Кроме списка пользователей можно указывать и типы доступа (чтение, запись, редактирование), которые разрешены конкретному пользователю в отношении этого набора данных.

10. Если набор состоит из неоднородных (в смысле защиты) данных, то для каждого типа данных устанавливается список пользователей и указывается, какие виды доступа разрешены каждому пользователю.

В процессе разработки системы в первую очередь должны быть определены требования к защите информации. Они заключаются в том, что всю

информацию по ее смысловому содержанию делят на группы различного уровня защиты, доступные: всем пользователям для чтения и обновления; всем для чтения и определенным категориям пользователей для обновления; определенным группам пользователей только для чтения и другим для чтения и обновления; определенным группам пользователей только для чтения и в индивидуальном порядке для чтения и обновления. Отнесение информации к той или иной группе секретности определяется тем, какой ущерб может быть нанесен системе при ее несанкционированном использовании.

Во всех случаях введения тех или иных методов защиты должны быть разработаны организационные меры, предусматривающие функционирование защиты, в частности хранение и смену паролей. С одной стороны, должна быть обеспечена "защита защиты" — например, невозможность кому-либо узнать пароль пользователя, в том числе сотрудникам ВЦ и эксплуатационному персоналу АСУ. С другой стороны, надо помнить о возможности такой ситуации, когда пользователь забыл или утерял свой пароль. В первом случае надо предусмотреть процедуру восстановления права пользователя на доступ к данным по старому или новому паролю. Во втором случае надо надежно "закрыть" данные, чтобы ими не могло воспользоваться лицо, нашедшее пароль.

Для маскировки наличия защиты может использоваться *метод правдоподобных ответов*. Он заключается в том, что лицу, обратившемуся с запросом к неразрешенным для него сведениям, выдается ответ, внешне совпадающий с запрашиваемым, но содержание заполняется случайным образом. Этот метод достаточно эффективен, так как не провоцирует на новые попытки "взлома" защиты. Недостаток метода заключается в том, что лица, имеющие право доступа, но обращающиеся с паролями небрежно, могут получить неверные сведения и, не подозревая об этом, использовать их в своей деятельности. Обычно эффективными являются "открытые" методы защиты, когда сам метод не представляет секрета, но воспользоваться им постороннему лицу невозможно.

Задачи синтеза систем защиты состоят в выборе совокупности методов защиты, обеспечивающей экстремальное значение определенной характеристики системы защиты при ограничениях на ряд других характеристик. В постановках задач синтеза систем защиты в качестве ограничивающих факторов и показателей эффективности защиты выступают стоимостные и временные затраты на разработку и эксплуатацию методов защиты, потери от "взлома" системы защиты, вероятность и среднее время несанкционированного доступа к объектам защиты.

Исходной информацией для постановки и решения задач синтеза являются следующие данные: W — множество объектов защиты; L_w — число структурных уровней защиты, которое необходимо последовательно преодолеть при доступе к объекту w ; характеристики k -го метода защиты, $1 \leq k \leq k_l$, использование которого возможно на l -м структурном уровне, т.е. $q_{kl}(w)$ — вероятность "взлома" (преодоления) l -го структурного уровня защиты объекта w при условии "взлома" предшествующих

уровней; $C_{kl}^{\pi}(w)$ и $C_{kl}^{\exists}(w)$ – стоимостные затраты соответственно на разработку и эксплуатацию; $\tau_{kl}^{\pi}(w)$ и $\tau_{kl}^{\exists}(w)$ – временные затраты соответственно на разработку и эксплуатацию; $d_{kl}(w)$ – потери в системе, вызванные преодолением ограничений рассматриваемого метода защиты; $t_{kl}(w)$ – время одной попытки несанкционированного доступа. Переменные: $x_{kl}(w) = 1$, если k -й метод защиты, относящийся к l -му уровню, закреплен за объектом w и $x_{kl}(w) = 0$, в противном случае $w \in W$, $1 < l < L_w$, $1 < k < k_l$.

Задача выбора оптимальной совокупности методов защиты, обеспечивающей максимальное среднее время несанкционированного доступа, записывается в виде

$$\max_{\{x_{kl}(w)\}} \left\{ \sum_{w \in W} \sum_{l=1}^{L_w} \sum_{k=1}^{k_l} t_{kl}(w) q_{kl}^{-1}(w) x_{kl}(w) \right\}$$

при ограничениях:

на структуру защиты

$$\sum_{k=1}^{k_l} x_{kl}(w) = 1, \quad w \in W, \quad l = \overline{1, L_w};$$

на вероятность несанкционированного доступа по структурным уровням и объектам защиты

$$\sum_{k=1}^{k_l} q_{kl}(w) x_{kl}(w) \leq q_l(w),$$

где $q_l(w)$ – допустимое значение вероятности несанкционированного доступа для l -го уровня и объекта w ;

по стоимостным затратам на проектирование и эксплуатацию системы защиты

$$\sum_{w \in W} \sum_{l=1}^{L_w} \sum_{k=1}^{k_l} C_{kl}^{\pi}(w) x_{kl}(w) \leq C^{\pi},$$

$$\sum_{w \in W} \sum_{l=1}^{L_w} \sum_{k=1}^{k_l} C_{kl}^{\exists}(w) x_{kl}(w) \leq C^{\exists},$$

где C^{π} , $C^{\exists}$ – выделяемые ресурсы на проектирование и эксплуатацию системы защиты;

по временным затратам на проектирование и эксплуатацию системы защиты

$$\sum_{w \in W} \sum_{l=1}^{L_w} \sum_{k=1}^{k_l} \tau_{kl}^{\pi}(w) x_{kl}(w) \leq T^{\pi},$$

$$\sum_{w \in W} \sum_{l=1}^{L_w} \sum_{k=1}^{k_l} \tau_{kl}^{\exists}(w) x_{kl}(w) \leq T^{\exists},$$

где T^{π} , $T^{\exists}$ – допустимые временные затраты на проектирование и эксплуатацию системы защиты;

на потери от "взлома" защиты

$$\sum_{w \in W} \sum_{l=1}^{L_w} \sum_{k=1}^{k_l} d_{kl}(w) x_{kl}(w) \leq D,$$

где D – допустимые суммарные потери от "взлома" защиты.

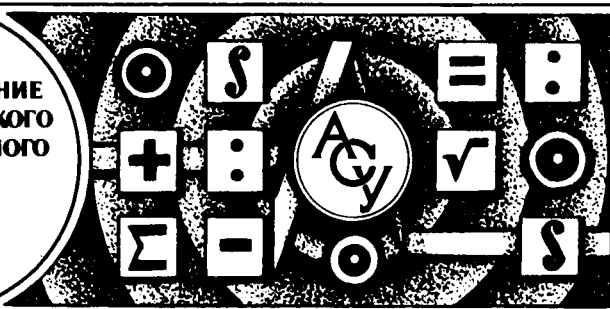
Решение рассмотренной задачи линейного целочисленного программирования булевыми переменными осуществляется методом "ветвей и границ" с использованием стандартных программных средств. Для сокращения размерности решаемой задачи целесообразно выделять подмножества идентичных (в смысле защиты) объектов и производить выбор методов защиты для "представителей" этих подмножеств.

Синтез динамических систем защиты осуществляется аналогично.

ПРОЕКТИРОВАНИЕ МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ГЛАВА

5



Характер математического и программного обеспечения АСУ существенно меняется по мере развития возможностей технических средств. В нашей стране принята программа всеобщей компьютерной грамотности и массового выпуска персональных микроЭВМ, обладающих высокой производительностью, развитой памятью и возможностью объединения в локальные вычислительные сети.

В этих условиях программное обеспечение направлено на работу с ЭВМ непрофессиональных программистов, самостоятельно создающих программы для своих задач в конкретной предметной области.

Разработчики АСУ, продолжая создавать алгоритмы и программы сложных и типовых задач, подготовка которых пользователями затруднительна, все в большей степени должны готовить программные средства поддержки для непрофессиональных программистов. Это повышает требования к разработчикам АСУ в части знания ими методов подготовки и отладки программ, автоматизации программирования.

Совершенствуются и развиваются языки программирования высокого уровня, появились непроцедурные языки, близкие к естественному. Облегчая работу пользователя, эти языки требуют развитого системного программного обеспечения, содержание которого все чаще становится различным для разных пользователей и должно определяться в процессе проектирования конкретной системы.

Структура математического обеспечения

Математического обеспечения вычислительных машин выполняет роль посредника между программистом и вычислительной техникой, обеспечивая удоб-

с
ем
лич
н
жеств.

ный для пользователя уровень общения с машиной; она включает в себя [2.19]:

операционную систему ОС ЕС, обеспечивающую эксплуатацию всех моделей ЕС ЭВМ (кроме ЕС 1011 и ЕС 1021) – однопроцессорных и многопроцессорных вычислительных установок с большим объемом оперативной памяти и полным набором внешних устройств в разнообразных режимах пользования вычислительных систем;

комплексы программ технического обслуживания – контрольно-наладочные тесты, обеспечивающие проверку правильности функционирования устройств и блоков во время наладки ЭВМ; дополнительные тесты, обеспечивающие классификацию отказов и локализацию места неисправности;

пакеты прикладных программ – пакеты, расширяющие возможности операционных систем, обеспечивающие работу диалоговых систем, работу в реальном масштабе времени, удаленную пакетную обработку; пакеты прикладных программ общего назначения, обеспечивающие различные применения алфавитно-цифровых дисплеев, графопостроителей; системы трансляции; программы для научно-технических расчетов, математического программирования, обработки матриц, автоматизации программирования и отладки и др.; пакеты прикладных программ, ориентированные на применение в АСУ – обобщенные системы обработки данных, информационно-поисковые системы общего назначения и системы обработки документов;

специальные программы пользователя; проектирование специального математического обеспечения является одним из основных этапов создания современных АСУ и в значительной степени определяет стоимость (50–60%) и длительность их разработки [2.18].

Основными характеристиками специального математического обеспечения АСУ являются время функционирования и объем обрабатываемой информации соответствующими комплексами программ, средний интервал времени между полным повторением решения функциональных задач. В зависимости от требований, предъявляемых к этим характеристикам, могут быть выделены следующие типы комплексов программ: АСУ организационного типа – АСУ управления отраслью народного хозяйства на уровне министерств и ведомств, АСУ предприятий и производственных объединений; оперативного управления производством на уровне предприятия, цеха; АСУ технологическими процессами, управления движением и т.п.

Комплексы программ первого типа характеризуются большими объемами поступающей и перерабатываемой информации, время обслуживания требования (сообщения) может составлять минуты и часы, интервал времени между повторными решениями задачи – несколько дней и даже месяцев. Комплексы программ второго типа обеспечивают решение задач с цикличностью смены или суток при допустимой задержке сообщений порядка долей минут или нескольких минут. Темп функционирования комплексов программ третьего типа определяется динамикой управляемых процессов. Как правило, допустимая периодичность решения задач в АСУ техноло-

гическими процессами составляет минуты, а допустимая задержка обработки сообщений и выдачи управляющих команд измеряется секундами. В автоматизированных системах управления движением цикл решения задач исчисляется секундами и даже десятими и сотыми долями секунды (системы управления воздушным движением), а допустимое время обработки сообщения находится в пределах долей секунды и миллисекунды.

Среднее время цикла решения задач зависит от производительности используемой вычислительной техники, объемов обрабатываемой информации и сложности решаемых задач. Задержка сообщений кроме этих параметров в значительной степени определяется типами внешних запоминающих устройств.

Разработка комплексов программ АСУ включает следующие этапы: 1) анализ требований к системе и разработка спецификаций; 2) разработка алгоритмов и блок-схем программ; 3) определение структуры программ (выделение программных модулей и их оптимизация); 4) кодирование (программирование); 5) отладка.

Временные затраты на реализацию этапов цикла разработки программного обеспечения составляют (в %): анализ требований и разработка спецификаций — 20; проектирование и программирование — 35; отладка — 45 [2.18].

§ 5.2. Операционные системы ЭВМ и трансляторы

Операционная система ОС ЕС предназначена для обеспечения пакетной обработки заданий, режима разделения времени и совмещения этих режимов. При этом осуществляется связь оператора с системой, протоколирование хода работы вычислительной системы, защита хранимых данных от несанкционированного доступа, работа с удаленными терминалами и абонентскими пунктами через каналы связи. Предусмотрена работа как с символьными, так и графическими устройствами ввода — вывода и ряд других функций [2.19].

Операционная система может работать как в однопрограммном режиме, так и в режиме мультипрограммирования. При *однопрограммном* режиме в оперативной памяти в каждый момент времени находится только одна выполняемая программа, все задания выполняются последовательно. Для работы в этом режиме достаточен объем оперативной памяти 64 Кбайт. Режим *мультипрограммирования* может выполняться с фиксированным и переменным числом задач.

В режиме мультипрограммирования с *фиксированным числом задач* оперативная память объемом не менее 128 Кбайт распределяется при генерации системы или оператором между одновременно выполняемым фиксированным числом заданий не более 15. Допускается в рамках задания распараллеливать процесс вычисления путем организации одновременно выполняемых задач общим числом не более 255. Между этими задачами, выполняемыми в рамках заданий, ресурсы распределяются динамически.

В режиме мультипрограммирования с *переменным числом задач* все ресурсы, включая оперативную память, минимальный объем которой составляет 256 Кбайт, распределяются между одновременно выполняемыми заданиями динамически. Число заданий может быть произвольным, но не более 15. Число задач, организуемых для распараллеливания вычислений, не лимитируется и определяется динамически в соответствии с наличием свободных ресурсов.

Пакетная обработка заданий осуществляется в виде их непрерывного потока. Все задания размещаются на диске и образуют входную очередь, из которой они выбираются практически без участия оператора, последовательно или по установленным приоритетам. Переход от обработки одного задания к другому осуществляется автоматически. Входные очереди могут пополняться в произвольные моменты времени.

Режим *разделения времени* реализуется путем выделения определенных интервалов времени, называемых *квантами*. Предназначенные для обработки в этом режиме задания, называемые *квантующимися*, находятся в оперативной памяти одновременно. В течение одного кванта времени обрабатывается одно задание, затем управление получает следующее квантующееся задание и т.д. Квантующиеся задания находятся в оперативной памяти без вытеснения во внешнюю память вплоть до завершения их обработки.

На старших моделях ЕС ЭВМ с объемом оперативной памяти не менее 512 Кбайт на базе мультипрограммного режима с переменным числом задач реализуется система разделения времени для обслуживания более 100 одновременно работающих удаленных абонентов. В оперативной памяти выделяются разделы, предназначенные для выполнения одного или нескольких заданий. Каждое задание получает управление в течение кванта времени, после чего оно вытесняется во внешнюю память, а на его место загружается следующее задание, получающее управление на свой квант времени. Таким образом, в каждый момент времени в соответствующем разделе оперативной памяти находится одно выполняемое задание, а все остальные, выполняемые в режиме разделения времени, сохраняются во внешней памяти. Последовательную загрузку заданий в оперативную память называют *свопингом*.

В режиме разделения времени возможна реализация диалога человека с ЭВМ, включая разработку и выполнение программ. Операционная система ОС ЕС позволяет вести диалог и без режима разделения времени; возможна разработка пакетов прикладных программ для программирования в режиме диалога на языках высокого уровня.

При разработке систем *реального времени* следует учитывать их специфические особенности, главная из которых заключается в том, что время реакции системы на поступающие сообщения и сигналы должно быть настолько мало, чтобы успеть обработать содержащиеся в них данные и использовать полученные результаты для управления процессом. Поскольку потоки данных носят случайный характер, система должна быть всег-

да в состоянии готовности получать входные сигналы и управляться ими. Повторить поступившие данные невозможно, поэтому потеря их недопустима. Высокую скорость реакции можно получить при максимальном распараллеливании процесса обработки данных, что требует высокого уровня мультипрограммирования, обеспечиваемого ОС ЕС.

Составной частью ОС ЕС является *система программирования*, в которую входят *трансляторы* с языков программирования АСЕМБЛЕР, ФОРТРАН IV, АЛГОЛ-60, КОБОЛ, ПЛ/1, РПГ, а также *редактор связей* и *тестран*. Программы готовятся с использованием модульного принципа. Большие программы сегментируются на меньшие, разрабатываемые и отлаживаемые самостоятельно с последующим объединением в процессе выполнения или перед ним.

Различают модули исходные, объектные и загрузочные.

Исходный модуль представляет собой программу, записанную на языке программирования.

Объектный модуль представляет собой результат трансляции набора данных, содержащихся в исходном модуле, на машинный язык. Формат объектных модулей одинаков для всех трансляторов, что позволяет объединять модули, написанные на разных языках программирования, и одинаковым образом обрабатывать их редактором связей.

Загрузочный модуль есть объединение объектных модулей, он уже может быть помещен в оперативную память для выполнения; редактор связей осуществляет редактирование — замену, исключение и перегруппировку секций программы по указанию программиста, создание необходимой структуры программы, формирование и запись характеристик загрузочного модуля в оглавление библиотеки и др.

В зависимости от структуры загрузочного модуля он может быть загружен в оперативную память целиком или по частям. В ОС ЕС допускаются следующие структуры модулей: простая, оверлейная или с перекрытием, динамическая последовательная и динамическая параллельная.

Модуль *простой структуры* не требует для выполнения задачи передачи управления никакому другому модулю. Он загружается в оперативную память как единое целое и выполняется от начала до конца, после чего занимаемое им место в оперативной памяти освобождается. Такая структура очень эффективна по скорости выполнения, так как не требует загрузки других частей и передачи им управления. Однако для больших задач загрузить в оперативную память одновременно все программы часто оказывается невозможным.

В *оверлейной структуре* программа делится логически на сегменты. В процессе выполнения программы некоторые сегменты загрузочного модуля, но не все, находятся в оперативной памяти одновременно. Загружаются они в одну и ту же область оперативной памяти последовательно по мере необходимости. При этом обычно один сегмент — *включающий* — находится в памяти в течение всего времени выполнения программы. Сменяющие друг друга сегменты — *исключающие* — обмениваются между

собой информацией через область памяти включающего сегмента. По мере усложнения задач, когда число сегментов очень велико или их выбор зависит от обрабатываемых данных, фиксированная структура запланированного редактором связей перекрытия сегментов может оказаться неэффективной.

В *динамических последовательных структурах* операционная система находит по имени вызываемый модуль, загружает его в выделенный для него участок памяти и передает ему управление. Этот модуль может вызывать другой модуль и т.д. Когда выполнение некоторого модуля закончено, управление передается вызвавшему модулю, а участок памяти, занимаемый выполненным модулем, освобождается. При необходимости многократного использования модуля он может сохраняться в оперативной памяти.

Динамическая параллельная структура применяется для параллельного выполнения программ в мультипрограммном режиме, если какая-либо из программ предусматривает при выполнении ожидания каких-либо событий, например, завершения операции ввода – вывода. В период ожидания может выполняться другая программа.

При необходимости повторного использования программ в режиме мультипрограммирования с переменным числом задач используют один из трех типов загрузочных модулей – однократно используемые, повторно используемые в реентерабельные. Тип модуля формируется редактором связей по указанию программиста.

Однократно используемые модули каждый раз необходимо заново вызывать из библиотеки, так как во время выполнения модули изменяются, что не позволяет осуществлять их повторное выполнение, это – простейший случай.

Повторно используемые модули обладают свойством самовосстановления. Любая их часть перед повторным использованием восстанавливается в исходное состояние. Это позволяет осуществить однократную загрузку модуля в оперативную память и многократное его использование. Однако изменение модуля в процессе выполнения не позволяет использовать его одновременно двумя или более пользователями или задачами. Поступающие на использование модуля запросы ставятся в очередь до освобождения и восстановления модуля.

Реентерабельные загрузочные модули никак не изменяются во время выполнения. Их можно загружать на защищенные участки памяти, чтобы исключить возможность случайных изменений со стороны какой-либо программы пользователя. Поэтому реентерабельная программа может использоваться одновременно несколькими задачами. До окончания выполнения модуля может по прерыванию начаться выполнение другой задачи, а затем продолжиться выполнение первой. Таким образом, реентерабельные программы позволяют работать с ними одновременно, параллельно и независимо многим пользователям, что является большим преимуществом и сильно расширяет возможности систем коллективного пользования.

Версия операционной системы ОС 7.0 ЕС реализует концепцию *вирту-*

альной машины, при которой каждому пользователю предоставляется функциональный эквивалент реальной ЭВМ – собственная виртуальная вычислительная машина. Система виртуальных машин с помощью операционной системы обеспечивает полную независимость функционирования отдельных виртуальных машин. Предусмотрены средства разграничения доступа к системе, хранимым или обрабатываемым в ней данным.

Специальная управляющая программа – *монитор виртуальных машин* – обеспечивает согласованное использование технических средств реальной ЭВМ и виртуальных машин, управляет их параллельной работой. Предусмотрен диалоговый режим работы любого одиночного пользователя, а также обмен данными как между различными удаленными абонентскими пунктами, так и между абонентскими пунктами и виртуальной машиной.

Каждая виртуальная машина содержит *виртуальный процессор* в виде части времени центрального процессора реальной ЭВМ, а также *виртуальную оперативную память* емкостью до 16 Мбайт, создаваемую с помощью средств динамического преобразования адресов реальной ЭВМ. Пользователю предоставляется *виртуальный пульт* в виде пультowego устройства реальной ЭВМ либо абонентского пункта со специальными языковыми средствами, а также *виртуальные каналы ввода – вывода*. Последние реализуются в виде части времени работы реальных каналов либо их функционирование моделируется программно.

Вторая составная часть ОС 7.0 ЕС – *базовая операционная система* – ориентирована на режим пакетной обработки. Она может работать на одной или нескольких виртуальных машинах. Обеспечивается выполнение программ пользователей, реализуемых с операционной системой ОС 6.1 ЕС. Важно отметить, что базовая операционная система обеспечивает работу нескольких виртуальных машин при использовании одной копии ее собственных программ. В качестве внешней памяти используются виртуальные накопители на магнитных дисках емкостью до 16 Мбайт.

Монитор виртуальных машин поддерживает двухпроцессорные конфигурации ЭВМ. Предусмотрен режим присоединенного процессора, при котором оба процессора работают с общим полем оперативной памяти, а периферийные устройства доступны только одному из них. Работу виртуальных машин моделируют оба реальных процессора, что увеличивает число или производительность виртуальных машин.

Операционная система ОС 7.0 ЕС реализуется с центральным процессором третьей очереди ЕС ЭВМ с оперативной памятью емкостью не менее 1 Мбайт.

Технические возможности малых ЭВМ позволяют создавать модульные системы, в которых отдельные модули выполняют некоторые выделенные функции. Такими модулями могут быть специализированные процессоры, универсальные микропроцессоры, интеллектуальные терминалы. На такие системы, называемые *системами с разделением функций*, ориентирована операционная система РАФОС (РАзделения Функций Операционная Система) [2.15].

Система РАФОС предназначена для организации вычислительного процесса в магистрально-модульных комплексах СМ ЭВМ, имеющих от 16 до 248 Кбайт оперативной памяти. Она ориентирована на АСУ ТП реального времени, автоматизацию научных исследований, решение информационных и управленческих задач в системах коллективного пользования и др. Может быть эффективно использована для микроЭВМ и персональных компьютеров.

В структуре РАФОС выделяют драйверы внешних устройств, мониторы, файловую систему, системные программы и библиотеки. Используя языки высокого уровня типа ФОРТРАН, ПАСКАЛЬ, БЕЙСИК, пользователи создают прикладные программы, пакеты и библиотеки.

Драйверы внешних устройств представляют собой специальные программы операционной системы, обеспечивающие доступ на физическом уровне ко всем периферийным устройствам и внешней памяти со стороны мониторов РАФОС, системных и прикладных программ. Система позволяет включать в нее новые драйверы для дополнительных внешних устройств, не предусмотренных в РАФОС, а также включать обращение к таким устройствам непосредственно в программу.

В состав РАФОС входит пять типов мониторов, программно совместимых снизу вверх:

RM-монитор, постоянно находящийся в оперативной памяти (резидентный) и предназначенный для исполнения в реальном времени отложенной программы;

SJ-монитор — монитор реального времени, обеспечивающий самую высокую скорость реакции на внешнее событие (прерывание). Он функционирует при минимальной конфигурации технических средств, требуя около 4 Кбайт оперативной памяти и обеспечивая работу системы в оперативной памяти от 16 до 56 Кбайт. Введение в его состав при генерации многотерминального драйвера позволяет программе работать с несколькими терминалами;

FB-монитор — простейший мультипрограммный монитор РАФОС, являющийся фоново-оперативным монитором реального времени. Все задачи получают один из восьми приоритетов. Фоновая задача имеет самый низкий приоритет 0, оперативная задача — самый высокий 7. Приоритеты от 1 до 6 имеют системные задачи, аналогичные оперативной задаче, но с более низким приоритетом. Все прикладные задачи и системные с низким приоритетом выполняются в фоновом режиме. Общее число задач — не более 8. Монитор использует оперативную память от 32 до 56 Кбайт, его резидентная часть занимает около 8 Кбайт.

XM-монитор является расширением FB-монитора и обслуживает до 8 задач реального времени. Он занимает около 14 Кбайт памяти и обеспечивает работу с оперативной памятью от 64 до 248 Кбайт.

TS-монитор является самым мощным монитором. Он обслуживает до 30 задач пользователей, работающих одновременно с разных терминалов

в режиме разделения времени. Предназначен для работы с системой, имеющей от 96 до 248 Кбайт памяти; его память оклепо 40—48 Кбайт.

Для вычислительных комплексов, реализуемых на базе микроЭВМ типа "Электроника" используется *операционная система ФОДОС* (Фоновая Основная Дисковая Операционная Система).

Вычислительный комплекс строится по модульному принципу, при котором отдельные функциональные устройства связаны между собой единым каналом обмена информацией.

Система ФОДОС предназначена для решения одним пользователем задач в реальном масштабе времени, в том числе разработки новых программ. При решении задач возможна пакетная обработка, а также выполнение одновременно двух задач. При подготовке программ на языке АС-СЕМБЛЕР имеется возможность создавать программы с оверлейной структурой и автоматизировать процесс отладки программ.

В результате трансляции программ, написанных на языках АССЕМБЛЕ-Ре и ФОРТРАНе, получают объектные форматы, состоящие из набора данных и машинных кодов и являющиеся объектными модулями. Последние используются в качестве входных для редактора связей.

Операционная система содержит ряд мониторов, имеющих модульную структуру.

Монитор одного задания (SJ) предназначен для выполнения одной программы пользователя. Он обслуживает все периферийные устройства и обеспечивает выполнение всех программ системы. Для работы монитора требуется не менее 8 Кслов; он позволяет использовать до 28 Кслов оперативной памяти.

Монитор основного фонового задания (FB) позволяет одновременно выполнять два независимых задания, из которых одно является основным и имеет приоритет по отношению ко второму — фоновому.

Монитор расширенной памяти (XM) дополнительно к возможностям монитора FB может обслуживать до 124 Кслов оперативной памяти. Кроме указанных мониторов существуют драйверы; программа обслуживания пользователей, обеспечивающая операции с файлами и загрузку драйверов; интерпретатор командной строки, являющийся составной частью программы обслуживания пользователя и осуществляющий прием и синтаксический анализ командной строки пользователя и некоторые другие.

Для динамического распределения памяти ФОДОС использует свопинг. Он заключается в том, что при необходимости выполнение фонового задания прерывается, часть его переписывается на диск и на освободившееся место загружается программа обслуживания пользователей. После обработки запроса пользователя восстанавливается то состояние, которое было до прерывания, т.е. соответствующая часть фонового задания загружается на то место, которое она занимала.

С переходом на магнитные носители и использованием режима коллективного доступа практически отпала необходимость в промежуточной подготовке данных. Основным средством ручного ввода данных является клавиатура дисплея, причем зачастую ввод осуществляется непосредственно пользователем, без посредников. В этих условиях основным для проектировщиков АСУ становится вопрос об организации планирования и выполнения заданий операционной системой ЭВМ, от чего, в свою очередь, зависит выбор системы приоритетов заданий в соответствии с важностью своевременного решения задач АСУ.

Режим разделения времени относится к методу обслуживания терминалов операционной системой. Каждому заданию, вводимому с определенного терминала, выделяется квант времени на выполнение. Все задания выполняются в определенной последовательности, процессор не может быть монополизирован одним заданием, а скорость процессора не дает возможность пользователям заметить совместную работу многих терминалов и почти одновременное выполнение многих заданий. Задания пользователей распределяются и накапливаются в очереди входных заданий. Результаты решения задач накапливаются в выходной очереди, откуда они рассылаются непосредственно на терминалы пользователей.

Одним из определяющих факторов эффективного применения ЭВМ в условиях коллективного использования вычислительных средств и при работе в режиме разделения времени являются используемые принципы планирования (распределения) ресурсов центрального процессора.

Существует множество алгоритмов распределения квантов времени между несколькими заданиями.

В мультипрограммных системах несколько активных заданий конкурируют из-за процессорного времени. Кроме процессоров пользователей существуют еще системные процессоры, которые также участвуют в использовании процессора. Все активные процессы, ожидающие, когда им будет выделен процессор, стоят в очереди *готовых процессов*. Процессы, которые ожидают, когда завершится ввод — вывод или заблокированы по другим причинам, стоят в отдельных очередях *заблокированных процессов*. Процесс переводится из очереди заблокированных процессов в очередь готовых, когда с него снимется блокировка.

При проектировании очередей готовых процессов необходимо учитывать многочисленные показатели производительности, такие, как хорошая загруженность ресурсов, малое время простоя, высокая пропускная способность и разумное время ожидания для заданий пользователей. Ни одно задание в системе не должно ждать "вечно". Правила планирования в реальных системах должны отражать стремление получить приемлемые эксплуатационные характеристики сразу в нескольких направлениях.

Хотя ни один показатель в отдельности не оптимизируется, достигается общая хорошая производительность системы.

В настоящее время наиболее часто используются три метода (класса алгоритмов) планирования: наивысшего приоритета, круговорота и очереди с обратной связью.

Метод наивысшего приоритета. Один из простых методов планирования очереди готовых процессов состоит в том, что процессор предоставляется тому из них, который имеет наивысший приоритет. Если вытеснение не допускается, то процесс с наивысшим приоритетом выполняется до тех пор, пока не кончится или не заблокирует сам себя. При поступлении в очередь процесса с более высоким приоритетом, чем у текущего, он должен ждать, пока текущий процесс не освободит процессор. Если вытеснение разрешено, то при появлении процесса с более высоким приоритетом текущий процесс прерывается и управление переходит к вновь прибывшему. Вытесненный процесс возвратится в очередь готовых процессов.

Пользуясь этой стратегией, прежде всего нужно решить, на каких характеристиках будет основан приоритет. Наиболее часто предлагаемый вариант — выбрать из очереди самое короткое задание, т.е. приоритет обратно пропорционален времени обработки. Для коротких процессов стратегия с вытеснением лучше, чем без вытеснения, так как в этом случае длинным процессам становится труднее удерживать процессор в своем распоряжении. При сильной загрузке системы длинные процессы обслуживаются хуже. Если система близка к пределу своих возможностей, то любое правило планирования работает не вполне хорошо.

Так как предварительная информация о времени выполнения обычно неточная, то при планировании можно предпринять попытки основывать выбор на информации о процессе, полученной за время его пребывания в системе. Каждому процессу при входе в систему присваивается некий приоритет. Приоритет возрастает с коэффициентом a во время ожидания в очереди готовых процессов и с коэффициентом b — во время выполнения. В зависимости от выбора коэффициентов a и b получаются разные правила планирования. Например, если $0 < a \leq b$, то очередь обслуживается в порядке поступления. Если $0 < b \leq a$, то получается обслуживание в противоположном порядке. Коэффициенты a и b могут зависеть от внешнего приоритета, т.е. от затрат, которые пользователь считает приемлемыми. Можно выбрать и нелинейные функции. Можно, чтобы приоритет убывал по линейному закону с течением времени. Когда достигается некое максимальное время, приоритет скачком возрастает до некоторой большой величины. Это благоприятствует коротким процессам и при этом соблюдается условие, что ни одному процессу не придется ждать обслуживания слишком долго. Избранный способ вычисления приоритетов определяется особенностями того типа процессов, с которыми имеет дело система. Например, если на ВЦ преобладают задачи обработки информационных потоков и эффективность определяется в единицах введенной и переработанной информации, то приоритет этих задач можно повысить даже в ущерб дру-

гим задачам. Такие задачи обычно используют много операций ввода — вывода. Поэтому процессор будет большею частью свободен и задачи с меньшим приоритетом, но не требующие устройств ввода — вывода будут использовать свободный процессор.

Если ВЦ обслуживает научное учреждение, основная загрузка которого состоит в выполнении большого числа не связанных между собой заданий, эффективность определяется количеством обработанных заданий. Тогда приоритеты можно назначать обратно пропорционально времени исполнения. При такой постановке может сложиться ситуация, когда будут выполняться только короткие задания. Для устранения этого каждой задаче при поступлении в очередь приписывают ранг от 1 до 9.

При просмотре очереди может оказаться, что задача с более низким рангом может быть обслужена, а для более приоритетной задачи ресурсов недостаточно. В этом случае всем задачам, находящимся во входной очереди выше выбранной, ранг увеличивается на 1. Таким образом, приоритет задачи постепенно повышается, если нужные ей ресурсы велики. Когда ранг достигнет значения 9, задача рассматривается как срочная. Для этой задачи все нужные ей свободные и освобождающиеся ресурсы резервируются до тех пор, пока она не будет обслужена. Можно использовать другую стратегию приоритетов, когда каждой задаче присваивается класс в зависимости от требуемой памяти. До 100 Кбайт — класс А, до 200 — класс В и более 200 — класс С. Задача поступает в очередь соответствующего класса на место, определяемое приоритетом. При освобождении ресурсов просматриваются очереди по каждому классу, а внутри очереди задачи просматриваются в порядке приоритета.

Одна из важных задач планирования состоит в том, чтобы обеспечить занятость внешних устройств. С этой целью процессам можно присваивать высокий приоритет в периоды, когда они интенсивно используют ввод — вывод.

Количество различных правил вычисления приоритета практически не ограничено. Несмотря на то что стратегия приносит желаемый результат, обеспечивая большинству процессов хорошее обслуживание, она дает и побочный эффект: долгое ожидание процессов с низким приоритетом. В некоторых системах это неприемлемо. Например, в системах с разделением времени пользователю гораздо удобнее получить по 1 с машинного времени каждые 5 с, чем 12-минутный интервал каждый час. Метод наимысшего приоритета не может обеспечить такой вид обслуживания.

Метод круговорота. Простой круговорот не использует никакой статической или динамической информации о приоритетах. Первый процесс из очереди готовых процессов получает квант времени длиной в q секунд, а затем отправляется снова в конец очереди, если только он себя не заблокирует. Если в очереди имеется K процессов, то каждый процесс получает q секунд из каждых Kq секунд процессорного времени. Следовательно, каждый процесс как бы выполняется процессором, скорость которого равна $1/K$ скорости физического процессора. Поэтому длина очереди — это важный параметр, определяющий быстроту продвижения процессов.

Величина кванта q определяет, насколько равномерно распределено время процессора по коротким периодам. С уменьшением q улучшается обслуживание более коротких процессов. Когда q мало, все готовые процессы обслуживаются одинаково и время ожидания прямо пропорционально объему требуемых услуг. Однако можно выбрать q слишком малым. Для квантов "разумной" величины можно пренебречь временем, необходимым для переключения процессора с одного процесса на другой. Если величина q того же порядка, что и время переключения, то задержка при переключении становится заметной. В самом деле, если время q слишком мало, то система может потратить время на переключение большее, чем на их выполнение.

Метод круговорота часто применяют в системах с разделением времени, рассчитанных на большое число пользователей. Такие системы должны гарантировать каждому процессу приемлемое время ответа. Иначе говоря, пользователи ожидают, что независимо от величины кванта их будут обслуживать по меньшей мере один раз каждые несколько секунд. Если q постоянно и в очереди K процессов, то время ответа приблизительно равно Kq . Для больших K это может быть слишком долго. Определенное время ответа можно гарантировать, если применить круговорот, ориентированный на цикл. Время цикла выбирается для очереди готовых процессов равным максимально достижимому времени ответа. Величина кванта вычисляется в начале каждого круга прохождения очереди. Процессы, вновь поступающие во время цикла, задерживаются до окончания цикла, прежде чем будут поставлены в очередь. Если очередь слишком длинная, то потери на переключение могут быть заметными, так как q будет совсем маленьким. Эту проблему можно урегулировать, задав минимальную величину кванта. При сильной загрузке системы пострадает время ответа, но, по крайней мере, издержки на переключение останутся на низком уровне.

Имея несколько очередей готовых процессов, система может извлечь выгоду из характера распределения времен работы. Процессор тратит большую часть своего времени на выполнение процессов из ведущей очереди, обеспечивая таким образом отличное обслуживание коротких процессов. В одном из алгоритмов, ориентированных на цикл, процессам из ведущей очереди предоставляется фиксированный квант времени. Если после того, как проработали все процессы ведущей очереди, до конца цикла еще остается время, то оно используется для выполнения фоновых процессов. Такую систему с двумя очередями можно обобщить до многоуровневого метода очередей с обратной связью.

Метод очередей с обратной связью. В основном алгоритме используется несколько очередей, каждая из которых обслуживается в порядке поступления. Новый процесс, входя в систему, попадает в первую очередь. После получения одного кванта он поступает во вторую очередь. После получения каждого последующего кванта он переходит в очередь со следующим номером. Время процессора планируется так, что он обслуживает непустую очередь с наименьшим номером.

При этом методе вновь поступивший процесс неявно получает высокий приоритет и выполняется подряд в течение стольких квантов времени, сколько успеет пройти до прихода следующего процесса, но не больше, чем успел проработать предыдущий. Это правило обеспечивает лучше, чем круговорот, обслуживание коротких процессов, не требуя предварительной информации о времени выполнения. Однако работа с несколькими очередями может повлечь большие издержки времени в начальный период, особенно если величина кванта мала.

При стратегии очереди с обратной связью пользователи могут извлечь для себя некоторую выгоду, разбивая свои задания на короткие процессы. Каждый из них будет обслуживаться с большим приоритетом, как вновь поступивший короткий процесс. Общее время обслуживания всех частей будет значительно лучше, чем если представить задание в виде одного длинного процесса, который будет переходить в очередь с более низким приоритетом и выполняться только при освобождении предыдущих очередей.

Основное преимущество очередей с обратной связью и круговорота по сравнению с обслуживанием по наивысшему приоритету состоит в том, что первые два метода позволяют хорошо обслуживать короткие задания, не требуя предварительной информации о времени выполнения процесса. С другой стороны, в тех случаях, когда приоритет вычисляется как функция от ресурсов, выделяемых процессу, часто желательно отдать предпочтение процессам с высоким приоритетом, чтобы увеличить загруженность ресурсов.

В большинстве систем процессоры взаимодействуют посредством прерываний. В центральном процессоре прерывание происходит, когда истекает установленный в системе времени или когда закончит работу канал, или когда выполняемой программе необходим ввод — вывод. Когда происходит прерывание, центральный процессор передается супервизору системы. Супервизор выполняет некоторые действия, такие, как запуск канала, разблокирование или создание нового процесса, а затем возвращает процессор готовому процессу из очереди. Так как прерывания происходят часто, невыгодно выполнять очень много команд каждый раз при предоставлении процессора очередному готовому процессу. Однако важно сохранить информацию, касающуюся текущего состояния процессора, а также выполнить некоторые вычисления, связанные с приоритетом и величиной кванта. Следовательно, одна из задач заключается в том, чтобы придумать такой метод работы с очередью готовых процессов, для которого эти вычисления нужно было бы делать лишь изредка.

Метод многоуровневого планирования. Принцип, лежащий в основе метода, состоит в том, что операции, которые встречаются часто, должны занимать меньше времени, чем те, которые встречаются редко. Этот принцип особенно удобен в применении к алгоритмам планирования, где операции разбиваются на уровни в зависимости от частоты выполнения.

Метод многоуровневого планирования реализуется как система, со-

стоящая из трех уровней: диспетчера, краткосрочного и долгосрочного планировщика. Диспетчер вызывается после того, как обработка прерывания завершена. Он выбирает следующий подлежащий выполнению процесс из очереди готовых процессов. Диспетчер работает очень часто, поэтому сам он должен быть очень коротким. Его задача — только взять первый процесс из очереди и предоставить ему процессор. Краткосрочный планировщик вызывается, чтобы вставить готовый процесс в очередь; хотя этому планировщику приходится, быть может, анализировать состояние процесса, уточнения приоритета на этом уровне должны быть сведены к минимуму. Процессы ставятся в очередь часто, возможно каждые несколько миллисекунд. Поэтому внесение любых сложных изменений в состояние процесса следует оставить долгосрочному планировщику. Долгосрочная корректировка, которая делается не так часто, может быть более сложной. Для вычисления нового приоритета может использоваться информация о состоянии процесса, например о предоставленных ему ресурсах. Так как долгосрочный планировщик вызывается только раз в несколько секунд, то допустимо потратить на его работу несколько миллисекунд.

Преимущество такой многоуровневой организации состоит в том, что она локализует издержки времени, связанные с планированием. Если приблизительная частота вызовов каждого уровня известна заранее, то ограничения на допустимый объем вычислений на каждом уровне тоже можно указать заранее. Всякий предполагаемый алгоритм планирования должен удовлетворять временным ограничениям на каждом уровне.

