

Глава 3. Структурный подход к разработке ПС

Проблема сложности является главной проблемой, которую приходится решать при создании больших и сложных систем любой природы (производственно-технических, организационно-экономических и др.). Для преодоления трудности разработки таких систем требуются специальные средства анализа и описания.

В структурном анализе выделяют два способа описания системы – либо с позиции функций, либо с позиции объектов. В первом случае строится функциональная модель, представляющая систему в виде набора функций, взаимодействующих через объекты системы. Во втором случае строится модель данных, представляющая систему в виде набора объектов, связанных функциями системы. Существует большое количество стандартов, позволяющих создавать как функциональные модели системы, так и модели данных.

Наиболее распространенные методологии, используемые для построения функциональных моделей системы: DFD, WFD, SADT, IDEF0, IDEF3, ORACLE, BAAN, ARIS и др. Несмотря на их разнообразие, все они основаны на двух классических подходах описания функций системы DFD и WFD. Методология DFD (Data Flow Diagram, диаграмма потоков данных) показывает функции верхнего уровня, и их взаимодействие через материальные или информационные потоки. Методология WFD (Work Flow Diagram, диаграмма потоков работ) показывает алгоритмы выполнения функций нижнего уровня. Современные методологии бизнес-моделирования (BAAN, ARIS и др.) помимо классических DFD и WFD моделей позволяют строить множество других моделей, отражающих такие аспекты системы, как стратегические цели, организационная структура, структура информации.

Для построения моделей данных обычно используются ER-диаграммы (Entity-Relationship Diagrams, диаграмма «сущность-связь»). Существуют несколько нотаций, применяемых для моделирования данных: П. Чена, Р. Баркера, IDEF1X и др. Как правило, различия между ними заключаются лишь в обозначениях. Многие CASE-средства позволяют автоматически переходить из одной нотации в другую.

3.1. Методология структурного анализа и проектирования SADT

3.1.1. Общие сведения

Методология структурного анализа и проектирования SADT (Structured Analysis and Design Technique) представляет модель системы в виде иерархической структуры взаимосвязанных диаграмм, отображающих функции системы и их взаимодействие через объекты системы. Она нашла свое применение для описания большого количества сложных систем из широкого спектра областей.

SADT разработана Д.Т. Россом, работа по ее созданию началась в конце 60-х гг. XX в. Первое значимое применение SADT было осуществлено в 1973 г. при разработке крупного аэрокосмического проекта. В 1974 г. SADT была использована в крупной европейской телефонной компании. От проекта к проекту SADT пересматривалась и совершенствовалась. Появление SADT на рынке произошло в 1975 г. после годичного оформления в виде продукта. К 1981 г. SADT использовалась уже в 50 компаниях при работе над 200 проектами с участием 2000 сотрудников. Проекты охватывали различные проблемные области: телефонные сети, аэрокосмическое производство, учет материально-технических ресурсов и др. [7].

В конце 70-х гг. Министерством обороны США была предложена и реализована Программа интегрированной компьютеризации производства *ICAM* (Integrated Computer Aided Manufacturing), направленная на увеличение эффективности промышленных предприятий посредством широкого внедрения компьютерных (информационных) технологий.

Реализация программы ICAM потребовала создания адекватных методов анализа и проектирования производственных систем и способов обмена информацией между специалистами, занимающимися такими проблемами. Для удовлетворения этой потребности в рамках программы ICAM была разработана методология моделирования *IDEF* (ICAM DEFinition), позволяющая исследовать структуру, параметры и характеристики производственно-технических и организационно-экономических систем. Общая методология IDEF состоит из нескольких частных методологий моделирования, основанных на графическом представлении систем: IDEF0, IDEF1, IDEF2, IDEF3 и др.

SADT реализована в одной из методологий IDEF – методологии функционального моделирования IDEF0. В 1993 г. методология IDEF0 была утверждена в качестве федерального стандарта США [12].