Организация вычислительного процесса на мини-ЭВМ. Применение мини-ЭВМ в АСУ имеет свои преимущества и свои недостатки. Системное программное обеспечение, имеющее направленность на диалоговый режим, дает определенные преимущества мини-ЭВМ в системах реального времени, но вызывает трудности реализации пакетного режима при большом объеме вычислений. Архитектура мини-ЭВМ позволяет создание действительно реентерабельного кода. Жесткое разделение кода от данных является базовой характеристикой архитектуры мини-ЭВМ. Эта особенность полезна в системах обработки запросов, где большое число операторов на терминалах решают одинаковые или близкие по содержанию задачи в отличие от систем коллективного доступа больших ЭВМ, где разные операторы решают различные задачи.

Сейчас большая часть мини-ЭВМ применяется в многозадачном режиме, однако есть возможность работы в режиме мультипрограммирования с полной аппаратной защитой между разделами.

Для избежания процесса создания большой программной системы планирования ОС мини-ЭВМ предлагают стандартные решения в области планирования памяти.

Требования небольшой стоимости привели к ограниченным возможностям операционной системы мини-ЭВМ, но эти ограничения отразились в положительном смысле на некоторых качествах систем. Основным результатом — повышение надежности ОС вследствие малого объема и небольшой

сложности. Большая надежность привела к тому, что в ОС мини-ЭВМ не получили достаточного развития процедуры восстановления после сбоев. На мини-ЭВМ универсального применения редко встречаются такие средства предотвращения и восстановления сбоев, как аппаратное резервирование, системы обнаружения ошибок и реконфигурации машины, учет отказавших блоков оперативной памяти.

Появление мини-ЭВМ связывалось для многих пользователей с возможностью уйти от слишком сложных ОС и реализовать удобные для применения несложные системы. Но в последние годы производители мини-ЭВМ расширяют возможности аппаратных конфигураций и в большей степени усложняют и расширяют возможности ОС. Для основных типов мини-ЭВМ уже имеются наборы ОС с разными функциональными возможностями, удовлетворяющие разным по объему и сложности требованиям конкретных применений.

В современных мини-ЭВМ хорошо развита работа с массивами. Имеются необходимые для этой цели методы доступа — последовательный, индексно-последовательный, прямой. Встречаются и другие методы, одни из которых являются частными случаями трех вышеуказанных, а другие — дальнейшим развитием этих методов (методы доступа к базе данных, телеобработки) в направлении аналогичных методов больших ЭВМ. Обеспечивается динамическое распределение памяти на магнитных дисках, которое является полезным и мощным средством повышения эффективности мини-ЭВМ. Операционные системы включают возможности работы на уровнях массива и записи, одновременный доступ нескольких пользователей к массиву и защиту их от несанкционированного доступа.

Системы управления данными и массивами являются позднейшим добавлением и расширением ОС мини-ЭВМ конкретно для их применения в АСУ. Они представляют собой программные системы, которые продаются отдельно от ОС и включаются в них в процессе генерирования.

§ 5.4. Алгоритмизация задач

На первом этапе разработки комплексов программ АСУ определяют требования, выполнение которых обеспечивает решение поставленных задач. Основными характеристиками являются время и стоимость обработки информации, вероятность ошибки, допустимые объемы памяти, неизбежность выполнения основных функций и т.п. Требования к системе содержатся в системных спецификациях, которые отображают предполагаемую реализацию системы с помощью ЭВМ. Спецификации являются основополагающим документом в процессе разработки системы. Точность и подробность составления спецификаций определяют вероятность возникновения ошибки на дальнейших этапах разработки. Большинство современных методов создания программного обеспечения допускает появление ошибок в спецификациях проекта, однако цена обнаружения и

корректировки ошибок возрастает по мере приближения разработки к завершению. В большинстве случаев ошибки в системе — результат неполных или противоречивых спецификаций (60–70% ошибок). В связи с этим возникает необходимость разработки и использования языков и формализованных методов разработки спецификаций, их анализа и контроля.

В настоящее время известен ряд методов формализации определения спецификаций. Первым шагом в направлении формализации процесса определения спецификаций явилось создание *HOS-методологии*, основанной на использовании *аксиом программного обеспечения высшего уровня*, которые определяют допустимые взаимодействия между процедурами задачи. Эти аксиомы предусматривают введение некоторых ограничений на использование процедур. Например, не допускаются ситуации, когда процедура контролирует сама себя, т.е. исключается использование рекурсии, процедура не может управлять пространством своих входных данных и т.п.

Эффективным средством формализации процесса определения спецификаций является *система PSL/PSA*, которая включает язык определения задач (*PSL*) и анализатор определения задач (*PSA*).

Язык определения задач предназначен для отображения функциональных требований к ресурсам. Язык содержит набор средств, позволяющих определять объекты системы, их свойства и взаимосвязи.

Анализатор определения задач осуществляет испытание предложений на языке *PSL*. Анализатор генерирует базу данных, отображающую системные требования, и осуществляет проверку последовательности и анализ полноты данных.

Система PSL/PSA позволяет получить формализованное представление проблемы, документирование системных требований в единообразной форме, обеспечивает выявление ошибок, обусловленных неполнотой информации.

Система структурного анализа и проектирования SADT представляет собой графическую систему, предназначенную для анализа и проектирования систем. *Графический язык* системы *SADT* — это иерархический структурированный набор диаграмм, причем каждый блок диаграммы раскрывается более детально с помощью другой диаграммы. Таким образом, задача представляется все с большей степенью детализации по мере разработки проекта. Система *SADT* позволяет повысить качество решений на ранних этапах проектирования на основе легко доступных средств контроля. Однако система не является автоматизированной.

Система SREM, используемая для автоматизации анализа требований, включает *язык определения требований RSL* для установления связи между объектами. Проверка последовательности предложений на языке *RSL* осуществляется с помощью процессора *REVS*. Система выполняет следующие операции: разработку требований, включающую описатели данных и этапы их разработки; разработку подробных проектов; моделирование

отдельных аспектов проектных решений с учетом принятых допущений; проверку пользователем требований, предъявляемых к системе.

Система структурного проектирования представляет собой неавтоматизированную систему, использование которой обеспечивает создание подробного проекта системы. Система состоит из следующих уровней: подсистема, процесс (часть подсистемы, представляющая отдельную функцию); работа (выполняемая программа или шаг задания); модуль (часть работы). Структурное проектирование выполняется на уровне процесса или работы. Документация включает базисную схему каждого модуля и его полное описание по установленной форме. Функция модуля описывается так, чтобы можно было осуществить его кодирование без использования дополнительной документации.

Приведенные средства определения спецификаций позволяют систематизировать процесс формирования требований к системе, автоматизировать процесс их контроля и снизить вероятность возникновения ошибок, вызванных некорректностью или неполнотой учета условий решаемых задач управления.

Вторым этапом процесса разработки программного обеспечения является составление алгоритма решения поставленной задачи.

При разработке алгоритма решения задачи прежде всего решаются следующие вопросы [2.6]: а) следует ли использовать имитационное моделирование или какой-либо из методов оптимизации; б) следует ли учитывать случайный характер переменных или процессов или использовать детерминированный подход; в) следует ли учитывать нелинейность некоторых соотношений или достаточно ограничиться их линейной аппроксимацией; г) можно ли использовать существующие методы решения или требуется разработать новый алгоритм; д) допустимо ли в данном конкретном случае использование известных оптимизационных методов или необходимо разработать эвристический алгоритм.

При решении приведенных вопросов возникает необходимость оценки качества алгоритмов. Существует много критериев. Чаще всего используется *критерий сложности алгоритма*, который характеризует время, затрачиваемое алгоритмом на получение решения в зависимости от размера задачи [2.17]. При этом под *размером задачи* понимается некоторое число, которое выражает меру количества входных или выходных данных.

Предел временной сложности при увеличении размера задачи называется *асимптотической временной сложностью*. Аналогично определяется *емкостная сложность* и *асимптотическая емкостная сложность*. Именно асимптотическая сложность алгоритма определяет в итоге размер задач, которые можно решить этим алгоритмом. Если алгоритм обрабатывает входы размера n за время cn^2 , где c — некоторая постоянная, то говорят, что временная сложность этого алгоритма есть $O(n^2)$ (читается "порядка n^2 "). Точнее, говорят, что неотрицательная функция $g(n)$ есть $O(f(n))$, если существует такая постоянная c , что $g(n) \leq cf(n)$ для всех, кроме конечного (возможно пустого) множества неотрицательных значений n .

Пусть дано пять алгоритмов $A_1 - A_5$ (табл. 5.1). Под *временной сложностью* здесь понимается число единиц времени, требуемого для обработки входа размера n . В табл. 5.1 приведены размеры задач, которые можно решить за 1 с, 1 мин и 1 ч каждым из этих пяти алгоритмов. Если в качестве основы для сравнения взять 1 мин, то, заменяя алгоритм A_4 алгоритмом A_3 , можно решить задачу в 6 раз большую, а заменяя A_4 на A_2 , можно решить задачу, большую в 125 раз. Если в качестве основы для сравнения взять 1 ч, то различие алгоритмов становится еще значительнее. Отсюда можно заключить, что асимптотическая сложность алгоритма служит важной мерой его качества, причем такой мерой, которая приобретает все большее значение с увеличением размера вычислений.

Сложность алгоритма определяет также то увеличение размера задачи, которого можно достичь с увеличением скорости машины. Увеличение скорости вычислений в 10 раз увеличивает размер задачи, которую можно решить, для алгоритма A_5 только на 3, тогда как для алгоритма A_3 — более чем втрое. Однако алгоритм с быстро растущей сложностью может оказаться предпочтительнее для задач с малым размером. Например, если временные сложности алгоритмов $A_1 - A_5$ равны соответственно $1000n$, $100n$, $\log n$, $10n^2$, n^3 и 2^n , то алгоритм A_5 будет наилучшим для задач размера $2 \leq n \leq 9$, A_3 — для задач размера $10 \leq n \leq 58$, A_2 — при $59 \leq n \leq 1024$, а $A_1 - n > 1024$.

Таблица 5.1

Алгоритм	Временная сложность	Максимальный размер задачи		
		1 с	1 мин	1 ч
A_1	n	1000	$6 \cdot 10^4$	$3,6 \cdot 10^6$
A_2	$n \log n$	140	4893	$2,0 \cdot 10^5$
A_3	n^2	31	244	1897
A_4	n^3	10	39	153
A_5	2^n	9	15	21

Для того чтобы показать, что не существует алгоритма, выполняющего решение задачи менее чем за определенное время, необходимо точное определение того, что есть алгоритм. Уточнение понятий алгоритма и его сложности требует привязки к некоторой конкретной схеме вычислений. Обычно в качестве такой схемы рассматривается машина Тьюринга или так называемая адресная машина. Однако можно принять модель, более близкую к реальному ходу вычислений в оперативной памяти машины с одним процессором, и считать, что вычисления реализуются программой, написанной на некотором упрощении (подмножестве) языка АЛГОЛ.

На первоначальных этапах развития теории сложности в ней чаще всего рассматривались задачи, на которые можно дать лишь один из двух ответов: ДА или НЕТ. Например, классическая "сложностная" постановка известной задачи о ранце (с ограничением в виде равенства) будет заключаться в выяснении того, существуют ли для заданных чисел a_j , $j = 1, 2, \dots$

..., n и b значения булевых переменных x_j , удовлетворяющие ограничению

$$\sum_{j=1}^n a_j x_j = b. \quad (5.1)$$

Обычная же формулировка задачи о ранце состоит в максимизации суммы

$$z = \sum_{j=1}^n c_j x_j \quad (5.2)$$

при условии (5.1). Однако легко заметить, что ответы на последовательность задач типа "существует ли набор булевых переменных x_j , удовлетворяющих (5.1), такой, что $z = k$ при $k = 0, 1, 2, \dots$ " дадут решение задачи о ранце в привычной формулировке. При этом под задачей понимается весь класс однотипных вопросов, различающихся входными данными. Например, в задаче (5.1), (5.2) вход представляет собой набор $\{k, \{a_j\}, \{c_j\}, b\}$, так что задача о ранце сводится к решению вопроса "существует ли значение целевой функции, равное k , в ситуации, описываемой остальными входными данными". Конкретный набор исходных данных определяет реализацию задачи.

Обозначим задачу через Π , ее реализацию — через π . Под *размером реализации* будем понимать число символов, необходимых для записи входных данных этой реализации. Размер реализации π будет обозначаться $|\pi|$. Под временем решения t_π реализации π понимается сумма времен выполнения всех команд программы.

Введем понятие *класса задач P , разрешимых за полиномиальное время*. Этот класс состоит из всех задач Π , обладающих следующим свойством: для любой $\Pi \in P$ существует АЛГОЛ-программа u_Π и полином $f_\Pi(\cdot)$, такие, что любая реализация u_Π за время $t_\pi(u_\Pi) \leq f_\Pi(|\pi|)$. Примерами задач из класса P являются задачи нахождения кратчайшего пути между двумя вершинами графа, минимального разреза в сети, минимального дерева.

Общепринято, что если задачу нельзя решить быстрее, чем за экспоненциальное время, то ее следует рассматривать как трудно разрешимую. Такие задачи образуют класс задач, недетерминированным образом разрешимых за полиномиальное время (NP -класс). Для определения NP класса необходимо расширить язык, добавив к нему новую команду $\text{CHOOSE}(x)$. Эта команда придает недетерминированному образом переменной x значение 0 или 1. Расширенный за счет этой команды язык назовем N -АЛГОЛ. Будем говорить, что программа на N -АЛГОЛе (N -программа) выдает на некоторую реализацию задачи ответ ДА, если N -программа выдает ответ ДА при некоторой комбинации значений, реализовавшихся при выполнении команд $\text{CHOOSE}(x)$. В этом случае под временем выполнения N -программы будем понимать минимальное время для всех детерминированных программ, полученных путем замены $\text{CHOOSE}(x)$ нулями или единицами, которые выдают ответ ДА. Если N -программа дает ответ НЕТ, то время ее выполнения для данной реализации не определяется.

Отметим, что приведенное определение времени для реализации с от-

ветом ДА является в значительной мере условным, так как любая реальная схема вычислений является детерминированной и для нее необходимо просмотреть все последовательности реализации CHOOSE(x), чтобы определить, выдает ли хоть одна из них ответ ДА. Это потребует времени, равного сумме времен, необходимых для каждого набора реализаций.

Определим класс NP как подкласс всех таких задач P , для которых существует N -программа u_P и полином $f_P(\cdot)$, такие, что для любой реализации π задачи P , для которой имеется ответ ДА, программа u_P выдает ответ ДА за время $t_\pi(u_P) \leq f_P(|\pi|)$. Так как любая программа является N -программой, то, очевидно, $P \subset NP$.

Дадим определение: множество A полиномиально сводимо к B (обозначается $A < B$), если существует функция φ , вычисляемая за полиномиальное время и такая, что при

$$a \in A \Leftrightarrow \varphi(a) \in B.$$

Если $A < B$ и $B < A$, то A и B полиномиально эквивалентны.

Множество E называется NP -полным (или универсальной переборной задачей), если $E \subset NP$ и для любого $A \subset NP$ имеем $A < E$. Множество NP -полных задач обозначается NPC , $NPC \subset NP$.

Поиск полиномиально разрешимых подклассов NP представляет большой теоретический и практический интерес. Класс NP -полных задач включает в себя многие классические задачи комбинаторики, такие, как задачу о коммивояжере, о гамильтоновом цикле, задачи целочисленного линейного программирования. Многолетнее изучение этих задач не привело к нахождению алгоритма полиномиальной сложности, и задачи рассматриваемого класса считаются теоретически трудно разрешимыми. Такими являются не только комбинаторные задачи, но и обычные задачи линейного программирования. При решении задачи линейного программирования стандартным симплекс-методом невозможно дать верхнюю оценку числа итераций, в виде полинома от параметров, характеризующих размер задачи.

Результаты машинных экспериментов и опыт решения прикладных задач показывают, что возможности точных методов ограничены и не удовлетворяют потребностям практики. Это подтверждается и теоретическим исследованием эффективности точных методов. Потребности практики чаще всего покрываются приближенными решениями, что объясняется неизбежной неточностью задания исходной информации во многих прикладных задачах и тем, что модели отражают реальные ситуации лишь с определенной степенью приближения.

Обычно различают два уровня повышения эффективности алгоритмов — локальный и глобальный. Под *локальным уровнем* понимаются любые приемы, позволяющие расширить имеющиеся на сегодняшний день возможности решения задач. *Глобальный уровень* предусматривает перевод определенных классов задач на более низкий уровень (например, из

NP в P). Такой переход обычно означает переход от точных методов к приближенным.

В тех случаях, когда рассматриваемая задача не имеет характера вопроса, а явным образом содержит целевую функцию, уместно ставить вопрос о нахождении не точных, а приближенных решений с оценкой их отклонения от оптимума.

Рассмотрим задачу нахождения оптимума функции $f(x)$ на допустимом множестве G . Пусть x^* — оптимальное решение этой задачи. Говорят, что допустимое решение x является ϵ -приближенным, если

$$\frac{|f(x) - f(x^*)|}{|f(x^*)|} \leq \epsilon, \quad (5.3)$$

где $\epsilon > 0$ — заданное число ($f(x^*) \neq 0$).

Величина $\epsilon > 0$ есть *гарантированная характеристика* приближенного алгоритма u , если для любой реализации задачи алгоритм дает приближенное решение x и выполняется неравенство (5.3). Сравнение приближенных алгоритмов проводится на основании гарантированных характеристик, которые позволяют выбрать оптимальный в определенном смысле эвристический алгоритм решения задачи.

Различные типы экстремальных задач из класса NPC ведут себя при нахождении ϵ -приближенного решения по-разному. Например, поиск ϵ -приближенного решения задачи коммивояжера снова представляет собой задачу класса NPC ; для задачи же о ранце и некоторых других приближенных задач существуют алгоритмы поиска ϵ -приближенного решения с полиномиальной сложностью. Таким образом, перспективным направлением повышения эффективности комбинаторных алгоритмов является выявление подклассов задач из класса NPC , для которых поиск ϵ -приближенных решений переводит их в класс P .

Для задач массового счета перспективным является применение статистически эффективных алгоритмов. Такие алгоритмы в подавляющем большинстве случаев оптимальны (ϵ -приближенное решение). Иначе говоря, с ростом размерности задачи доля точно оптимизируемых задач (ϵ -оптимизируемых) среди всех задач стремится к 1.

При оценке эффективности алгоритмов теоретическими методами следует иметь в виду, что полученные результаты дают представление о поведении задач в наихудших возможных случаях, тогда как для вычислительной практики существенно более важно их поведение в среднем. Действительно, экспериментально известно, что для решения задачи линейного программирования с m ограничениями и n переменными обычно требуется от m до $3m$ итераций; как правило, для задач не слишком больших размеров число итераций близко к $3m/2$. Кроме того, теоретические результаты практически ничего не говорят о том, как будет вести себя конкретный алгоритм на конкретной задаче. Например, среднее время прямого поиска в списке пропорционально $N/2$, где N — число элементов списка, а двоичного поиска (список упорядочен по ключу) — пропорционально $\log_2 N$. Однако двоичный поиск неэффективен для небольших списков, которые подлежат частому изменению, так как введение нового элемента

может вызвать переписывание всех элементов. Разработка алгоритмов является в основном творческой деятельностью, хотя существует множество типовых методов и алгоритмов, которые могут применяться для решения задач, возникающих в АСУ. К таким методам прежде всего относятся методы исследования операций.

Наиболее разработанными приближенными методами решения экстремальных задач являются комбинаторные методы, основная идея которых состоит в использовании конечности множества решений и в замене полного перебора направленным перебором. При этом главную роль в сокращении перебора играет оценка и отбрасывание неперспективных (т.е. заведомо не содержащих оптимума) подмножеств решений. Общая схема комбинаторных методов (методы типа схемы "ветвей и границ") применима практически ко всем задачам дискретного программирования независимо от их линейности или иных свойств. Эта схема может быть применена непосредственно к исходной естественной формулировке экстремальной задачи, без сведения ее к целочисленной задаче линейного программирования. Кроме того, эти методы чаще всего обеспечивают конечность вычислительного процесса по самому своему построению.

Долгое время единственным способом оценки эффективности комбинаторных методов являлось численное экспериментирование. Основной тенденцией, выявленной в достаточно представительных сериях экспериментов, является тенденция к экспоненциальному возрастанию трудоемкости вычислений при росте числа переменных. Теоретические исследования комбинаторных задач также приводят к заключению об экспоненциальной трудоемкости решения комбинаторных задач. Однако практика показывает, что комбинаторные методы позволяют решать задачи АСУ достаточно больших размеров.

Использование комбинаторных методов типа "ветвей и границ" положено в основу стандартного пакета ЛП АСУ (для решения задач линейного программирования).

§ 5.5. Структурное программирование

Эффективным методом повышения качества комплексов программ является использование при их подготовке принципов *структурного программирования* [2.20]. При организации проектирования программ и процесса кодирования это позволяет предотвратить большинство логических ошибок и обнаружить те, которые допущены. Структурное программирование включает три составляющие: проектирование сверху — вниз; модульное программирование; структурное кодирование.

Проектирование сверху — вниз. По существу это иерархический подход к решению поставленной задачи. Метод предусматривает на первом этапе определение задачи в общих чертах и последовательное уточнение ее структуры путем детализации основных функций. На каждом шаге детализации

выясняются основные функции, т.е. задача разбивается на ряд подзадач, пока эти подзадачи не станут настолько простыми, что каждая из них может быть представлена несложным программным модулем, действие которого описывается одной фразой.

Метод предусматривает описание данных, их структуры и основных процессов обработки. Описание данных должно включать тщательно отобранные примеры, демонстрирующие выполнение основных функций системы и их наиболее существенные варианты. При описании модуля должны быть описаны тестовые данные.

При проектировании сверху — вниз создание проекта системы по уровням, начиная с верхнего, облегчает реализацию поставленных целей и выявление ошибок на ранних стадиях создания программного обеспечения.

Модульное программирование. В его основе лежит разделение программы на логические части (модули) и последовательное программирование каждой части. Если проведено проектирование всей задачи сверху — вниз, то она разбивается на подзадачи, соответствующие возможным модулям. Разбиение задачи на модули должно обеспечивать независимость реализации каждого программного модуля. Последующее формирование комплекса программ не должно требовать знаний о внутренней работе модуля, т.е. необходимо обеспечить простоту сопряжений программных модулей комплекса.

При разбиении задачи на модули следует учитывать их размер. Обычно размер модуля составляет около 60 строк. Такая длина удобна для восприятия, так как она легко охватывается и запоминается. Это число строк помещается на одной странице, его легко прочесть на терминале вычислительной машины. Однако целесообразно выбирать размеры модуля так, чтобы он соответствовал одной подзадаче.

Следует стремиться к независимости программных модулей. Для этого необходимо, чтобы модуль не зависел от источника входных данных, места назначения выходных данных и от предысторий. Каждый модуль должен иметь свое назначение, отличающееся от назначения других модулей. Это должен быть замкнутый блок, вход и выход которого точно определены. Функциональная независимость модулей сокращает вероятность воздействия изменений в одной подпрограмме на остальные подпрограммы комплекса.

Воздействие изменений в одном модуле на другие модули называют *волновым эффектом*. Этот эффект можно уменьшить, сведя к минимуму связь между модулями, т.е. сократив количество путей, вдоль которых изменения или ошибки влияют на другие части программы. Простейший путь уменьшения волнового эффекта состоит в сокращении использования глобальных переменных и в сокращении размеров модуля. Минимизация взаимосвязей между модулями проводится обычно за счет усиления связей между элементами одного модуля (модульной прочности). Высокая степень независимости, характерная для структурного программирования, и минимальная связь модулей приводят к ограничению влияния ошибок

в пределах отдельных модулей и к уменьшению как функциональной сложности модуля, так и сложности связей.

Формализация и оптимизация процесса разбиения системы программного обеспечения на модули с учетом приведенных факторов позволяет существенно повысить качество создаваемых комплексов программ и сократить вероятность появления ошибок на дальнейших этапах разработки.

Рассмотрим постановку и решение задачи разбиения сложной системы программного обеспечения на функциональные модули, имеющие минимальное число информационных связей, при ограничении на общее число выделяемых модулей, на число процедур и информационных элементов в каждом модуле, на сложности связей между отдельными модулями [2.16]. Исходными данными для задачи являются: множество различных типов входных, промежуточных и выходных переменных — информационных элементов; множество необходимых процедур обработки (преобразования) информационных элементов; взаимосвязи процедур обработки информационных элементов.

Отображение множества процедур на множество информационных элементов удобно представить в виде двудольного графа, дуги которого соединяют процедуры с соответствующими информационными элементами, причем дуги, инцидентные различным информационным элементам, могут быть помечены различными цветами (рис. 5.1).

Задача разбиения программного обеспечения на модули может быть сформулирована как задача разбиения заданного множества процедур обработки данных на непересекающиеся подмножества, имеющие минимальное число попарно общих информационных элементов. Наиболее наглядно такая постановка иллюстрируется с использованием графа, вершинами которого являются процедуры, а связывающие их дуги различного цвета соответствуют имеющимся информационным элементам. В этом случае в графовой интерпретации данная задача формулируется как задача разбиения раскрашенного графа на подграфы (модули), обеспечивающие минимум суммарного числа дуг различного цвета, связывающих выделенные подграфы, при ограничении на общее число выделяемых подграфов, число вершин и дуг в каждом подграфе, число связей между ними.

Для удобства постановки, описания методов и результатов решения задачи сведем информацию о связи процедур и данных (элементов записей), содержащуюся в графовой модели, в специальную табличную форму. По строкам таблицы расположим выполняемые системой функции или процедуры (запись, сортировка, контроль, обновление, выборка, печать и т.д.), а по столбцам — элементы множества идентификаторов исходных, промежуточных и выходных данных, связанных с выполнением этих процедур. На пересечении столбца и строки таблицы записывается "1" — признак зависимости или связи рассматриваемой функции и информационного элемента, если таковая существует.

В этом случае критерий оптимального разбиения на модули информационного и программного обеспечения, содержание которого описано в табличной форме, формулируется следующим образом. Найти разбиение множества функций или процедур таблицы на непересекающиеся подмножества, минимизирующее суммарное число информационных связей между модулями при

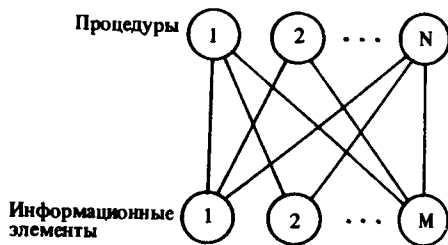


Рис. 5.1

ограничениях: на число выделяемых модулей; на число информационных элементов, входящих в один модуль, определяющее сложность разработки блок-схемы подсистемы; на число процедур каждого модуля, определяющее, в основном, сложность кодирования блок-схемы; на число информационных связей между выделяемыми модулями; на совместимость отдельных процедур.

Введем необходимые обозначения:

$D = \{d_1, d_2, \dots, d_l, \dots, d_i\}$ – множество информационных элементов системы;
 $\|w_{rl}\|$ – матрица взаимосвязи процедур и информационных элементов, где

$$w_{rl} = \begin{cases} 1 & \text{– если элемент } d_l \text{ используется при выполнении процедуры } a_r, \\ 0 & \text{– в противном случае;} \end{cases}$$

$$x_{rv} = \begin{cases} 1 & \text{– если процедура } a_r \text{ входит в модуль } v, \\ 0 & \text{– в противном случае, } r = \overline{1, R}, \quad v = \overline{1, V}, \quad V < R; \end{cases}$$

$$y_{lv} = \begin{cases} 1 & \text{– если } \sum_{r=1}^R w_{rl} x_{rv} > 1, \\ 0 & \text{– если } \sum_{r=1}^R w_{rl} x_{rv} = 0, \quad l = \overline{1, L}. \end{cases}$$

Суммарное число S различных информационных элементов, являющихся попарно общими для различных модулей, равно

$$S = \sum_{l=1}^L \sum_{v=1}^V \sum_{v'=v+1}^V y_{lv} y_{lv'}. \quad (5.4)$$

При разработке программного обеспечения АСУ требуется минимизировать S при следующих основных технологических ограничениях:
 на число процедур в модулях

$$1 < \sum_{r=1}^R x_{rv} < \overline{M}_v, \quad v = 1, 2, \dots, V, \quad (5.5)$$

где $\overline{M}_v$ – мощность множества процедур в v -м модуле;
 на число информационных элементов, используемых модулями,

$$\sum_{l=1}^L y_{lv} < L_v, \quad (5.6)$$

где L_v – максимальное число информационных элементов, используемых v -м модулем;

на интерфейс между отдельным модулем и другими модулями системы

$$\sum_{l=1}^L \sum_{v=1}^V y_{lv} y_{lv'} < S_{vv'} \quad \text{для заданного } v', \quad (5.7)$$

где $S_{vv'}$ – максимально возможное число информационных связей между модулем v' и остальными модулями системы;

на интерфейс между двумя заданными модулями системы

$$\sum_{l=1}^L y_{lv} y_{lv'} < S_{vv'} \quad \text{для заданных } v \text{ и } v', \quad (5.8)$$

где $S_{vv'}$ – максимально возможное число информационных связей между модулями v и v' ;

на разделение отдельных процедур системы по различным модулям

$$x_{rv} + x'_{r'v} < 1, \text{ где } v = 1, 2, \dots, V \text{ для заданных } r \text{ и } r'. \quad (5.9)$$

Поставленная задача является задачей квадратичного целочисленного программирования и может быть сведена к линейной форме путем линеаризации выражений (5.4), (5.7), (5.8) и введением связи между вспомогательными и основными переменными. В этом случае она может быть решена с использованием имеющихся пакетов прикладных программ.

В условиях, когда задано информационное обеспечение, т.е. определено содержание каждого информационного массива и его динамические характеристики, решаются динамические задачи разбиения системы программного обеспечения на модули. В качестве критерия оптимизации используются минимум общего времени обмена с внешней памятью и минимум межмодульного интерфейса.

Исходными данными для постановки и решения динамических задач синтеза системы модулей являются множество информационных массивов системы, для которых определены: входные, выходные и промежуточные данные; множество альтернативных процедур обработки данных, входящих в состав алгоритмов, обеспечивающих решение задач системы; возможные последовательности выполнения процедур в процессе обработки; способы обмена с внешней памятью модульной системы обработки данных; взаимосвязи процедур с информационными массивами; временные характеристики обращения к массивам.

Задачи синтеза систем модулей программного обеспечения СОД сведены к задачам нелинейного целочисленного программирования. Для их решения предложены алгоритмы, использующие схему "ветвей и границ" и особенности функционирования модульных систем обработки данных [2.16].

Структурное кодирование. Третьим составным элементом структурного программирования является структурное кодирование, которое представляет собой метод написания хорошо структурированных программных модулей, удобных для тестирования, модификации и использования. Метод предусматривает написание программных модулей произвольного размера и сложности на основе ограниченного множества базисных логических структур. Метод аналогичен принципу, положенному в основу проектирования схем, где любая логическая структура может быть создана из элементарных структур И, ИЛИ и НЕ. Структурное кодирование базируется на строго доказанной теореме о структурировании [2.1], которая утверждает, что любую правильную программу (с одним входом и одним выходом, без заикливаний и недостижимых команд) можно написать с использованием следующих логических структур: последовательности двух или более операторов; выбора одного из двух операторов (IF THEN, ELSE); повторения (или управления циклом) оператора, пока выполняется некоторое условие (DO WHILE).

На рис. 5.2 показаны основные логические структуры: последовательности (рис. 5.2, а); выбора (рис. 5.2, б); повторения (рис. 5.2, в). Каж-

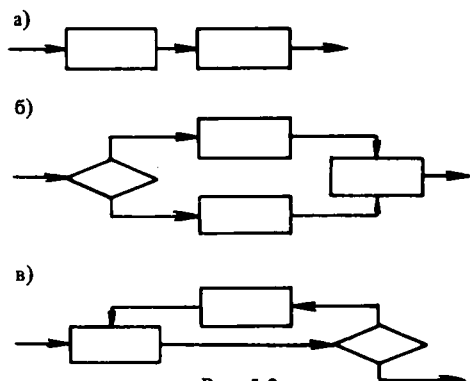


Рис. 5.2

дая структура имеет один вход и один выход. Комбинации правильных программ, полученные с использованием трех основных логических структур, также являются правильными программами. При использовании только указанных структур отпадает необходимость в безусловных переходах и метках. Применение структурного кодирования в значительной степени уменьшает сложность программ. Существенным моментом является то, что при замене любого функционального блока в схеме правильной программы на любую из трех основных логических структур программа остается в классе правильных программ. При структурном кодировании *структура последовательности* формализует тот факт, что операторы программы выполняются в порядке их появления, *структура выбора* – выбор одного из двух действий исходя из выполнения некоторого условия (оператор IF THEN, ELSE), *структура повторения* – выполнение группы операторов до тех пор, пока выполняется некоторое условие (оператор DO WHILE или итеративный DO).

Хотя теоретически возможно написать все правильные программы, используя только перечисленные структуры, можно расширить это ограничение добавлением структур, имеющих один вход и один выход и формализующих операторы DO UNTIL и CASE.

Таким образом, чтобы получить структурированную программу, следует избегать операторов GO TO. Каждый программный модуль должен состоять из последовательности операторов IF, THEN, ELSE, циклов или операторов CASE. В случае, если язык не имеет достаточного количества языковых конструкций для структурного программирования (ФОРТРАН, КОБОЛ), рекомендуется применять операторы GO TO только для переходов вперед и там, где возможно заменять их операторами CALL или PERFORM. В языках ПЛ/1 и АЛГОЛ имеется большая часть рекомендованных конструкций, что позволяет легко писать структурированные программы.

Применение принципов структурного программирования позволяет существенно уменьшить сложность проектируемых комплексов программ

и облегчить процесс их отладки за счет более детального проектирования. Основным преимуществом структурного программирования является то, что программа может быть проанализирована путем проверки структуры и становится возможным обнаруживать ошибки уже на этапе программирования, что значительно снижает затраты на процесс отладки.

§ 5.6. Отладка программ и программных комплексов

Характерной особенностью современных АСУ является значительное усложнение используемого в них программного обеспечения, которое представляет собой большие комплексы взаимодействующих программ объемом до десятков и сотен тысяч команд. С ростом надежности вычислительной техники программное обеспечение становится основным источником неправильного функционирования систем управления. В процессе его разработки допускается большое количество ошибок, стоимость обнаружения и устранения которых составляет 40–50% общих затрат на разработку и внедрение АСУ.

Технология разработки и внедрения программного обеспечения АСУ предусматривает проведение следующих этапов: постановка задач и выбор алгоритмов их решения, разработка системных и программных спецификаций задач, разработка комплекса программ, его отладка, опытная и промышленная эксплуатация.

Ошибки программного обеспечения могут быть классифицированы следующим образом: *системные ошибки*, обусловленные неправильным пониманием содержания задачи и условий ее реализации в АСУ; *алгоритмические ошибки*, связанные с некорректной формулировкой и реализацией алгоритмов программным путем; *программные ошибки* (описки, ошибки в логике, кодировании и т.д.); *технологические ошибки* в процессе подготовки документации на программу и перевода ее на машинные носители.

Основным этапом формирования разрабатываемых комплексов программ, на котором обнаруживаются, локализируются и устраняются ошибки проектирования, является отладка. Процесс отладки по уровню сложности и связи с реальными данными делится на программную и системную отладку.

Программная отладка включает проверку общей логики программы и правильности ее записи (камеральная проверка), подготовку к вводу в ЭВМ, трансляцию, индивидуальную (автономную) отладку отдельных частей программы. В процессе программной отладки обнаруживаются и устраняются различные алгоритмические, программные и технологические ошибки.

Системная отладка предназначена для проверки соответствия логической схемы комплекса программ ее функциональному назначению с использованием специально подготовленных массивов и в условиях, модели-

рующих процесс функционирования АСУ. В процессе системной отладки устраняется большинство сложных алгоритмических и системных ошибок. Общее число выявляемых в процессе системной отладки ошибок определяется ограничениями на временные и стоимостные ресурсы и исходным количеством ошибок в комплексе программ, поступающем на этап системной отладки. При достижении выбранного уровня показателей качества системной отладки комплекс программ передается в опытную эксплуатацию для проверки его функционирования с использованием реальных полноразмерных массивов и в реальном масштабе времени.

Обнаружение ошибок в программах на этапе эксплуатации системы может привести к существенным потерям времени и средств на их устранение, а в отдельных случаях к потере работоспособности системы. Поэтому с целью повышения качества комплексов программ, поступающих в эксплуатацию, необходимо улучшить планирование и организацию отладки, что позволит в конечном счете повысить эффективность функционирования АСУ за счет снижения затрат, связанных с наличием необнаруженных ошибок и получением недостоверных результатов; уменьшения времени и затрат на устранение и выявление ошибок; уменьшения вероятности наличия ошибок в программах.

Качество комплекса программ, поступающего на этап отладки, определяется в основном эффективностью методов проектирования и программирования, которые используются при его разработке. Использование модульно-иерархического построения программ с применением принципов структурного программирования позволяет существенно снизить сложность компонентов и их взаимодействия, упростить общее координирование компонентов, повысить адаптивность и модернизируемость комплекса программ, минимизировать пути распространения ошибок за счет минимизации или ограничения числа связей между модулями и, таким образом, повысить качество создаваемого программного обеспечения.

Общими задачами отладки являются генерация множества тестов, анализ работоспособности программ и их комплексов, обнаружение ошибок различных типов, диагностика и локализация ошибок, устранение ошибок, корректировка алгоритмов, программ и соответствующей документации. Известные в настоящее время модели и методы отладки ориентированы на формализацию и автоматизацию отдельных ее операций.

Рассмотрим существующие методы обнаружения и локализации ошибок в комплексах программ. Наиболее разработанными методами обнаружения ошибок на этапе отладки являются: формальное доказательство правильности программ; методы *N*-версионного программирования; методы тестирования программ.

Формальное доказательство правильности программ основано на анализе исполняемых программой действий и состоит в доказательстве того, что утверждения на выходе программы будут удовлетворяться при любом разрешенном входе. Среди методов этого класса можно выделить аксиоматическое доказательство правильности, доказательство правильности

ти с использованием аннотаций, которые представляют собой утверждения, связанные с каждым принципиальным узлом программы, доказательство корректности программных свойств.

В общем случае доказательство корректности составления программных модулей — очень сложная процедура. Для доказательства корректности программного модуля объемом около 100 инструкций требуется один человеко-месяц работы.

Основными ограничениями широкого использования методов формального доказательства правильности являются сложность формализации условий в программах, отсутствие развитых языков утверждений для программ с разнообразными типами данных, наличие ошибок в данных, большое число ограничений на семантику языков и работу компилятора и т.п. Кроме того, доказательство правильности может подтвердить правильность только действительно верных программ. Вместе с тем после проведения формального доказательства правильности нельзя утверждать, что программа всегда будет выполняться верно. Применение методов формального доказательства правильности программ в настоящее время ограничивается небольшими и несложными программами.

Методы N-версионного программирования, получившие развитие в последнее время, являются в определенном смысле аналогами методов резервирования, применяемых в технических системах. При использовании этих методов ресурсы на проведение отладки минимальны. Создается несколько версий программы, каждая из которых допускает наличие некоторого числа ошибок. Правильность функционирования набора версий достигается за счет их совместного использования. При этом используется одна из двух стратегий: резервная программа включается в работу при обнаружении ошибки; выбирается верное решение параллельно работающих программ. Согласованным функционированием программ управляет программа-супервизор, а версии программ разрабатываются различными программистами. Программные спецификации при использовании данного метода должны составляться таким образом, чтобы в различных версиях программы использовались различные структуры и приемы программирования.

Однако в настоящее время не существует методики формирования соответствующих спецификаций и оценки эффективности метода N-версионного программирования. Следует отметить, что N-версионное программирование не применимо в системах реального времени с жестким ограничением ресурсов, в задачах с большими объемами выходных данных, а также в тех задачах, где трудно определить допустимую разницу результатов, получаемых различными версиями программы.

Метод тестирования программ является основным методом отладки. Тестирование обеспечивает проверку того, что постановка задачи правильно понята системщиками и программистами, обеспечивает конкретное доказательство возможности получения решения, по крайней мере, части задачи, концентрирует внимание на небольшом числе наиболее сложных

процедур обработки данных и причинах появления ошибок. Тестирование есть процесс оценки степени готовности модулей программного обеспечения к передаче в режим эксплуатации, основанный на сборе и анализе данных о характеристике поведения программного обеспечения по результатам его реализации в известных условиях для заданных входов (тестов).

При отладке комплексов программ различают методы статистического и детерминированного тестирования.

При *статистическом тестировании* характеристики тестов описываются некоторыми распределениями или статистическими параметрами, а признаком ошибки является отклонение параметров или законов распределения выходных данных от ожидаемых. Статистическое тестирование применяется в основном для определения надежностных характеристик комплекса программ на этапе опытной эксплуатации, так как не несет в себе информации для локализации и устранения ошибок, а характеризует только их наличие.

При использовании *методов детерминированного тестирования* важнейшим вопросом на этапе отладки является выбор модели исследуемого комплекса программ и степени ее детализации. В качестве модели программы используют ее блок-схему, причем наибольшей степени детализации моделирования соответствует использование в качестве блока оператора программы, наименьшей — всей программы. В последнем случае структура программы представляет собой "черный ящик", а модель программы — отображение пространства входов X в пространство выходов Y : $x_i \xrightarrow{\pi} y_j$; $\forall x_i \in X; \forall y_j \in Y$. Степень детализации модели комплекса программ следует выбирать в зависимости от цели тестирования (тестирование основных функций, тестирование связей по информации и управлению и т.п.). При этом возникает проблема оценки полноты тестирования.

В качестве критериев оценки полноты тестирования используют максимизацию числа проверяемых при тестировании путей. В качестве ограничений обычно используют требование прохождения на множестве тестовых данных каждой вершины, ветви или пути хотя бы один раз.

Необходимо отметить, что исчерпывающее тестирование программы на всех допустимых множествах входных данных неосуществимо, так как даже для программы с двумя целочисленными входами (32-битовых входов) общее число входных наборов составляет $2^{32} \times 2^{32} = 2^{64}$. При наличии в программе циклов, число итераций которых зависит от исходных данных, количество элементов или структур, которые необходимо проверить, становится неопределенно большим. При использовании схем в качестве моделей программ не могут быть выявлены ошибки в неисполнимых операторах, используемых для описания переменных, выделения им определенных областей оперативной памяти и т.п.

Основной проблемой при проведении отладки является проблема генерации необходимых тестовых данных. Существует два подхода к их генерации: *вероятностный*, при котором входные последовательности формируются с помощью генератора случайных чисел; *детерминированный*,

при котором входные последовательности данных определяются на основе анализа структуры и характеристик комплекса программ.

Существуют вероятностные методы генерации тестовых данных, основанные на разделении входной области программы на подобласти S_j , для которых известна вероятность их появления $p(S_j)$. Множество $\{p(S_j)\}_j = \overline{1, r}$ называют *операционным профилем программы*. Из каждого множества S_j случайным образом выбираются данные для n_j тестов. Количество тестов n_j определяется в соответствии с вероятностями $p(S_j)$.

Одним из наиболее перспективных методов вероятностного тестирования является метод эффективных входных векторов. При использовании этого метода из допустимой входной области выделяется некоторая ее часть, представляющая собой список значений для каждой входной переменной x . Полученная область определения $\hat{Q}$ содержит входные векторы, называемые *эффективными*. Для каждой координаты эффективного входного вектора с помощью генератора случайных чисел выбирается некоторое значение a_β из x и строится тест $\hat{t}_i$, который реализуется программой. В процессе реализации теста $\hat{t}_i$ определяется множество дуг γ_i на графе возможных реализаций программы, которые входят в расчетный путь теста $\hat{t}_i$, и объединение множеств $\gamma = \bigcup_1^i \gamma_i$. Среди этих дуг определяют дуги,

не входившие в множество γ , полученное на предыдущем шаге. Если такие дуги есть, проверяют, не превышает ли отношение общего числа пройденных дуг ко всему множеству дуг графа возможных реализаций программы заданной величины ϵ . Если этот критерий не удовлетворяется, пытаются найти данные для непокрытых проведенными тестами дуг графа методами, аналогичными методам символического тестирования, которые описываются ниже.

Характерным представителем детерминированного подхода к генерации тестовых данных является *метод мутации программ*. Реализация этого метода основана на использовании *мутационного оператора*, который представляет собой некоторое видоизменение, трансформацию операторов исходной программы. Обычно вводится несколько мутационных операторов, каждый из которых соответствует различным классам простых ошибок языка L программы P . Мутационные операторы генерируют мутанты программы.

На практике обычно используется множество $M^*(P)$, содержащее мутанты P , полученные многократным применением мутационных операторов к P . Считается, что программа прошла мутационный тест с входным набором $T \subset D$, если: 1) программа P функционирует верно на данных T ; 2) для каждого t мутанта из $M^*(P)$ либо t не проходит на данных P , либо t эквивалентен P . Здесь D — входная область программы P . Таким образом, если программа P проходит мутационный тест, то она свободна от простых ошибок. Относительно комплексных ошибок делается предположение, что вероятность того, что композиция нескольких простых

ошибок приведет к успешному выполнению на данных T соответствующего мутанта, равна нулю.

Основными недостатками данного метода отладки являются невозможность проверки сложных программ, содержащих комбинации простых ошибок, а также ограничения на использование конструкций языков программирования. Кроме того, тестирование при таком подходе не может гарантировать отсутствия ошибок в программе, так как ее корректность устанавливается лишь для весьма ограниченного множества мутационных тестов.

Наиболее перспективными методами детерминированного подхода к генерации тестовых данных являются методы, основанные на использовании *символического исполнения программы*. При таком исполнении вместо действительных значений переменных используют их символические значения, а множество прогонов с различными входными значениями заменяется одним символическим выполнением некоторого пути программы. Последняя представляется в виде направленного графа, вершинами которого являются предложения программы, а дугами — передачи управления между ними. Символическому выполнению пути p соответствует система неравенств $B(p)$ относительно входных переменных, называемых условиями пути p . Решение t_p системы неравенств $B(p)$ определяет условия выполнения исследуемого пути, т.е. любые исходные данные, удовлетворяющие полученному решению, вызовут выполнение этого пути.

Методы, использующие символическое исполнение программы, являются промежуточными между методами формального доказательства правильности программ и методами эмпирического тестирования. Эти методы дают возможность значительно сократить число используемых тестовых прогонов и позволяют обнаруживать до 70% ошибок.

Автоматизация описанных методов отладки является необходимым условием повышения эффективности отладочных работ, так как позволяет сократить стоимость и время, необходимые на их проведение. Кроме того, автоматизация отладки комплексов программ позволяет сократить число невыявленных ошибок за счет более полной и систематической проверки и реализации значительно большего числа тестов, чем при ручной отладке; помогает освободить программиста от повторяющейся работы, позволяя сконцентрировать усилия на более сложных задачах проверки.

Ожидаемые результаты прохождения тестов в таких системах определяют вручную, автоматизируется лишь генерация тестов и сравнительный анализ ожидаемых значений и результатов тестирования. Наиболее сложным вопросом при автоматизации тестирования является определение его *достаточности* и *полноты*. В современных системах автоматического тестирования критерием останова является число проверенных инструкций, условных переходов, путей программ и т.п. Наиболее перспективным для определения достаточности тестирования представляется подход, основанный на минимизации суммарных потерь на тестирование и последствий от оставшихся в программном обеспечении ошибок.

Рассмотренные методы и системы автоматизации тестирования не обеспечивают *локализации ошибок*, т.е. точного установления мест их расположения в комплексе программ. Некоторое исключение составляют методы, использующие при генерации тестовых данных символическое исполнение программы, в результате которого формируются сведения о путях обработки информации, на которых результаты тестов отличаются от ожидаемых. Но и в этих случаях процесс локализации и устранения ошибок остается наиболее сложным и трудноформализуемым процессом системной отладки, который в настоящее время проводится в основном вручную.

На первом этапе процесса локализации обычно выделяются действия, выполняемые программой неверно, с целью выработки одной или нескольких версий о причине ошибки. На следующем этапе разрабатывают план проверки гипотез и создают соответствующие тесты для этой проверки.

Известные методы и модели формализации процесса локализации ошибок обеспечивают лишь автоматизацию процедур выделения областей локализации ошибок. При этом используется диагностическая модель программы, задаваемая совокупностью $\langle X, V, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4 \rangle$ и представляемая графом, где X — множество вершин, соответствующих переменным комплекса программ; V — множество дуг, соответствующее линейным участкам команд; Γ_1 — отображение множества V в V ; Γ_2 и Γ_3 — отображения множества X в V и V в X соответственно; Γ_4 — отображение множества X в X . Параметры тестового набора задаются точкой входа v_b , точкой выхода v_l , множеством вершин исходных данных X_b и множеством вершин результатов X_l . Кроме того, задана булева функция правильности результатов $F(x_j)$ для $\forall x_j \in X_b$ и правильности точки выхода $F(v_b)$.

Решение задачи определения области локализации ошибок сводится к выделению подграфа, связывающего вершины v_l и v_b . Дальнейшие операции по установлению места расположения ошибки проводят вручную. Недостатком рассмотренного метода локализации является то обстоятельство, что выделенные подграфы локализации ошибки имеют, как правило, весьма большие размеры и в предельных случаях включают все элементы диагностической модели. Это объясняется тем, что в данном методе не учитывается конкретное содержание тестовых наборов и их направленность на выявление определенных свойств проверяемой программы.

Значительную помощь при локализации ошибок в программах оказывает рациональное размещение дампов, т.е. команд или подпрограмм печати, печатающих данные для распознавания и обнаружения ошибок. Например, при размещении дампов для проверки правильности организации циклов места их расположения целесообразно выбирать таким образом, чтобы они идентифицировали максимально возможное число циклов, а число точек, в которых расположены дампы, было при этом минимальным. Задача формулируется с использованием диагностической модели программы $\langle X, V, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4 \rangle$ как задача о покрытии множества дуг, образующих циклы, таким образом, чтобы число дуг покрывающего набора

было минимальным и не было ни одного цикла, для которого хотя бы одна дуга не вошла в набор.

Процесс локализации ошибок на этапе отладки является наиболее сложным для автоматизации. Системы автоматизации процессов локализации ошибок, известные в настоящее время, позволяют организовать прерывание программы, проводить выдачу или изменение значений переменных, моментальные снимки путей при реализации программы и т.п.

Одним из наиболее перспективных методов локализации ошибок по результатам тестирования является чтение программы в обратном направлении. Этот метод позволяет формализовать процессы определения областей локализации ошибок и определения места расположения ошибки в комплексе программ с применением аппарата специальных сетей Петри. Сети строятся на основании графовой модели блок-схемы комплекса программ, на которой отображены информационные связи. Введение специальной системы маркировки сетей Петри, отображающей вершины реализованного маршрута обработки информации и ошибочные результаты теста, позволяет на основе построения деревьев достижимости автоматически определить область локализации и место расположения ошибки в комплексе программ.

Большое число, разнообразие типов и трудоемкость обнаружения ошибок в сложных комплексах программ АСУ обуславливают повышенные требования к этапу системной отладки, целью которой является проверка выполнения комплексом программ всех функций, предусмотренных техническим проектом системы, и установления соответствия полученных результатов поставленным целям и задачам.

Системная отладка проводится путем последовательного сопряжения программных модулей. При этом выбор последовательности объединения модулей является одним из важнейших решений, поскольку он определяет сложность и трудоемкость тестирования, типы необходимых инструментов тестирования, а также тщательность и экономичность всего этапа системной отладки. Особенностью этого этапа является необходимость выявления не только программных, но и сложных алгоритмических ошибок. Такие ошибки возникают при объединении отдельных программных модулей в комплексы из-за неправильной организации передач управления и информации между модулями, отсутствия достоверной информации о необходимых временных ресурсах ЭВМ и ресурсах памяти для отдельных задач и их совокупностей, неправильного распределения функций между модулями комплекса и др.

Основными методами, используемыми на этапе системной отладки, являются методы анализа и тестирования.

Методы анализа применяют на начальном и завершающем этапах системной отладки. На начальном этапе анализируется текст программы, строится ее модель, определяются некоторые характеристики (сложность, избыточность), составляется план проведения отладочных работ. На завершающем этапе анализируется эффективность комплекса программ, полученного в процессе системной отладки.

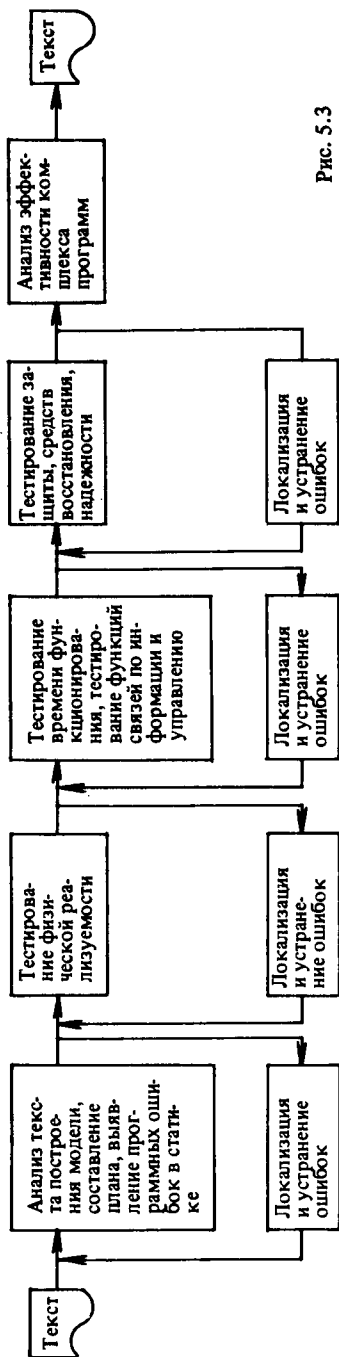


Рис. 5.3

Методы тестирования используют для оценки степени готовности комплекса программ к передаче в режим эксплуатации; в основе методов лежит сбор и анализ данных о характеристиках поведения комплекса по результатам его реализации в известных условиях при заданных входах (тестах).

На этапе системной отладки тестирование обеспечивает исследование причин различных аномалий в поведении программы, оценку времени, объема памяти и других ресурсов, необходимых на различных этапах реализации программы, оценку эффективности систем защиты и средств восстановления, оценку поведения программы при неблагоприятных внешних условиях и недостоверных исходных данных и т.п.

Анализ целей, технологии, моделей и методов проведения системной отладки позволяет выделить ряд этапов отладочных работ и последовательность их проведения (рис. 5.3).

На первом этапе на основании анализа текста программы выявляются программные ошибки, возникающие при объединении программных модулей в комплекс или не выявленные в процессе программной отладки (в том числе некоторые ошибки заикливания, лишние и тупиковые операторы и т.п.). Эти ошибки обнаруживаются без применения ЭВМ в статике либо при наличии соответствующего программного обеспечения на основании информации, полученной при трансляции с помощью ЭВМ. Кроме того, на этом этапе строится модель программного комплекса. В качестве моделей, используемых при составлении тестов и локализации ошибок, применяют графовые модели: детальные блок-схемы комплекса программ и его частей, детальные блок-схемы отдельных программных модулей.

Составление плана системной отладки предусматривает выбор страте-

гии проведения отладочных работ, т.е. определение состава и последовательности отлаживаемых подструктур, обеспечивающих оптимальные временные или стоимостные характеристики системной отладки. В настоящее время применяются различные стратегии системной отладки: "сверху-вниз", "снизу — вверх" и их различные модификации. Реализация любой стратегии требует разбиения комплекса программ на части, автономной отладки полученных частей, последовательного объединения автономно отлаженных частей и их совместной отладки вплоть до исходного комплекса программ.

Множество стратегий системной отладки определяется множеством возможных вариантов разбиения и объединения структурного графа G , вершинами которого являются программные модули комплекса, а дугами — связи по управлению между ними. Любая mn -стратегия задает множество подграфов разбиения графа $GP^m = \{P_1, \dots, P_v, \dots, P_n^m\}$ и последовательность объединения этих подграфов до исходной графовой структуры $\hat{P}^{mn} = \{\hat{P}_1^{mn}, \dots, \hat{P}_k^{mn}, \dots, \hat{P}_N^{mn}\}$. При этом задача выбора оптимальной стратегии системной отладки по критерию минимума времени на ее проведение формулируется следующим образом:

найти

$$\min \sum_{mn} (T_{mn}^{\text{разб}} + T_{mn}^{\text{об'ед}}) x_{mn}$$

при ограничении

$$\sum_{mn} (C_{mn}^{\text{разб}} + C_{mn}^{\text{об'ед}}) x_{mn} \leq C_{\text{зад}},$$

где

$$x_{mn} = \begin{cases} 1, & \text{если выбирается } mn\text{-стратегия системной отладки,} \\ 0 & \text{— в противном случае;} \end{cases}$$

$T_{mn}^{\text{разб}}, C_{mn}^{\text{разб}}$ — время и стоимость разбиения при использовании mn -стратегии; $T_{mn}^{\text{об'ед}}, C_{mn}^{\text{об'ед}}$ — время и стоимость объединения при использовании mn -стратегии; $C_{\text{зад}}$ — допустимые затраты на проведение системной отладки.

На втором этапе системной отладки выявляются ошибки, связанные с просчетами в использовании ресурсов вычислительной техники по памяти (физической реализуемости комплекса программ). Эти ошибки обнаруживают путем анализа спецификаций комплекса программ (использование внешних запоминающих устройств), а также на основании информации компиляторов об объемах оперативной памяти, необходимой для размещения комплекса программ в процессе его функционирования.

На третьем этапе системной отладки на основании анализа результатов тестирования времени функционирования, тестирования согласованности по управлению и информации, тестирования функций

выявляются и устраняются ошибки, связанные с просчетами в использовании ресурсов вычислительной техники по времени, ошибки сопряжений и ошибки в выполнении основных функций комплекса программ.

На четвертом этапе осуществляется проверка требований по обеспечению защиты от несанкционированного доступа (тестирование защиты), по обеспечению сохранности программных модулей и информационных массивов (тестирование средств восстановления), по обеспечению надежных характеристик (тестирование надежности). Кроме того, на этом этапе может быть проведена проверка поведения системы при неблагоприятных внешних условиях (тестирование конфликтных ситуаций) и недостоверных исходных данных.

На пятом этапе проводится анализ эффективности комплекса программ, полученного в процессе системной отладки. При этом обычно используется подход, когда эффективность комплекса программ характеризуется суммарными затратами на его проектирование и отладку.

Некоторые ошибки сопряжений программных модулей могут быть выявлены в процессе компиляции комплекса программ. По результатам тестирования комплекса программ на этапе системной отладки осуществляется предварительная оценка характеристик эффективности защиты, средств восстановления, надежности; полная оценка этих характеристик может быть проведена лишь на этапе опытной эксплуатации на основе статистических характеристик тестирования на полноразмерных массивах исходных данных.

Одним из основных направлений сокращения трудоемкости и времени этапа подготовки программ является *автоматизация программирования*. Системы автоматизации программирования представляют собой совокупность алгоритмических и программных средств, предназначенных для автоматизации процесса разработки программ, т.е. перевода исходных алгоритмов на машинный язык. В общем случае система автоматизации программирования состоит из одного или нескольких входных языков (проблемно- и машинно-ориентированных), системы трансляции программ с этих языков, системы структурного контроля и системы выпуска технической документации. Система автоматизации программирования обеспечивает получение синтаксически, семантически и структурно корректных программ, а также технической документации на программы и их характеристики.

Система автоматизации программирования решает следующие задачи: контроля входного текста; оптимизации программы; использования библиотеки стандартных программ; выпуска документации. Система предусматривает синтаксический, семантический и структурный контроль комплекса программ.

Задачей *синтаксического контроля* является проверка входного текста на соответствие синтаксису языка, заданному в его формальном описании. Синтаксический контроль обеспечивает выдачу информации о месте ошибки в форме, облегчающей программисту определение ошибочных кон-

струкций и причин ошибок, а также информацию о содержании ошибок и способе их устранения.

Задачей *семантического контроля* является проверка правильности применения языковых конструкций. Например, семантический контроль проверяет допустимость применения в операторах переменных с заданными в их описаниях характеристиками (тип, размерность).

Структурный контроль состоит в проверке правильности последовательностей языковых конструкций, отсутствия заикливания и тупиков, использования оперативной памяти, сопряжений программных модулей.

Оптимизация программ включает оптимизацию распределения памяти, оптимизацию выражений, оптимизацию циклов.

Оптимизация распределения памяти предусматривает размещение переменных, имеющих пересекающиеся области действия, в одних и тех же ячейках памяти. Таким образом решается задача оптимизации использования индексных регистров. Если в программе используется подряд несколько одинаковых модификаций, то многократное вычисление и засылка на индексный регистр для сокращения времени счета исключается.

Оптимизация выражений состоит в однократном вычислении значений выражения, запоминании его в рабочем регистре и дальнейшем многократном использовании этого значения в процессе вычислений.

Оптимизация циклов, сводится к вынесению за пределы цикла выражений, не зависящих от параметров цикла.

Использование библиотеки стандартных программ система автоматизации программирования обеспечивает посредством обращения к ней с применением различных языковых и программных средств.

Система *выпуска документации* осуществляет редактирование и компиляцию исходных данных и подготавливает информацию для распечатки различных документов: о программах, константах и переменных.

Отладка в КОБОЛе. Средства отладки КОБОЛа предназначены для обеспечения возможностей как слежения за логическими шагами работающей программы, так и вывода значений определенных данных в процессе вычислений.

Средства для осуществления отладки, предоставляемые КОБОЛОм программисту, реализуются в виде строк отладки и секций отладки. *Строки отладки* могут быть любыми статьями или операторами КОБОЛа, добавляемыми в программу для наблюдения за промежуточными значениями и операциями во время выполнения программы. Такие строки будут исключены из процесса выполнения программы после того, как она будет отлажена и готова для использования. При этом необязательно фактически удалять их из программы. Они просто не будут компилироваться при окончательной компиляции готовой программы. *Секции отладки* – это целые секции, включаемые в секции декларатив разделов процедур. Строки отладки выполняются каждый раз, когда им естественным образом передается управление. Секции отладки выполняются, как только в работающей программе происходит обращение к определенным идентификаторам файлов или процедур. Секции отладки также не включаются при окончательной компиляции готовой программы.

Из секций отладки возможны обращения к специальному *данному* с именем DEBUG-ITEM (данные-отладки). Никакие внешние по отношению к секциям отладки операторы, включая операторы строк отладки, не могут обращаться к данному

DEBUG-ITEM, а само это данное не должно описываться в разделе данных, оно предусматривается самой системой, а не программой. Значения DEBUG-ITEM устанавливаются автоматически при передаче управления секции отладки. Они содержат: 1) номер, идентифицирующий конкретный оператор исходной программы; 2) идентификатор — имя файла или имя процедуры, к которым произошло обращение, вызвавшее передачу управления секции отладки; 3) значения индексов или имена индексов; 4) информацию о значениях, связанных с причиной передачи управления секции отладки.

Автоматическая передача управления секции отладки и установка значений в данное DEBUG-ITEM происходит следующим образом.

1. Для идентификатора — непосредственно после любого выполняемого оператора, за исключением оператора WRITE, или непосредственно перед выполнением оператора WRITE.

2. Для имени процедуры — непосредственно перед операторами GO TO, PERFORM, SORT или перед непосредственной передачей управления, или непосредственно после оператора ALTER.

3. Для имени файла — после выполнения операторов OPEN или CLOSE и любого выполнения оператора READ, не приводящего к выполнению связанных с этим оператором повелительных операторов.

Отладка в ПЛ/1. Естественный порядок выполнения программы может быть нарушен прерыванием, возникшим вследствие ошибок в программе и в некоторых специальных случаях. Состояния прерывания в ПЛ/1 называют исключительными. Типичным исключительным состоянием присвоены стандартные имена состояний, указывающие причину прерывания. Программист может запретить или, наоборот, разрешить обработку определенных исключительных состояний.

Исключительные состояния, установленные в ПЛ/1, охватывают не все причины прерываний. Для прерываний, которым не сопоставлены специальные исключительные состояния, создается системное состояние ERROR (ошибка). В ряде случаев одно и то же исключительное состояние может соответствовать разным причинам прерываний. Для уточнения причины прерывания служит встроенная функция ON CODE. Значение этой функции — число в форме BINARY FIXED (15,0), указывающее код прерывания. Обычно функцию ON CODE используют в утверждении ON ERROR для выдачи на печать кода прерываний, однако ее можно использовать и для определения варианта действий по исправлению ошибки или ликвидации ее последствий.

Специально для отладки программ служат три исключительных состояния: CHECK (список имен), STRING RANGE и SUBSCRIPT RANGE. Эти состояния по умолчанию выключены, но их можно включить префиксами состояний.

"Список имен" в префиксе CHECK может содержать неиндексированные имена элементарных и агрегатных переменных, метки и имена входов, разделенные запятыми.

Состояние CHECK создается, если в сфере действия CHECK — префикса для имени, указанного в списке имен этого префикса — произошло одно из следующих событий: изменилось значение переменной (прерывание возникает после изменения значения), встретилось утверждение с указанной меткой или произошел вызов процедуры с указанным именем входа (прерывание возникает перед выполнением помеченного утверждения или перед вызовом процедуры).

В ON-единице для состояния CHECK можно предусмотреть контроль присваемого значения или "перехват" выполнения помеченного утверждения. В частности, "перехват" передачи управления и вызов процедур применяется при автономной отладке отдельных модулей, которые в дальнейшем объединяются в единую программу.

При нормальном возврате из ON-единицы выполнение программы продолжается с точки прерывания, но, используя утверждение GO TO ON-единицы, можно перейти в любую другую точку программы.

Состояние CHECK обычно используют для печати трассы программы (последовательности меток выполняемых утверждений и имен входов выполняемых процедур) и значений присваиваемых переменным.

Состояние STRING RANGE возникает, если в сфере действия префикса (STRING RANGE) обнаруживаются неправильные аргументы встроенной функции SUBSTR. В ON-единице можно исправить неправильную длину строки, после этого функция SUBSTR вычисляется повторно.

Состояние SUBSCRIPT RANGE возникает, если в сфере действия префикса (SUBSCRIPT RANGE) значение индекса элемента массива оказалось вне установленных для него границ.

Употребление префиксов отладочных состояний замедляет выполнение программы и увеличивает потребность в памяти, поэтому, завершив отладку, эти префиксы целесообразно удалить.

Отладка в ФОРТРАНе. Важнейшая системозависимая особенность языка ФОРТРАН-IV в ОС ЕС заключается в том, что фортранные программы пользователя не содержат элементов, зависящих от ОС непосредственно. Такая зависимость может проявляться в программах лишь косвенно — через ограничения на допустимые программисту размеры внешней и внутренней памяти, через эксплуатационные характеристики внешних устройств, через соглашения об организации оверлейных программ с запланированным перекрытием и т.д. Прямая зависимость от ОС вынесена из фортрановских программ в сопровождающие их программы, написанные на языке управления заданиями.

Отладка в АСЕМБЛЕРе. Наиболее простым средством получения промежуточных данных о работе ассемблеровской программы является использование макрокоманды SNAP. Эта команда обладает широкими возможностями. Кроме распечатки содержимого регистров и полей памяти в состав выводимых данных могут быть включены слово состояния программы (PSW), информация о связях, содержимое общей области памяти, управляющие поля супервизора и т.д.

Для более серьезной отладки программ, написанных на АСЕМБЛЕРе, в составе ОС ЕС имеется специальный транслятор — *тестран*. Язык отладочных директив тестрана предусматривает не только возможность выполнить те или иные отладочные функции в заданных точках отлаживаемой программы. В задании на отладку могут быть определены логические и арифметические переменные, значения которых можно изменять в зависимости от поведения отлаживаемой программы. Эти переменные могут быть подвергнуты анализу, по результатам которого можно изменить порядок выполнения отладочных акций.

Тестран позволяет сопровождать распечатки, получаемые в процессе отладки, комментариями, а также собрать информацию о логике выполнения заданных участков отлаживаемой программы. Для этой цели используются модификации директивы TRACE, которые позволяют накапливать сведения о состоявшихся переходах на участке отлаживаемой программы, ограниченном специальными метками; запоминать переходы на подпрограммы, состоявшие в указанном сегменте; фиксировать команды, которые могут изменить содержимое области основной памяти, заданной начальными и конечными адресами; отменять трассировку отдельных или всех участков.

§ 5.7. Оценка качества программных комплексов

Качество комплекса программ на каждом этапе системной отладки целесообразно оценивать группами показателей, характеризующих его с точки зрения наличия ошибок, а также трудоемкости их обнаружения и устранения.

Рассмотрим некоторые из возможных показателей качества по группам, используемые при системной отладке в соответствии с рис. 5.3. К первой группе относятся показатели сложности и избыточности; ко второй — показатель физической реализуемости комплекса программ; к третьей — показатели времени функционирования, функциональной корректности, согласованности по управлению и согласованности по информации; к четвертой — показатели, характеризующие надежность комплекса программ, эффективность средств восстановления, эффективность системы защиты; к пятой — показатель эффективности комплекса программ, характеризующий затраты на проектирование и отладку комплекса программ.

Следует отметить, что хотя основное значение системной отладки состоит в проверке правильности работы всего комплекса программ, начинать ее надо с проверки работы каждого модуля. Поэтому выделяют следующие уровни проверок: единичный модуль, подсистему, внутренний интерфейс, внешний интерфейс, комплекс программ. Уровень подсистем целесообразно выделять при проверке сложных систем программного обеспечения. Обычно в отдельные подсистемы группируются модули по функциональному признаку.

Показатели качества оцениваются как для всего комплекса программ, так и для его частей (отдельных программных модулей и их совокупностей, подлежащих совместной системной отладке в соответствии с планом отладочных работ) и являются векторными величинами.

Среди количественных оценок приведенных показателей качества наиболее широко используемым является показатель сложности.

Сложность комплекса программ включает структурную и статистическую сложность.

Для анализа *структурной сложности* важнейшее значение имеет степень взаимосвязи модулей. При этом структурная сложность может быть оценена:

а) *по степени отличия структуры от древовидной* $\rightarrow C = A - G$, где G — число связей, образующих древовидную структуру; A — реальное число связей между модулями комплекса;

б) *сложностью связей по управлению* $\rightarrow \Lambda = \sum_i \Lambda_i = \sum_i \sum_j (\vec{a}_{ij} + \overleftarrow{a}_{ij})$, где $\vec{a}_{ij}$ — управляющие связи i -го модуля, которые вызывают его для исполнения; $\overleftarrow{a}_{ij}$ — связи, посредством которых i -й модуль вызывает другие модули; Λ_i — сложность управляющих связей i -го модуля;

в) *сложностью связей по информации*: $\Phi = \sum_i \Phi_i = \sum_i \sum_k (\vec{b}_{ik} + \overleftarrow{b}_{ik})$, где

$\vec{b}_{ik}$ — количество имен переменных на входе i -го модуля, необходимых для его нормального функционирования; $\overleftarrow{b}_{ik}$ — количество имен переменных, подготавливаемых i -м модулем; Φ_i — сложность информационных связей i -го модуля.

Для оценки *статистической сложности* комплекса программ используется количество составляющих его программных модулей N .

Сложность программных модулей также включает понятия структурной и статистической сложности.

Структурная сложность программных модулей, как правило, определяется на основании учета количества маршрутов, исполняемых программой, и числа условных операторов: $\bar{\epsilon}_j = \sum_i^{M_j} \epsilon_{ij}$, где ϵ_{ij} — количество условий, определяющих i -й маршрут j -го модуля; M_j — количество маршрутов j -го модуля.

Статистическая сложность программных модулей обычно определяется путем подсчета количества операторов N_{1j} и операндов $N_{2j} - N_{cj} = N_{1j} + N_{2j}$. Таким образом показатель сложности можно представить вектором $\bar{C} = \{C, \Lambda, \Phi, \{\epsilon_j\}, N_{ij}\}$; компоненты вектора сложности служат ориентирами при планировании отладочных работ и сравнении различных версий комплекса программ.

Показатель избыточности характеризует наличие нереализуемых структурных элементов в комплексе программ и его программных модулях.

Избыточность комплекса программ определяется количеством нереализуемых программных модулей I_1 и количеством неиспользуемых информационных элементов I_2 .

Избыточность программных модулей определяется количеством лишних операторов и тупиков в программах $\{I_{3j}\}$; здесь j — индекс программного модуля.

Показатель избыточности представляется вектором; $\bar{I} = \{I_1, I_2, \{I_{3j}\}\}$.

Показатель физической реализуемости характеризует ресурсы вычислительной техники по памяти, необходимые для функционирования комплекса программ. Он включает: объемы оперативной памяти, необходимые комплексу программ при различных режимах его работы $\{V_m\}$; корректность вызова управляющей программой совокупностей программных модулей, соответствующих различным режимам работы $\{P_m\}$, объемы внешней памяти по различным типам запоминающих устройств $\{W_k\}$.

Корректность вызова в оперативную память группы модулей, соответствующих m -му режиму работы комплекса программ, может быть определена на основе анализа спецификаций, а соответствующие компоненты физической реализуемости представлены булевыми величинами:

$$P_m = \begin{cases} 1, & \text{если управляющая программа осуществляет вызов в оперативную память группы модулей для } m\text{-го режима работы комплекса программ в соответствии с требованиями спецификаций;} \\ 0 & \text{— в противном случае.} \end{cases}$$

Таким образом, показатель физической реализуемости есть вектор:

$$\bar{P}_p = \{\{V_m\}, \{P_m\}, \{W_k\}\}.$$

Функциональная корректность комплекса программ есть вектор $\bar{K}_\Phi = \{f_1, \dots, f_a\}$, компоненты которого характеризуют корректность программной реализации функций комплекса программ относительно требований спецификаций. Здесь Q — количество функций, выполняемых комплексом в соответствии с системными спецификациями. Компоненты вектора $\bar{K}_\Phi$ могут быть представлены булевыми величинами и определяются следующим образом:

$$f_i = \begin{cases} 1, & \text{если функция, соответствующая } i\text{-й компоненте показателя,} \\ & \text{выполняется в соответствии с требованиями спецификаций;} \\ 0 & \text{— в противном случае.} \end{cases}$$

Функциональная корректность отдельных программных модулей определяется аналогичным образом, соответствующие компоненты могут быть включены в показатель функциональной корректности.

Показатель времени функционирования комплекса программ включает средние времена функционирования основных режимов работы комплекса $\{T_m\}$ и средние времена функционирования отдельных программных модулей и их совокупностей $\{t_r\}$: $\bar{B}_\Phi = \{\{T_m\}, \{t_r\}\}$.

Согласованность по управлению включает согласованность по управлению комплекса программ и согласованность по управлению его программных модулей.

Согласованность по управлению комплекса программ определяет корректность реализации основных последовательностей программных модулей относительно требований спецификаций и корректность множества сопряжений между парами модулей. Она может быть охарактеризована булевыми величинами исходя из следующих условий:

$$y_i = \begin{cases} 1, & \text{если } i\text{-я последовательность реализации программных модулей} \\ & \text{выполняется в соответствии с требованиями спецификаций;} \\ 0 & \text{— в противном случае,} \end{cases}$$

или

$$y_i = \begin{cases} 1, & \text{если } i\text{-е сопряжение выполняется без ошибок;} \\ 0 & \text{— в противном случае.} \end{cases}$$

Ошибки в сопряжениях модулей — y_i — обычно сводятся к следующим: отсутствует вызов необходимого модуля; вызывается модуль, не подлежащий исполнению; управление передается в точку, не предназначенную для возврата управления.

Согласованность по управлению программных модулей может быть оценена отношением числа проверенных маршрутов к общему числу маршрутов модуля.

Согласованность по информации включает согласованность по информации комплекса программ и согласованность по информации программных модулей.

Согласованность по информации комплекса программ определяет корректность связей между программными модулями и может быть оценена отношением числа проверенных маршрутов обработки информации к общему числу маршрутов, реализующих информационную связь по k -му информационному элементу.

Согласованность по информации программных модулей определяет корректность информационных связей по внутренним информационным элементам и оценивается аналогично согласованности по информации комплекса программ.

Показатель, характеризующий надежность комплекса программ, может быть описан вектором $\vec{N} = \{O, \lambda, P\}$, компонентами которого являются: O — общее количество выявленных ошибок, λ — интенсивность появления невыявленных ошибок, P — вероятность безотказной работы в течение заданного интервала времени. Точное определение количества ошибок в программе и вероятностей появления ошибок различных типов прямыми методами измерений получить невозможно, существуют лишь косвенные пути статистической оценки этих величин в процессе отладки и опытной эксплуатации комплекса программ. Эти оценки основаны на математических моделях, предусматривающих жесткую корреляцию приведенных выше компонентов вектора надежности. Надежные оценки хорошо согласуются со статистикой тестирования программ и используются для предсказания вероятности безотказной работы комплекса программ заданный промежуток времени:

$$P(t) = \exp\left(-\int_0^t \lambda(t) dt\right).$$

Показатель, характеризующий эффективность системы защиты, есть вектор $\vec{Z} = \{P_{н.д}, C_{н.д}\}$, компонентами которого являются: $P_{н.д}$ — вероятность несанкционированного доступа к модулям и информационным элементам комплекса программ; $C_{н.д}$ — потери от "взлома" системы защиты.

Показатель, характеризующий эффективность средств восстановления, есть вектор $\vec{B} = \{K_r, \sigma, T_{р.з}, P_{р.з}\}$, компонентами которого являются: K_r — коэффициент готовности системы; σ — вероятность сохранности информации в системе; $T_{р.з}$ — среднее время решения задачи; $P_{р.з}$ — вероятность успешного решения задачи. Перечисленные компоненты используются для оценки эффективности оперативного (K_r, σ) и восстановительного ($T_{р.з}, P_{р.з}$) резервирования программных модулей и информационных массивов.

Показатель эффективности комплекса программ, полученного в процессе системной отладки, есть вектор $\vec{Z} = \{C_{пр}, C_{п.о}, C_{с.о}\}$, компонентами которого являются: $C_{пр}$ — затраты на проектирование комплекса программ; $C_{п.о}$ — затраты на программную отладку; $C_{с.о}$ — затраты на системную отладку. В рамках разработанной методологии затраты на проведение системной отладки определяются выражением $C_{с.о} = \sum_{j=1}^n C_j +$

$+ C_{н.о}(P_{н.о})$, где C_j — затраты на проведение j -го этапа системной отладки;

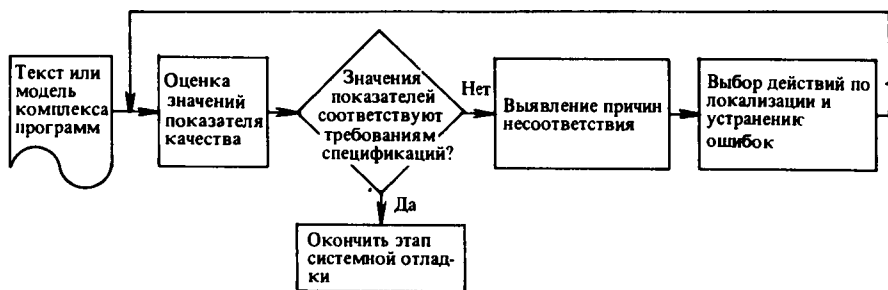


Рис. 5.4

$C_{н.о}(P_{н.о})$ — потери от необнаруженных ошибок; $P_{н.о}$ — вероятность появления необнаруженной ошибки после проведения системной отладки.

Таким образом, *качество комплекса программ*, получаемого в процессе системной отладки, может быть описано совокупностью рассмотренных выше векторных показателей.

Каждый этап системной отладки (кроме анализа эффективности комплекса программ) с использованием приведенных групп показателей качества может быть представлен как итерационный процесс, состоящий на каждой итерации из оценки фактических значений показателей качества, сравнения их с требованиями спецификаций, выбора действий по установлению причин выявленных несоответствий, их локализации и устранению (рис. 5.4).

Оценка значений показателей качества проводится на основании анализа текста комплекса программ и его различных графовых моделей и на основании результатов тестирования.

Спецификации на комплекс программ служат основным эталоном для определения достаточности проведенных отладочных работ и должны содержать необходимую информацию о требуемых значениях показателей качества, которые необходимо обеспечить в процессе системной отладки. В спецификациях на комплекс программ должны быть сформулированы следующие требования: допустимые значения компонентов векторов сложности и избыточности, характеристики выполняемых функций в терминах их входов и выходов, максимально допустимый объем оперативной памяти, который может быть отведен исследуемому комплексу в процессе его функционирования, максимально допустимые объемы внешней памяти для размещения информационных элементов комплекса, максимально допустимые времена функционирования комплекса программ и его структурных элементов, основные последовательности реализации программных модулей, характеристики связей по информации, характеристики надежности, системы защиты и средств восстановления, эффективность комплекса программ. Несоответствие какого-либо компонента показателей качества требованиям спецификаций является либо следствием недостатков проектирования или программирования, либо следствием наличия ошибок.

В процессе исследования возможных вариантов повышения значений показателей качества комплекса программ может быть обнаружено, что ни один из них не позволяет обеспечить соответствие показателя требованиям спецификаций. В таком случае следует провести корректировку требований спецификаций с учетом реальных возможностей решения исследуемой задачи.

На основании сравнения значений компонентов и показателей качества с требованиями спецификаций может быть оценена общая отлаженность комплекса программ:

$$\hat{O} = \sum_i \frac{\sum_j p_{ij}}{n_i} / I,$$

где n_i — количество компонентов i -го показателя качества; I — количество рассматриваемых показателей качества;

$$p_{ij} = \begin{cases} 1, & \text{если } j\text{-й компонент } i\text{-го показателя качества соответствует} \\ & \text{требованиям спецификаций;} \\ 0 & \text{— в противном случае.} \end{cases}$$

Выбор действий по локализации и устранению несоответствий требованиям спецификаций состоит в выборе множества показателей качества, для которых будет проведена локализация и устранение соответствующих ошибок, а также выбор для этого множества последовательности действий, которая обеспечит локализацию и устранение ошибок с минимальными затратами времени и средств.

После выполнения действий по локализации и устранению несоответствий значений показателей качества требованиям спецификаций осуществляется переход к новой итерации системной отладки по исследуемой группе показателей качества.

Таким образом, процесс выявления ошибок в комплексе программ (в том числе и сложных алгоритмических) с использованием системы показателей качества может быть в достаточной степени формализован, а его основные операции — автоматизированы.

В соответствии с вышеизложенным процесс отладки сложных комплексов программ должен включать такую последовательность действий.