В 2001 г. в связи с расширением применения информационных технологий в народном хозяйстве Российской Федерации были изданы Рекомендации по стандартизации Р 50.1.028–2001 [11]. Рекомендации содержат основные сведения о методологии IDEF0 и графическом языке описания моделей, а также практические указания по ме-

тодике разработки таких моделей. Рекомендации по стандартизации Р 50.1.028–2001 предназначены для использования при анализе и синтезе производственно-технических и организационно-экономических систем методами функционального моделирования в различных отраслях экономики. Рекомендации содержат описание комплекса средств для наглядного представления широкого спектра деловых, производственных и других процессов и операций предприятия на любом уровне детализации, а также организационные и методические приемы применения этих средств.

Создание SADT способствовало в 60 – 70-е гг. XX в. решению проблемы неудовлетворительного описания и анализа сложных систем. Д.А. Марка так описывает эту проблему: «Эксплуатационные расходы, возникающие после сдачи систем, существенно превышали расходы на их создание и росли с большой скоростью из-за низкого качества разработанных систем. Исследования показывали, что основная часть ошибок в системах возникала в процессе анализа и проектирования, тогда как при реализации и тестировании их допускалось гораздо меньше. ... Наиболее катастрофические ошибки совершались на ранних этапах создания систем. Часто эти ошибки являлись следствием неполноты функциональных спецификаций или несогласованности между спецификациями и результатами проектирования» [7]. Ошибочное определение систем являлось результатом несовершенства методов, применяемых для их создания, особенно тех методов, которые использовались на ранних стадиях процесса. Появление SADT способствовало лучшему пониманию проблем на начальных этапах создания систем. Это позволило сократить затраты на создание и эксплуатацию систем, а также повысить их надежность.

Широкое применение SADT объясняется следующими особенностями этой методологии. Во-первых, SADT позволяет достаточно легко отражать такие системные характеристики, как управление, обратная связь и исполнители. Это объясняется тем, что SADT изначально возникла для проектирования систем из любой предметной области в отличие от других структурных методов, разработанных для проектирования только программных систем (например, метода М. Джексона). Графический язык и набор процедур анализа SADT позволяет составить описание более общей системы, частью которой является программная система. Поэтому SADT успешно применяется на ранних этапах жизненного цикла систем. Во-вторых, в SADT имеются развитые процедуры поддержки коллективной работы участников проекта, что дает определенные преимущества, особенно на ранних стадиях создания системы. Кроме того, широкое использование

SADT показало, что ее можно сочетать с другими структурными методами. Это достигается использованием графических SADT-описаний в качестве схем, связывающих воедино различные методы и применяемых для описания частей системы с различным уровнем детализации.

SADT использовали при решении широкого круга проблем. Ее основное применение связано с созданием автоматизированных систем и интегрированных компьютеризированных производств. Области эффективного применения SADT: планирование промышленного производства, организация материально-технического снабжения, очистка нефти, банковское дело, конфигурация компьютерных систем, разработка программных средств для оборонных систем и телефонных сетей, обучение персонала, автоматизация процессов производства и проектирования, системная поддержка и диагностика, долгосрочное и стратегическое планирование и др. Кроме того, под названием IDEF0 SADT применялась многими специалистами в военных и промышленных организациях. Широкий спектр областей указывает на универсальность и эффективность методологии SADT.

Стандарт IDEF0 основан на методологии SADT, их ключевые понятия идентичны, поэтому для простоты изложения далее будет использоваться название SADT.

Методология SADT предназначена для описания функций верхнего уровня и не отражает последовательности выполнения функций во времени. Для описания временной последовательности и алгоритмов выполнения функций системы разработан стандарт IDEF3 [14].

3.1.1.1. Основные термины, используемые SADT

Приведем основные термины, используемые SADT [11].

Блок (activity box) – прямоугольник, используемый для описания функции; блок содержит имя и номер.

Ветвление – разделение дуги на несколько сегментов.

Внутренняя дуга – входная, управляющая или выходная дуга, а также дуга механизма, которые связывают блоки одной диаграммы. Отличается от граничной дуги.