1. Программная отладка: использование возможностей трансляторов и компиляторов для автоматического выявления синтаксических и других программных ошибок.

2. Составление плана отладки (выбор стратегии).

3. Автономная отладка программных модулей в соответствии с выбранной стратегией отладочных работ.

- 3.1. Проверка физической реализуемости исследуемой совокупности программных модулей: а) оценка объемов оперативной и внешней памяти, необходимой для размещения комплекса программ (физической реализуемости); б) исследование возможностей сокращения объема оперативной памяти на размещение комплекса программ при превышении им допусти-

мой величины (перераспределение памяти комплекса программ за счет использования общих областей для размещения информационных элементов, сокращения блоков обмена с внешней памятью, организация оверлейной структуры и т.п.); в) проверка реентерабельности (самовосстанавливаемости) программных модулей путем проверки получения устойчивых результатов при многократном к ним обращении; г) локализация и исправление ошибок, связанных с использованием ресурсов вычислительной системы по памяти.

3.2. Проверка времени функционирования: а) оценка времени функционирования методами тестирования; б) исследование возможностей сокращения времени функционирования при превышении допустимых значений (распараллеливание процесса вычислений, организация параллельного выполнения операций ввода – вывода и вычислительных операций и т.п.); в) локализация и устранение обнаруженных ошибок.

3.3. Проверка правильности реализации основных функций, сопряжений программных модулей, информационных связей: а) определение маршрутов обработки информации, характеризующих основные функции, сопряжения, информационные связи; б) генерация тестовых данных для полученного набора маршрутов; в) проверка правильности выполнения маршрутов по результатам тестирования и оценка компонентов функциональной корректности, согласованности по управлению и информации; г) локализация и устранение обнаруженных ошибок.

4. Связная (комплексная) отладка автономно отлаженных совокупностей программных модулей и всего комплекса программ.

4.1. Проверка физической реализуемости.

4.2. Проверка времени функционирования.

4.3. Проверка правильности реализации основных функций, сопряжений, информационных связей.

5. Проверка требований по обеспечению защиты от несанкционированного доступа (тесты, основанные на попытках "взлома" защиты), по обеспечению сохранности программных модулей и информационных массивов (тестирование средств восстановления), по обеспечению надежностных характеристик (статистическое тестирование); проверка поведения комплекса программ при неблагоприятных внешних условиях (тестирование на больших объемах исходных данных, при пиковых нагрузках и т.п.) и недостоверных исходных данных.

6. Оценка эффективности комплекса программ по затратам на его проектирование и отладку. Оформление программной документации.

В программную документацию включаются:

1) спецификация, определяющая состав комплекса программ и документацию;

2) общее описание комплекса программ, содержащее сведения о назначении программы, области применения, используемых методах, ограничениях на применение;

3) руководство программиста, содержащее описание используемых

средств программирования (включая сведения об используемой версии ОС) и требования к конфигурации технических средств, описание интерфейсов, имен программных модулей, логической и физической структуры базы данных, описание используемых пакетов прикладных программ и средств диагностики и т.п.;

4) руководство оператора, содержащее описание внешних носителей, правила организации программ на внешних носителях, состав сообщений оператору, описание входного потока заданий и т.п.;

5) текст комплекса программ, содержащий тексты программных модулей с комментариями, данные контрольного примера, управляющие операторы подготовки и запуска контрольного примера, значения основных показателей качества комплекса программ;

6) описание контрольного примера, содержащее описание проверяемых функций и параметров, входной информации, результаты прогона по данным контрольного примера, управляющих операторов подготовки и запуска контрольного примера.

После завершения отладки комплекс программ и его программная документация передаются в эксплуатацию.

§ 5.8. Организация и планирование процесса программирования

Подготовленные для программирования материалы — алгоритмы и укрупненные схемы программ, получает руководитель группы программистов. Он распределяет задания между программистами, руководствуясь установленными сроками работы над созданием АСУ, параметрами, характеризующими сложность и трудоемкость разрабатываемых программ, квалификацией программистов и другими факторами. Руководствуясь нормами затрат времени на программирование и необходимого машинного времени ЭВМ, составляют график выполнения работ по этапам. В дальнейшем руководитель группы контролирует соответствие фактических затрат времени предусмотренным в графике, в необходимых случаях консультирует программистов и координирует их работу, участвует в системной отладке.

Нормирование труда программистов. Временные типовые нормы времени на программирование задач для ЭВМ рассчитаны на комплексы задач или задачи и указаны в человеко-днях при пятидневной рабочей неделе продолжительностью 41 ч в неделю.

Нормы времени определены по факторам, в наибольшей степени влияющим на трудоемкость программирования. К этим факторам относятся: количество наборов данных (макетов) входной информации, включая текущие данные, нормативно-справочную и извлекаемую из банков данных информацию; число типов форм выходной информации; степень новизны комплекса задач; сложность алгоритма; вид используемой инфор-

мации; сложность контроля входной и выходной информации; язык программирования; степень использования типовых модулей и пакетов прикладных программ.

Предусмотрены четыре степени новизны комплексов задач: А — задачи, предусматривающие применение принципиально новых методов разработки; Б — разработка типовых проектных решений, оригинальных задач; В — привязка типовых проектных решений, требующая их изменения, разработка задач, имеющих аналоги; Г — использование типовых проектных решений, разработка задач, имеющих аналогичные решения.

По сложности алгоритмы разделены на три группы: оптимизации и моделирования систем и объектов; учета, отчетности, статистики и информационного поиска; стандартных методов решения задач, не требующих применения сложных численных и логических методов.

По виду используемой информации определены три группы: переменной или текущей информации; нормативно-справочной информации; информационной базы.

Сложность контроля информации определена четырьмя группами: перекрестного контроля с учетом связей между группами входных данных и документов; формального контроля однотипных входных данных и документов; контроля при выводе данных сложной многоуровневой структуры, разнообразной формы и содержания; вывода данных на машинные носители, печати документов однообразной формы и содержания.

Нормы времени определены для комплексов задач, имеющих степень новизны В и использующих текущую информацию. При задачах с другими характеристиками используют для пересчета поправочные коэффициенты, в частности для задач группы А—2—5; группы Г—0,4—0,8.

Предполагается использование языков верхнего уровня типа ПЛ/1: для языков низкого уровня применяется коэффициент 1,15. При использовании типовых модулей или пакетов прикладных программ применяется коэффициент 0,6—0,8, а при разработке стандартного математического обеспечения — 1,2—1,6.

Трудоемкость технорабочего проекта определяется как сумма 85% трудоемкости технического и 100% трудоемкости рабочего проектов.

Общая трудоемкость программирования определяется как сумма трудоемкостей отдельных этапов работы.

Нормы времени учитывают работы, сопутствующие программированию, такие, как согласование работ внутри группы, проверка и приемка работ, внесение исправлений и дополнений по указанию руководителя и т.п., а также время на обслуживание рабочего места, отдых и личные надобности в размере 10% от оперативного времени. Если в организации более прогрессивные нормы времени, их не следует изменять на основании приводимых ниже.

Нормы времени в человеко-днях на различных стадиях проектирования приведены в табл. 5.2.

Таблица 5.2

Число входных наборов данных	Стадии проектирования		
	техническое задание	технический проект	рабочий проект
До 5	18	40–55	150–520
" 10	20	60–80	180–700
" 15	22	75–95	200–850

Нормирование затрат машинного времени. Нормы затрат времени работы ЭВМ определяются в зависимости от тех же приведенных выше основных факторов, которые влияют в наибольшей степени и на трудоемкость программирования, с той разницей, что дополнительно учитывается быстродействие ЭВМ.

Приведенные в табл. 5.3 для ориентации нормы времени в часах определены для комплексов задач группы В, стандартных методов решения и языков программирования высокого уровня.

Время работы ЭВМ на стадии ввода в эксплуатацию составляет от 10 до 25% времени, указанного в табл. 5.3. Если внедрение осуществляется заказчиком, затраты машинного времени на отладку должны быть снижены на 10–25% по усмотрению руководителя работ.

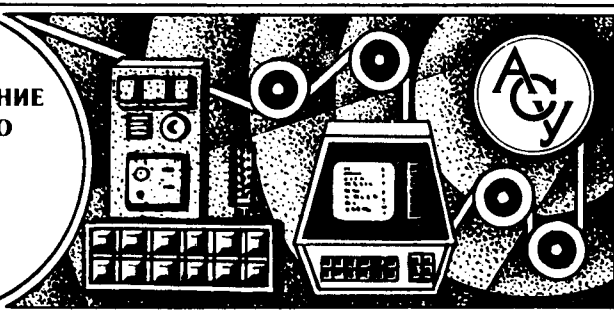
Таблица 5.3

Число входных наборов данных	Число типов выходных форм		
	1–9	10–20	21–40
До 5	30–90	100–200	150–330
" 10	50–180	150–280	200–400
" 15	65–220	190–350	270–500

ПРОЕКТИРОВАНИЕ ТЕХНИЧЕСКОГО ОБЕСПЕЧЕНИЯ

ГЛАВА

6



Используемые в АСУ технические средства, особенно вычислительные машины, развиваются с невиданной в технике скоростью. За 20–30 лет производительность ЭВМ возросла на три порядка, а габариты и стоимость при сравнимой производительности уменьшились в 50–100 раз. Это существенно расширило возможности разработчиков по созданию высокоэффективных АСУ.

Разработчикам необходимо принимать проектные решения, по возможности не зависящие от возможностей конкретных технических средств, чтобы их можно было модернизировать или заменять на более совершенные в процессе эксплуатации системы.

Вместе с тем хорошая информация о перспективных вычислительных средствах позволит проектировать системы на принципиально ином уровне, с гораздо большими возможностями использования ЭВМ.

Массовый выпуск персональных компьютеров позволяет в процессе проектирования дифференцировать набор технических средств вплоть до отдельного типового пользователя, определяя нужную комплектацию периферийными устройствами.

§ 6.1. Структура комплекса технических средств

Иерархическая структура управления требует для ее реализации соответствующей структуры системы передачи и обработки данных. По мере развития ЭВМ совершенствовались внешние запоминающие устройства на магнитных лентах (НМЛ) и магнитных дисках (НМД), устройства управления (УУ), устройства ввода и вывода данных, однако сохранялась присущая системам 60–70-х годов звездообразная структура под-

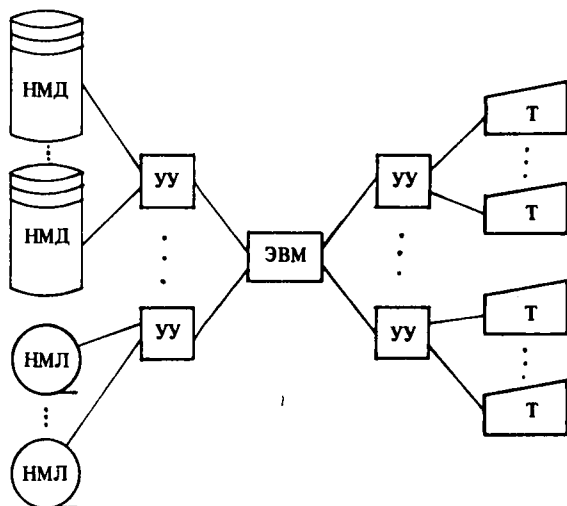


Рис. 6.1

ключения терминалов (Т) к центральному процессору (рис. 6.1). Передача данных между установленными на различных уровнях иерархии ЭВМ осуществлялась путем физического перемещения носителей, как правило, магнитных лент, или передач по телетайпу с последующим вводом данных в ЭВМ с получаемой на телетайпе перфоленты. Использовались также традиционные способы — почтовая, телефонная и телеграфная связь, посылка курьеров. У разработчиков не было технической возможности проектировать структуру комплекса технических средств (КТС) с непосредственным подключением через каналы связи удаленных друг от друга ЭВМ.

В настоящее время следует проектировать систему с использованием новых технических возможностей. К сожалению, по различным обстоятельствам продолжают встречаться случаи, когда приобретение технических средств предшествует разработке системы. Это приводит к несоответствию между возможностями технических средств и требованиями к ним системы. В результате эффективность системы снижается, а затраты возрастают, так как приходится докупать некоторые технические средства, что обходится дороже, чем единовременная закупка всего необходимого оборудования.

Звездообразное подключение терминалов к ЭВМ имеет ряд существенных недостатков: не может быть обеспечена достаточная удаленность терминалов от центрального процессора, так как подключение устройства управления через каналы ввода — вывода возможно лишь на небольших расстояниях (десятки метров); каждый терминал подключен к устройству управления, что приводит к большой суммарной длине кабельных связей, их физической сложности, а порой и запутанности; подключение терминала возможно одновременно только к одной ЭВМ; терминалы не обладают па-

мью и программируемой логикой, поэтому вся обработка информации при обмене сообщениями между пользователями и ЭВМ идет через центральный процессор, что резко снижает производительность ЭВМ. Подключение внешних запоминающих устройств характеризуется теми же недостатками.

В середине 70-х годов появились существенные положительные сдвиги в области микроэлектроники и техники связи, которые обусловили возможность перехода от традиционных схем построения вычислительных систем к более гибким и развитым структурам. К основным таким достижениям можно отнести: создание и серийное производство дешевых высокоскоростных (1–10 Мбит/с) электрических и оптических кабелей; резкое повышение степени интеграции электронных схем, массовое производство сверхбольших интегральных схем (СБИС) и микропроцессоров (МП), позволяющих создавать большое количество сложных по организации и интенсивно взаимодействующих между собой устройств; освоение и использование в широких масштабах методов протокольного обмена информацией между устройствами.

Понятие *протокольного обмена* связано с появлением вычислительных сетей; *протокол обмена* описывает правила, по которым происходит взаимосвязь одноименных уровней программной структуры вычислительной сети. Данные правила учитывают обязательные характеристики элементов вычислительной сети и процедуры их взаимодействия.

Взаимосвязь соседних (в одной ЭВМ) уровней программной структуры определяется стандартами, именуемыми интерфейсами. *Интерфейсы* определяют стыковку соседних уровней программной структуры и поэтому имеют локальное значение.

Протоколы характеризуют функционирование всей вычислительной сети. в целом и имеют определенную иерархию. Наиболее распространенной в настоящее время является семиуровневая модель процесса обмена информацией, развитие которой исторически связано с появлением сетей, обеспечивающих взаимодействие ЭВМ, расположенных в различных городах. Поскольку процесс обмена для таких вычислительных сетей хорошо изучен, на их основе создаются способы обмена и в локальных вычислительных сетях.

Локальной вычислительной сетью (ЛВС) называют распределенную вычислительную систему, построенную на базе общей среды передачи данных, обеспечивающей физическую полносвязность всех компонентов системы, простоту реконфигурации системы и охватывающей несколько сравнительно близко расположенных помещений.

В локальной сети, как и в любой вычислительной системе, выделяют семь уровней обработки информации: прикладной, представительный, сеансовый, транспортный, сетевой, канальный (информационный) и физический.

Прикладной (седьмой) уровень охватывает: общие служебные функции (подключение и отключение пользователя к прикладной программе,

проверка санкционированного доступа и т.д.); общие базовые функции (доступ к файлам и их передача, удаленный ввод заданий и т.д.); проблемно-ориентированные функции (информационный поиск, обработка заказов на авиабилеты, подбор библиографии и т.д.).

Представительный (шестой) уровень связан с выполнением необходимых преобразований информации для устранения синтаксических разногласий (элементарные синтаксические и формальные преобразования символов, полей данных, страниц экранов терминалов и т.д.).

Сеансовый (пятый) уровень обеспечивает поддержку диалога (управление началом и завершением сеанса, управление межпроцессорными связями, восстановление после ошибок и т.п.).

Транспортный (четвертый) уровень обеспечивает двухточечное (между двумя соседними абонентскими пунктами) соединение пользователей, вне зависимости от реализации этого соединения на 3-м (сетевом) уровне, и служит для оптимизации использования связанных ресурсов и минимизации стоимости обмена информацией в системе за счет учета всех заявок и наличия ресурсов. Кроме того, данный уровень обеспечивает сервис по установлению соединений объектов сеансового уровня и обслуживание передачи данных вплоть до ее завершения.

Сетевой (третий) уровень выполняет работы, связанные с выбором маршрутов, коммутацией соответствующих линий связи и т.п.

Канальный (второй) уровень обеспечивает: установление, проведение и завершение обмена информацией по информационному каналу; установление соединений на физическом уровне, при этом обнаруживает и устраняет ошибки данного уровня.

Физический (первый) уровень представляет собой набор средств для физического соединения объектов друг с другом. При этом допускается как непосредственная связь источника сообщения с приемником, так и связь через промежуточные ретрансляторы.

Совокупность уровней с первого по четвертый образует транспортную сеть, назначение которой — организация обмена информацией между уровнями с пятого по седьмой.

Протоколы обмена для всех семи уровней имеют некоторые отличия. Наибольшую популярность в настоящее время имеют протоколы обмена по стандартам X.21 и X.25, распространение которых связано с тем, что в 80-х годах Международной организацией по стандартизации (ISO) была разработана эталонная модель соединения открытых систем, имеющая фундаментальное значение. Международный консультативный комитет по телефонии и телеграфии (МККТТ) рекомендовал ряд стандартов серии X для подключения абонентского оборудования к сетям коммутации каналов и пакетов.

Рекомендации X.21 и X.25 определяют соответственно: стандарты на универсальный интерфейс для синхронной работы в общей сети передачи данных и стандарты взаимодействия абонентских машин с коммуникационной сетью.

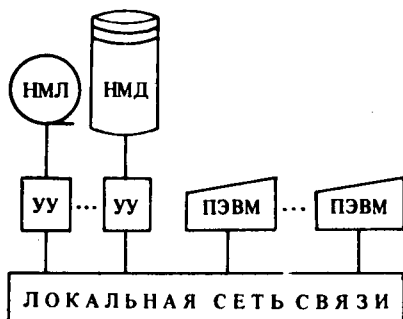


Рис. 6.2

Рекомендация X.21 описывает используемые цепи обмена, схему сигналов в точке интерфейса и метод синхронизации символов между вычислительной машиной и каналом связи (в дополнение к этому МККТТ разработал и утвердил рекомендацию X.21-бис, введение которой позволяет использовать в сети связи стандартные модемы к аналоговым телефонным каналам связи), а также характеризует управление каналами физического уровня.

Рекомендация X.25 представляет собой стандарт трехуровневой структуры, удовлетворяющей требованиям эталонной модели, т.е. описывает протоколы физического, канального и сетевого уровней в точках сопряжения с сетью коммутации пакетов (взаимодействия абонентских машин с коммутационной сетью). Эта рекомендация определяет также основные требования к процедурам транспортировки пакетов через всю коммутационную сеть — от одной абонентской машины к любой другой.

Применение данных стандартов улучшает использование пропускной способности канала.

Такой подход обеспечивает возможность подключения к каждому терминалу дешевого микропроцессора и не менее дешевого блока оперативной памяти. То же самое касается и внешних запоминающих устройств. Это упрощает структуру связи всех компонентов вычислительной системы при небольшой суммарной длине соединяющих кабелей. Созданные таким образом системы являются локальными вычислительными сетями, а система их связи — *локальной сетью связи (ЛСС)* (рис. 6.2).

Сравнительные характеристики локальных сетей связи. Основными показателями, характеризующими ЛСС, являются: физическая среда передачи данных; методы доступа к физической среде; топология среды передачи данных.

Физическая среда передачи данных может быть реализована с применением витых пар (ВП), коаксиального кабеля с дискретной сигнализацией (КДС), коаксиального кабеля с аналоговой сигнализацией (КАС), оптоволоконного кабеля (КО). Сравнительные качественные характеристики перечисленных физических сред приведены в табл. 6.1. Стоимость дорогих кабелей имеет тенденцию к снижению, что надо принимать во внимание при разработке перспективных ЛСС.

Основным достоинством витых пар наряду с их низкой стоимостью является легкость подключения новых узлов и наращивания длины пары с помощью повторителей. Основной недостаток — низкая скорость передачи данных (до 1 Мбит/с). Поэтому витые пары чаще всего исполь-

зуют для подключения к ЭВМ периферийных устройств (устройств печати, накопителей на гибких магнитных дисках и лентах).

Таблица 6.1

Тип среды	Параметры					
	полоса пропускания	число подключаемых узлов	протяженность без повторителей	трудоемкость прокладки	помехозащищенность	стоимость (без учета прокладки)
Витые пары	Узкая	Малое	Малая	Средняя	Малая (более высокая для экранированных ВП)	Низкая
Коаксиальный кабель с дискретной сигнализацией	Узкая — средняя	Среднее	Средняя	Малая	Средняя — высокая	Средняя
Коаксиальный кабель с аналоговой сигнализацией	Широкая	Большое	Большая	”	Высокая	Средняя — высокая
Оптоволоконный кабель	Очень широкая	Малое	Очень большая	Средняя	Очень высокая	Очень высокая

Коаксиальный кабель с дискретной сигнализацией обеспечивает более высокую скорость передачи данных (до 10 М бит/с), но стоимость его значительно выше. При увеличении размера сети свыше 2 км (но не более 10 км) используют повторители, при этом сохраняется высокая помехоустойчивость.

Для построения сетей на базе коаксиальных кабелей с аналоговой сигнализацией используют в основном технику кабельного телевидения и радиочастотные модемы. Несмотря на высокую стоимость как прокладки, так и эксплуатации ЛСС на основе КАС, они имеют два существенных преимущества: возможность в разных полосах частот передавать одновременно речь, изображение и данные; допустимая длина сети — порядка 50 км.

Основные преимущества оптоволоконных кабелей — высокая скорость передачи (до 200 М бит/с) и высокая помехозащищенность. Основной недостаток — затруднено подключение новых абонентских устройств из-за неотработанной технологии ветвления оптоволоконных кабелей.

Топология сети передачи данных может быть одного из следующих типов: шина (моноканал); кольцо (петля); звезда; дерево (поликанал).

Сравнительные качественные характеристики приведенных топологий представлены в табл. 6.2. Каждая из существующих топологий ЛСС определяет как метод доступа к каналу ЛСС, так и физическую среду передачи данных.

Таблица 6.2

Топология	Параметры			
	сложность интерфейса	наращиваемость	надежность	стоимость
Шина	Малая – средняя	Высокая	Высокая	Низкая
Кольцо	Малая	Средняя	”	Средняя
Звезда	”	Малая	Средняя – высокая	Высокая
Дерево	Средняя	Высокая	Высокая	Низкая

Для топологии типа "шина" используются КДС и следующие методы доступа: множественный доступ с контролем несущей и обнаружением конфликтов (МДКН/ОК); селективный опрос; адресуемая явная передача права на работу в сети.

Для топологии типа "кольцо" применяются в основном ВП и используются следующие методы доступа: заполнение регистра; групповой опрос; неадресуемая неявная передача права.

Для топологии типа "звезда" применяются как ВП, так и КО. При использовании ВП центром звезды является электронная АТС или коммутатор пакетов и используются методы доступа X.21 или X.25. Во втором случае в центре звезды располагается оптический смеситель и используется метод доступа МДКН/ОК.

Топология типа "дерево" характерна для среды передачи данных вида КАС. Из-за поликанальности данной среды возможно одновременное использование сразу нескольких методов доступа: МДКН/ОК, частотное разделение каналов; адресуемая передача права.

Поскольку все методы доступа зависят от топологии ЛСС, их самостоятельное количественное и качественное сравнение затруднительно; некоторые их качественные характеристики приведены в табл. 6.3.

Способы доступа можно разделить на четыре основных класса.

Способы случайного доступа. При этом способе начало обмена информацией не координируется абонентами и используются процедуры, позволяющие уменьшать вероятность конфликтов (например, временной сдвиг в начале каждой попытки начала обмена).

Таблица 6.3

Класс метода доступа	Топология локальной сети связи			
	шина	кольцо	звезда	дерево
Случайный	МДКН/ОК	Заполнение регистра	МДКН/ОК	МДКН/ОК
Детерминированный	Централизованный	Селективный опрос	Групповой опрос	Коммутация физического (Х.21) или логического (Х.25) соединения
	Децентрализованный	Адресуемая передача права	Неявная передача права	Коммутация частотных полос

Способы пропорционального доступа. Конфликтные ситуации исключаются путем заранее заданного порядка работы абонентов.

Способы приоритетного доступа. Конфликтные ситуации устраняются путем обмена специальными управляющими сигналами до начала передачи и установкой приоритетов на порядок передачи.

Способы локально-приоритетного доступа. Доступ предоставляется на основе приоритетов, но при этом источник учитывает только состояние линии в точке его подключения.

Сравнительный анализ локальных сетей связи. В настоящее время отсутствуют формальные способы выбора среды передачи данных, поэтому топология ЛВС выбирается в зависимости от того, какие эксплуатационные характеристики необходимо получить в каждом конкретном случае реализации ЛВС. Тем не менее, подходя к данному вопросу неформально, можно существенно сократить круг рассматриваемых в дальнейшем локальных сетей связи.

Так, можно исключить из рассмотрения древовидную топологию и соответствующие ей локальные сети на базе техники кабельного телевидения, как не получившие пока широкого распространения в нашей стране, а также звездообразные ЛСС, поскольку суммарная длина соединительных кабелей в них очень велика, а отказ центрального узла выводит из строя всю сеть. Поэтому рассмотрим два типа топологий – “шина” и “кольцо” по следующим вариантам сетей:

- 1) число узлов (станций, ЭВМ, абонентских пунктов) сети – 100;
- 2) метод доступа сети типа “кольцо” – неявная передача права; метод доступа сети типа “шина” – либо МДКН/ОК, либо явная передача права по логическому кольцу узлов (в дальнейшем будем подразумевать, что для шины передача права явная, а для кольца неявная);

- 3) в сети имеется либо один активный узел (т.е. узел, передающий па-

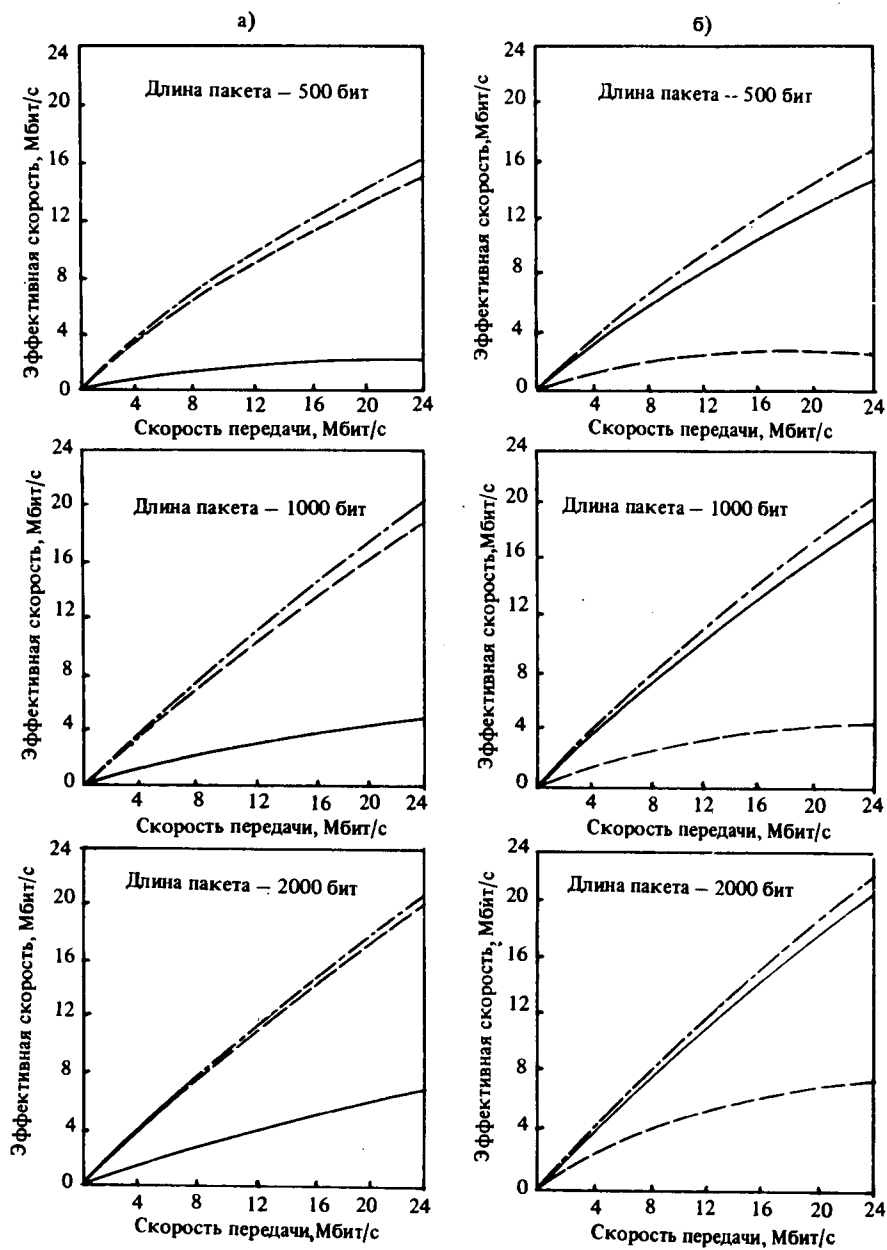


Рис. 6.3

кет в другой и имеющий после этого еще один пакет), либо активны все узлы сети;

4) каждый пакет передается в составе кадра данных с заголовком, состоящим из 96 бит, длина пакета: 500, 1000 и 2000 бит.

На рис. 6.3 приведены аналитические зависимости средней эффективной скорости передачи кадров данных от скорости передачи в физической среде.

На рис. 6.3 штрихпунктирная линия соответствует сети типа "кольцо" с передачей права, пунктирная — сети типа "шина" с методом доступа МДКН/ОК и сплошная линия — сети типа "шина" с передачей права, а — в каждый момент активна одна из 100 станций, в — в каждый момент активны все 100 станций.

Основной вывод из показанных на рисунке графиков состоит в том, что кольцевая сеть с передачей права менее всего чувствительна к нагрузке и обладает более высокой эффективной скоростью передачи данных в диапазоне скоростей до 24 Мбит/с и при длине пакета до 2000 бит.

Другой сравнительный анализ кольцевой сети с передачей права и сети типа "шина" с методом доступа МДКН/ОК подтверждает сделанный вывод. На рис. 6.4 приведены графики зависимости средней задержки передачи кадров данных от нагрузки на физическую среду для двух скоростей: 1 и 10 Мбит/с. Средняя задержка и нагрузка нормированы соответственно средним временем передачи кадра данных в среде и скоростью передачи среды. Средняя длина пакета 1000 бит, заголовок кадра 10 байт, число узлов 50. При скорости передачи 1 Мбит/с средние задержки в обеих сетях практически одинаковы. При скорости 10 Мбит/с задержка в кольцевой сети значительно меньше в широком

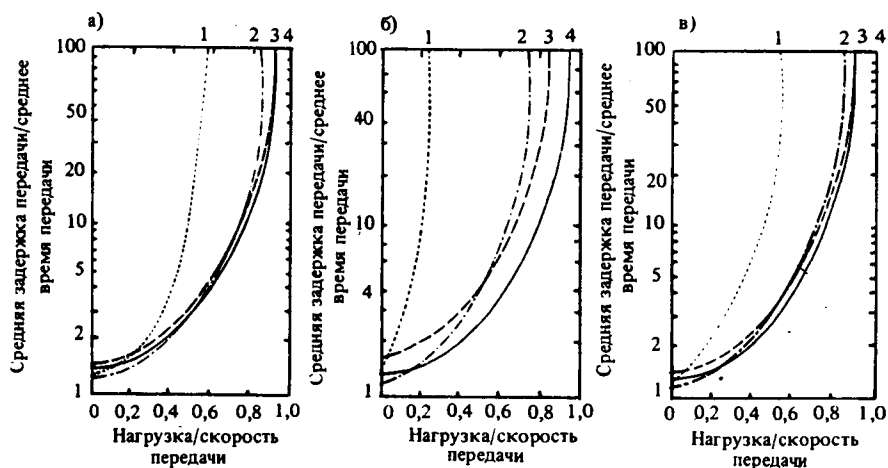


Рис. 6.4

диапазоне параметров (длины кабелей, нагрузка, типы распределений длин пакетов).

Критическим параметром для сети типа "шина" с методом доступа МДКН/ОК, который определяет ее производительность, является отношение τ_p/τ_n , где τ_p — время распространения сигнала по физической среде (кабелю), а τ_n — среднее время передачи данных в физической среде. Поскольку τ_p не зависит от скорости передачи, а определяется длиной кабеля, то при одной и той же длине кабеля теоретически показано, что шина с методом доступа МДКН/ОК имеет удовлетворительные характеристики до тех пор, пока отношение τ_p/τ_n достаточно мало. При отношении $\tau_p/\tau_n > 0,05$ частота конфликтов практически сводит производительность сети на нет. Связано это с тем, что при больших нагрузках системы высокая частота конфликтов совместно с алгоритмом откладывания повторных передач приводит к увеличению средней задержки передачи. Практически это означает блокировку узла на неопределенный период времени. В кольцевой сети с передачей права даже при максимальной нагрузке обеспечивается равноправное распределение полосы пропускания среды между всеми активными узлами, так как право передачи сообщения передается каждым узлом сразу после передачи одного кадра. Правильность изложенного иллюстрируется графиками рис. 6.4, б, в. Кривая 1 показывает влияние времени распространения и подтверждает важность отношения τ_p/τ_n для шины с методом доступа МДКН/ОК, что означает практическую невозможность использования данного метода при длине кабеля ≥ 10 км.

Проведенные сравнительные оценки ЛСС по электрическим характеристикам, которые имеют не меньшее значение, чем производительность, показывают, что:

- 1) сети с топологией типа "шина" и методом доступа МДКН/ОК должны строиться как на базе одного отрезка кабеля (сегмента), так и на основе нескольких отрезков кабеля, соединенных повторителями и имеющих одни и те же характеристики, т.е. в таких сетях нельзя использовать различные среды передачи данных;

- 2) длина сети типа "шина" зависит от используемых схем обнаружения конфликтов;

- 3) приемопередатчики непосредственно подключаются к кабелю (при топологии шина) и должны быть электрически изолированы от аппаратуры узлов;

- 4) кольцевая сеть должна состоять из синхронных двухточечных каналов, которые допускают большое ослабление сигналов (и соответственно большие длины кабелей); в одном кольце возможно одновременное использование различных сред передачи, согласованных по скорости;

- 5) в кольцевой сети можно использовать двухточечные оптоволоконные кабели;

- 6) сетевой адаптер и соответствующие схемы реализации метода досту-

па при топологии кольцо могут быть собраны на одной плате (БИС или СБИС), которая может устанавливаться непосредственно в узле.

Основные эксплуатационные характеристики ЛСС — высокая надежность и готовность к использованию — являются ключевыми при их оценке.

Требования к изготовителям ЛСС можно сформулировать следующим образом: необходима простота установки и возможность планируемого развертывания сети в большом числе помещений (прокладка и распайка кабелей, установка розеток). Со стороны пользователя требования следующие: автоматическое обнаружение и устранение отказов и возможность обслуживания неквалифицированным персоналом. Такому уровню в настоящее время соответствуют только телефонные учрежденческие сети и коммунальные абонентские сети, поэтому технология построения ЛСС должна стремиться к этому уровню.

Топология типа "шина" не удовлетворительна к отказам питания и отказам соединительных кабелей.

При топологии типа "кольцо" устойчивость к отказам достигается использованием распределительных панелей с проходными реле. В случае, когда проходные реле остаются без питания, местный отвод пропускается, т.е. не включается в сеть. Таким образом, отказ блока питания узла или обрыв соединительного кабеля автоматически отключает узел от сети, после чего сеть восстанавливает свою работоспособность. Отказ в основном кольце приводит к частичному отказу всей сети связи, превращая ее в линейную последовательность узлов. В этом случае данные, передаваемые узлом ниже обрыва, могут быть получены узлами, расположенными ниже передающего. При выходе из строя узла, обеспечивающего синхронизацию сети, эту функцию берет на себя другой узел.

Рассмотренные особенности следует учитывать при выборе топологии локальной сети связи в каждом случае реализации локальной вычислительной сети с учетом как пожеланий заказчика, так и возможностей изготовителя.

Необходимость создания распределенных систем управления, включающих ЛВС, вызвана тем, что управляемые объекты территориально распределены на большом пространстве. В подобных случаях удобно строить систему так, чтобы основной объем информации обрабатывался в непосредственной близости от источников информации. Примером может служить АСУ предприятия, где ЛВС охватывает ряд цехов и время от времени ЭВМ, входящие в одну сеть, обмениваются сообщениями с целью координации локальных производственных процессов в масштабе всего предприятия. Подсистема, являющаяся либо ЛВС, либо ее частью, может самостоятельно управлять производством в данном цехе, взаимодействуя при этом с другими подсистемами для получения информации о нормах выпуска изделия, о поставке комплектующих деталей, сырья и т.д.

Распределенная обработка позволяет упростить установку системы, монтаж и ввод ее в эксплуатацию за счет возможности поэтапного запуска

модулей системы и сокращения общего числа и протяженности каналов связи. Внедрение на предприятии системы распределенной обработки информации способствует созданию организационной структуры, в которой специалисты, отвечающие за ход технологического процесса, отвечают и за выполнение связанных с ним вычислительных задач. Создание многомашинных комплексов позволяет повысить производительность системы за счет перераспределения вычислительных ресурсов при перегрузке одного из звеньев системы, а также возможности планирования очередности решения задач, требующих большого количества расчетов, используя высокоскоростной процессор, который получает данные от других ЭВМ, обрабатывает их по заданной стратегии и передает обратно.

Распределенные вычислительные системы коллективного пользования. Типичная организация распределенной вычислительной системы (РВС) для создания территориально распределенной АСУ показана на рис. 6.5. РВС состоит из трех групп вычислительных систем (ВС), которые в свою очередь могут представлять собой ЛВС, расположенные в различных точках. Первая группа включает машины, объединенные в локальную сеть типа "шина", во вторую группу входят ЭВМ, соединенные "кольцом", и третья группа — центральная вычислительная система, в функции которой входит сбор данных и обработка информации, поступающей от других ВС и операторов, а также общая координация работы всей системы. Все локальные ВС объединяются в центральную ВС через шлюзы — межсетевой интерфейс.

Техника построения ЛВС (рис. 6.5) позволяет строить ВС с различной степенью централизации и иерархии, называемые в настоящее время распределенными вычислительными системами коллективного пользования (РВСКП).

В настоящее время под РВСКП обычно понимается многомашинальная система с ориентацией отдельных ЭВМ либо на приложения (обработка данных, изображения и т.д.), либо на функции, выполняемые в традиционных ЭВМ их операционными системами (например, загрузчик прикладных программ, файловая система, планировщик времени центрального процессора и т.д.), либо на то и другое. В состав РВСКП могут включаться как однотипные, так и разнотипные ЭВМ.

При таком подходе к созданию ВС возможно распределение между пользователями не времени центрального процессора, а компонентов "прикладных" процессоров, т.е. происходит как бы распределение пользователей в зависимости от их приоритета и типа решаемых задач по ресурсам РВСКП (в простейшем случае — выделение пользователю одного из процессоров сети для монопольной работы).

Объединение вычислительных средств в составе РВСКП позволяет: предоставить пользователям более мощный набор аппаратных и программных средств хранения, обработки и отображения информации (манипулирования); увеличить загрузку уникальных специализированных устройств обработки данных за счет передачи им информации от территориально уда-

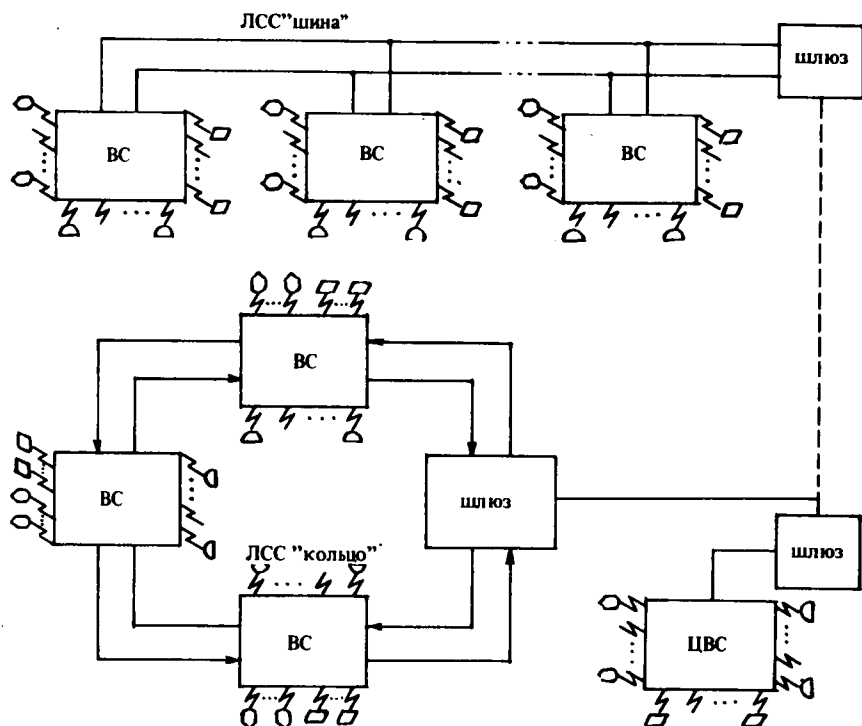


Рис. 6.5

ленных пользователей; обеспечить преемственность в аппаратных и программных средствах для ЭВМ различных поколений; наращивать мощность РВСКП по мере увеличения нагрузки на нее.

Важно отметить, что РВСКП предъявляет некоторые требования к локальной сети связи, а именно: ЛСС должна обеспечивать нормальную передачу данных для всех входящих в сеть устройств и ЭВМ, при этом необходимо учитывать такие высокоскоростные устройства, как канал прямого доступа в память и устройство управления дисковыми; в большинстве случаев надежность ЛСС должна быть такой, чтобы обеспечить круглосуточное безостановочное функционирование РВСКП в течение рабочей недели; стоимость подключения устройств и ЭВМ к ЛСС должна быть приемлемой.

Наряду с использованием ЛСС при создании РВСКП применяют также традиционные методы подключения терминалов (как интеллектуальных, представляющих собой персональные ЭВМ, так и стандартных устройств ввода – вывода) и других периферийных устройств. При этом используют два основных способа подключения периферийных устройств к центральному процессору любого класса: подключение посредством мультиплексного канала; подключение посредством селекторного канала.

Мультиплексные каналы обеспечивают обмен данными с относительно медленно действующими периферийными устройствами и имеют два режима работы: *монопольный*, когда канал обеспечивает работу только одного устройства, и *мультиплексный*, когда одновременно обеспечивается поочередный обмен информацией ряда устройств.

Селекторные каналы предназначены для организации обмена данными между оперативной памятью процессора и быстродействующими периферийными устройствами и работают только в монопольном режиме.

Количество мультиплексных и селекторных каналов зависит от модели ЭВМ и может варьироваться от двух селекторных и одного мультиплексного канала с числом подключаемых к каналу устройств ввода — вывода, не превышающем 10; до десятка селекторных и нескольких мультиплексных каналов с количеством подключаемых устройств до 300. При этом пропускная способность мультиплексных каналов варьируется от десятков до нескольких сотен килобайт в секунду в мультиплексном режиме. Сравнительные данные для некоторых каналов ЕС ЭВМ приведены в табл. 6.4.

Т а б л и ц а 6.4

Сравнительная таблица технических характеристик
мультиплексных и селекторных каналов ЕС ЭВМ

Технические характеристики	ЕС-1022	ЕС-1033	ЕС-1035	ЕС-1040	ЕС-1045	ЕС-1050	ЕС-1052	ЕС-1055	ЕС-1060
Мультиплексные каналы	1	1	1	1	2	1	1	1	1
Число подканалов	128	128	256	128–256	256	196	224	256	224
Максимальная суммарная скорость в мультиплексном режиме, К байт/с	70–80	70	40	20–25	40	110–670	425	40	110
Максимальная суммарная скорость в монопольном режиме, К байт/с	400	350	200–600	180–720	160	180–720	—	1500	670
Селекторные каналы									
Число каналов	2	3	4	6	5	6	2	4	2–6
Скорость канала, М байт/с	0,7	0,8	0,75–1,0	1,3	0,5–1,5	1,3	1,25	1,5–3,0	1,2

Применение тех или иных способов связи ЭВМ как друг с другом так и с периферийными устройствами зависит, как было отмечено выше, в первую очередь от быстродействия периферийных устройств. При этом надо помнить, что реальное быстродействие многих устройств (например, абонентских терминалов) ставится в ущерб их функциональному размещению из-за отмеченных ранее ограничений на скорости передачи по ЛСС при удаленном расположении терминалов. Иногда подключение таких устройств в РВСКП осуществляется через специально выделенную для этой цели ЭВМ, которая занимается обслуживанием обращений к "быстрым" устройствам и оптимизацией их работы с точки зрения уменьшения времени доступа к данному классу устройств.

При включении в РВСКП супер-ЭВМ необходимо продумать не только вопросы физической стыковки входящих в сеть ЭВМ, но и распределения функций между ними, поскольку общая производительность вычислительной системы, включающей супер-ЭВМ, зависит от внешнего окружения.

Основными функциями вычислительной системы являются следующие.

Ввод — вывод. При резком повышении производительности ЭВМ возникают проблемы ввода и вывода данных, связанные с необходимостью загрузить супер-ЭВМ. Эти трудности разрешаются двумя путями: совершенствованием программного обеспечения с целью сокращения объема вводимой и выводимой информации за счет ее установки, и разработкой специальных аппаратных средств ввода — вывода вплоть до использования ЭВМ.

Неарифметическая обработка данных. К этому классу задач относятся трансляция, редактирование, обработка текстов и т.п. В таких случаях удобно использовать ЭВМ, специально ориентированную на работу со списочными структурами или таблицами.

Управление базами данных. Работа с базами данных предполагает выполнение следующих операций: создание и уничтожение библиотек данных, запоминание объектов по мнемоническим названиям вместе с определенными их свойствами, поиск по ключам и контексту, защита от несанкционированного доступа и т.д. Решение этого класса задач в настоящее время лежит в области программных средств, но уже ведутся работы по аппаратной реализации баз данных на основе микропроцессоров.

Управление процессами. К этому классу задач относится управление различными вычислительными и другими процессами, протекающими в вычислительной системе.

Для поддержания высокой производительности системы необходима развитая операционная система, согласующая работу всех ее компонентов и решающая задачи планирования и распределения работ и ресурсов, отработки сбоев и даже реконфигурации системы.

Поскольку операционная система универсальной ЭВМ использует до 50% ее мощности, в настоящее время ряд функций операционных систем решаются аппаратными средствами. Таким же образом решается проблема совместимости программного обеспечения различных ЭВМ. Например, если персональная ЭВМ (ПЭВМ) имеет два микропроцессора, они могут

работать под управлением так называемой "гибкой" операционной системы, позволяющей автоматически определять, какой именно процессор нужен для исполнения той или иной загружаемой программы.

На основе так называемых "одноплатных микроЭВМ приставок" к персональным компьютерам возможно дать пользователям доступ к программному обеспечению практически всех популярных разнотипных ПЭВМ. Такие приставки позволяют использовать не только программное обеспечение различных ПЭВМ, но и решают вопрос использования ранее разработанного математического обеспечения на новых моделях и, наоборот, адаптацию нового программного обеспечения на старых моделях.

Кроме такого ярко выраженного аппаратного подхода к вопросу совместимости программного обеспечения существует и программный подход. Этот подход состоит в создании так называемых интерпретаторов *p*-кода, развитие которого относится к рубежу 70-х годов, когда было предложено первое радикальное средство для решения проблемы программной совместимости ЭВМ. Суть данного подхода — создание операционных систем и трансляторов с языков высокого уровня для некой абстрактной ЭВМ с системой команд, удовлетворяющих ряду выбранных из теоретических соображений принципов. Эта система команд, получившая название *p*-код (от слова *portability*) позволяет разработчику любой вновь создаваемой ЭВМ, снабдив ее интерпретатором *p*-кода, решить проблему базового программного обеспечения, написанного в UCSD-версиях (по имени *University of California, — San Diego*).

Совокупность аппаратно-программных решений совместимости ЭВМ сводится к следующему.

Аппаратные решения: 1) создание ЭВМ с несколькими микропроцессорами, способными работать с различными операционными системами, и, следовательно, с возможностью использования программного обеспечения данных ОС; 2) подключение к ЭВМ "приставок", которые исполняют программы, не доступные данной ЭВМ, используя при этом все ее возможности по связи с периферийными устройствами.

Программные решения: 1) оснащение ЭВМ системами на базе интерпретаторов *p*-кода для использования в ЭВМ машино-независимых трансляторов с языков высокого уровня; 2) оснащение ЭВМ стандартной операционной системой, применяемой независимо от класса используемой ЭВМ; 3) разработка трансляторов для автоматического перевода прикладных программ, написанных в одной операционной системе, в другую.

В настоящее время, когда целые операционные системы реализуются аппаратным способом путем "прожигания" в микропроцессоре, разница между аппаратными и программными средствами подчас существует лишь для специалистов в области разработки ЭВМ. Для проектирования сетей коллективного пользования важно лишь правильно выбрать комплексы технических средств, совместимые либо по программному, либо аппаратному интерфейсу, или осуществляющие обмен по международному стандарту локальных вычислительных систем.

Создание КТС является сложным итерационным процессом, который в настоящее время не может быть полностью формализован из-за отсутствия единого подхода к выбору КТС для АСУ различных объектов. В связи с этим важно определить основные принципы проектирования КТС, которые на основе отечественного и зарубежного опыта формулируются следующим образом. В процессе проектирования КТС необходимо:

1) провести четкое разграничение функций между отдельными подсистемами КТС, определить их специализацию делением на основные, вспомогательные и обслуживающие подсистемы; при этом особое внимание должно быть уделено вопросам непрерывности процесса обработки информации;

2) учесть распределение обязанностей по управлению объектом при выборе устройств ввода и отображения информации, что связано со сложно формализуемым процессом управления, удобством работы операторов и непредвиденностью ситуаций при функционировании АСУ;

3) обеспечить резервирование технических средств исходя из требуемой надежности и работоспособности комплекса в различных экстремальных ситуациях;

4) предусмотреть концентрацию родственных задач на отдельных ЭВМ многомашинного комплекса и их продуманное перераспределение;

5) обеспечить модульность разработки всех подсистем КТС, что позволяет осуществить автономную поставку, монтаж и наладку оборудования по функциональным подсистемам и таким образом сократить сроки разработки и учесть потребности развития объектов; модульное построение вычислительных систем позволяет создавать экономически оправданные системы, развитие которых полностью соответствует этапам внедрения АСУ;

6) предусмотреть возможность как программной, так и аппаратной защиты системы от несанкционированного доступа;

7) соотнести возможности терминального оборудования с возможностью вычислительных средств, производительность входящих в КТС ЭВМ с пропускной способностью линий связи (при проектировании распределенных вычислительных систем);

8) учесть преемственность как основных вычислительных средств, так и периферийного оборудования по стыковке вновь приобретаемых с уже действующими устройствами ВС.

Необходимость учета при выборе КТС АСУ не только технических и экономических характеристик технических средств, но и организационных и конъюнктурных факторов обуславливает высокую сложность данной проблемы.

Для реализации изложенных принципов проектирования КТС иполь-

зуется так называемый "трехступенчатый подход", заключающийся в следующем. На первом этапе проектируют "семейство компонентов ВС". В это оборудование входят: процессоры, устройства внешней памяти, оборудование связи и передачи данных, каналы ввода — вывода, контроллеры устройств и т.д. Эти компоненты являются составными частями для проведения второго этапа — выбора варианта установки КТС в соответствии с требованиями заказчика. К этому этапу системотехник располагает значительно более детальным знанием рабочей нагрузки будущей ВС. Сюда же входит выбор не только аппаратных, но и программных компонентов системы. Третий этап — настройка системы — выполняется при реальной рабочей нагрузке ВС после ее монтажа и запуска.

Независимо от подхода на практике существуют две принципиально различные методологии проектирования систем. Одна называется проектированием "снизу вверх", а другая — "сверху вниз".

Проектирование *снизу вверх* состоит в итеративной выработке решения проблемы для получения системы, удовлетворяющей заданным условиям.

Проектирование *сверху вниз* основано на получении как характеристик компонентов системы, так и системы в целом непосредственно из проектных спецификаций, что в результате позволит обойтись без проверок, кроме необходимых для устранения возможных сомнений относительно корректного применения процедур проектирования.

В настоящее время эти методологии, к сожалению, развиты и реализованы лишь для простых случаев, в то время как их основное достоинство — обеспечение корректности сложных проектов.

При использовании обеих методологий важно помнить, что постоянно необходимо искать компромисс между аппаратными и программными средствами при реализации конкретных систем управления. С появлением интеллектуальных терминалов и персональных ЭВМ времена выбора КТС без учета создаваемого программного обеспечения АСУ прошли.

§ 6.3. Специальные технические средства

Специальные, или специализированные, устройства предназначены для выполнения особых функций при взаимодействии пользователя с ЭВМ. Данный термин в большей мере отражает цель, нежели конструкцию устройства. Требования, предъявляемые к устройствам данного класса, различны, и акцент на наиболее важные из них делает заказчик.

В общем случае специализированное (специальное) устройство должно быть: более экономичным, чем устройство общего назначения при его конкретном применении; серийно производимым, что отражается как на его стоимости, так и на надежности (оборудование, проходящее, как правило, несколько итерационных этапов от опытного образца до серийного изготовления, обладает большей надежностью и ремонтоспособностью, чем мелкосерийное и уникальное оборудование); эффективно используемым, т.е.

эффективность специализированного устройства всегда должна сопоставляться с эффективностью реализации тех или иных операций с помощью стандартных средств.

В первом приближении затраты на реализацию специализированных аппаратных средств могут быть оценены, как стоимость заменяющих их программ и соответствующих ресурсов. Если стоимость разработки и эксплуатации программных средств несущественно превышает стоимость специального устройства, необходимо принимать во внимание тот факт, что в перспективе стоимость программного обеспечения будет только увеличиваться, тогда как аппаратные средства постоянно дешевеют. Специальное (специализированное) устройство должно быть удобным в эксплуатации, что подразумевает не только хорошую конструкцию и дизайн данного устройства, но и удобство его применения в конкретных решаемых пользователем задачах.

Таким образом, выбор аппаратных средств специального или общего назначения должен основываться на сравнительной оценке настоящей или прогнозируемой стоимости с учетом изложенных требований, причем стоимость должна определяться в соответствии с продолжительностью жизни комплекса.

Примером средств специального назначения, получивших широкое применение в настоящее время, являются устройства ввода графической информации, которые обеспечивают два класса взаимодействия с ЭВМ: указание на элементы изображений и определение положения элементов изображения, которые в свою очередь подразделяются на устройства позиционирования (определения координат точек на экране дисплея) и устройства ввода (кодирования) информации с графических документов.

В качестве устройства указания элементов изображения наиболее широко применяется световое перо, реагирующее на свечение той области экрана графического дисплея, куда оно направлено.

Для определения координат точки на экране дисплея применяются прямые методы, когда рабочий орган устройства устанавливается в требуемую точку, а следящая система вычисляет его положение (например, при сенсорном методе в качестве рабочего органа используется палец, а следящая система состоит из матрицы фото- и светодиодов, окружающих экран).

Значительно чаще применяются косвенные методы определения координат точки на экране. В этом случае перемещение изображения знака слежения (курсора) на экране дисплея обеспечивается изменением характеристик устройства позиционирования. При попадании знака слежения в нужную точку по команде оператора вычисляются искомые координаты.

Такие устройства ввода обычно имеют две степени свободы (рис. 6.6) и реализуются в виде шарового (рис. 6.6, а), рычажного (рис. 6.6, б) указателей или устройства типа "мышь" (рис. 6.6, в). На клавиатуре большинства современных дисплеев выделены специальные клавиши для указания четырех направлений перемещения курсора (по осям координат).

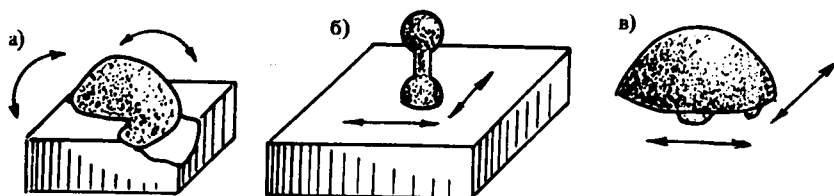


Рис. 6.6

Отметим, что в зависимости от способов обработки получаемых от устройств позиционирования параметров одни и те же устройства ввода используются как указатели позиции точки на экране, так и для ввода числовых параметров.

Устройства кодирования графической информации предназначены для автоматического или полуавтоматического ввода точек с графических документов. Автоматические устройства на основе сканирующего блока считывают графические изображения в память ЭВМ. При использовании устройства полуавтоматического типа оператор перемещает рабочий орган (зонд) вручную по рабочей поверхности устройства (планшета), при этом координаты зонда определяются автоматически. Во всех подобного типа устройствах осуществляется фиксация считанных координат либо на внешние носители, либо непосредственно в память ЭВМ.

Обычно такие устройства состоят из планшета, зонда (в виде карандаша), алфавитно-цифровой и функциональной клавиатуры, блока индикации и обрабатывающего устройства. Современные устройства кодирования определяют координаты точки на основе сигналов между излучателем (зондом) и приемником (планшетом) и по числовым значениям характеристик принятых сигналов вычисляют координаты зонда.

При этом используются электромагнитные, световые, ультразвуковые и звуковые сигналы; точность вычисления координат составляет более десятых долей миллиметра.

Подобные устройства имеют два режима работы: дискретный — когда оператор после установки зонда в требуемой точке дает команду на определение координат зонда, и непрерывный, когда оператор перемещает зонд вдоль некоторой линии, а устройство ввода автоматически вычисляет и фиксирует координаты некоторой последовательности точек. Для идентификации объектов в состав устройств кодирования включаются различные клавиатуры и функциональные кнопки. В табл. 6.5 представлены характеристики наиболее распространенных устройств кодирования информации.

Описываемыми устройствами ввода информации обычно комплектуют "интеллектуальные" терминалы или персональные ЭВМ, которые программирует пользователь, что позволяет изменять свойства терминалов не только аппаратным путем (добавлением или модификацией блоков), но и изменением программного обеспечения данного терминала. Терминалы данного класса имеют различные возможности: начиная от терминалов,

Таблица 6.5

Наименование показателей	Тип			
	ЭМ 709	ПК ГИО	ГАРНИ	СМ 6402
Размер рабочего поля, мм	900x1200	850x618	1055x780	850x600
Координатные сетки, мм	0,1; 1; 2;	0,1	0,5	1; 1,25
Разрешающая способность, мм	0,1	0,1	—	0,1
Точность определения координат, мм	0,2	0,4	—	0,5
Толщина носителя документа, мм	—	2,5	—	1,0
Производительность (точек в мин) в режиме:				
дискретном	—	—	50	80
непрерывном	—	180	—	6000
Масса, кг	300	500	—	180
Потребляемая мощность, кВт	0,5	0,8	—	0,48

Таблица 6.6

Марка	Тип	Рабочее поле, мм	Разрешающая способность, мм	Точность позиционирования, мм	Количество цветов	Скорость, мм/с
БС-7051	Планшет	1000x1200	0,05 (0,25)	—	3	50
БС-7052	Рулон	380x600	0,1 (0,05)	—	3	200
БС-7053	"	841x1600	0,1 (0,05)	—	3	150
АП-7251	Планшет	1189x841	0,05 (0,025)	0,2	3	130 (100)
АП-7252	Рулон	594x420	0,05 (0,025)	0,2	3	280 (250)
АП-7253	"	1189x841	0,05 (0,025)	0,2	3	280 (250)
"Итекан-2М"	Планшет	594x841	0,1	—	2	70
"Итекан-4"	Рулон	841x6500	0,05 (0,025)	—	3	400 (200)
"Итекан-6"	Планшет	2000x1600 (3600x5600)	0,25	0,2 (0,4)	3	85
ЭМ 722	"	1200x1600	—	0,2	3	250
ЭМ 7022	"	1200x1600	0,1	0,15	1	250
ЭМ 732	"	1200x1600	0,025	0,2	4	1000

имеющих жесткую структуру, ограниченный набор функций (их схема управления выполняет минимум команд, а обмен данными с ВС осуществляется посимвольно), и кончая терминалами с гибкой модульной структурой, позволяющей легко модифицировать его свойства в соответствии

с конкретными задачами пользователя, обеспечивающие возможность предварительного редактирования вводимой информации, что позволяет значительно разгрузить линии связи и центральный процессор ВС, повышая таким образом ее производительность.

Наряду со специальными устройствами ввода широко используются специальные устройства вывода информации, такие, как графопостроители (они бывают рулонные и планшетные) и табло различного класса (электромеханические, на электронно-лучевых трубках, люминесцентных источниках, светодиодах и жидкокристаллических индикаторах). Характеристики основных отечественных графопостроителей приведены в табл. 6.6.

Перспективы развития описанных выше устройств ввода — вывода и предварительной обработки информации связываются в настоящее время с "интегральными" терминалами, включающими в себя различные по функциональным возможностям и комплектуемым составляющим наборы, komponуемые по спецификации заказчика.

§ 6.4. Надежность технических средств

Одна из основных причин широкого применения РВС в АСУ производством — их высокая надежность. При делении системы на ряд автономно работающих ЛВС сбой в одной машине не влечет за собой отказ всей системы. Для того чтобы система функционировала непрерывно, необходимо иметь не только резервные средства обработки, но и обеспечить надежность всей системы в целом — от датчиков до исполнительных органов, поскольку ЭВМ, получающая неверную информацию от датчиков, даже при ее полной исправности может принести больший ущерб, чем просто неисправная ЭВМ. Поэтому главная цель мероприятий по повышению надежности — обеспечение непрерывной работы системы, на которую не должны влиять ни ошибки, ни сбои.

Жизнеспособность вычислительного комплекса. Практика внедрения информационных, управляющих и других систем реального времени показывает, что недооценка жизнеспособности системы на стадии ее проектирования ведет порой к катастрофическим результатам — провалу всего проекта.

В отличие от систем пакетной обработки, рассматривавшихся ранее, к комплексам технических систем реального времени (СРВ) предъявляются дополнительные требования, связанные с особенностью данных систем, а именно: комплексы программ, работающие в реальном масштабе времени, обмениваются данными многими различными способами как в одной ЭВМ, так и по линиям связи, образуя сложные интерфейсы; сообщения поступают в систему независимо друг от друга и в случайные моменты времени; нарушение связи между программными модулями или ошибка в данных даже в одной ЭВМ могут вызвать непоправимые нарушения и не только в работе остальных входящих в вычислительную систему ЭВМ

и периферийного оборудования, но и в деятельности всего предприятия или даже объединения, эксплуатирующего данную систему управления. Тем не менее, при всей очевидной важности проблемы обеспечения жизнеспособности комплекса технических средств при проектировании и создании АСУ различного профиля, данный вопрос редко когда решается более серьезно, чем простым резервированием некоторых наиболее "ненадежных", с точки зрения разработчиков, технических средств.

Рассмотрим компоненты, определяющие жизнеспособность вычислительной системы.

Жизнеспособность является интегральной мерой возможностей системы, которая количественно связывает три следующих фактора: надежность, ремонтпригодность и технические возможности оборудования.

Надежность в приложении к ВС часто количественно определяют средним временем между отказами (СВМО) или наработкой на отказ, т.е. как ожидаемое время между ближайшими последовательными сочетаниями событий, приводящих к отказу.

Ремонтпригодность статистически выражается средним временем восстановления (СВВ), которое необходимо для того, чтобы устранить те причины, которые привели к возникновению отказа.

Технические возможности системы определяются как степень удовлетворения системой требований со стороны задач, для решения которых она предназначена.

В основе высокой жизнеспособности КТС лежит его способность "деградировать" постепенно, т.е. способность продолжать свое хотя бы частичное функционирование, несмотря на то, что со временем технические параметры устройств ухудшаются, до тех пор, пока не перестанет работать его основное ядро.

Среднее время между отказами. Используя СВМО, можно характеризовать надежность от отдельных элементов до системы в целом. При этом для оценки СВМО используют перечень приводящих к отказу событий и функцию, описывающую вероятность наступления таких событий. Надежность выражается СВМО, измеряемым в часах или его обратной величиной — частотой отказов.

По мере сборки блоков из элементов вплоть до устройства в целом все сложнее становится идентификация событий, составляющих отказ. Тем не менее в большинстве случаев можно применить эффективные меры для выяснения того, произошел отказ в системе или нет. Такие меры составляют важную часть технических условий на систему.

Основная трудность, с которой сталкивается проектировщик АСУ при определении надежности технических средств, заключается в том, что расчетные данные достоверны лишь в той степени, в какой достоверны принятые исходные значения частоты отказов элементов. Серийно выпускаемые в настоящее время элементы вычислительных систем и средств автоматизации имеют достаточно высокую надежность (например, частота отказов интегральных микросхем составляет от 0,01 до 0,4 отказа на миллион

часов работы). В силу этого достоверные данные по надежности отдельных устройств и системы могут быть получены только после длительных испытаний.

Кроме того, само понятие отказа вычислительной системы нуждается в уточнении. Различают отказы элементов системы и отказы системы с точки зрения пользователя. Данные об отказах первого типа, как было отмечено, содержатся в паспортных данных. Отказы второго типа не всегда вызываются отказами компонентов системы. Причинами системных отказов, с точки зрения пользователя, могут быть не только перемежающиеся отказы и сбои в работе компонентов, но также отказы программного обеспечения. Поэтому не всегда верна трактовка отказа системы, заключающаяся в том, что дефектный компонент дает всего один отказ, приводящий к системному, после чего он заменяется. Ниже приведен пример ситуации (табл. 6.7), когда 50 дефектных компонентов привели к 150 случаям вызова наладчиков и инженерного персонала пользователями системы, кроме того, к 50 случаям бесполезного поиска неисправных компонентов и 100 случаям замены компонентов, половина из которых на самом деле исправны.

Таблица 6.7

Событие	Число событий (на 10 000 ч)	Наработка на событие, ч
Отказы с точки зрения пользователя	350	СВМО системы – 28,6
Вызов наладчиков	150	Среднее время между обслуживанием – 66,7
Число ремонтируемых или замененных компонентов	100	Среднее время между ремонтами – 100
Число замененных компонентов действительно оказавшихся неисправными	50	СВМО компонентов – 200

Значение СВМО системы зависит в определенной мере от пользователей; некоторые из них при отказе перезапускают ее процедурами рестарта, в то время как другие прибегают к помощи наладчиков и требуют поиска неисправностей. В результате, с точки зрения пользователя, СВМО системы окажется отличным от рассчитанного разработчиком и указанного в технической документации.

Ремонтопригодность системы. Статистически ремонтпригодность выражается средним временем восстановления системы, которое зависит от контекста еще в большей степени, чем средняя наработка на отказ.

Среднее время восстановления само по себе является величиной, определяемой средними временами выполнения следующих основных операций: обнаружения факта появления отказа; выделения отказавшего эле-
240

мента; удаления отказавшего элемента; получения, замены или ремонта данного элемента; монтажа заменяемого элемента; проверки работы после замены; инициализации ВС; возобновления работы программного обеспечения эксплуатируемой системы.

Все эти операции, кажущиеся простыми, на самом деле взаимосвязаны. Например, замена отказавшего элемента может привести к отказу другого; на поиск отказа в ВС может уйти непредсказуемо долгое время, особенно если этот отказ не выявлен сразу, а повлек за собой серию лавинообразных изменений в системе программного обеспечения; выявление отказавшего элемента вызывает побочные действия, приводящие к отказу уже не одного блока, а всей вычислительной системы в целом. В частности, отключение питания на отказавшем устройстве в случае отсутствия отдельного разъема может привести к необходимости отключения стойки, в которой находится ряд исправных и не подлежащих выключению устройств; проверка работоспособности отремонтированного блока вне системы не является гарантией того, что блок является исправным; успешная инициализация системы после ремонта одного из ее блоков может подчас говорить не об успешном включении данного блока в работу всей ВС, а лишь о слабой его загрузке и т.д.

Поэтому ремонтоспособность системы зависит в первую очередь от следующих "неизмеряемых" факторов: организации обслуживания на месте, средств обслуживания, квалификации обслуживающего персонала, места расположения неисправного блока, его окружения, удобства замены блока. Все эти факторы могут вызвать заметное отклонение от среднего времени восстановления системы. Игнорировать эти кажущиеся "мелочи" проектировщик систем реального времени не имеет права.

В вычислительной системе можно довести детализацию любого из ее блоков до элементов, с которыми не происходит постепенной "деградации". Такой элемент находится всегда в одном из двух альтернативных состояний — ВКЛ или ВЫКЛ. Вероятность того, что элемент находится в состоянии ВКЛ называется *готовностью элемента*. Аналогичный параметр ВС называется *готовностью системы*.

Готовность элемента определяется формулой $P_{\text{вкл}} = \text{СВМО} / (\text{СВМО} + \text{СВВ})$. Данная формула является приближенной, тем не менее она является весьма эффективной при оценке жизнеспособности системы. Вероятность того, что элемент находится в состоянии ВЫКЛ, определяется формулой $P_{\text{выкл}} = 1 - P_{\text{вкл}} = \text{СВВ} / (\text{СВМО} + \text{СВВ})$.

Для работоспособности системы нет никакой разницы между отключением элемента из-за его выхода из строя или отключения в профилактических целях. Следовательно, СВМО и СВВ должны отражать организацию профилактических работ, предусмотренных для данной системы. Если эти времена существенно зависят от профилактических работ (например, для лентопротяжных механизмов), имеет смысл оценить это влияние.

Пусть система состоит из n элементов. Так как каждый из них может находиться в одном из двух состояний (ВКЛ или ВЫКЛ), то имеется

2^n возможных или конфигурационных состояний системы и с каждым состоянием связана вероятность нахождения ВС в этом состоянии. Для подсистемы i получим: $P_{i\text{вкл}} = \text{CBMO}_i / (\text{CBMO}_i + \text{CBV}_i)$, а $P_{i\text{выкл}} = \text{CBV}_i / (\text{CBMO}_i + \text{CBV}_i)$.

Вероятность данного состояния системы P_S определяется для независимых подсистем как произведение вероятностей для подсистем, отвечающих данному состоянию:

$$P_S = \prod_{i=1}^n \text{CBX}_i / (\text{CBV}_i + \text{CBMO}_i),$$

где

$$\text{CBX}_i = \begin{cases} \text{CBMO}_i & \text{— если подсистема } i \text{ включена в состоянии } S, \\ \text{CBV}_i & \text{— если подсистема } i \text{ выключена в состоянии } S. \end{cases}$$

Предположим, что для каждого конфигурационного состояния системы, т.е. для каждого интересующего нас параметра технических возможностей, его величина, определяемая в числовом виде, имеет особое значение. Тогда анализ жизнеспособности ВС будет состоять из следующих шагов: 1) разработка модели конфигурационного состояния ВС; 2) определение на основе элементов СВМО и СВВ коэффициента готовности для каждого конфигурационного состояния.

Функция ограничения возможностей системы. Предположим, что нас интересуют k параметров, определяющих технические возможности ВС. С каждым конфигурационным состоянием S_x связана определенная величина каждого из рассматриваемых параметров A_{kx} , с которым в свою очередь связана функция ограничения ($F_{\text{огр}}$), определяющая относительную важность соответствующего параметра. Эта функция $F_{\text{огр}} = f\{S_x; A_{kx}; \tau_x\}$, где τ_x — длительность нахождения системы в данном состоянии. Задержка выполнения функции обработки может оказаться не столь важной, пока не достигнет пороговой величины. Повышение (относительно расчетных) технических возможностей никак не учитывается, тогда как уменьшение их ниже допустимого предела резко ограничивает возможности ВС. Например, для нормального функционирования операционной системы UNIX на ЭВМ класса CM-4 требуется объем оперативной памяти — 1 Мбайт, в случае уменьшения ее до 256 Кбайт возможности данной ОС резко падают и она практически теряет все преимущества перед другими ОС, значительно более слабыми (RSX — 11 M).

На рис. 6.7 приведен вид некоторых типичных функций ограничения, где по оси X откладывается значение параметра (в нормализованном от некоторого заданного уровня виде), а по оси Y — значение параметра, определенное функцией ограничения.

На рис. 6.7, а приведена функция ограничения параметра, для которого превышение технических возможностей игнорируется по каждому их уровню вплоть до максимума, которому присваивается полный вес. Например, дисковая память емкостью 29 Мбайт не дает никаких практических преимуществ, если для создания базы данных системы со всеми словарями необходимо 10 Мбайт.

На рис. 6.7, б приведена функция ограничения для нежелательного параметра. Номинальному значению соответствует полное отсутствие данного фактора. Примером такого параметра является время реакции в системе, которое в идеальном случае должно выражаться несколькими миллисекундами. В случае его увеличения до величины, вызывающей у пользователя нежелание работать с предлагаемой ему системой, дальнейшее снижение значения функций ограничения приостанавливается.

На рис. 6.7, в приведена функция ограничения параметра, который получает положительную оценку лишь по достижении определенной минимальной величины (например, система сервоуправления, где для обеспечения стабильности каждую секунду необходимо выполнять минимальное число итераций).

На рис. 6.7, г приведен параметр с функцией ограничения типа "окно". Примером может служить устройство построчной печати, при уменьшении скорости печати которого до некоторой минимальной величины его полезность равна нулю, поскольку для этой скорости печати существуют знакосинтезирующие устройства, значительно более дешевые. Если же устройство печатает с очень большой скоростью все время, то это также не имеет смысла, поскольку гору бумаги, которую в состоянии напечатать современное устройство печати даже за 1 ч непрерывной работы, невозможно просмотреть в приемлемое время ни одному пользователю.

На рис. 6.7, д приведена двоичная функция ограничения, которая характеризуется тем, что, как только рассматриваемый параметр превзойдет заданный минимальный уровень, функция ограничения сразу приобретает максимальное значение. Пример такой системы — синхронная передача данных: при стыковке дисплея с ЭВМ совершенно неважно, что линия может работать со скоростью в несколько млн. бод, поскольку дисплейный интерфейс работает со значительно меньшей скоростью (9600 бод), а на любой другой скорости передачи данных возникают искажения.

Приведенные примеры не описывают весь возможный диапазон функций ограничения, которая может быть непрерывной, дискретной, разрывной, нелинейной и т.д. Кроме того, $F_{огр}$ может меняться с течением жизни вычислительной системы или программного обеспечения, реализованного на данной ВС.

Распределение уровней технической возможности КТС. Поскольку каждому конфигурационному состоянию системы ставится в соответствие некоторая вероятность, а каждому параметру технических возможностей — некоторая величина, то вероятность достижения любого заданного уровня

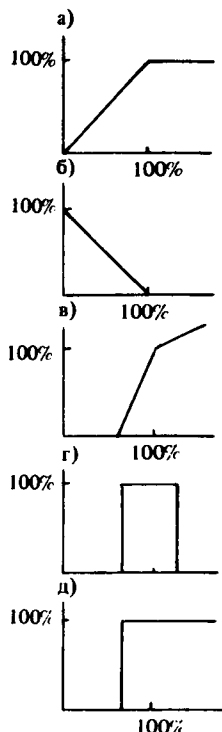


Рис. 6.7



Рис. 6.8

технической возможности определяется как сумма вероятностей пребывания в каждом из состояний с таким уровнем. Как правило, состояний, в которых достигается данный уровень некоторой технической возможности, достаточно много.

Суммируя вероятности, относящиеся к каждому уровню параметра технических возможностей, можно построить функцию распределения вероятностей, подобную изображенной на рис. 6.8. Практически такая функция не может быть непрерывной, однако в таком виде ею значительно легче пользоваться.

Для каждого параметра технических возможностей имеется по одному такому распределению. Система, у которой отсутствует постепенная "деградация", на таком графике будет представлена двумя точками. Система более общего вида характеризуется подобной кривой, и именно это распределение описывает жизнеспособность системы.

Математическое ожидание распределения выражает ожидаемое значение рассматриваемого параметра, получаемое усреднением по всей совокупности возможных состояний. Отношение среднего значения параметра к максимальному называется *эффективной жизнеспособностью* ВС. Второй момент этого распределения характеризует спорадичность поведения системы, третий — показывает, насколько может ухудшиться положение перед окончательным выходом ВС из строя.

Показанное на рис. 6.8 распределение типично для системы с хорошей жизнеспособностью. Чем больше μ , тем меньше σ , и чем меньше пик при нулевом уровне рассматриваемого параметра, тем лучше качество системы. Подъем вблизи нуля показывает, что ВС имеет больше состояний, в которых она не работоспособна, чем это определяется уровнем технических возможностей.

Еще более удобный способ представления данных о жизнеспособности системы — построение кривых распределения интегральной вероятности различных параметров технических возможностей. В результате получим кривые, показанные на рис. 6.9. Кривая А типична для систем с постепенной

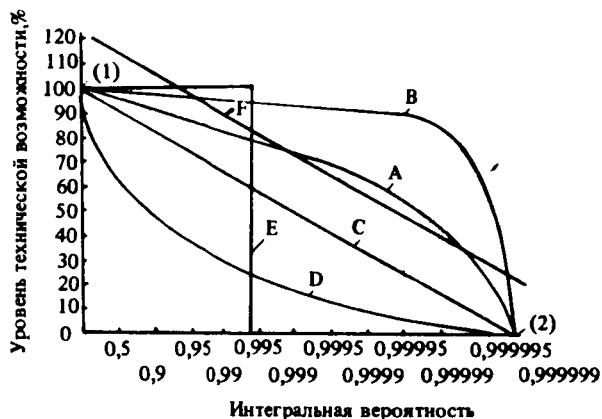


Рис. 6.9

деградацией; *B* — отражает более высокий уровень жизнеспособности; кривая *D* характерна для ВС, в которой высокая надежность отдельных блоков сочетается с малой собственной жизнеспособностью; *E* — отражает работу системы в режиме двоичного отказа при нахождении точки разрыва далеко от начала, такая система близка к идеальной; кривая *F* является иллюстрацией получения высокой жизнеспособности за счет большого запаса надежности.

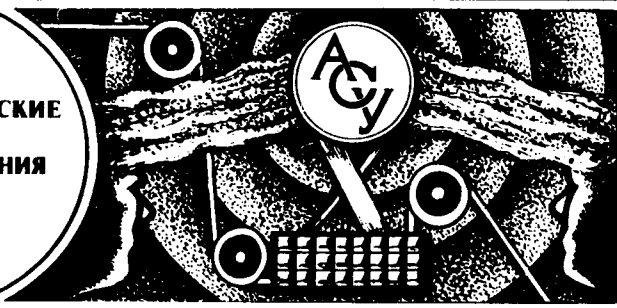
Необходимая в каждом случае форма кривой зависит от характера применения ВС, что должно быть оговорено в технических условиях на нее. Следует задавать, по крайней мере, два уровня каждого параметра технических возможностей, а также значение вероятности для каждого из этих уровней.

В настоящее время эффективное значение жизнеспособности, равное 0,99, в случае достаточно больших систем обеспечивается при условии минимальной избыточности (20–30%) при весьма скромных требованиях к среднему времени между отказами в сотни часов и среднему времени восстановления, составляющему десятки минут. Надо обратить внимание на то, что СВМО в данном случае рассматривается при довольно высоком (90%) уровне достоверности, что не часто бывает на практике.

ПСИХОЛОГИЧЕСКИЕ АСПЕКТЫ ПРОЕКТИРОВАНИЯ

ГЛАВА

7



Автоматизированные системы управления имеют существенное отличие от технических систем, заключающееся в необходимости учета психологии человека. Это относится в равной степени к психологии руководителя, использующего результаты решения задач в АСУ; пользователя, обращающегося к услугам АСУ, но не являющегося ее участником; самого разработчика.

Достаточно развитая инженерная психология, определяющая поведение и характеристику человека при взаимодействии с техническими средствами, имеет лишь очень ограниченное отношение к работе человека в АСУ.

Разработчикам необходимо уделять психологическим аспектам особое внимание, тщательно продумывая, как будут практически реализованы проектные решения в действующей системе с учетом человеческого фактора. Большое значение при этом имеют психологические тесты и эксперименты, предварительное "проигрывание" возможных реальных ситуаций.

Без совместной работы с психологами провести такое исследование на хорошем уровне трудно, если не невозможно, однако для достижения успеха специалист-системотехник должен и сам обладать определенным уровнем знаний в этой области, сочетая методы психологического и системного анализа.

§ 7.1. Принятие решений руководителем

В административно-организационных системах наряду с формальными критериями и оценками эффективности функционирования предприятия существуют различные неформальные оценки деятельности руководителя и качества принимаемых им решений. При этом не исключена возможность существенных противоречий между этими оценками. Технически грамотный руководитель может не обладать достаточными волевыми

качествами, умением настоять на быстром и четком исполнении принятых им решений. Бывает и обратная ситуация, когда хороший администратор и организатор коллектива оказывается не в должной мере владеющим всеми тонкостями технологии руководимого им предприятия. Но и в максимально благоприятном случае, когда руководитель во всех аспектах соответствует своей должности, существует различие между оценкой руководителя и реальной эффективностью принимаемых им решений.

Стремление повысить свою оценку приводит к некоторому давлению, иногда не вполне осознанному, определенному перекосу принимаемых решений в сторону повышения оценок, хотя бы в ущерб действительной эффективности организации. Иногда на этой основе возникают уродливые явления — попытки прямого или скрытого обмана, приписки, искусственное занижение возможностей с целью получить ненапряженный план и т.д.

Большое психологическое значение имеют стереотипы, носящие характер привычных лозунгов типа "план — досрочно", "выпустить сверх плана" и т.п. При этом не анализируется, нужна ли в действительности сверхплановая продукция, готовы ли потребители принять ее досрочно. Несмотря на очевидную вредность такого подхода, во многих случаях руководители руководствуются не производственной необходимостью, а подобными стереотипами.

Затраты любых видов ресурсов в системе управления существенно меньше, чем в управляемой. Простой поворот выключателя приводит в действие мощный электродвигатель, небольшое изменение нажима на педаль газа заметно ускоряет движение многотонного грузовика, устное распоряжение директора приводит к перемене условий работы десятков и сотен людей. Происходит как бы многократное усиление энергии управляющего воздействия.