Входная дуга – класс дуг, отображающих вход блока, т.е. информационные или материальные объекты, которые преобразуются функцией в выход. Входные дуги связываются с левой стороной блока.

Выходная дуга – класс дуг, отображающих выход блока, т.е. ма-

териальные или информационные объекты, произведенные функцией. Выходные дуги связываются с правой стороной блока.

Глоссарий (glossary) – список определений для ключевых слов, фраз и аббревиатур, связанных с узлами, блоками, дугами или с моделью SADT в целом.

Граничная дуга – дуга, один из концов которой связан с блоком, а другой не присоединен ни к какому блоку на диаграмме. Отображает связь диаграммы с другими блоками системы и отличается от внутренней дуги.

Декомпозиция (decomposition) – разделение моделируемой функции на функции-компоненты.

Дерево узлов – представление отношений между родительскими и дочерними узлами модели SADT в форме древовидного графа. Имеет то же значение и содержание, что и термин «перечень узлов».

Диаграмма – часть модели, описывающая декомпозицию блока.

Диаграмма А-0 (А дефис ноль) – специальный вид (контекстной) диаграммы, состоящей из одного блока, описывающего функцию верхнего уровня, ее входы, выходы, управление и механизмы, вместе с формулировками цели модели и точки зрения, с которой строится модель.

Диаграмма-иллюстрация FEO (For Exposition Only diagrams) – графическое описание, используемое для сообщения специфических фактов о диаграмме. При построении диаграмм FEO можно не придерживаться правил SADT.

Дочерний блок (child box) – блок на дочерней (порожденной) диаграмме.

Дочерняя диаграмма (child diagram) – диаграмма, детализирующая родительский (порождающий) блок.

Дуга (arrow) – направленная линия, состоящая из одного или нескольких сегментов, которая моделирует канал, передающий информационные или материальные объекты от источника (начальная точка дуги), к приемнику (конечная точка дуги со стрелкой). Имеется четыре класса дуг: входная, выходная, управляющая и дуга механизма (в т.ч. вызова). (См. термины «сегмент дуги», «граничная дуга», «внутренняя дуга»).

Дуга вызова – вид дуги механизма, который обозначает обращение из блока данной модели (или части модели) к блоку другой модели (или другой части той же модели) и обеспечивает связь между модулями или между разными частями одной модели.

Дуга механизма – класс дуг, которые отображают механизмы, т.е. средства, используемые для выполнения функции; включает спе-

специальный случай дуги вызова. Дуги механизмов связываются с нижней стороной блока.

Дуга, помещенная в туннель (туннельная дуга) – граничная дуга со специальной нотацией в виде круглых скобок в начале или в конце дуги. Граничная дуга, оканчивающаяся скобками там, где она присоединяется к блоку, означает, что данные, выраженные этой дугой, не обязательны на следующем уровне декомпозиции. Граничная дуга, начинающаяся скобками на свободном конце, означает, что представляемые ею данные отсутствуют на родительской диаграмме.

Имя блока – глагол или глагольный оборот, помещенный внутри блока и описывающий моделируемую функцию.

Интерфейс – разделяющая граница, через которую проходят материальные или информационные объекты; соединение между двумя или большим числом компонентов модели, передающее материальные или информационные объекты от одного компонента к другому.

Код ICOM (аббревиатура Input – вход, Control – управление, Output – выход, Mechanism – механизм) – код, обеспечивающий соответствие граничных дуг дочерней диаграммы с дугами родительского блока; используется для ссылок.

Контекст – окружающая среда, в которой действует функция (или комплект функций на диаграмме).

Контекстная диаграмма – диаграмма, имеющая узловой номер A–n (A дефис n) ($n \geq 0$), которая представляет контекст модели. Диаграмма A–0, состоящая из одного блока, является обязательной контекстной диаграммой; диаграммы с узловыми номерами A–1, A–2, ..., – дополнительные контекстные диаграммы.

Метка дуги (arrow label) – существительное или оборот существительного, связанные с дугой или сегментом дуги и определяющие их значение.

Модель SADT – графическое описание системы, разработанное с определенной целью (см. термин «цель») и с выбранной точки зрения (см. термин «точка зрения»). Комплект документов, которые изображают объекты и функции системы с помощью графических диаграмм, текста и глоссария.

Номер блока – число, помещаемое в правом нижнем углу блока и однозначно идентифицирующее блок на диаграмме.

Перечень узлов – многоуровневый список, показывающий узлы модели SADT в упорядоченном виде. Имеет то же значение и содержание, что и термин «дерево узлов».

Примечание к модели – текстовый комментарий, являющийся

частью диаграммы и используемый для записи факта, не нашедшего графического изображения.

Родительская диаграмма (parent diagram) – диаграмма, которая содержит родительский блок.

Родительский блок (parent box) – блок, который подробно описывается дочерней диаграммой.

Связывание/развязывание – объединение значений дуг в составное значение (связывание) или разделение значений дуг (развязывание), выраженные синтаксисом слияния или ветвления дуг.

Сегмент дуги – сегмент линии, который начинается или заканчивается на стороне блока, в точке ветвления или слияния, или на границе (несвязанный конец дуги).

Семантика – содержание (значение) синтаксических компонентов языка.

Синтаксис – структурные компоненты языка, их характеристики и правила, определяющие связи между компонентами.

Слияние – объединение двух или большего числа сегментов дуг в один сегмент.

С-номер – номер, создаваемый в хронологическом порядке и используемый для идентификации диаграммы и прослеживания ее истории; может быть использован в качестве ссылочного выражения при определении конкретной версии диаграммы. Обычно С-номер состоит из инициалов автора модели и хронологических данных (даты создания очередной версии диаграммы).

Текст – любой текстовый (не графический) комментарий к графической диаграмме SADT.

Тильда – небольшая ломаная молниеобразная линия, используемая для соединения метки с конкретным сегментом дуги или примечания модели с компонентом диаграммы.

Точка зрения (viewpoint) – указание на должностное лицо или подразделение организации, с позиции которого разрабатывается модель. Для каждой модели точка зрения единственная.

Узел (node) – блок, порождающий дочерние блоки; родительский блок. (См. термины «перечень узлов», «дерево узлов», «узловой номер», «узловая ссылка», «номер узла диаграммы»).

Узловая ссылка – код, присвоенный диаграмме для ее идентификации и определения положения в иерархии модели; формируется из сокращенного имени модели и узлового номера диаграммы с дополнительными расширениями.

Узловой номер – код, присвоенный блоку и определяющий его положение в иерархии модели; может быть использован в качестве

подробного ссылочного выражения.

Узловой номер диаграммы – часть узловой ссылки диаграммы, которая соответствует номеру родительского блока.

Управляющая дуга – класс дуг, которые в SADT отображают управления, т.е. условия, при выполнении которых выход блока будет правильным. Данные или объекты, моделируемые как управления, могут преобразовываться функцией, создающей соответствующий выход. Управляющие дуги связываются с верхней стороной блока.

Функция (activity) – деятельность, процесс или преобразование (моделируемые блоком), идентифицируемое глаголом или глагольной формой, которая описывает, что должно быть выполнено.

Цель (purpose) – краткая формулировка причины создания модели.

3.1.1.2. Основные концептуальные положения SADT

Определение. Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Методология SADT основана на следующих концептуальных положениях:

1. *Модель SADT* – графическое описание системы, разработанное с определенной целью и с выбранной точки зрения. Модель состоит из комплекта документов, которые изображают объекты и функции системы с помощью графических диаграмм, текста и глоссария.

2. *Блочное моделирование и его графическое представление.* В SADT все, что происходит в системе, принято называть функциями. На диаграмме функция отображается в виде блока. Интерфейсы, посредством которых блок взаимодействует с другими блоками или с внешней по отношению к моделируемой системе средой, представляются дугами, входящими в блок или выходящими из него.