В этих условиях иногда возникает иллюзия легкости реализации решений, особенно в организационно-административных системах управления. Принятие решения или изменение уже принятого решения на другое, возможно противоположное, дается лицу, принимающему решение, относительно малой ценой. Затраты ресурсов в управляемой системе, связанные с реализацией принятого решения, при этом, как правило, недооцениваются. Кажется, что осуществить решение ненамного труднее, чем его принять. Особенно часто в таких случаях учитывают только прямые затраты, игнорируя побочные эффекты. Между тем дополнительные затраты, связанные с такими побочными эффектами, иногда настолько превышают основные, что приходится отменять ранее принятое решение. Поэтому одним из условий осуществимости принятых решений является тщательный анализ необходимых ресурсов, реальности их получения и использования с учетом как основных, так и дополнительных затрат на реализацию решений.

Управляемые системы, как правило, являются динамическими. Их состояние, свойства и характеристики меняются во времени. Система управления должна учитывать динамичность управляемой системы. Реше-

ние должно быть принято в такой момент времени, чтобы его реализация обеспечила желаемое изменение состояния системы не вообще, а к определенному сроку.

В технических системах время реакции системы, т.е. интервал времени от момента выдачи управляющего воздействия до перехода системы в соответствующее этому воздействию новое состояние, обычно детерминировано или имеет небольшой разброс. Это позволяет определить предельно допустимый срок принятия решения.

В организационно-административных системах время реализации решения чаще всего может меняться в достаточно широких пределах. Для ускорения выполнения решения можно выделить больше людей, создать им условия для более интенсивной работы, ввести сверхурочные часы и т.д. Такими мерами можно компенсировать задержку принятия решения. Поэтому окончательный срок, когда решение должно быть принято, гораздо менее критичен. Это приводит к тому, что принятие решения многократно откладывают. Причем чем сложнее проблема, чем труднее и важнее ее решение, тем более оно затягивается. При этом игнорируется или недооценивается то обстоятельство, что неизбежная необходимость ускорить реализацию решения приводит к снижению качества и повышению риска задержки реализации сверх допустимого срока.

В то же время не следует принимать решение слишком рано, намного ранее срока, необходимого для его реализации. В таких случаях, как правило, начало реализации решения откладывают до тех пор, пока это представляется возможным. Другими словами, исполнитель решает другую проблему — когда надо начинать реализацию решения. Здесь возникает опасность чрезмерной задержки начала реализации решения со всеми вытекающими из этого последствиями. Нежелательность слишком раннего принятия решения объясняется не только появлением избыточного интервала между принятием решения и началом его реализации, ведущим, как правило, к задержке реализации. Не менее важно, что в этом интервале может поступить новая входная информация, содержание которой требует коренного пересмотра уже принятого решения. Однако наличие готового решения часто приводит к нежеланию его пересмотра или частичному пересмотру, в результате чего дополнительные сведения в большей или меньшей степени игнорируются.

Случайные воздействия, поступающие как одновременно с управляющими воздействиями, так и в течение всего времени движения системы в интервале между двумя последовательными моментами их выдачи, отклоняют движение системы от выбранной траектории. Чем полнее сведения о возможных случайных воздействиях внешней среды и чем лучше известно, как поведет себя система под их влиянием, тем более правильно могут

быть выбраны управляющие воздействия. Необходимо не только иметь сведения о поведении и местонахождении системы в пространстве состояний до и к моменту получения ею управляющих воздействий, но по воз-

248

возможности и прогнозировать как состояние внешней среды, так и поведение системы под ее влиянием в будущем.

Трудно представить себе автомобиль, которым водитель управляет, глядя назад и видя путь, который пройден автомобилем, а не тот, по которому он движется. При движении задним ходом водитель всегда смотрит в зеркало заднего вида, оборачивается назад, высовывается из кабины, чтобы видеть, куда он подает автомобиль. В случаях затруднений обзора участка пути, по которому предстоит двигаться автомобилю, правила дорожного движения предписывают обратиться к помощи второго лица, которое помогает водителю, наблюдая за состоянием участка. Несоблюдение этих предосторожностей может привести к авариям и несчастным случаям.

Между тем такая ситуация зачастую встречается в системах управления производством или другими экономическими объектами. Сотрудники управления такими системами иногда не имеют четкого представления о лежащем впереди пути, будущем состоянии внешней среды, возможных воздействиях с ее стороны на управляемую систему, о том, какова будет в этих случаях ее реакция. Приходится полагаться на информацию об уже совершившихся событиях, поступающую, к тому же, зачастую не в полном объеме, с задержкой во времени и не всегда достоверную. Существующие автоматизированные системы ориентированы в основном на задачи учета и отчетности, а не на прогнозирование; на выдачу информации о том, что происходило в прошлом, а не что произойдет в будущем. Несмотря на трудность прогнозирования будущего, ограниченную точность существующих методов, без решения этих задач трудно осуществлять высокоэффективное управление.

Значение информации в системах управления неразрывно связано с ее достоверностью и своевременностью. Любые искажения поступающей в систему информации приводят к тому, что принимаемые решения являются в большей или меньшей степени ошибочными. Такие искажения могут быть как случайными, так и преднамеренными. В последнем случае эти сведения называют дезинформацией.

Этапы принятия решения. Принятие решения не является каким-то единовременным, обособленным актом. Это процесс, протекающий во времени и состоящий из нескольких этапов.

Первым этапом этого процесса является осознание того состояния или ситуации, в которой находятся управляемая система и принимающий решение человек. В общем случае можно говорить о накоплении системой управления определенного объема данных о состоянии управляемой системы и ее внешней среды. Начальный этап можно считать в определенном смысле предварительным, предшествующим процессу решения. На этом этапе выявляется удовлетворенность или неудовлетворенность тем состоянием, в котором находится система. В случае удовлетворенности, как указывалось выше, проблема принятия решения не существует, если не считать решения "не принимать никаких решений". Разумеется, может

возникнуть необходимость принятия решений, направленных на сохранение существующего удовлетворительного состояния. Однако выявление такой ситуации, ее осознание тоже входят в содержание рассматриваемого начального этапа.

Вторым этапом формируется желание изменить (или сохранить) существующее состояние системы определенным образом. Иными словами, устанавливается некоторая цель, которую желательно достичь. Это важнейший этап, на котором в результате анализа существующего состояния, причин неудовлетворенности им синтезируется представление о некотором ином состоянии, в которое желательно перейти, являющимся целью. Для принятия решения человеком или системой управления совершенно необходимо существование желания осуществить какие-то действия для достижения сформулированной цели. Без определения цели такое желание сформулировано быть не может. Даже когда в сказке говорится: "Пойди туда, не знаю куда, принеси то, не знаю что", герой чудесным образом догадывается о скрытом смысле этих неопределенных слов, о той цели, которая за ними кроется, идет куда надо, удовлетворяя высказанное в такой форме вполне определенное желание посылающего его лица.

Третий этап принятия решения состоит в определении всех возможных способов или путей достижения цели, перехода в желаемое состояние. Здесь важно в максимальной степени обеспечить полноту этого множества возможных решений вплоть до допустимости его избыточности. Лучше включить для последующего рассмотрения несколько решений, кажущихся на первый взгляд непривлекательными, чем пропустить хотя бы одно действительно эффективное решение. Потери, связанные с анализом большого числа решений, чаще всего несоизмеримо меньше возможных потерь системы в результате случайного исключения решения, которое на самом деле является наилучшим.

Четвертый этап заключается в выборе из множества возможных решений наилучшего в смысле эффективности достижения цели с соблюдением при этом некоторых правил выбора. Результатом именно этого этапа является единственное принятое решение. В этом смысле можно говорить, что в нем и заключается собственно акт принятия решения. Однако хотя этот этап действительно центральный, без трех предыдущих этапов его осуществить невозможно, они все вместе составляют единый процесс. Иногда три первых этапа называют *подготовкой решения*, а четвертый — *актом принятия решения*.

Завершает весь процесс заключительный, пятый этап — реализация принятого решения. Хотя этот этап следует за уже принятым решением, он включается в процесс решения. Без него решение теряет смысл. Если управление системой продолжается, что происходит всегда, пока цель не достигнута, пятый этап замыкает цикл управления. В результате реализации принятого решения изменяется состояние системы. Информация об этом новом состоянии, его осознание являются начальными этапами следующего цикла.

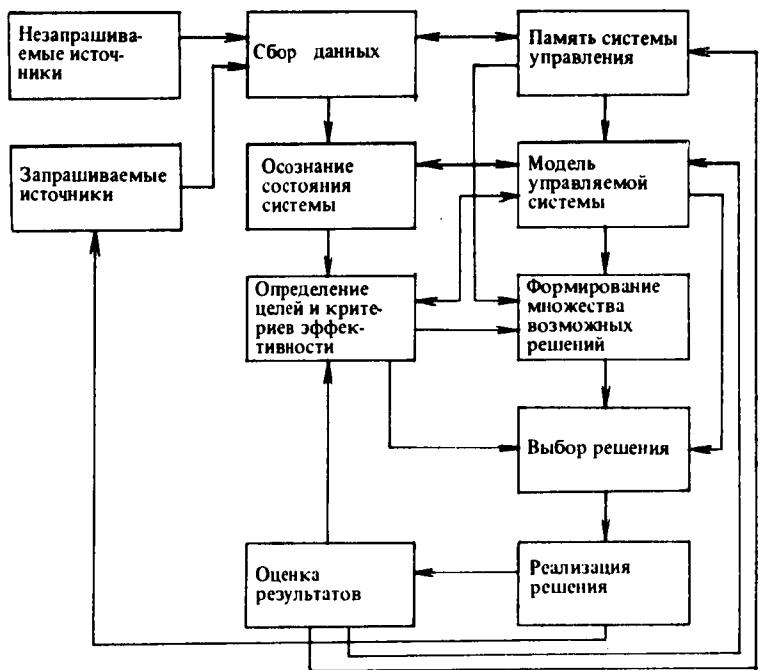


Рис. 7.1

На рис. 7.1 приведена схема функционирования системы управления. Сбор данных осуществляется от запрашиваемых и незапрашиваемых источников. Первая группа включает источники, связанные с управляемой системой, и сообщает сведения о результатах реализации предыдущего решения. Собранные данные запоминаются системой управления для последующей обработки и использования.

Для анализа поведения управляемой системы, проверки реакции на предполагаемые решения и других целей создается модель управляемой системы, имеющая двусторонние связи с памятью системы управления и блоком осознания состояния системы. Результаты, полученные в этом блоке, вместе с оценкой результатов реализации предыдущего решения и информацией о модели управляемой системы используются для определения текущих целей и критериев эффективности. В свою очередь цели и критерии вместе с данными о модели управляемой системы и необходимыми сведениями из памяти позволяют сформировать множество возможных решений. На основе этого множества, зная цели и критерии эффективности и используя модель управляемой системы, осуществляется выбор решения. Полученное решение затем реализуется, а результаты реализации оцениваются. Сведения о реализации решения вновь поступают на вход системы, а оценки результатов запоминаются и используются для определе-

ния целей и критериев эффективности в следующем цикле управления. Наличие памяти и модели позволяет анализировать поведение системы за длительный период времени, включающий ряд циклов управления. Такой анализ позволяет уточнить определение целей и критериев эффективности, облегчает формирование множества возможных решений, поскольку многие из них просто хранятся в памяти, и создает условия для корректировки модели.

Приведенная схема отражает процесс принятия решений как сложной системой управления, так и отдельным человеком. В последнем случае все этапы принятия решения происходят в его сознании. Выяснив (что-либо) с помощью своих органов чувств и осознав ситуацию, в которой он находится, человек сознательно или подсознательно моделирует свое поведение. Он определяет цели и критерии эффективности или, попросту говоря, выясняет для себя, чего же он хочет и что он понимает под хорошим результатом своих действий. Обращаясь к памяти, он вспоминает, какие решения принимал в подобных ситуациях, старательно обдумывает, какие действия он мог бы предпринять для получения желаемого результата. Сравнивая возможные пути, оценивает их эффективность и, останавливаясь на одном из них, начинает его реализацию.

Рассмотренная схема процесса принятия решений не только весьма условна, но и сознательно упрощена, чтобы сконцентрировать внимание на основных этапах. На самом деле каждый из этапов представляет собой сложный комплекс процедур, связи между которыми являются не столь определенными.

Наличие случайных воздействий на управляемую систему приводит к тому, что результат управляющих воздействий не детерминирован. Конечно, система управления должна учитывать возможные влияния случайных воздействий, но именно в силу их случайного характера сделать это в должной степени не всегда удастся. Дополнительные затруднения в оценке качества управления возникают также из-за временных задержек между управляющими воздействиями и реакцией на них. Поэтому оценка качества управления сложной системой по эффективности ее функционирования может быть получена, как правило, лишь по усредненным за достаточно продолжительный период данным. Однако за продолжительное время происходят существенные изменения как в системе управления, так и в управляемой системе, что еще более затрудняет оценку изменения их функционирования во времени.

Если вспомнить об эффекте усиления, присущем каждой системе управления, то становится ясно, что небольшие изменения качества управления приводят к существенным изменениям в функционировании управляемой системы. Не так важно, как изменится фонд заработной платы системы управления при сокращении или увеличении ее штата на одного человека, гораздо важнее определить, к каким изменениям эффективности управляемой системы это приведет.

Связь эта редко бывает очевидной, в действительности ухудшение

качества управления при сокращении одного-двух человек в системе управления в явном виде установить трудно, если не невозможно, а прямые затраты на их зарплату сокращаются, что и определяет стремление к сокращению аппарата управления. Следует отметить, что это стремление в большинстве случаев вполне оправдано, ибо любой живой организм, каким является и система управления, обладает тенденцией к росту, далеко не всегда сопровождающемуся пропорциональным увеличением его возможностей. Более того, чрезмерный рост аппарата управления не только увеличивает прямые затраты на его содержание, но и снижает эффективность управления системы. Поэтому всегда существуют нижняя и верхняя границы размера системы управления, за пределами которых эффективность управления снижается. Вся трудность заключается в том, что границы эти размыты и отстоят достаточно далеко друг от друга, а внутри них эффективность функционирования управляемой системы не критична к числу сотрудников.

Психологические аспекты принятия решений. Человеку биологически свойственно очень осторожно принимать решения. Нерешительный человек опасается всего: он не уверен, будут ли получены именно те результаты, на которые рассчитано принимаемое решение; насколько полезными эти результаты окажутся; какие трудности придется преодолевать на пути реализации решения; не будет ли побочных нежелательных последствий, например в виде осуждения людей, и т.п. Чем более важной является решаемая проблема, чем значительнее последствия принимаемого решения, чем к большим изменениям существующего состояния оно может привести, тем больше неуверенность, тем труднее принять решение. Известно, что на деловых совещаниях очень часто решение наиболее важных вопросов стремятся отложить, передать в специально создаваемую комиссию, под любым предлогом уклониться от немедленного решения. При этом забывают, что отсутствие решения во многих случаях гораздо хуже пусть даже не оптимального решения. Нерешительность, длительные колебания, имеющие следствием затяжку решения на неопределенный срок, могут приводить к катастрофическим последствиям. Буриданов осел, стоя между двумя одинаковыми стогами сена, не мог решить, с какого из них начать, и в результате умер от голода.

Как правило, в молодости человек бывает более решительным. Он энергично ищет друзей, интересную работу, хорошее место жительства, готов переехать в любую другую часть страны, переменить профессию, коренным образом изменить свой образ жизни, стремится познать новое в самых различных областях знаний и человеческой культуры. С годами человек приобретает определенные привычки и навыки, с которыми ему трудно расстаться, стремится сохранить и ограничить круг друзей и знакомых, не хочет менять свою квартиру на другой район даже в пределах одного города. Ему психологически свойственно подсознательное чувство сопротивления всему новому, неизвестному, что может изменить уже достигнутое им положение. В ряде случаев он отказывается принимать да-

же такие решения, которые в принципе могут улучшить его положение.

В какой-то степени это объясняется накапливаемым с годами опытом. Человеку свойственно в гораздо большей степени учиться на собственных ошибках, чем воспринимать чужой опыт. Ребенок, несмотря на все наставления старших, пока не обожжется, серьезно не опасается горячей плиты. Правила дорожного движения представляют собой сконцентрированный и осмысленный опыт печальных происшествий многих поколений водителей, но молодой по стажу и по возрасту автомобилист ездит лихо и позволяет себе не очень с ними считаться. Если же он в результате нарушения какого-либо правила попадает в неприятную ситуацию, то обычно начинает именно это (и только это) правило соблюдать. Лишь после нескольких таких случаев он понимает, что надо соблюдать правила полностью.

Принимая решение, человек убежден в его правильности, т.е. в том, что оно обеспечит желаемый результат. Поэтому если фактический результат совпадает с ожидаемым, это считается естественным, так и должно было случиться. Само собой разумеется, что при этом не надо анализировать свое решение, возвращаться к мотивам, по которым оно было принято, тем более что уже ждут своего разрешения новые проблемы. Но если фактический результат не совпадает с желаемым, противоположен ему — тут не до новых задач. Начинаются мучительные переживания неудачи, анализ ее причин, поиск людей, которых можно было бы в ней обвинить, и затем обдумывание новых возможных решений для достижения желаемого результата, выбор наилучшего из них.

Таким образом, ошибки воспринимаются гораздо более остро, чем оказавшиеся правильными решения, а потому и запоминаются сильнее, заставляя по мере их накопления все осторожнее и осторожнее принимать решения. Одинаковая по содержанию неудача воспринимается гораздо легче в молодости, когда все впереди и можно начинать жизнь сызнова, и ранит намного сильнее в зрелом или пожилом возрасте. В юности при выборе решения часто берут верх эмоции, а в зрелости — рассудительность. Эти и другие причины приводят к тому, что человек с возрастом становится осторожнее, осмотрительнее, больше опасается ошибочных решений и старается не принимать решений, которые могут привести к изменению существующей обстановки.

В этом заключается одно из диалектических противоречий в развитии человечества — стремление отдельного человека к стабильности, сохранению такого достигнутого им положения, которое он сам считает для себя удовлетворительным, а человеческого общества в целом — к непрерывному развитию и совершенствованию.

Опасения и колебания при принятии решения человеком чаще всего неоправданы. Действительно, если есть несколько возможных решений, близких по степени достижения цели, вероятности осуществления желаемого результата, то можно спокойно принять просто любое из них. Если оно

даже не наилучшее, то проигрыш будет не настолько велик, чтобы вызвать серьезные сожаления. Если же среди возможных решений есть одно более предпочтительное, то его и следует принять. Конечно, задача затрудняется необходимостью оценить качество решения по конечному результату, который будет получен при его реализации, что трудно сделать в наш динамичный, существенно недетерминированный век.

В обыденной жизни, где велика неопределенность и влияние случайных факторов, это, быть может, сделать еще труднее, чем в деловой, служебной области. Здесь часто приходится иметь дело с другими людьми, поведение которых слабо регламентировано — лишь пределами закона и общественной морали. Человек представляет собой чрезвычайно сложную систему, закономерности его действий в житейских обстоятельствах, мотивация поведения во многих ситуациях слабо изучены и плохо предсказуемы. Принимая решение, связанное с другим человеком, мы ожидаем от него одного, а он ведет себя совсем по-другому. Вступая в брак, каждый питает некоторые иллюзии о том, как будет развиваться совместная семейная жизнь. Если эти иллюзии обеих сторон оказываются различными и ни одна из них не совпадает с реальной действительностью, брак распадается.

Служебная деятельность несколько более регламентирована, но и в ней участвуют другие люди, а случайные факторы определяются не только наличием людей, но и многими другими причинами.

Если обратиться к схеме процесса принятия решения, рассмотренной выше (см. рис. 7.1), причины психологической осторожности становятся яснее.

В случае, когда однотипные решения принимаются достаточно часто, сведения о них и об их реализации свежи в памяти, сбор данных налажен, модель управляемой системы сформирована, проверена и соответствует действительности. Осознание состояния системы не вызывает затруднений, цель и критерии эффективности сформированы на предыдущих этапах и являются удовлетворительными. Множество возможных решений практически сформировано. В такой ситуации расхождения с оптимальным решением на каждом этапе невелики, выигрыш в среднем удовлетворителен, в противном случае вносятся изменения во все перечисленные блоки схемы принятия решения. Это означает, что выбор решения на каждом этапе не вызывает особых затруднений.

Если же решаемая проблема представляет собой единичный случай или ее надо решать очень редко, или она решается впервые, то все обстоит совсем по-другому. Неясно, как организовать сбор данных — какие источники запрашивать, какие данные, поступающие от незапрашиваемых источников, использовать. Неясно, какие из содержащихся в памяти сведений могут и должны быть использованы при решении проблемы, а модель предстоит создавать и она неизбежно является весьма грубой, очень приближенной представляет действительность. Множество возможных решений надо каким-то образом формировать. Неопределенность критерия затрудняет получение оценок результативности решений. Все это заставляет действо-

вать очень осторожно. Прежде чем окончательно принять решение, проверяется, правильно ли осознано состояние системы, собираются дополнительные данные, уточняющие модель, что, в свою очередь, требует новых данных. Возможные решения проверяются на модели, условно оценивается их эффективность.

§ 7.2. Психологические факторы при поиске новых решений в процессе проектирования

Формирование множества возможных решений является одним из наиболее творческих этапов, требующих больших интеллектуальных усилий. Ясно, что чем больше первоначально сформированное множество, тем больше вероятность того, что в нем содержатся наиболее эффективные решения. Недостаточно эффективные решения легко потом отбросить, гораздо труднее ввести новые, более эффективные, хотя бы потому, что надо каким-то способом узнать, что это надо сделать.

Сент-Экзюпери сказал, что вещь становится совершенной не тогда, когда к ней нельзя уже ничего прибавить, а когда от нее уже ничего нельзя отнять. В применении к системным исследованиям это означает, что исходное множество, будь то множество возможных решений, входных или выходных параметров, существенных взаимосвязей или какое-либо иное, сначала должно быть как можно более полным, чтобы в дальнейшем не потребовалось в него ничего добавлять. Затем из него последовательным анализом удаляются элементы, не существенные для решения рассматриваемой задачи, пока не останется только то, что нужно. Например, при оптимизации многокритериальных систем это может быть множество решений, оптимальных по Парето. Оно является совершенным в смысле сентенции Сент-Экзюпери, поскольку из него уже больше ничего нельзя исключить. Попытки действовать в обратной последовательности — сначала найти один элемент, который должен войти в конечный вариант искомого множества, затем добавить еще один и т.д.; как правило, приводят к неудаче.

Поиск принципиально новых методов решения для включения в исходное множество является творческим процессом, их надо изобрести, придумать. Если их можно извлечь из какой-либо памяти, они уже не являются принципиально новыми.

Существует ряд психологических факторов, препятствующих или способствующих успеху творческого мышления, выходящих за рамки решения задач управления.

Когда задача сформулирована и накоплены необходимые для ее решения знания, начинается интенсивный мыслительный процесс. Для серьезных задач, которые не удастся решить сразу, как говорят, "с ходу", концентрированные усилия над решением задачи должны обязательно чередоваться с расслаблением, полным отвлечением от ее решения. Такая передышка может быть заполнена чем угодно — физической нагрузкой, сном,

”ничего неделанием”, решением не менее трудной, но обязательно совершенно иной по смыслу и методам решения задачи; важно, что при этом сознание полностью отключается от решаемой задачи, начисто ”забывает” о ней. Предполагается, что во время этих перерывов решение задачи продолжается в подсознательной области. Может быть, поэтому после отдыха мысль работает острее, чем раньше, открываются новые аспекты задачи и возможности ее решения. Во всяком случае, без такой периодической смены этапов интенсивного мышления и отключения трудно добиться успеха — мысль работает все медленнее и наконец ”зацикливается”, полностью перестает генерировать новые варианты, тупо повторяя одно и то же.

Одним из главных препятствий на пути получения нового решения является *психологическая инерция*. Она заключается в том, что у человека, когда он знакомится с условиями задачи, возникает некоторая модель, образ задачи, мысленные условия, не совпадающие с фактической задачей и ее условиями. Это происходит потому, что формулировка задачи построена так, что вызывает в памяти некоторый стереотип, знакомый образ, от которого уже трудно отказаться. Иногда формулировка задачи просто настраивает человека на определенный мыслительный процесс, и он уже не может перестроиться.

Психологическая инерция в определенной степени является следствием обучения, хорошего знакомства с каким-либо предметом. Детально зная и многократно используя определенные объекты или процессы, человек применяет их в ситуации, кажущейся ему аналогичной этим хорошо знакомым задачам, не обращая внимания на то, что начальные условия являются совсем другими. На этом основан *эффект функциональной устойчивости*, когда используют предмет в одном определенном качестве, не задумываясь о том, что у него может быть иное применение.

Психологическая инерция усиливается при ряде обстоятельств: нетерпении, опасности, в состоянии напряжения, при волнении. Усиливают ее также отвлекающие действия. Она как щоры, сужает область, в которой ищется решение, не позволяя найти лежащее за ее пределами простое и эффективное решение.

Как же бороться с психологической инерцией? Нельзя ведь советовать забыть все, что знаешь, или при решении любой задачи никогда не прибегать к ранее известным, привычным методам. Вместе с тем преодолеть психологическую инерцию можно довольно просто: надо помнить, что она существует, что мы ей подвержены. Решив задачу или думая, что она решена, надо сопоставить решение с условием — действительно ли в условии задачи содержатся те ограничения, которые мы старались выполнить в процессе решения, проверить, нет ли принципиально иных подходов к ее решению.

Найти новый эффективный метод, новое решение помогает так называемая *инверсия*. Она заключается в том, что известное решение какой-либо проблемы надо попытаться изменить на прямо противоположное.

Не один век баржи тянули бурлаки, затем мощные буксиры. Между тем оказалось, что гораздо эффективнее их толкать, жестко закрепив буксир сзади баржи. Аналогично этому авиаконструкторы привычно располагали двигатели в передней части самолета, чтобы они тянули его вперед. Между тем расположение двигателей в хвосте, толкающих самолет вперед, дало определенные преимущества. В пишущих машинках буква через красящую ленту прижимается к бумаге, оставляя отпечаток. Хорошее быстродействие выходных печатающих устройств ЭВМ удалось получить при прямо противоположном действии — бумага прижимается к букве. Перечень таких примеров удачных решений с помощью инверсии можно продолжать очень долго.

Для поиска новых решений полезна *аналогия* — использование аналогичных ситуаций, причем главным образом не в похожих задачах, а в далеких областях, не имеющих, как представляется с первого взгляда, ничего общего с решаемой задачей. Многие инженерные решения представляют собой аналоги объектов или процессов в живой природе. Считается, что одним из величайших изобретений человека является колесо с осью, потому что ему нет аналогов в живой природе.

Особенно эффективна аналогия, когда мысль о ее наличии и возможности использования возникает неожиданно, как *п р о з р е н и е*, при взгляде на совершенно иное по природе явление, в беседе на отвлеченные темы, при чтении художественной литературы. Так делаются многие открытия. Чаще всего такой случай возникает, когда человек увлечен решением сложной задачи, непрерывно о ней думает. По-видимому, при отвлечениях подсознательно продолжают поиски решения, что позволяет его увидеть в совершенно новой обстановке.

Другое явление, полезное при генерировании новых идей, особенно при разработке организационно-административных систем, называют *эмпатией*. Под этим понимают отождествление личности одного человека с личностью другого, проникновение его в чувства другого лица. В обыденной жизни это явление используют, когда просят: "войди в мое положение" или "представь себя на моем месте". Эмпатия требует от человека умения войти в образ, мыслить и чувствовать себя другим человеком. Для этого нужна природная одаренность и нескованность, некоторое специальное обучение и тренировка. Надо уметь думать не о себе, а о другом, аналогично тому, как хороший рыбак думает не о том, какой вкусной будет для него уха после рыбалки, а будет ли вкусным для рыбки червяк, насаживаемый на крючок. Наиболее простым случаем является отождествление разработчиком организационно-административной системы управления себя с сотрудником будущей системы или ее клиентом.

Более сложны так называемые *деловые игры*, получившие широкое распространение за рубежом и в некоторых областях управления в нашей стране, в частности в строительстве. Участники деловой игры распределяют между собой "роли" сотрудников некой системы управления: директора, начальников различных отделов и цехов, а также заказчиков и

258

поставщиков. Получая вводные условия, они начинают управлять жизнедеятельностью предприятия, которое в игре моделируется комплексом программ, решаемых на ЭВМ. Этот комплекс программ имитирует реакцию отдельных участков или всего предприятия в целом на решения, принимаемые участниками игры. Эта реакция анализируется участниками, которые принимают новые решения, каждый раз получая поощрительные или штрафные очки, пока не становится ясной степень эффективности управляющих воздействий каждого участника игры. Необходимым условием успешного проведения игры является как можно более полное вхождение каждого ее участника в образ того сотрудника, роль которого он исполняет.

Целовые игры успешно используют в процессе обучения администраторов, в меньшей степени — для оценки степени подготовленности и квалификации участников и лишь иногда — для моделирования новых ситуаций и поиска решений.

Большое значение при поиске принципиально новых решений имеет хорошо развитое чувство воображения, элементы фантазии. Полезно представить себе идеальную ситуацию, когда сняты ограничения на ресурсы, есть возможность использовать технические средства с любыми возможностями, в том числе реально не существующие. Довольно часто решения, найденные для такой идеальной системы, в большей или меньшей степени могут быть реализованы и в обычной. Иногда таким способом удается выяснить, что небольшие изменения некоторых ограничений, в принципе возможные и допустимые, позволяют применить найденные новые, более эффективные решения.

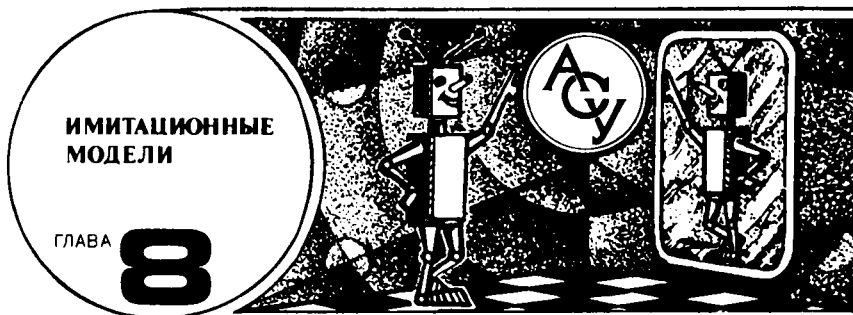
Для поиска новых решений в сложных случаях оказываются полезными *систематизированные исследования*. Под этим понимают выделение нескольких основных составляющих, направленных на решение задачи. По каждой составляющей стараются найти как можно больше новых решений. Затем все составляющие и найденные частные решения сводят в таблицу. Комбинируя в различных сочетаниях частные решения, получают принципиально новые решения сложной задачи.

Для коллективного поиска новых решений полезен метод, получивший название *мозговой атаки* или *мозгового штурма*. Он заключается в том, что группе специалистов дается задание — найти решение сложной задачи. Каждый участник группы может высказывать любые идеи в этом направлении, независимо от возможности их реализации, в том числе разрешается выдвигать идеи, кажущиеся фантастическими, нелепыми, совершенно нереальными. Какое-либо осуждение выдвинутых идей, тем более их критика в любой форме, не допускаются. Высказанные идеи можно только подхватывать — развивать, дополнять или выдвигать новые. Необходимо, чтобы все мысли высказывались свободно, поток их был интенсивным, одна следовала за другой без пауз.

Возникает своеобразный психологический настрой на генерацию новых идей, коллективное творчество заставляет каждого участника мыслить,

гораздо интенсивнее, чем в одиночку. Результат группы оказывается больше, чем мог бы быть получен как сумма результатов ее участников. Такой психологический настрой возникает только при соблюдении некоторых условий: группа должна быть не слишком большой и не слишком маленькой, оптимальный состав 6—8 человек; участники группы должны иметь примерно одинаковый уровень подготовленности и образования; в составе группы не должно быть начальника или неформального лидера и ряд других.

Эффект интенсивной генерации новых идей возникает вскоре после начала работы группы и продолжается в пределах часа. Все высказанные идеи фиксируют. После окончания генерации идей начинается второй этап — их анализ и критика. Этот этап напоминает предыдущий с той разницей, что группе дается задание раскритиковать все идеи, высказанные на первом этапе. Допускаются любые возражения, пусть даже несуществующие или легко опровергаемые, можно только их развивать или выдвигать новые, но нельзя спорить и доказывать справедливость рассматриваемой идеи. Те предложения и идеи, которые выявлены в результате обоих этапов мозговой атаки, оказываются, как правило, наиболее плодотворными, результативными и очень часто принципиально новыми.



Одна из важных особенностей АСУ — принципиальная невозможность проведения реальных экспериментов до завершения проекта. Возможным выходом является использование имитационных моделей. Однако их разработка и использование чрезвычайно сложны, возникают затруднения в достаточно точном определении степени адекватности моделируемому процессу. Поэтому важно принять решение — какую создать модель.

Другой важный аспект — использование имитационных моделей в процессе эксплуатации АСУ для принятия решений. Такие модели создаются в процессе проектирования, чтобы их можно было непрерывно модернизировать и корректировать в соответствии с изменяющимися условиями работы пользователя.

Эти же модели могут быть использованы для обучения персонала перед вводом АСУ в эксплуатацию и для проведения деловых игр.

§ 8.1. Общие сведения

Имитационное моделирование — это метод исследования, заключающийся в имитации на ЭВМ с помощью комплекса программ процесса функционирования системы или отдельных ее частей и элементов. Сущность метода имитационного моделирования заключается в разработке таких алгоритмов и программ, которые имитируют поведение системы, ее свойства и характеристики в необходимом для исследования системы составе, объеме и области изменения ее параметров.

Принципиальные возможности метода весьма велики, он позволяет при необходимости исследовать системы любой сложности и назначения с любой степенью детализации. Ограничениями являются лишь мощность используемой ЭВМ и трудоемкость подготовки сложного комплекса программ.

В отличие от математических моделей, представляющих собой аналитические зависимости, которые можно исследовать с помощью достаточно мощного математического аппарата, имитационные модели, как правило, позволяют проводить на них лишь одиночные испытания, аналогично однократному эксперименту на реальном объекте. Поэтому для более полного исследования и получения необходимых зависимостей между параметрами требуются многократные испытания модели, число и продолжительность которых во многом определяются возможностями используемой ЭВМ, а также свойствами самой модели.

Использование имитационных моделей оправдано в тех случаях, когда возможности методов исследования системы с помощью аналитических моделей ограничены, а натурные эксперименты по тем или иным причинам нежелательны или невозможны.

Даже в тех случаях, когда создание аналитической модели для исследования конкретной системы в принципе возможно, имитационное моделирование может оказаться предпочтительным по затратам времени ЭВМ и исследователя на проведение исследования. Для многих задач, возникающих при создании и функционировании АСУ, имитационное моделирование иногда оказывается единственным практически реализуемым методом исследования. Этим в значительной степени объясняется непрерывно возрастающий интерес к имитационному моделированию и расширение класса задач, для решения которых оно применяется.

Методы имитационного моделирования развиваются и используются в основном в трех направлениях: разработка типовых методов и приемов создания имитационных моделей; исследование степени подобия имитационных моделей реальным системам; создание средств автоматизации программирования, ориентированных на создание комплексов программ для имитационных моделей.

Различают два подкласса систем, ориентированных на системное и логическое моделирование. К подклассу *системного моделирования* относят системы с хорошо развитыми общеалгоритмическими средствами; с широким набором средств описания параллельно выполняемых действий, временных последовательностей выполнения процессов; с возможностями сбора и обработки статистического материала. В таких системах используют специальные языки программирования и моделирования — СИМУЛА, СИМСКРИПТ, GPSS и др. Первые два из этих языков являются подмножествами процедурно-ориентированных языков программирования типа ФОРТРАН, ПЛ/1, расширенными средствами динамических структур данных, операторами управления квазипараллельными процессами, специальными средствами сбора статистики и обработки списков. Эти дополнительные возможности позволяют вести статистические исследования моделей, поэтому такие системы иногда называют *системами статистического моделирования*.

К подклассу *логического моделирования* относят системы, позволяющие в удобной и сжатой форме отражать логические и топологические

особенности моделируемых объектов, обладающие средствами работы с частями слов, преобразования форматов, записи микропрограмм. К этому подклассу систем относят языки программирования АВТОКОД, ЛОТИС и др.

В большинстве случаев при имитационном моделировании экономических, производственных и других организационных систем управления исследование модели заключается в проведении стохастических экспериментов. Отражая свойства моделируемых объектов, эти модели содержат случайные переменные, описывающие как функционирование самих систем, так и воздействия внешней среды. Поэтому наибольшее распространение получило статистическое моделирование.

Имитационная модель характеризуется наборами входных переменных

$$X(t) = \{x_1(t), x_2(t), \dots, x_m(t)\};$$

наблюдаемых или управляемых переменных

$$Y(t) = \{y_1(t), y_2(t), \dots, y_k(t)\};$$

управляющих воздействий

$$R(t) = \{r_1(t), r_2(t), \dots, r_l(t)\};$$

возмущающих воздействий

$$W(t) = \{w_1(t), w_2(t), \dots, w_p(t)\}.$$

Состояние системы в любой момент времени

$Z(t) = \{z_1(t), z_2(t), \dots, z_n(t)\}$ и начальные условия $Y(t_0)$, $R(t_0)$, $W(t_0)$ могут быть случайными величинами, заданными соответствующим распределением вероятностей. Соотношения модели определяют распределение вероятностей величин в момент $t + \Delta t$:

$$Z(t + \Delta t) = \{z_1(t + \Delta t), z_2(t + \Delta t), \dots, z_n(t + \Delta t)\}.$$

Существуют два основных способа построения моделирующего алгоритма — принцип Δt и принцип особых состояний [1.2].

П р и н ц и п Δt . Промежуток времени (t_0, t) , в котором исследуется поведение системы, разбивают на интервалы длиной Δt . В соответствии с заданным распределением вероятностей для начальных условий по априорным соображениям или случайным образом выбирают для начального момента t_0 одно из возможных состояний $z^0(t_0)$. Для момента $t_0 + \Delta t$ вычисляется условное распределение вероятностей состояний (при условии состояния $z^0(t_0)$). Затем аналогично предыдущему выбирают одно из возможных состояний $z^0(t_0 + \Delta t)$, выполняют процедуры вычисления условного распределения вероятностей состояний для момента $t_0 + 2\Delta t$ и т.д.

В результате повторения этой процедуры до момента $t_0 + n\Delta t = T$ получают одну из возможных реализаций исследуемого случайного процесса. Таким же образом получают ряд других реализаций процесса. Описанный способ построения моделирующего алгоритма занимает много машинного времени.

Принцип особых состояний. Все возможные состояния системы $Z(t) = \{ z_i(t) \}$ разбивают на два класса — обычные и особые. В обычных состояниях характеристики $z_i(t)$ меняются плавно и непрерывно. Особые состояния определяются наличием входных сигналов или выходом, по крайней мере, одной из характеристик $z_i(t)$ на границу области существования. При этом состояние системы меняется скачкообразно.

Моделирующий алгоритм должен предусматривать процедуры определения моментов времени, соответствующих особым состояниям, и величин характеристик системы в эти моменты. При известном распределении вероятностей для начальных условий выбирают одно из возможных состояний и по заданным закономерностям изменений характеристик $z_i(t)$ находят их величины перед первым особым состоянием. Таким же образом переходят ко всем последующим особым состояниям. Получив одну из возможных реализаций случайного многомерного процесса, с использованием аналогичных процедур строят другие реализации. Затраты машинного времени при использовании моделирующего алгоритма по принципу особых состояний обычно меньше, чем по принципу Δt .

Имитационное моделирование используют в основном для следующих применений:

- 1) при исследовании сложных внутренних и внешних взаимодействий динамических систем с целью их оптимизации. Для этого изучают на модели закономерности взаимосвязи переменных, вносят в модель изменения и наблюдают их влияние на поведение системы;

- 2) для прогнозирования поведения системы в будущем на основе моделирования развития самой системы и ее внешней среды;

- 3) в целях обучения персонала, которое может быть двух типов: индивидуальное обучение оператора, управляющего некоторым технологическим процессом или устройством, и обучение группы людей, осуществляющих коллективное управление сложным производственным или экономическим объектом.

В системах обоих типов комплекс программ задает некоторую обстановку на объекте, однако между ними имеется существенное различие. В первом случае программное обеспечение имитирует функционирование объектов, описываемых технологическими алгоритмами или передаточными функциями; модель ориентирована на тренировку психофизиологических характеристик человека, поэтому такие модели называются *тренажерами*. Модели второго типа гораздо сложнее. Они описывают некоторые аспекты функционирования предприятия или фирмы и ориентированы на выдачу некоторых технико-экономических характеристик при воздействии на входы чаще всего не отдельного человека, а группы людей, выполняющих различные функции управления;

- 4) для макетирования проектируемой системы и соответствующей части управляемого объекта с целью прикидочной проверки предполагаемых проектных решений. Это позволяет в наиболее наглядной и понятной заказчику форме продемонстрировать ему работу будущей системы,

что способствует взаимопониманию и согласованию проектных решений. Кроме того, такая модель позволяет выявить и устранить возможные неувязки и ошибки на более ранней стадии проектирования, что на 2—3 порядка снижает стоимость их исправления.

§ 8.2. Имитационные модели производственных процессов

Вид модели производственного процесса зависит в значительной степени от того, является ли он дискретным или непрерывным. В дискретных моделях переменные изменяются дискретно в определенные моменты имитационного времени. Время может приниматься как непрерывным, так и дискретным в зависимости от того, могут ли дискретные изменения переменных происходить в любой момент имитационного времени или только в определенные моменты. В непрерывных моделях переменные процесса являются непрерывными, а время может быть как непрерывным, так и дискретным в зависимости от того, являются ли непрерывные переменные доступными в любой момент имитационного времени или только в определенные моменты. В обоих случаях в модели предусматривают блок заданного времени, который имитирует продвижение модельного времени, обычно ускоренного относительно реального.