3. *Лаконичность и точность* документации, описывающей систему. Графический язык позволяет лаконично, однозначно и точно показать элементы системы и связи между ними, выявить ошибочные, лишние или дублирующие связи.

4. *Передача информации.* Средства SADT облегчают передачу информации от одного участника разработки модели к другому. К

числу таких средств относятся:

- диаграммы, основанные на простой графике блоков и дуг, легко читаемые и понимаемые;
- метки на естественном языке для описания блоков и дуг, а также глоссарий и сопроводительный текст, уточняющие смысл элементов диаграммы;
- последовательная декомпозиция диаграмм, строящаяся по иерархическому принципу, при котором на верхнем уровне отображаются основные функции, а на нижних уровнях происходит их детализация и уточнение;
- древовидные схемы иерархии диаграмм и блоков, обеспечивающие обозримость модели в целом и входящих в нее деталей, что особенно важно при моделировании больших систем.

5. *Строгость и формализм.* Разработка моделей SADT требует соблюдения ряда строгих формальных правил, обеспечивающих однозначность, точность и целостность сложных многоуровневых моделей. Для этого на всех этапах разработки должны строго, формально соблюдаться синтаксические и семантические правила графического языка, а результаты тщательно документироваться.

6. *Итеративное моделирование.* Разработка модели в SADT представляет собой пошаговую, итеративную процедуру. На каждом шаге итерации разработчик предлагает вариант модели, который подвергают обсуждению, рецензированию и последующему редактированию, после чего цикл повторяется. Это позволяет точно описывать системы с помощью модели.

7. *Отделение функций системы от ее организационной структуры.* При разработке моделей следует отделять функции моделируемой системы от существующей организационной структуры (предприятия, фирмы). Это позволит избежать субъективной точки зрения предприятия и ее руководства. Организационная структура должна явиться результатом применения модели. Сравнение результата с существующей организационной структурой позволяет, во-первых, оценить адекватность модели, а во-вторых – предложить решения, направленные на совершенствование этой структуры.

3.1.1.3. SADT-модели

Рассмотри основные понятия, на которых базируется SADT.

Под термином *система* понимается совокупность взаимодействующих компонент и взаимосвязей между ними. Компонентами си-

системы могут быть любые комбинации разнообразных сущностей, включающие людей, информацию, программные средства, оборудование, изделия, сырье или энергию (энергонасосители). Например, планеты в составе Солнечной системы; системы молекулярных взаимодействий в живых организмах; телефонные сети и др.

Под термином *моделирование* понимается процесс точного описания системы. Модель описывает, что происходит в системе, как ею управляют, что она преобразует, какие средства использует для выполнения своих функций и что производит. Модель разрабатывают для понимания, анализа и принятия решений при проектировании новой системы или реконструкции и замене существующей.

С точки зрения SADT модель может быть сосредоточена либо на функциях системы, либо на ее объектах. SADT-модели, ориентированные на функции, принято называть функциональными моделями, а ориентированные на объекты системы – моделями данных. *Функциональная модель* представляет набор функций системы и их взаимодействие через объекты системы. *Модели данных* дуальны по отношению к функциональным моделям и представляют собой набор объектов системы, связанных функциями системы. Полная методология SADT поддерживает создание множества моделей для более точного описания сложной системы.

SADT-модель является описанием системы, у которого существует единственная *цель* (purpose) и одна *точка зрения* (viewpoint). Целью служит набор вопросов, на которые должна ответить модель. Точка зрения – позиция, с которой описывается система. Цель и точка зрения помогают руководить разработкой модели и вести этот процесс в определенных рамках. Рассмотрим эти понятия подробнее.

SADT-модель дает точное и полное описание системы, имеющее конкретное назначение. Это назначение, называемое *целью* модели, вытекает из формального определения модели в SADT: *М есть модель системы S, если М может быть использована для получения ответов на вопросы относительно S с точностью А*. Таким образом, *цель* модели – получение ответов на определенную совокупность вопросов относительно моделируемой системы с заданной степенью точности. Если модель отвечает не на все вопросы или ее ответы недостаточно точны, то считается, что модель не достигла своей цели.

Вопросы для SADT-модели формулируются на самом раннем этапе проектирования. Цель модели – основная суть этих вопросов, выраженная в одной-двух фразах. На рис. 3.1 показан пример определения цели модели экспериментального механического цеха (далее ЭМЦ). Вверху слева приведен список вопросов, ответы на которые

должна содержать модель. Вверху справа список вопросов сведен в одно предложение. Это предложение становится целью модели, а список вопросов сохраняется как детализация этого предложения.

Используется в:	Автор: Магса	Дата: 20/12/99	х	Рабочая версия	Читатель	Дата	Контекст: Тор
	Проект: Экспериментальный механический цех	Пересмотр:		Проект			
	Замечания: 1 2 3 4 5 6 7 8 9 10			Рекомендовано			
				Публикация			
ВОПРОСЫ: Каковы обязанности мастера? Каковы обязанности механика? Кто контролирует задания? Как продвигаются по цеху материалы? На каких этапах требуется чертеж? В какой момент на процесс влияют стандарты качества? На каких этапах требуются инструменты? Что происходит с забракованными изделиями?			ЦЕЛЬ: Определить обязанности каждого работника экспериментального механического цеха и показать, как эти обязанности взаимосвязаны между собой для того, чтобы составить описание должностных обязанностей сотрудников цеха.				
ПРЕТЕНДЕНТЫ: Мастер Механик Контролер Начальник цеха			ТОЧКА ЗРЕНИЯ: Начальника цеха				
Узел: А-0	Название: Цель и точка зрения модели экспериментального механического цеха					Номер: DAM002	

Рис. 3.1. Пример определения цели и точки зрения модели

Поскольку качество описания системы снижается, если оно не сфокусировано на чем-нибудь конкретном, SADT требует, чтобы модель рассматривалась все время с одной и той же позиции. Эта позиция называется *точкой зрения* модели. С фиксированной точки зрения можно создать описание системы, которое не содержит ничего лишнего, и в котором не смешиваются не согласованные понятия.

Точка зрения – указание на должностное лицо или подразделение организации, с позиции которого разрабатывается модель. Для каждой модели должна существовать единственная точка зрения. Обычно наименованием точки зрения является название должности или подразделения.

Точка зрения непосредственно связана с конкретной ролью, выполняемой человеком или подразделением, которые являются частью системы. На рис. 3.1 внизу слева перечислены претенденты (мастер, механик, контролер, начальник цеха), с точки зрения которых можно описать ЭМЦ.

Выбор единой точки зрения означает выделение определенных аспектов системы и применение определенной терминологии. Без

правильно расставленных акцентов и терминологии согласованное описание системы невозможно. Остановимся на этом подробнее.

Во-первых, точка зрения выделяет одни аспекты системы и игнорирует другие. Изменение точки зрения приводит к рассмотрению других аспектов объекта. Аспекты, важные с одной точки зрения, могут не появиться в модели, разрабатываемой с другой точки зрения на тот же самый объект. При этом точка зрения не ограничивает предмет рассмотрения, но заставляет аналитика учитывать приоритеты различных аспектов системы. Например, если для создания модели ЭМЦ принять точку зрения либо мастера, либо механика, либо контролера, то ни одна из них не даст модели, которая позволила бы описать должностные инструкции всех сотрудников цеха. Только с позиции начальника цеха можно увидеть все виды работ, выполняемых в цехе, и проследить взаимосвязи обязанностей различных работников. Точка зрения начальника цеха позволяет создателю модели определить роль каждого работника в изготовлении отдельных деталей и описать координацию обязанностей персонала.

Во-вторых, выбор определенной точки зрения означает использование соответствующего набора терминов. Например, с точки зрения контролера употребляются термины *стандарты качества, принятое/забракованное задание*; с позиции механика используются термины *сырье/заготовки, станки, инструменты*.