Разработка имитационной модели и проведение моделирующих экспериментов в общем случае могут быть представлены в виде нескольких основных этапов, приведенных на рис. 8.1.

Компонента модели, отображающая определенный элемент моделируемой системы, описывается набором характеристик количественного или логического типа. В зависимости от длительности существования различают компоненты условно-постоянные и временные. Условно-постоянные компоненты существуют в течение всего времени эксперимента с моделью, а временные — генерируются и уничтожаются в ходе эксперимента. Компоненты имитационной модели делят на классы, внутри которых они

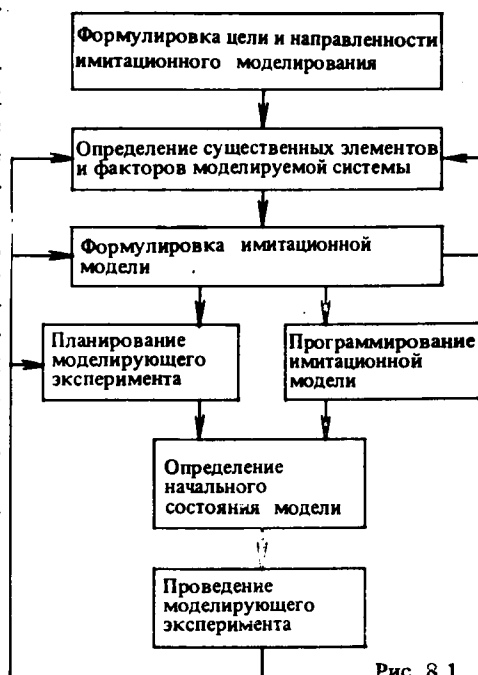


Рис. 8.1

имеют одинаковый набор характеристик, но отличаются их значениями.

Состояние компоненты определяется значениями ее характеристик в данный момент модельного времени, а совокупность значений характеристик всех компонент определяет состояние модели в целом.

Изменение значений характеристик, являющееся результатом отображения в модели взаимодействия между элементами моделируемой системы, приводит к изменению состояния модели. Характеристика, значение которой в ходе моделирующего эксперимента изменяется, является переменной, в противном случае это параметр. Значения дискретных переменных не изменяются в течение интервала времени между двумя последовательными особыми состояниями и меняются скачком при переходе от одного состояния к другому.

Моделирующий алгоритм представляет собой описание функциональных взаимодействий между компонентами модели. Для его составления процесс функционирования моделируемой системы разбивается на ряд последовательных событий, каждое из которых отражает изменение состояния системы в результате взаимодействия ее элементов или воздействия на систему внешней среды в виде входных сигналов. Особые состояния возникают в определенные моменты времени, которые планируются заранее, либо определяются в ходе эксперимента с моделью. Наступление событий в модели планируется путем составления расписания событий по временам их свершения либо проводится анализ, выявляющий достижение переменными характеристиками установленных значений.

Для этой цели наиболее удобно использовать СИВС, описанные в гл. 2. Представленные на них материальные и информационные потоки легко анализировать для выявления особых состояний. Такими состояниями являются отражаемые на СИВС моменты окончания обработки изделия на каждом рабочем месте или его транспортировки; приема и выдачи на постоянное или временное хранение; сборки деталей в узлы, узлов в изделие и т.п. Для дискретного производства изменение характеристик между особыми состояниями можно также считать дискретным, имея в виду переход условным скачком от исходного материала к заготовке, от заготовки к полуфабрикату, от полуфабриката к детали и т.д.

Таким образом, каждая производственная операция рассматривается как оператор, изменяющий значение характеристик изделия. Для простых моделей последовательность состояний можно принимать детерминированной. Лучше отражают действительность случайные последовательности, которые можно формализовать в виде случайных приращений времени, имеющих заданное распределение, либо случайного потока однородных событий, аналогично потокам заявок в теории массового обслуживания. Аналогичным образом можно проанализировать и выявить с помощью СИВС особые состояния при движении и обработке информации.

На рис. 8.2 представлена структура обобщенной имитационной модели.

При моделировании непрерывных производственных процессов по

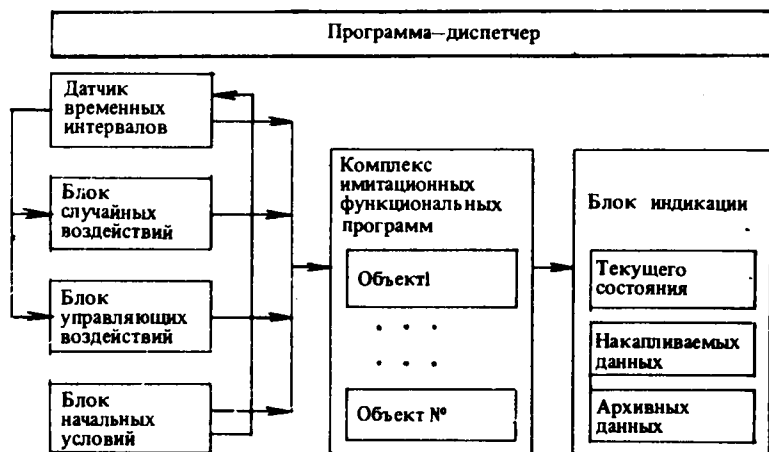


Рис. 8.2

принципу Δt датчик временных интервалов выдает тактовые импульсы для работы моделирующего алгоритма. Блоки случайных и управляющих воздействий, а также начальных условий служат для ручного ввода условий проведения очередного модельного эксперимента.

Комплекс имитационных функциональных программ по каждому моделируемому объекту определяет условное распределение вероятностей состояний объекта к окончанию каждого момента Δt . При случайном выборе одного из возможных состояний это осуществляется функциональной подпрограммой; при выборе экспериментатором — программой, заложенной в блоке управляющих воздействий, или, при желании осуществлять этот выбор вручную на каждом такте, вводом новых начальных условий исходя из текущего состояния, определяемого с помощью блока индикации.

Функциональная программа определяет параметры технологической установки на каждом такте в зависимости от заданных начальных условий — характеристик сырья, заданного режима, свойств и условий работы установки. Из модели технологической части программным путем могут быть добавлены соотношения весового и объемного баланса.

Координацию и взаимодействие всех блоков и программ осуществляет программа-диспетчер.

При моделировании дискретных процессов, при котором обычно используют принцип особых состояний, структура имитационной модели изменяется незначительно. Вместо датчика временных интервалов вводится блок, определяющий наличие особого состояния и выдающий команду на переход к следующему. Функциональная программа имитирует на каждом переходе одну операцию на каждом рабочем месте. Характеристики таких операций могут быть детерминированными во времени, например

при работе станка-автомата, либо случайными с заданными распределениями. Кроме времени могут имитироваться и другие характеристики — наличие или отсутствие брака, отнесение к некоторому сорту или классу и т.п.

Аналогично имитируются сборочные операции, с той разницей, что на каждой операции изменяются не характеристики обрабатываемого материала, а вместо одних наименований — детали, узлы — появляются другие — узлы, изделия — с новыми характеристиками. Однако принципиально операции сборки имитируются аналогично операциям обработки — определяются случайные или детерминированные затраты времени на операцию, значения физических и производственных характеристик.

§ 8.3. Имитационные модели предприятий и производственных объединений

Для имитации сложных производственных систем требуется создание логико-математической модели исследуемой системы, позволяющей проведение с нею экспериментов на ЭВМ. Модель реализуют в виде комплекса программ, написанных на одном из универсальных языков программирования высокого уровня либо на специальном языке моделирования. С развитием имитационного моделирования появились системы и языки, сочетающие возможности имитации как непрерывных, так и дискретных систем, что позволяет моделировать сложные системы типа предприятий и производственных объединений.

При построении модели прежде всего следует определить ее назначение. В модели должны быть отражены все существенные с точки зрения цели ее построения функции моделируемого объекта и в то же время в ней не должно быть ничего лишнего, иначе она будет слишком громоздкой и мало эффективной.

Основным назначением моделей предприятий и объединений является их исследование с целью совершенствования системы управления либо обучения и повышения квалификации управленческого персонала. При этом моделируется не само производство, а отображение производственного процесса в системе управления.

Для построения модели используется укрупненная СИВС. Методом единичной нити выявляют те функции и задачи, в результате которых может быть получен искомый результат в соответствии с назначением модели. На основании логико-функционального анализа строят структурную схему модели. Построение структурной схемы позволяет выделить ряд самостоятельных моделей, входящих в виде составных частей в модель предприятия. На рис. 8.3 приведен пример построения структурной схемы моделирования финансово-экономических показателей предприятия. Модель учитывает как внешние факторы — спрос на продукцию, план поставок, так и внутренние — затраты на производство, существующие и планируемые производственные возможности.

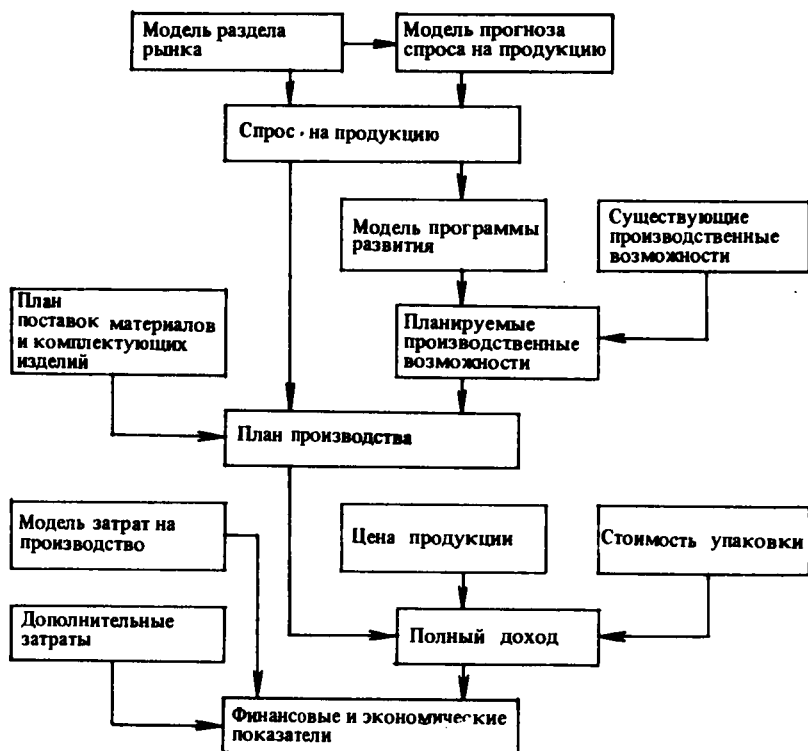


Рис. 8.3

Некоторые из моделей являются детерминированными — расчет планируемого полного дохода по номенклатуре и количествам в соответствии с планом производства при известных ценах и стоимости упаковки. Модель плана производства является оптимизационной, настраиваемой на один из возможных критериев — максимизацию дохода или использования производственных мощностей; наиболее полное удовлетворение спроса; минимизацию потерь поставляемых материалов и комплектующих изделий и пр. В свою очередь модели спроса на продукцию, планируемых производственных мощностей и плана поставок являются вероятностными с различными законами распределения.

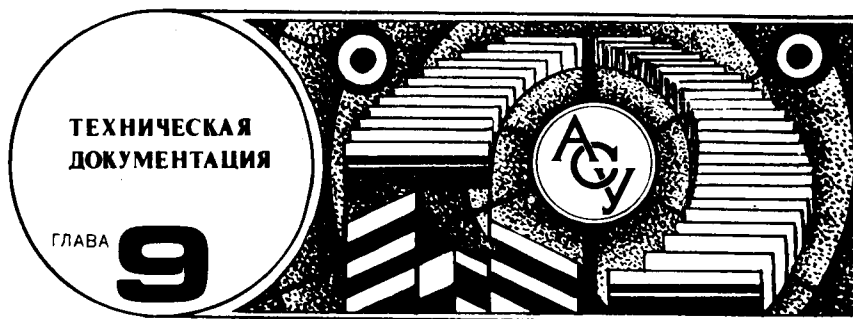
Взаимосвязь между моделями, координация их работы и связь с пользователями осуществляется с помощью специальной программы, которая на рис. 8.3 не показана. Эффективная работа пользователей с моделью достигается в режиме диалога.

Построение структурной схемы модели не формализовано и во многом зависит от опыта и интуиции ее разработчика. Здесь важно соблюдать общее правило — лучше на первых этапах составления схемы включить в нее большее число элементов с последующим их постепенным сокраще-

нием, чем начать с некоторых, кажущихся основными, блоков, намереваясь в последующем их дополнять и детализировать.

После построения схемы, обсуждения ее с заказчиком и корректировки переходят к построению отдельных моделей. Необходимая для этого информация содержится в системных спецификациях — перечень и характеристики задач, необходимые для их решения исходные данные и выходные результаты и т.д. Если системные спецификации не составлялись, эти сведения берут из материалов обследования, а иногда прибегают к дополнительным обследованиям.

Важнейшими условиями эффективного использования моделей являются проверка их адекватности и достоверность исходных данных. Если проверка адекватности осуществляется известными методами, то достоверность имеет некоторые особенности. Они заключаются в том, что во многих случаях исследование модели и работу с ней лучше проводить не с реальными данными, а со специально подготовленным их набором. При подготовке набора данных руководствуются целью использования модели, выделяя ту ситуацию, которую хотят промоделировать и исследовать.



Разработка проекта сопровождается на каждой стадии соответствующей проектной технической документацией. Ее цель — формальное представление проектных решений для последующего использования при их реализации. Сложность создаваемой системы, большое количество участников разработки различных специальностей требуют регламентации проектной документации. Руководитель разработки, регламентируемой ГОСТами, определяет степень детализации, следит за соответствием между отдельными частями проектной документации. Это особенно важно при внесении каких-либо изменений в проект.

Переход к безбумажной технологии документооборота в АСУ пока не касается проектной документации. Однако постепенно объем ее должен уменьшаться, а характер — изменяться, так как многие проектные решения, получаемые при автоматизации проектирования, могут оставаться на магнитном носителе.

§ 9.1. Общие положения и общесистемная документация

Техническая документация на АСУ — это комплекс взаимосвязанных документов, в которых полностью описаны все решения по созданию и эксплуатации АСУ.

Общие положения. Разрабатываемая при создании АСУ всех видов для всех уровней управления, начиная с отрасли и ниже, техническая документация, а также комплектность проектной документации (кроме организационно-распорядительных документов, разрабатываемых на предпроектной и проектной стадиях) регламентированы следующими государственными стандартами.

Виды и комплектность документов определяются ГОСТ' 24.101–80; обозначение документов – ГОСТ 24.102–80; общесистемные документы – ГОСТ 24.203–80.

Требования к содержанию документов определяются ГОСТами: "Техническое задание" – ГОСТ 24.201–79; "Технико-экономическое обоснование" – ГОСТ 24.202–80; "Описание постановки задачи" – ГОСТ 24.204–80; "Ввод в эксплуатацию" – ГОСТ 24.208–80; по информационному обеспечению – ГОСТ 24.205–80; по техническому обеспечению – ГОСТ 24.206–80; по программному обеспечению – ГОСТ 24.207–80; по организационному обеспечению – ГОСТ 24.209–80.

Требования к выполнению текстовых документов определены ГОСТ 24.301–80; к выполнению схем – ГОСТ 24.302–80; обозначения условные графические технических средств – ГОСТ 24.303–80.

Внесение изменений в техническую документацию определено ГОСТ 24.401–80; учет, хранение и обращение документации – ГОСТ 24.402–80.

Все документы допускается разрабатывать самостоятельными частями, которые должны удовлетворять требованиям к разделяемому на части документу. В первой части разделяемого документа должен быть приведен полный перечень всех частей. К наименованию типа документа и его коду должны быть добавлены соответственно наименование и номер самостоятельной части.

Номенклатура типов документов может быть расширена относительно установленной стандартом, если этого требует специфика создаваемой АСУ. В этом случае номенклатура дополнительно выпускаемых документов должна быть установлена отраслевым стандартом.

Документы в проектах АСУ укомплектовывают на систему или составляющие ее части: подсистему, комплекс задач или задачу, комплекс программ или программу, а также по видам обеспечения. На каждый комплект составляют ведомость документов. На предпроектной стадии и при вводе в эксплуатацию документы в комплекты не объединяют.

Каждый документ должен иметь самостоятельное обозначение, которое указывается на всех листах этого документа. Структура обозначения документа приведена на рис. 9.1.

Общесистемная документация. Выпускается для общего описания и обоснования решений, принятых в проекте АСУ. В ее состав входят пояснительная записка к проекту; общее описание АСУ; смета затрат; расчет экономической эффективности; план мероприятий по вводу АСУ в эксплуатацию; ведомости документов технического и рабочего (технорбочего) проектов, эксплуатационных документов; патентный формуляр; формуляр системы; ведомость держателей подлинников.

Пояснительная записка должна включать: основание для разработки; краткую характеристику объекта управления; проектные решения по АСУ; состав проекта.

В разделе "Основание для разработки" приводят: наименование объек-

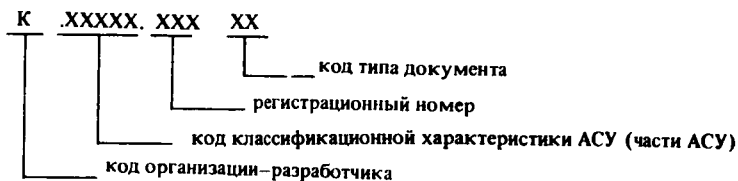


Рис. 9.1

та, на котором создают АСУ; наименование документов, на основании которых ведут разработку АСУ, в соответствии с техническим заданием; перечень организаций, участвующих в разработке АСУ с указанием заказчика, головного разработчика и соисполнителей; сроки выполнения стадий создания АСУ.

В разделе "Краткая характеристика объекта управления" должны быть указаны: характеристика объекта управления (предприятия, подразделения предприятия, установки и т.д.); основные технико-экономические показатели объекта управления.

В разделе "Проектные решения по АСУ" должны быть указаны: перечень НИР, использованных при разработке проекта; обоснование и краткая характеристика основных решений по функциональной и обеспечивающей частям АСУ; описание общих принципов функционирования АСУ; общий регламент функционирования АСУ; сведения о соответствии показателей, характеризующих качество создаваемой АСУ, установленным в техническом задании; сведения об обеспечении совместимости системы с АСУ других уровней и других функциональных назначений.

В разделе "Состав проекта" следует указывать место нахождения основных проектных решений в соответствующих документах проекта АСУ.

Общее описание АСУ состоит из следующих разделов: назначение АСУ; описание АСУ; место АСУ в системе управления объектом; связь АСУ с другими системами управления.

В разделе "Назначение АСУ" должны быть приведены назначение АСУ и ее основные функции.

В разделе "Описание АСУ" должны быть приведены: описание АСУ в целом и ее частей; перечень основных проектных решений, обеспечивающих достижение поставленных целей; назначение каждой из основных частей (комплексов, задач, устройств и т.д.).

В разделе "Место АСУ в системе управления объектом" должны быть указаны: место АСУ в общей структуре системы управления объектом, на котором создают АСУ; перечень подразделений, объектов управления и функций, на которые распространяется действие АСУ; описание связей основных частей АСУ с подразделениями объекта управления в процессе функционирования.

В разделе "Связь АСУ с другими системами управления" должны быть

приведены: перечень систем, с которыми связана разрабатываемая АСУ, с описанием связей между системами; описание регламента связей.

Смета затрат составляется в соответствии с СН 202—76 и нормативными документами Госплана СССР, Госстроя СССР, Госбанка СССР.

Расчет экономической эффективности содержит следующие разделы: исходные данные для расчета; расчет экономической эффективности системы; результаты расчета.

В разделе "Исходные данные для расчета" должны быть приведены: ссылка на методику определения экономической эффективности; перечень факторов, обуславливающих повышение эффективности функционирования объекта управления при создании АСУ; исходные данные, необходимые для расчета согласно принятой методике; ссылка на источники получения исходных данных; обоснование выбора базы для сравнения.

В разделе "Расчет экономической эффективности системы" должны быть приведены: расчет затрат на создание АСУ; расчет затрат на содержание и эксплуатацию системы; расчет ожидаемой экономии по основным технико-экономическим показателям и ожидаемого годового экономического эффекта внедрения АСУ в целом; расчет коэффициента эффективности и срока окупаемости затрат.

В разделе "Результаты расчета" должны быть приведены основные результаты расчетов: затрат на создание системы; затрат на содержание и эксплуатацию; ожидаемого годового экономического эффекта от создания системы; коэффициента эффективности капитальных вложений; срока окупаемости затрат; сравнения расчетного коэффициента эффективности и срока окупаемости затрат с нормативными.

План мероприятий по вводу АСУ в эксплуатацию должен содержать перечень основных работ, выполнение которых обеспечивает подготовку объекта к вводу АСУ в эксплуатацию. По каждой работе указывают ее содержание, количественную характеристику, срок исполнения, организации-исполнители и форму завершения работы.

Ведомость документов технического проекта содержит перечень документов, включаемых в комплект документов данного технического проекта, в том числе заимствованных из других проектов. Ведомость заполняют по разделам — частям проекта АСУ в последовательности: общесистемная документация; документация функциональной части; документация информационного обеспечения; документация технического обеспечения; документация математического обеспечения; документация программного обеспечения; документация организационного обеспечения. В каждый раздел включают документы, относящиеся к частям проекта АСУ, в соответствии с ГОСТ 24.101—80.

Ведомость документов рабочего (технорабочего) проекта содержит перечень документов, включаемых в разрабатываемый проект, в том числе заимствованных из других проектов, аналогично предыдущему.

Ведомость эксплуатационных документов содер-

жит перечень эксплуатационных документов согласно ГОСТ 24.101–80.

П а т е н т н ы й ф о р м у л я р должен содержать разделы и включать сведения, установленные ГОСТ 2.110–68.

Ф о р м у л я р с и с т е м ы должен включать разделы: основные параметры и технические характеристики АСУ; сведения о состоянии технического обеспечения АСУ.

В разделе "Основные параметры и технические характеристики АСУ" приводят: общие указания по эксплуатации АСУ и ведению формуляра, включая перечень технической документации, с которой должен быть ознакомлен персонал для обеспечения функционирования АСУ, требования к ведению формуляра и месту его хранения; общие сведения о системе с указанием ее наименования, вида и даты сдачи в эксплуатацию; основные технические данные и характеристики АСУ, а также перечень фактически реализуемых функций системы; комплектность АСУ, включая перечень носителей данных и программ, необходимых для реализации функций АСУ, организации вычислительного процесса и решения контрольных примеров; перечень эксплуатационных документов АСУ; гарантийные обязательства разработчиков с указанием перечня устройств, гарантийные сроки службы которых меньше гарантийных сроков службы системы в целом; акты о приемке АСУ в опытную и промышленную эксплуатацию.

В разделе "Сведения о состоянии технического обеспечения АСУ" приводят: сведения о рекламациях с указанием номера, даты и краткого содержания рекламационного акта, а также сведения об устранении замечаний, указанных в акте; сведения по учету работы АСУ с указанием продолжительности работы за месяц и с момента ввода в опытную эксплуатацию; учет неисправностей с указанием даты, времени, характера, причин их возникновения и лиц, устранивших неисправность; замечания по эксплуатации и аварийным ситуациям, принятых мерах; сведения о проведении проверок измерительных устройств и о точностных характеристиках измерительных каналов (для АСУ ТП); сведения о ремонте устройств и изменениях схем и программного обеспечения с указанием основания, даты и содержания изменения; сведения о выполнении регламентных (профилактических) работ и их результатах; сведения о регистрации оборудования контрольными органами (для АСУ ТП).

В разделах документа допускается вместо приведения сведений давать ссылки на эксплуатационные документы, в которых эти сведения содержатся.

В е д о м о с т ь д е р ж а т е л е й п о д л и н н и к о в выполняют в соответствии с требованиями ГОСТ 2.112–70.

§ 9.2. Состав и содержание документов предпроектной стадии

Основными документами на предпроектной стадии являются технико-экономическое обоснование, техническое задание и описание постановки задач.

Технико-экономическое обоснование. Предназначено для обоснования производственно-хозяйственной необходимости и технико-экономической целесообразности создания или развития АСУ.

Документ ТЭО АСУ должен состоять из следующих разделов: введение; характеристика объекта и существующей системы управления; цели, критерии и ограничения создания АСУ; функции и задачи создаваемой АСУ; ожидаемые технико-экономические результаты создания АСУ, выводы и предложения.

Раздел "Введение" должен содержать следующие сведения по технико-экономическому обоснованию создания АСУ: основание для проведения работ, наименование организации-заказчика; наименование организаций — участников работ; сроки начала и окончания работ; источники, объемы, порядок финансирования работ; перечень нормативно-технических документов, методических материалов, использованных при проведении ТЭО.

Раздел "Характеристика объекта и существующей системы управления" должен содержать: общую характеристику объекта; характеристику производственно-хозяйственной деятельности, организационной и производственной структуры объекта; характеристику существующей системы управления и ее структурных элементов с указанием распределения функций управления между элементами организационной структуры; характеристику функций управления, используемых методов и средств управления; перечень и характеристику недостатков в организации и управлении объектом (в методах управления, организационной структуре управления, выполнении функций управления, обеспечении информацией и т.д.); оценку производственных потерь, возникающих из-за недостатков в организации и управлении по объекту в целом и его частям (ухудшение технико-экономических и социальных показателей деятельности объекта и его частей); характеристику готовности объекта к созданию АСУ.

Раздел "Цели, критерии и ограничения создания АСУ" должен содержать: формулировку производственно-хозяйственных, научно-технических и экономических целей и критериев создания АСУ; характеристику ограничений по созданию АСУ.

Раздел "Функции и задачи создаваемой АСУ" должен содержать: обоснование выбора перечня автоматизированных функций и комплексов задач (задач) управления с указанием очередности внедрения; требования к качественным характеристикам реализации функций и задач управления в соответствии с действующими нормативно-техническими документами, определяющими общие технические требования к АСУ конкретного вида; дополнительные требования к АСУ в целом и ее частям, учитывающие специфику объекта управления и создаваемой АСУ.

Раздел "Ожидаемые технико-экономические результаты создания АСУ" должен содержать: перечень основных источников экономической эффективности, получаемой в результате создания АСУ (в том числе экономия производственных ресурсов, улучшение качества продукции, повышение производительности труда и т.д.), и оценку ожидаемых изменений ос-

новых технико-экономических и социальных показателей производственно-хозяйственной деятельности объекта (например, показателей по номенклатуре и объемам производства, себестоимости продукции, рентабельности, отчислениям в фонды экономического стимулирования, уровню социального развития); оценку ожидаемых затрат на создание АСУ с распределением их по очередям создания АСУ и по годам; ожидаемые обобщающие показатели экономической эффективности АСУ.

Раздел "Выводы и предложения" должен состоять из следующих подразделов: выводы о производственно-хозяйственной необходимости и технико-экономической целесообразности создания АСУ; предложения по совершенствованию организации и управления; рекомендации по созданию АСУ.

Подраздел "Выводы о производственно-хозяйственной необходимости и технико-экономической целесообразности создания АСУ" должен содержать: сопоставление ожидаемых результатов создания АСУ с заданными целями и критериями создания АСУ (по целевым показателям и нормативным требованиям); принципиальное решение вопроса о создании АСУ (положительное или отрицательное).

Подраздел "Предложения по совершенствованию организации и управления" должен содержать предложения: по совершенствованию производственно-хозяйственной деятельности; по совершенствованию организационной и функциональной структур системы управления, методов управления; по развитию видов обеспечения АСУ и т.д.

Подраздел "Рекомендации по созданию АСУ" должен содержать рекомендации: по виду создаваемой АСУ, ее совместимости с другими АСУ и неавтоматизируемой частью существующей системы управления; по организационной и функциональной структуре создаваемой АСУ; по составу и характеристикам подсистем и видов обеспечения АСУ; по организации использования имеющихся и приобретению дополнительных средств вычислительной техники; по составу организаций-разработчиков, которые необходимо привлечь к созданию АСУ; по рациональной организации разработки и внедрения АСУ; по определению основных и дополнительных, внешних и внутренних источников, видов и объемов финансирования и материального обеспечения разработок АСУ; по обеспечению производственных условий создания АСУ; другие рекомендации по созданию АСУ.

Техническое задание — основной исходный документ для создания АСУ. При приемке АСУ в эксплуатацию проверяется ее соответствие техническому заданию. Допускается разрабатывать отдельные технические задания на части АСУ.

Техническое задание на создание АСУ должно содержать: введение; характеристику объекта управления; назначение АСУ; основные требования к АСУ; технико-экономические показатели АСУ; состав, содержание и организацию работ по созданию АСУ; порядок приемки АСУ.

Раздел "Введение" должен содержать: полное наименование и условное обозначение АСУ; основание для создания АСУ (перечень документов);

наименование и условное обозначение темы или разработки; сроки начала и окончания работ по созданию АСУ; наименование организаций, участвующих в создании АСУ (исполнителей, соисполнителей), и их реквизиты; сведения об источниках и порядке финансирования.

Раздел "Характеристика объекта управления" должен содержать описание состава объекта управления (производственную или иную структуру) в зависимости от его типа; характеристики входных и выходных материальных потоков; описание особенностей объекта управления, определяющих основные требования к создаваемой АСУ (регламент, режим работы и т.п.).

Раздел "Назначение АСУ" должен содержать: назначение АСУ, основные цели ее создания, критерии эффективности функционирования объекта в условиях автоматизированного управления; перечень выполняемых функций, необходимых для достижения целей; описание общей структуры системы управления объектом с указанием места АСУ в ней; взаимосвязи создаваемой АСУ с системами управления других уровней; перспективы развития АСУ.

Раздел "Основные требования к АСУ" должен содержать следующие подразделы: требования к системе и ее частям; требования к качеству выполнения функций АСУ; требования к видам обеспечения АСУ.

Подраздел "Требования к системе и ее частям" должен содержать: основные показатели (параметры), которые должны быть достигнуты в условиях автоматизированного управления объектом; структуру АСУ и входящих в нее частей; требования к функционированию АСУ или ссылку на документы, регламентирующие: режим работы; способы обмена информацией со смежными системами с указанием режима обмена, объема, содержания, системы кодирования и при необходимости формы представления информации; эргономические требования в части рациональной компоновки технических средств, удобств обслуживания, комфортности пунктов управления и при необходимости эстетического решения; требования к сохранности информации при авариях в системе энергоснабжения.

Подраздел "Требования к качеству выполнения функций АСУ" должен содержать перечень функций управления и решаемых задач или комплекса задач с указанием для каждой задачи основных входных и выходных показателей и потребителей информации. По каждой функции управления указывают периодичность и формы представления информации, режимы выполнения и требования к результатам.

Подраздел "Требования к видам обеспечения АСУ" должен содержать требования к программному, информационному, организационному, техническому и другим видам обеспечения в соответствии с требованиями, установленными стандартами на АСУ конкретных видов. Допускается приводить требования по применению типовых решений, составу и структуре используемых технических средств, включая специальные технические средства, разрабатываемые для АСУ конкретного вида. Состав требований к частям системы, разрабатываемым по отдельным техническим заданиям,

определяют в зависимости от выделяемой части системы : подсистемы АСУ, ИВЦ, банка данных и т.п.

Раздел "Технико-экономические показатели АСУ" должен содержать: технико-экономические показатели, которые должны быть достигнуты в результате создания АСУ, с указанием максимально допускаемой суммы одновременных затрат на ее создание; годовой экономической эффект и источники его возникновения (повышение производительности, улучшение качества и т.п.); коэффициент экономической эффективности затрат. К техническому заданию следует прилагать расчет экономической эффективности и при необходимости расчет научно-технического уровня создаваемой АСУ.

Раздел "Состав, содержание и организация работ по созданию АСУ" должен содержать перечень стадий и этапов выполнения работ; перечень работ по стадиям и этапам, сроки их выполнения и организации – исполнители работ; форму завершения стадий и этапов создания АСУ. Стадии, этапы и перечень выполняемых работ указывают в соответствии с нормативно-технической документацией на АСУ конкретного вида. Кроме того, в разделе необходимо приводить перечень мероприятий по подготовке объекта к внедрению АСУ с указанием сроков выполнения работ и исполнителей. Эти сведения могут быть представлены в виде планов-графиков. В разделе может быть указана последовательность внедрения частей или задач АСУ.

Раздел "Порядок приемки АСУ" должен содержать указания о составе и объеме приемо-сдаточных испытаний, которые проводят при вводе системы в эксплуатацию или предъявлении системы Государственной (межведомственной, внутриведомственной) комиссии. Состав и объем испытаний определяют по нормативно-технической документации на АСУ конкретного вида.

Описание постановки задачи. Содержит характеристики комплекса задач или задачи: условий, необходимых для решения; входной и выходной информации.

В разделе "Характеристика комплекса задач" следует приводить: цель, назначение, технико-экономическую (организационно-техническую) сущность комплекса задач и обоснование целесообразности его решения (в частности, для задач оптимизации – критерий управления и ограничения); перечень объектов (технологических объектов управления, подразделений, предприятий и т.д.), при управлении которыми решают комплекс задач, при необходимости – описание структуры объектов управления и перечень показателей, характеризующих их состояние; описание процедур использования выходной информации; периодичность решения и ограничения по срокам выдачи выходной информации; требования к организации сбора и передачи в обработку входной информации (с указанием сроков ее поступления), к порядку ее контроля и корректировки; условия, при которых прекращается решение комплекса задач автоматизированным способом; связи данного комплекса задач с другими комплексами (за-

дачами) АСУ; должности лиц и (или) наименования подразделений, определяющих условия и временные характеристики конкретного решения задачи (если они не определены общим алгоритмом функционирования системы); распределение функций между персоналом и техническими средствами при различных ситуациях решения комплекса задач.

Раздел "Выходная информация" должен содержать: перечень и описание выходных сообщений; перечень и описание имеющих самостоятельное смысловое значение структурных единиц информации выходных сообщений — показателей, реквизитов и их совокупностей, сигналов управления. В описании по каждому выходному сообщению следует указывать: идентификатор; форму представления сообщения (документ, видеограмма, сигнал управления) и требования к ней; периодичность выдачи; сроки выдачи; получателей информации. В описании по каждой структурной единице информации следует указывать: наименование; идентификатор выходного сообщения, содержащего структурную единицу информации; требования к точности и надежности вычисления (при необходимости).

Раздел "Входная информация" должен содержать: перечень и описание входных сообщений; перечень и описание структурных единиц информации входных сообщений. В описании по каждому входному сообщению следует указывать: идентификатор; форму представления сообщения и частоту поступления. В описании по каждой структурной единице информации входных сообщений следует указывать: наименование; требуемую точность ее числового значения (при необходимости); источник информации (документ, видеограмма, устройство, кодограмма, информационная база на машинных носителях и т.д.); идентификатор источника информации.

В приложение включают чертежи форм документов, видеограмм для входной и выходной информации, а также другие материалы по усмотрению разработчика (например, описания массивов).

§ 9.3. Состав и содержание документов проектной стадии по видам обеспечения

Документы по информационному обеспечению. Предназначены для описания проектных решений и включают: описание информационного обеспечения АСУ; описание организации информационной базы; описание системы классификации и кодирования; чертежи форм документов (видеограмм); описание массива информации; перечень входных сигналов и данных; перечень выходных сигналов (документов); описание технологического процесса обработки данных.

Описание информационного обеспечения АСУ состоит из следующих разделов: принципы организации информационного обеспечения; организация сбора и передачи информации; построение системы классификации и кодирования; организация внутримашинной информационной базы; организация внешнемашинной информационной базы.

В разделе "Принципы организации информационного обеспечения"

должны быть приведены: состав, структура и принципы организации информационного обеспечения; обоснование выбора носителей данных и принципы распределения информации по типам носителей; описание принятых видов и методов контроля в маршрутах обработки данных при создании и функционировании внешнимашинной и внутримашинной информационных баз с указанием требований, на соответствие которым проводят контроль; описание решений, обеспечивающих информационную совместимость АСУ с другими связанными с ней системами управления по источникам, потребителям информации, по сопряжению применяемых классификаторов (при необходимости), по использованию в АСУ унифицированных систем документации.

В разделе "Организация сбора и передачи информации" должны быть приведены перечни источников, носителей информации, оценка интенсивности и объема информации, описание общих требований к организации сбора и передачи информации.

В разделе "Построение системы классификации и кодирования" должны быть приведены: описание принятых систем классификации объектов; методы кодирования объектов классификации; перечень применяемых общесоюзных, отраслевых и других зарегистрированных классификаторов.

Раздел "Организация внутримашинной информационной базы" должен содержать описание принципов построения базы, характеристики ее состава и объема, структуру информационной базы на уровне баз данных с описанием характера взаимосвязей баз данных и указанием функций АСУ, при реализации которых используют каждую базу данных, характеристики данных, содержащихся в каждой базе.

Раздел "Организация внешнимашинной информационной базы" должен содержать характеристики состава, объема и принципы построения базы, в том числе основные положения по организации и обслуживанию фонда нормативно-справочной информации во взаимосвязи с автоматизируемыми функциями управления.

В приложениях следует приводить справочные и другие дополнительные материалы и сведения (систематизированный перечень наименований структурных единиц информации с присвоенными им обозначениями и описаниями их сущности).

Описание организации информационной базы составляют на внутримашинную и внешнимашинную информационные базы. Каждое описание должно состоять из следующих разделов: состав и структура информационной базы; организация ведения информационной базы.

При описании внутримашинной информационной базы:

в разделе "Состав и структура информационной базы" должны быть приведены перечни баз данных и массивов информации с указанием логических связей между ними; для каждой базы данных приведен перечень массивов информации и логические связи между ними; для массива информации указана логическая структура данных внутри массива или дана

ссылка на документ "Описание массива информации"; описание баз данных выполняют в соответствии с требованиями системы управления базами данных;

в разделе "Организация ведения информационной базы" должна быть приведена последовательность процедур при создании и обслуживании базы с указанием, при необходимости, регламента выполнения процедур и средств защиты базы от разрушения и несанкционированного доступа, а также с указанием связей между массивами баз данных и массивами входной информации.

При описании внемашиной информационной базы:

в разделе "Состав и структура информационной базы" должен быть приведен перечень документов и других информационных сообщений с указанием автоматизируемых функций управления, при реализации которых формируют или используют данный документ; описание каждого документа, информационного сообщения в перечне дают с указанием его обозначения, подразделений, ответственных за его формирование, и других сведений по усмотрению разработчика; допускается приводить ссылки на документы проекта, в которые включены эти сведения; структуру базы данных приводят с разбивкой на виды информации и указанием связей между ними;

в разделе "Организация ведения информационной базы" должна быть приведена последовательность процедур по маршруту движения для групп документов до передачи их в ВЦ.

Описание системы классификации и кодирования должно содержать по каждому классифицируемому объекту описание метода кодирования, структуру и длину кода, указания о системе классификации и другие сведения по усмотрению разработчика.

Чертежи форм документов (видеограмм) включают изображения форм документов или видеограмм в соответствии с требованиями государственных стандартов унифицированной системы документации и необходимые пояснения.

Описание массива информации должно содержать: наименование массива; обозначение массива; наименование носителя информации; перечень реквизитов в порядке их следования в записях массива с указанием по каждому реквизиту: обозначения алфавита, длины в знаках, диапазона изменения (при необходимости), логических и семантических связей с другими реквизитами данной записи и другими записями массива; оценку объема массива; другие характеристики массива (при необходимости).

Перечень входных сигналов и данных должен состоять из перечня входных сигналов и перечня входных данных.

В разделе "Перечень входных сигналов" должны быть приведены: для каждого аналогового сигнала — наименование измеряемой величины, единицы измерения, диапазон изменения, требования к точности и периодичности измерения, тип сигнала; для каждого дискретного сигнала —

наименование, разрядность и периодичность, тип сигнала; для каждого сигнала типа "да — нет" о наличии или отсутствии некоторого события — указание источника формирования.

В разделе "Перечень входных данных" должны быть приведены входные данные с указанием их наименований, кодовых обозначений и значности реквизитов, а также наименования и кодовые обозначения документов или сообщений, содержащих эти данные.

Перечень выходных сигналов (документов) должен состоять из перечня выходных сигналов и перечня выходных документов.

В разделе "Перечень выходных сигналов" должен быть приведен перечень выходных сигналов с указанием их наименований, назначения, единиц измерения и диапазонов изменения, способа представления, пользователей информации.

В разделе "Перечень выходных документов" должен быть приведен перечень выходных документов с указанием их наименований, кодовых обозначений, перечня и значности реквизитов, пользователей информации.

Описание технологического процесса обработки данных должно состоять из следующих разделов: технологический процесс сбора и обработки данных; технологический процесс обработки данных на ВЦ.

В разделе "Технологический процесс сбора и обработки данных" должны быть приведены: состав и последовательность выполнения операций по сбору, регистрации, обработке, контролю и передаче данных в ВЦ; перечень документации, сопровождающей данный технологический процесс.

В разделе "Технологический процесс обработки данных на ВЦ" должны быть приведены: состав и последовательность выполнения операций по приему, контролю, обработке, выдаче результатов обработки и других операций, выполняемых техническими средствами ВЦ; перечень документации, сопровождающей данный технологический процесс.

Документы по техническому обеспечению. Предназначены для описания проектных решений по техническому обеспечению и включают: описание комплекса технических средств; структурную схему комплекса технических средств; план расположения; перечень заявок на разработку новых технических средств; перечень заданий заказчику АСУ (Генпроектировщику) на проектирование в смежной части проекта объектов, связанных с созданием АСУ; ведомость оборудования и материалов; технические требования к технологическому объекту управления; задания на проектирование в смежной части проекта объектов, связанных с созданием АСУ; проектную оценку надежности комплекса технических средств; принципиальную схему; схему автоматизации; таблицу соединений и подключений; схему соединений внешних проводов; чертеж общего вида; заказную спецификацию; чертеж установки технических средств.

Описание комплекса технических средств должно содержать: общие положения; структуру комплекса технических

средств; вычислительный комплекс; абонентские пункты; аппаратуру передачи данных.

В разделе "Общие положения" должны быть приведены исходные данные, использованные при проектировании технического обеспечения АСУ.

В разделе "Структура комплекса технических средств" должны быть приведены: обоснование выбора структуры комплекса технических средств, в том числе технических решений по обмену данными с объектом управления и техническими средствами других АСУ, использованию технических средств ограниченного применения (в соответствии с перечнями, утвержденными в установленном порядке) и ссылки на документы, подтверждающие согласование их поставки; описание функционирования КТС, в том числе в пусковых и аварийных режимах; описание размещения КТС на объектах управления и на производственных площадях с учетом выполнения требований техники безопасности и соблюдения технических условий эксплуатации; обоснование применения и технические требования к оборудованию, предусмотренному в утвержденных проектах и сметах на строительство или реконструкцию предприятия и изготавливаемому в индивидуальном порядке по заказным спецификациям и чертежам проектных организаций как не повторяющееся, не имеющее отраслевой принадлежности по изготовлению и применяемое в силу особых технических решений в проекте; обоснование методов защиты технических средств от механических, тепловых, электромагнитных и других воздействий, защиты данных, в том числе от несанкционированного доступа к ним, и обеспечения заданной достоверности данных в процессе функционирования КТС (при необходимости); результаты проектной оценки надежности КТС (при необходимости).

В разделе "Вычислительный комплекс" должны быть приведены: обоснование и описание основных решений по выбору типа ЭВМ; описание структурной схемы технических средств, размещенных в ВЦ; описание основных решений по выбору типов периферийных технических средств ЭВМ, в том числе средств получения, контроля, подготовки, сбора, регистрации, хранения и отображения информации; результаты расчета или расчет числа технических средств ВЦ и потребности в машинных носителях данных; обоснование численности персонала, обеспечивающего функционирование ВЦ в пусковом, нормальном и аварийном режимах; расчет площадей и технические решения по оснащению рабочих мест; описание особенностей функционирования технических средств ВЦ в пусковом, нормальном и аварийном режимах; принятая технология подготовки машинных носителей данных, решения по контролю данных при вводе в ЭВМ и в процессе их хранения.

В разделе "Абонентские пункты" должны быть приведены: обоснование и описание решений по выбору технических средств абонентских пунктов (АП); обоснование и описание решений по выбору периферийных технических средств АП, в том числе средств получения, контроля, подготовки, сбора, регистрации, передачи, отображения информации и воздействия на объект управления; результаты расчета (или расчет) числа технических

средств каждого АП и потребности в машинных носителях данных; обоснование численности персонала, обеспечивающего функционирование каждого АП; технические решения по оснащению рабочих мест оперативно-го персонала АП, включая описание рабочих мест; описание особенностей функционирования АП в пусковом, нормальном и аварийном режимах.

В разделе "Аппаратура передачи данных" должны быть приведены: обоснование и описание решений по выбору средств телеобработки и передачи данных, в том числе решения по выбору каналов связи и результаты расчета (при необходимости — расчет) их числа; решения по выбору технических средств, обеспечивающих сопряжения с каналами связи, в том числе результаты расчета (или расчет) их потребности; требования к арендуемым каналам связи; сведения о размещении абонентов и объемно-временных характеристиках передаваемых данных; основные показатели надежности, достоверности и других технических характеристик средств телеобработки и передачи данных.

Структурная схема комплекса технических средств должна содержать комплекс технических средств и связи между его отдельными элементами или группами элементов, объединенными по каким-либо логическим признакам (например, совместному выполнению отдельных или нескольких функций, одинаковому назначению и т.д.). На схеме допускается указывать основные характеристики технических средств. Структура КТС АСУ (при необходимости) может быть представлена несколькими схемами, первой из которых является укрупненная схема КТС АСУ в целом.

План расположения средств технического обеспечения, выполняемый при разработке технического проекта, должен определять расположение пунктов управления и средств технического обеспечения, требующих специальных помещений или отдельных площадей для размещения. При разработке рабочего и технорабочего проектов на нем должны быть показаны планы и разрезы помещений с указанием размещения средств технического обеспечения: датчиков, исполнительных механизмов, устройств телемеханики и связи, средств вычислительной техники, кабельных и трубных проводок и т.п. На плане указывают также установочные размеры, необходимые для монтажа технических средств.

Перечень заявок на разработку новых технических средств (форма ТК-1 по ГОСТ 16084—75) должен содержать наименование и назначение разработки, наименование организации-разработчика (предполагаемой), ориентировочную стоимость и объем разработки, сроки выполнения работ.

Перечень заданий заказчику АСУ (Генпроектировщику) на проектирование в смежных частях проекта объекта, связанное с созданием АСУ, должен содержать наименование всех заданий, их назначение, даты выдачи и сроки выполнения работ.

Ведомость оборудования и материалов должна содержать сведения, необходимые для составления смет на приобретение

и монтаж средств технического обеспечения системы, и соответствовать утвержденным в установленном порядке требованиям по составлению заказных спецификаций и ведомостей к проектам АСУ.

Технические требования к технологическому объекту управления содержат требования к технологическому оборудованию (агрегату, установке и т.д.) по изменению, доработке и модернизации, связанным с установкой на объекте управления регулирующих органов, отборных устройств, датчиков и т.п., а также необходимые сведения для проведения этих работ.

Задание на проектирование в смежной части проекта объекта, связанное с созданием АСУ, составляют при необходимости выполнения проектных работ в смежных частях проекта объекта.

Заказная спецификация содержит сведения, необходимые для заказа средств технического обеспечения АСУ, и должна соответствовать утвержденным в установленном порядке требованиям по составлению заказных спецификаций к проектам АСУ.

Документы по программному обеспечению. Документация программного обеспечения предназначена: для описания проектных решений по программному обеспечению в документе "Описание программного обеспечения АСУ"; для определения требований к программе (комплексу программ) в документе "Техническое задание"; для описания решений, обеспечивающих сопровождение, изготовление и эксплуатацию программы (комплекса программ) в документах "Пояснительная записка", "Общее описание", "Описание программы", "Спецификация", "Руководство программиста", "Руководство оператора", "Текст программы", "Формуляр", "Порядок и методика испытаний"; для проверки работоспособности программы (комплекса программ) в документе "Описание контрольного примера".

Описание программного обеспечения АСУ должно содержать: вводную часть; структуру программного обеспечения; основные функции частей программного обеспечения; методы и средства разработки программного обеспечения; операционную систему; средства, расширяющие возможности операционной системы.

Раздел "Вводная часть" должен содержать основные сведения о техническом, информационном и других видах обеспечения АСУ, необходимые для разработки программного обеспечения, или ссылку на соответствующие документы проекта АСУ.

Раздел "Структура программного обеспечения" должен содержать перечень частей программного обеспечения с указанием взаимосвязей и обоснованием выделения каждой из них.

Раздел "Основные функции частей программного обеспечения" должен содержать подразделы, в которых для каждой части программного обеспечения приводят назначение и описание основных функций, а также состав проектных документов, подлежащих разработке.

Раздел "Методы и средства разработки программного обеспечения" должен содержать перечень методов программирования и средств разработки программного обеспечения АСУ с указанием частей программного обеспечения, при разработке которых следует использовать соответствующие методы и средства.

Раздел "Операционная система" должен содержать: наименование, обозначение и краткую характеристику выбранной операционной системы и ее версии, в рамках которой будут выполняться разрабатываемые программы, с обоснованием выбора и указанием источников, где дано подробное описание выбранной версии; наименование руководства, в соответствии с которым должна осуществляться генерация выбранного варианта операционной системы; требования к варианту генерации выбранной версии операционной системы.

Раздел "Средства, расширяющие возможности операционной системы" должен содержать подразделы, в которых для каждого используемого средства, расширяющего возможности операционной системы, следует указать: наименование, обозначение и краткую характеристику средства с обоснованием необходимости его применения и указанием источника, где дано подробное описание выбранного средства; наименование руководства, в соответствии с которым следует настраивать используемое средство на конкретное применение; требования к настройке используемого средства.

Описание программы должно соответствовать ГОСТ 19.402-78. Для программы (комплекса программ), получаемой за счет использования ранее разработанных программных средств, документ "Описание программы" следует дополнять разделом "Настройка программных средств".

Раздел "Настройка программных средств" должен содержать: наименование, обозначение использованных программных средств, описание процедур, необходимых для их настройки, или ссылки на эксплуатационную документацию этих средств; перечень элементов использованных программных средств, необходимых для получения программы (комплекса программ); описание настройки на языке, предусмотренном в эксплуатационной документации на используемые программные средства.

Руководство программиста по составу разделов и их содержанию должно соответствовать ГОСТ 19.504-79 и, кроме того, включать раздел "Сведения о форме представления программы (комплекса программ)".

Раздел "Сведения о форме представления программы (комплекса программ)" должен содержать сведения о носителе, на котором записана программа, о содержании и системе кодирования информации, записанной на носителе, а также сведения, необходимые для чтения информации с носителя.

Для программы (комплекса программ), допускающей настройку на условия конкретного применения, в документ "Руководство программиста" включают разделы: структура программы; настройка программы; дополнительные возможности; сообщения системному программисту.

Содержание этих разделов должно соответствовать требованиям ГОСТ 19.503—79. Разрешается для программы (комплекса программ), допускающей настройку на условия конкретного применения, вместо разделов, перечисленных выше, разрабатывать отдельный документ "Руководство системного программиста", удовлетворяющий требованиям ГОСТ 19.503—79.

Описание контрольного примера должно содержать разделы: назначение, исходные данные; результаты расчета; проверка программы (комплекса программ).

Раздел "Назначение" должен содержать перечень параметров и краткую характеристику функций из числа реализуемых программой (комплексом программ), проверяемых контрольным примером.

Раздел "Исходные данные" должен содержать описание исходных данных для проверки программы (комплекса программ) и сами исходные данные. Допускается исходные данные представлять в виде распечатки с АЦПУ.

Раздел "Результаты расчета" должен содержать результаты обработки исходных данных программой (комплексом программ), позволяющие оценить правильность выполнения проверяемых функций и значений проверяемых параметров. Допускается результаты расчета приводить в виде распечатки с АЦПУ.

Раздел "Проверка программы (комплекса программ)" должен содержать: описание состава технических средств, необходимых для работы программы (комплекса программ), или ссылку на соответствующие программные документы; описание процедур формирования исходных данных для проверки программы (комплекса программ), вызова проверяемой программы (комплекса программ) и получения результатов расчета; описание действий оператора при подготовке исходных данных и проверке программы (комплекса программ) на контрольном примере.

Документация по организационному обеспечению. Предназначена для описания изменений организационной структуры управления объектом, связанных с созданием АСУ (схема организационной структуры, описание организационной структуры); описаний действий персонала по обеспечению функционирования АСУ (технологические инструкции, инструкции по эксплуатации); установления функций, прав и обязанностей должностных лиц по обеспечению функционирования АСУ (должностные инструкции). Содержит схему и описание организационной структуры; технологические инструкции процесса обработки данных; инструкции по эксплуатации технических средств (не включая сведений, имеющихся в документации предприятия-изготовителя); должностные инструкции.

Документы по вводу в эксплуатацию. Документы этой стадии относятся к приемо-сдаточной документации на АСУ. В нее входят следующие документы.

Акт завершения работ для фиксации факта завершения отдельной работы при создании АСУ (не распространяется на строительно-монтажные и пусконаладочные работы).

Акт приемки в опытную эксплуатацию для фиксации факта завершения ввода АСУ в целом или ее частей в опытную эксплуатацию.

Акт приемки в промышленную эксплуатацию для фиксации факта ввода АСУ (или ее части) в промышленную эксплуатацию. К "Акту приемки в промышленную эксплуатацию" прилагают программу и протоколы испытаний, протоколы заседаний комиссии, акты приемки в промышленную эксплуатацию принятых ранее частей АСУ, перечень технических средств, которые использовала комиссия при приемке АСУ.

План-график работ устанавливает перечень работ, сроки выполнения и исполнителей работ, связанных с созданием АСУ.

Кроме того, должны быть подготовлены приказы: о проведении работ; о готовности объекта к проведению строительно-монтажных и наладочных работ; о начале опытной эксплуатации и вводе в промышленную эксплуатацию АСУ или ее частей; о составе приемочной комиссии.

§ 9.4. Порядок утверждения документов и ввода АСУ в эксплуатацию

Порядок утверждения документов. Техничко-экономическое обоснование и техническое задание на создание АСУ согласовываются с головной организацией по созданию АСУ в отрасли, генеральным проектировщиком предприятия (при строительстве или реконструкции), головной организацией Минприбора, закрепленной за данной отраслью, отделом отраслевого министерства, который осуществляет руководство работами по созданию АСУ в отрасли, а также с Всесоюзным промышленным объединением, в ведении которого находится предприятие, и утверждаются руководством министерства (ведомства) заказчика.

ТЭО и ТЗ на создание АСУ на базовых предприятиях (объединениях) в соответствии с утвержденными ГКНТ программами работ по решению научно-технических проблем дополнительно согласовываются с главным управлением вычислительной техники и систем управления ГКНТ и утверждаются руководством министерства (ведомства) заказчика.

Утвержденное ТЗ направляется (по одному экземпляру) в головную организацию Минприбора, организации-разработчику, головной организации министерства заказчика, а по АСУ базовых предприятий, кроме того, — в ГКНТ. Корректировка ТЗ на последующих стадиях создания АСУ осуществляется в порядке, установленном для вновь разрабатываемых технических заданий.

Технический проект согласовывается разработчиком с заказчиком, головной организацией по созданию АСУ в отрасли, головной организацией Минприбора, закрепленной за данной отраслью. Если проект АСУ является частью проекта строительства или реконструкции предприятия, на котором создается АСУ, он дополнительно согласовывается с генеральным проектировщиком предприятия.

Рассмотрение, обсуждение и согласование документации технического проекта заказчиком осуществляются в соответствующие сроки, но не более месяца со дня получения комплекта документации, другими организациями — не более двух недель.

Решения технического проекта по отдельным объектам АСУ (общестроительным, противопожарным, радиотехническим и др.), разработанные в соответствии с действующими нормами и правилами (о чем должна быть соответствующая запись в материалах проекта), не подлежат согласованию с органами государственного надзора.

В отдельных необходимых случаях частичные отступления от действующих норм и правил должны быть предварительно согласованы разработчиком с соответствующими органами в порядке, установленном министерством или ведомством, утвердившим эти правила и нормы. При отсутствии утвержденных норм и правил по отдельным вопросам проектные решения согласуются с соответствующими органами государственного надзора.

Головные организации, осуществляющие согласование технической документации, не должны требовать от разработчика представления технической документации в большем объеме и составе, а также с большей степенью детализации, чем это предусмотрено методическими материалами или действующими государственными стандартами по составу и содержанию проектов АСУ.

Разработчик передает заказчику полный комплекс согласованного технического проекта в двух экземплярах, заказную документацию и требования к архитектурно-строительной части проекта в четырех экземплярах. Организации-соисполнители передают генеральному разработчику соответствующую документацию технического проекта в трех, а заказную документацию и требования к архитектурно-строительной части проекта в пяти экземплярах.

Заказчик составляет и подписывает акт утверждения технического проекта, который должен содержать результаты обсуждения и согласования, в том числе оценку полноты и качества разработки. Акт утверждения и технический проект представляются заказчиком для утверждения в министерство заказчика.

Разработчик принимает участие в защите проекта в организациях, осуществляющих согласование и утверждение проекта. Переутверждение технических проектов, в которые внесены изменения, направленные на улучшение технико-экономических показателей АСУ, снижение затрат и сокращение сроков разработки и внедрения, проводится в порядке, установленном для вновь разработанных проектов.

Полный комплект утвержденного технического проекта вместе с актом о его утверждении должен храниться у заказчика и в организации, разработавшей проект.

Все программы и инструкции, проведенные разработчиком на контрольном примере, сдаются заказчику по акту, удостоверяющему их прием-

ку в опытную эксплуатацию. Сдаваемые заказчику программы, инструкции и описания алгоритмов должны соответствовать требованиям, предъявляемым к составу и содержанию рабочего проекта АСУ. Заказчику передаются программы, записанные на магнитных лентах. Заказчик обеспечивает разработчика магнитной лентой и машинным временем, необходимыми для сдачи программ в опытную эксплуатацию и их дублирования.

Приемка комплексов задач (подсистем) в опытную эксплуатацию заключается в решении контрольного примера специально подготовленным персоналом заказчика в присутствии представителей разработчика с последующим анализом результатов. По взаимному согласованию контрольный пример может выполняться разработчиком в присутствии заказчика.

По результатам приемки подписывается акт сдачи-приемки программ в опытную эксплуатацию. Обнаруженные ошибки разработчика в программах и другой технической документации устраняются в процессе сдачи в опытную эксплуатацию.

Порядок ввода АСУ в эксплуатацию. Ввод в эксплуатацию АСУ и ее отдельных элементов представляет собой процесс постепенного перехода от существующих методов "ручного" управления к методам автоматизированного управления. Проводится он силами заказчика при участии разработчика и организаций-соисполнителей. Взаимодействие организаций заказчика, разработчика и соисполнителей осуществляется на основании договорных условий и плана-графика сдачи АСУ в промышленную эксплуатацию.

Ввод в эксплуатацию осуществляется поэтапно, начиная со стадии разработки технического проекта по мере готовности рабочей документации и ввода в эксплуатацию технических средств, обеспечивающих внедрение очередей или объектов АСУ, способных к самостоятельному функционированию.

К вводу АСУ в эксплуатацию следует приступать при наличии: оформленных документов о выполнении плана мероприятий по подготовке объекта; рабочей документации на внедрение выделенной очереди или АСУ в целом; обученного персонала, обеспечивающего подготовку к вводу АСУ в эксплуатацию; принятых в эксплуатацию технических средств АСУ, обеспечивающих функционирование внедряемых комплексов задач и подсистем.

На стадии ввода АСУ в эксплуатацию осуществляются: опытная эксплуатация отдельных задач и их комплексов; приемка комплексов задач в промышленную эксплуатацию; проведение приемо-сдаточных испытаний; приемка АСУ в промышленную эксплуатацию.

Состав и очередность выполнения работ определяются согласованными планами-графиками ввода в эксплуатацию, в которых указываются состав и сроки выполнения работ: по строительству, монтажу, наладке и испытанию объектов АСУ с момента получения рабочей документации до сдачи объектов в промышленную эксплуатацию; по проведению опытной эк-

сплуатации, приемо-сдаточных испытаний комплексов задач; по обеспечению перехода от существующих методов управления к методам, предусмотренным проектом АСУ.

Обязанности заказчика: завершить выполнение организационно-технических мероприятий по подготовке предприятия к внедрению и оформить их актами; обеспечить выполнение персоналом предприятия должностных и технологических инструкций; ввести в эксплуатацию технические средства, необходимые для внедряемого технологического процесса обработки данных; издать приказ с планом-графиком о проведении опытной эксплуатации АСУ и проанализировать совместно с разработчиком результаты опытной эксплуатации; завершить опытную эксплуатацию комплексов задач АСУ и приемку их в промышленную эксплуатацию; внести изменения в организационную структуру предприятия в соответствии с проектом АСУ; разработать проект приказа по составу приемочной комиссии; разработать и согласовать с разработчиком проект программы приемо-сдаточных испытаний; организовать работу приемочной комиссии, представить ей требуемую документацию, провести испытания АСУ; проверить эффективность реализованных решений в условиях промышленной эксплуатации, по результатам анализа функционирования системы разработать рекомендации по ее дальнейшему развитию.

Обязанности разработчика: скорректировать техническую документацию по результатам опытной эксплуатации АСУ; принимать участие в разработке проекта программы приемо-сдаточных испытаний АСУ; осуществлять методическое руководство и принимать участие в сдаче задач (комплексов задач) в промышленную эксплуатацию; участвовать в работе комиссии по приемке АСУ в промышленную эксплуатацию.

При поручении на договорных условиях работ по анализу функционирования АСУ разработчику заказчик обязан: подготовить и издать приказ о проведении работы, собрать и представить данные и документы, необходимые для анализа системы, обеспечить проведение работ по проверке функционирования АСУ. На основе анализа разработчик обязан разработать и передать заказчику предложения по основным направлениям развития АСУ.

Опытная эксплуатация задач (комплексов задач) осуществляется заказчиком с участием разработчика в соответствии с программой опытной эксплуатации, должностными и технологическими инструкциями. Возникшие в процессе опытной эксплуатации дополнительные требования заказчика, не предусмотренные в техническом задании и техническом проекте, не являются основанием для отрицательной оценки результатов опытной эксплуатации и могут быть удовлетворены по дополнительному соглашению в согласованные сроки.

При положительных результатах опытной эксплуатации комплексов задач (подсистем) составляется двусторонний акт о приемке их в промышленную эксплуатацию, после чего ответственность за их функционирование несет заказчик.

Приемка АСУ в промышленную эксплуатацию производится по завершению приемки заказчиком всех комплексов задач в промышленную эксплуатацию. Задачи, комплексы задач и технические средства АСУ, не предусмотренные в техническом задании, но внедренные заказчиком самостоятельно, могут быть включены в комплекс сдаваемой АСУ только по согласованию с разработчиком и после внесения в ТЗ на создание АСУ соответствующих изменений.

Заказчик предъявляет АСУ приемочной комиссии, назначаемой распоряжением министерства (ведомства) заказчика по согласованию с министерством (ведомством) разработчика. Помимо членов комиссии к приемке могут привлекаться с правом совещательного голоса эксперты по отдельным вопросам создания АСУ. Он обязан обеспечить нормальные условия работы комиссии в соответствии с принятой программой приемки АСУ. Для оперативного решения организационных вопросов, возникающих в процессе приемки АСУ, приказом руководителя организации-разработчика выделяется ответственный представитель — заместитель директора, главный инженер или заместитель главного инженера.

Совместно с разработчиком заказчик подготавливает проект программы проведения испытаний и приемки АСУ и передает его приемочной комиссии для рассмотрения и утверждения. В программе указывается: наименование сдаваемой АСУ, директивные документы, на основании которых разрабатывалась система, состав приемочной комиссии и номер приказа о ее назначении, цель, объекты, объем, место и последовательность испытаний, методика проведения испытаний и оценки результатов. Кроме того, во временное пользование комиссии передаются следующие документы: приказы, распоряжения, планы, договоры, предусматривающие создание АСУ; технико-экономическое обоснование, техническое задание, технический и рабочий проекты АСУ; акты рассмотрения и утверждения технического проекта; акты, подтверждающие выполнение мероприятий по подготовке предприятия к внедрению АСУ; акты об окончании монтажа технических средств АСУ; двусторонние акты заказчика и разработчика по сдаче задач, комплексов задач, устройств и их комплексов в промышленную эксплуатацию в соответствии с утвержденным техническим заданием; приказы по предприятию о вводе в промышленную эксплуатацию отдельных задач, комплексов задач; штатное расписание подразделений предприятия, обслуживающих АСУ, сведения об их укомплектованности специалистами и квалификации последних.

Приемочная комиссия обеспечивает: проверку документации и функционирования АСУ в сроки, определенные приказом министерства (ведомства), назначившего комиссию; организацию рабочих групп и распределение обязанностей среди членов комиссии для проверки отдельных подсистем; проверку расчета экономической эффективности АСУ и научно-технического уровня; организацию рабочих совещаний и подготовку актов приемки АСУ.

Проверка условий эксплуатации, режима работы технических средств

АСУ проводится одновременно с проверкой функционирования комплексов задач.

Подготовленность персонала, ответственного за функционирование АСУ, определяется в соответствии с программой обучения и должностными инструкциями, входящими в состав рабочего проекта. Результаты проверки обсуждаются на рабочих совещаниях и оформляются протоколами.

Завершающим этапом работы комиссии является составление акта, в котором указываются: состав комиссии, должности и места работы членов комиссии; срок (дата) приемки; состав исполнителей (организаций, предприятий), принимавших участие в создании АСУ; основания для проведения приемки (приказы, распоряжения и т.д.); перечень предъявленной документации АСУ и оценка ее соответствия действующим нормативно-техническим документам; соответствие фактически выполненных и внедренных работ техническому заданию; готовность всех видов обеспечения и структурных подразделений заказчика к внедрению и эксплуатации АСУ; сведения об эффективности АСУ (сопоставление имеющихся фактических или ожидаемых данных по объему и источникам получаемой экономии с расчетными данными); выводы комиссии о возможности приемки АСУ; основные рекомендации комиссии по дальнейшему развитию АСУ и возможности использования проектных решений АСУ, создаваемых на других предприятиях.

Акт приемки в пяти экземплярах подписывается председателем и всеми членами комиссии. Дата подписания акта комиссией считается датой ввода АСУ в эксплуатацию.

Бурное развитие информатики и вычислительной техники, происходящее в мире, приводит к кардинальным изменениям содержания и форм труда миллионов людей. По данным, приведенным в [2.3], уже в настоящее время в сфере обработки информации занято около 50% трудоспособного населения промышленно развитых стран. В США по прогнозам ожидалось, что число системотехников и программистов возрастет с 180 тыс. в 1970 г. до 500 тыс. человек к 1990 г. Однако уже в 1980 г. их число составило 530 тыс. человек, причем в основном за счет системотехников, число которых возросло за десять лет в 3 раза, составив 300 тыс. человек. Уровень их оплаты является наиболее высоким, наравне с системными программистами, т.е. на 15–25% выше оплаты прикладных программистов.

В нашей стране начата обширная программа всеобщего компьютерного обучения, которая приведет к большим трудно предсказуемым социально-экономическим последствиям, сопоставимым со значением всеобщей грамотности населения. Уже через несколько лет появятся специалисты самых различных профессий, способные самостоятельно формализовать свои профессиональные знания и создавать программы для решения возникающих у них задач на персональных компьютерах. Это обстоятельство не может не сказаться на характере деятельности системотехников-проектировщиков АСУ. Представляется, что развитие будет идти в двух направлениях.

Первое направление в основном сохранит характер проектирования современных сложных АСУ административного и организационно-технологического типа на уровне предприятия и выше. Общая компоновка и идеология системы, проектирование технического и информационного обеспечения многоуровневого и сетевого типа, выявление и постановка новых задач, профессиональный выбор методов решения сложных задач управления и другие основные решения остаются прерогативой специалистов-системотехников.

В то же время неуклонное усложнение объектов, вовлекаемых в сферу автоматизации, объективно отражается на характере задач АСУ, методах и способах их решения и в конечном счете на общей методологии проектирования АСУ.

Установлено, например, что по мере усложнения объектов автоматизации группирование факторов, существенно влияющих на эффективность работ по созданию АСУ, все больше смещается к начальным этапам проектирования и для сложных, крупномасштабных систем соотносится в основ-

ном со стадиями внешнего и концептуального проектирования (стадия концептуального проектирования для сложных систем образует своего рода мостик, на котором имеется компромисс между потребностями заказчика и возможностями их удовлетворения разработчиком в данных конкретных условиях). Здесь необходимо особое внимание уделить поиску обоснованных ответов на вопросы: возможно ли в принципе создание АСУ, удовлетворяющей исходным требованиям и будет ли созданная АСУ эффективной? И чем сложнее проектируемая АСУ, тем весомее цена ошибок, допущенных на начальных стадиях проектирования.

Важным моментом является возрастающая активность пользователя, его умение самостоятельно обращаться к системе за получением нужной информации в удобном для него виде, формулировать и решать относительно простые задачи. Это также должно учитываться при проектировании АСУ.

Второе направление заключается в разработке систем, обеспечивающих сервис активному пользователю. В это достаточно широкое понятие входят средства создания независимых от пользователя, но предоставляющих ему возможности активного использования информационных систем; специализированные проблемно-ориентированные комплексы программ самообучения пользователя и помощи ему при самостоятельной постановке своих задач и их программировании; предназначенные для определенных групп пользователей интерфейсы человек — ЭВМ, включая графические образы, речь и т.д.

В перспективе — это мощная инструментальная система, обеспечивающая режим "автопроектирования" проблемно-ориентированных АСУ (информационного, контрольного, контрольно-аналитического и других ориентаций). Созданная на основе единых средств представления знаний о предметной области и проблемной среде, единых средств описания и манипулирования данными, обладая возможностью формировать и актуализировать банк методов решения типовых задач, такая инструментальная система позволит непрограммирующему профессионалу самому "проектировать АСУ", настраивая последнюю на свои информационные потребности.

Изменится сам характер труда проектировщика, который будет работать за пультом персонального компьютера, используя хранящуюся в обширной памяти мощную базу знаний. Постепенно изменится традиционное представление о технической документации — проектирование будет осуществляться по "бесбумажной" технологии.

Переход к гибким автоматизированным производствам с безлюдной технологией, развитие систем автоматизированного проектирования выпускаемой этими производствами продукции с непосредственной связью рабочего места проектировщика и производственного комплекса приведут к коренному изменению содержания многих традиционных подсистем АСУП, таких, как оперативное управление, технологическая подготовка производства, материально-техническое снабжение и др.

Сообразно изменениям характера производственных процессов, развитию средств вычислительной техники, расширению их вычислительных и логических возможностей будут меняться и совершенствоваться методы и средства проектирования.

Хотелось бы надеяться, что хотя бы некоторые изложенные в учебнике соображения и рекомендации принципиального, концептуального характера сохранятся и при переходе на будущую, более высокую ступень проектирования.

Рекомендуемая (1)

- 1.1. Автоматизированные системы управления предприятиями / Под ред. *В.Н. Четверикова*. — М.: Высшая школа, 1979.
- 1.2. *Бусленко Н.П.* Моделирование сложных систем. — М.: Наука, 1978.
- 1.3. *Липаев В.В.* Качество программного обеспечения. — М.: Финансы и статистика, 1983.
- 1.4. *Мамиконов А.Г.* Основы построения АСУ. — М.: Высшая школа, 1981.
- 1.5. Общеотраслевые руководящие методические материалы по созданию автоматизированных систем управления предприятиями и производственными объединениями. — М.: Статистика, 1977.
- 1.6. *Овчаров Л.А., Селетков С.Н.* Автоматизированные банки данных. — М.: Финансы и статистика, 1982.
- 1.7. Основы построения АСУ / *Т.Н. Рахимов, О.А. Заикин, Б.Я. Советов*. — Ташкент: Укитувчи, 1984.
- 1.8. Проектирование автоматизированных систем управления / *А.А. Аишмов, В.В. Кульба, А.Г. Мамиконов*. — Алма-Ата: Мектеп, 1983.

- 1.9. Справочник разработчика АСУ / *А.А. Модин, Е.Г. Яковенко, Е.П. Погребной*. — М.: Экономика, 1978.
- 1.10. Справочник проектировщика АСУ ТП / Под ред. *Г.Л. Смилянского*. — М.: Машиностроение, 1983.

Использованная (2)

- 2.1. *Ван Тассел Д.* Стиль, разработка, эффективность, отладка и испытание программ: Пер. с англ. — М.: Мир, 1981.
- 2.2. Внедрение типовой системы управления предприятием / *В.М. Португал, А.И. Семенов, А.Л. Марголин*. — М.: Наука, 1981.
- 2.3. *Громов Г.Р.* Национальные информационные ресурсы: проблемы промышленной эксплуатации. — М.: Наука, 1984.
- 2.4. *Дейт К.* Введение в системы баз данных: Пер. с англ. — М.: Наука, 1980.
- 2.5. *Евдокимов В.В., Рейнер В.А.* Машинный синтез АСУП. — М.: Статистика, 1980.
- 2.6. Исследование операций: Пер. с англ. *Майзер Х., Эйджин Н., Тролл Р.* и др. — М.: Мир, 1981. Т. 1.
- 2.7. *Липаев В.В.* Проектирование математического обеспечения АСУ. — М.: Сов. радио, 1977.
- 2.8. Локальные микропроцессорные вычислительные сети / *И.В. Прангизвили, В.С. Подлазов, Г.Г. Стецюра*. — М.: Наука, 1984.
- 2.9. *Майерс Г.* Искусство тестирования программ: Пер. с англ. — М.: Финансы и статистика, 1982.
- 2.10. *Мартин Дж.* Организация баз данных в вычислительных системах: Пер. с англ. — М.: Мир, 1980.

- 2.11. Мельцер М.И. Разработка алгоритмов АСУП. — М.: Статистика, 1975.
- 2.12. Методика определения экономической эффективности автоматизированных систем управления предприятиями и производственными объединениями / ГКНТ СССР. — М.: Статистика, 1979.
- 2.13. Модель АСУ промышленными предприятиями / В.А. Кирлев, Л.Н. Куранова, Е.А. Перфильева и др. — М.: Статистика, 1978.
- 2.14. Олле Т.В. Предложения Кодасил по управлению базами данных: Пер. с англ. — М.: Финансы и статистика, 1981.
- 2.15. Операционная система СМ ЭВМ РАФОС: Справочник / Под ред. Семика В.П. — М.: Финансы и статистика, 1984.
- 2.16. Оптимальные модульные системы обработки данных / А.А. Ашимов, А.Г. Мамиконов, В.В. Кульба. — Алма-Ата: Наука, 1981.
- 2.17. Построение и анализ вычислительных алгоритмов / А.Ахо, Дж. Хонкрофт, Дж. Ульман. — М.: Мир, 1979.
- 2.18. Принципы разработки программного обеспечения: Пер. с англ. / М. Зелкович, А. Шоу, Дж. Геннон. — М.: Мир, 1982.
- 2.19. Система математического обеспечения ЕС ЭВМ / В.Г. Лесюк, А.С. Марков, Г.В. Пеледов, Л.Д. Райков. — М.: Статистика, 1974.
- 2.20. Структурное программирование: Пер. с англ. / У. Дал, Э. Дейкстра, К. Хоор. — М.: Мир, 1975.
- 2.21. Тиори Т., Фрай Дж. Проектирование структур баз данных: Пер. с англ. В 2 кн. Кн. 1, 2 — М.: Мир, 1985.
- 2.22. Типовые проектные решения АСУП: Подсистема бухгалтерского учета. — М.: Статистика, 1975.
- 2.23. Узлдон Дж.-Л. Администрирование баз данных в вычислительных системах: Пер. с англ. — М.: Мир, 1980.
- 2.24. Феррари Д. Оценка производительности вычислительных систем: Пер. с англ. — М.: Мир, 1981.
- 2.25. Хаббард Дис. Автоматизированное проектирование баз данных: Пер. с англ. — М.: Мир, 1984.
- 2.26. Шаймарданов Р.Б. Моделирование и автоматизация проектирования структур баз данных. — М.: Радио и связь. 1984.
- 2.27. Эпштейн В.Л. АРИУС: Автоматизация проектирования баз данных и пакетов прикладных программ: Препринт. — М.: Институт проблем управления, 1983.

Предметный указатель

Акцепт 89

База данных 126

— — реляционная 131

Банк данных 120

Ведомость платежная 98

Время реакции 51

Граф индикаторный 106

— информационный 142

Дамп 198

Дескриптор 119

Документ 109

Документация программная 286

Достоверность информации 138

Доступ к данным 135

Драйвер 171

Жизнеспособность эффективная 239

Избыточность информационная 142

— прагматическая 142

Калькуляция нормативная 68

Классификатор 122

Коэффициент готовности системы 51

Матрица достижимости 106

— смежности 106

Модель информационная матричная 37

Модуль загрузочный 168

— объектный 168

— — реентерабельный 169

Номенклатура 85

— специфицированная 85

Пакет прикладных программ 64

Память виртуальная 170

План снабжения 88

Подсистема АСУ 45

Полномочий делегирование 44

Потребность специфицированная 86

Прирост прибыли годовой 54

Проектирование сверху — вниз 181

— структурное 186

Реализуемость физическая 207

Режим разделения времени 173

Резервирование информационных массивов 155

Реквизит документа 109

Решения проектные типовые 65

Свопинг 167

Себестоимость нормативная 68

Сеть локальная вычислительная 218

Система классификации 122

Сложность алгоритма 181

– комплекса программ 206

Спецификации системные 37

Способность пропускная 51

Срок окупаемости 54

Структура организационная 43

– функциональная 44

– языковая 119

Схема обработки данных 151

– структурно-информационно-
временная 33

Счет балансовый 93

Тестран 205

Трудоемкость нормативная 69

Функция АСУ 44

Эффект волновой 187

Эффективность капитальных
вложений 54

Язык информационный 115

Оглавление

Предисловие	3
Введение	6
Глава 1. Основные стадии создания АСУ	9
§ 1.1. Регламентация порядка разработки	9
§ 1.2. Предпроектная стадия создания АСУ	12
§ 1.3. Стадия разработки проектов	16
§ 1.4. Стадия ввода в эксплуатацию	19
§ 1.5. Неформальные этапы разработки систем	22
§ 1.6. Разработка систем, не имеющих аналогов	26
§ 1.7. Распределение функций между специалистами	28
Глава 2. Проектирование систем на макроуровне	31
§ 2.1. Методы изучения и анализа существующих систем управления	~~31~~
§ 2.2. Структуризация системы	~~49~~
§ 2.3. Выбор задач, подлежащих автоматизации, и их постановка	46
§ 2.4. Общие характеристики системы	50
§ 2.5. Техничко-экономическая эффективность системы	53
§ 2.6. Научно-технический уровень АСУ	56
Глава 3. Проектирование функциональных подсистем	63
§ 3.1. Общие положения	~~63~~
§ 3.2. Техничко-экономическое планирование	66
§ 3.3. Календарное планирование	70
§ 3.4. Управление технической подготовкой производства	78
§ 3.5. Управление основным производством	81
§ 3.6. Управление материально-техническим снабжением	85
§ 3.7. Бухгалтерский учет	93
§ 3.8. Управление сбытом продукции	99
Глава 4. Проектирование информационного обеспечения	101
§ 4.1. Основные понятия и определения	~~101~~
§ 4.2. Методы анализа информационных потоков	105
§ 4.3. Проектирование форм входных данных и выходных результатов	108
§ 4.4. Информационные языки и классификаторы	115
§ 4.5. Организация баз данных и выбор систем управления базами данных	125
§ 4.6. Достоверность информационного обеспечения	138
§ 4.7. Защита информации	158

Г л а в а 5. Проектирование математического и программного обеспечения . . .	164
§ 5.1. Структура математического обеспечения	164
§ 5.2. Операционные системы ЭВМ и трансляторы	166
§ 5.3. Организация информационно-вычислительного процесса	173
§ 5.4. Алгоритмизация задач	179
§ 5.5. Структурное программирование	186
§ 5.6. Отладка программ и программных комплексов	192
§ 5.7. Оценка качества программных комплексов	205
§ 5.8. Организация и планирование процесса программирования	213
Г л а в а 6. Проектирование технического обеспечения	216
§ 6.1. Структура комплекса технических средств	226
§ 6.2. Методологические принципы проектирования комплекса технических средств	233
§ 6.3. Специальные технические средства	234
§ 6.4. Надежность технических средств	238
Г л а в а 7. Психологические аспекты проектирования	246
§ 7.1. Принятие решений руководителем	246
§ 7.2. Психологические факторы при поиске новых решений в процессе проектирования	256
Г л а в а 8. Имитационные модели	261
§ 8.1. Общие сведения	261
§ 8.2. Имитационные модели производственных процессов	265
§ 8.3. Имитационные модели предприятий и производственных объединений	268
Г л а в а 9. Техническая документация	271
§ 9.1. Общие положения и общесистемная документация	271
§ 9.2. Состав и содержание документов предпроектной стадии	275
§ 9.3. Состав и содержание документов проектной стадии по видам обеспечения	280
§ 9.4. Порядок утверждения документов и ввода АСУ в эксплуатацию	289
Заключение	295
Литература	297
Предметный указатель	300

Учебное издание

Акоп Гаспарович Мамиконов

ПРОЕКТИРОВАНИЕ АСУ

Заведующая редакцией Н.И. Хрусталева

Научный редактор А.Г. Зудин

Младший редактор Е.В. Судьенкова

Художественный редактор В.И. Мешалкин

Технический редактор И.А. Балелина

Корректор Р.К. Косинова

Оператор С.Г. Кучирь

ИБ № 6070

Изд. № Стд-512. Сдано в набор. 20.08.86. Подп. в печать 19.05.87. Т-10470.
Формат 60x88¹/₁₆. Бум. офсет. № 1. Гарнитура Пресс-Роман. Печать офсетная.
Объем 18,62 усл. печ. л. + 0,25 усл. печ. л. форзац. 19,11 усл. кр.-отт.
22,29 уч.-изд. л. + 0,43 уч.-изд. л. форзац.

Тираж 21 000 экз.

Заказ № 1299.

Цена 1 р. 10 к.

Издательство "Высшая школа", 101430, Москва, ГСП-4, Неглинная ул.,
д. 29/14.

Набрано на наборно-пишущих машинах издательства.

Отпечатано в Московской типографии № 8 Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли. 101898, Москва, Центр, Хохловский пер., 7.