Четкое фиксирование точки зрения позволяет разгрузить модель, отказавшись от детализации и исследования отдельных элементов, не являющихся необходимыми, исходя из выбранной точки зрения на систему. Одна из сложнейших задач аналитика в SADT – оставаться в рамках выбранной точки зрения.

Может оказаться необходимым построение моделей с разных точек зрения для детального отражения всех особенностей моделируемой системы. Тогда в прикрепленных диаграммах кратко документируют и другие точки зрения.

3.1.1.4. Принцип декомпозиции. Иерархия диаграмм

Одним из основных понятий методологии SADT является *декомпозиция (decomposition)*. Принцип декомпозиции применяется при разбиении сложного процесса на составные части. Декомпозиция позволяет постепенно представлять модель системы в виде иерархической структуры взаимосвязанных диаграмм.

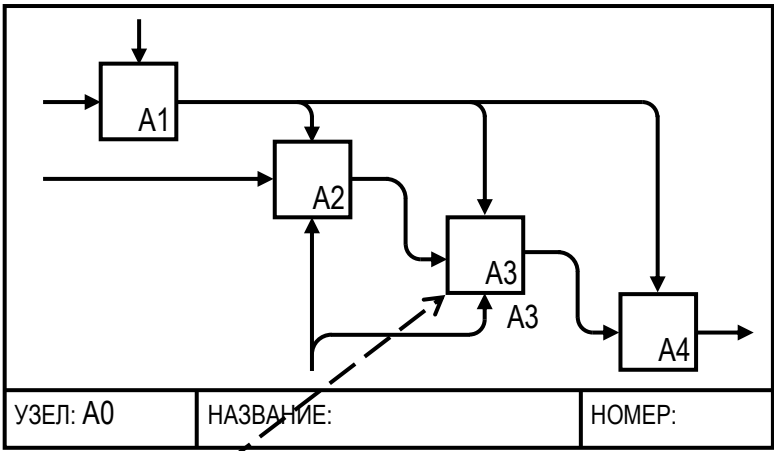
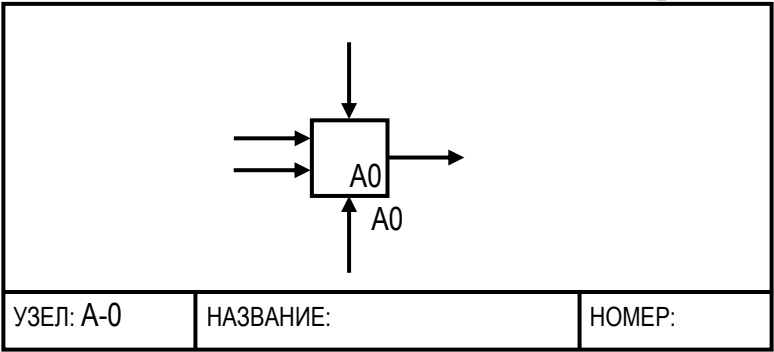
Построение модели системы начинается с описания общей функции, выполняемой системой. Эта функция представляется в виде

единственного блока с граничными дугами на контекстной диаграмме самого верхнего уровня (рис. 3.2). Блок определяет границу системы: все, что внутри него, является частью моделируемой системы, а все, что лежит вне него, образует внешнюю среду системы. Граничные дуги отображают связи системы с внешней средой. На рис. 3.31 показан верхний уровень типовой модели промышленного предприятия. Блок с названием *выпускать и реализовывать заданную номенклатуру изделий, оказывать услуги по их обслуживанию и ремонту с получением максимально возможной прибыли* описывает самую общую функцию промышленного предприятия. Дуга управления (*нормативные акты, регламентирующие производство, международные и внутренние стандарты*) определяет, как промышленное предприятие преобразует объекты, заданные четырьмя входными дугами (*информация о потребности в продукции предприятия, заказы; средства от заказчика, кредиты, капиталовложения; изделия на обслуживание и в ремонт; сырье, материалы, комплектующие изделия*), в объекты, заданные четырьмя выходными дугами (*готовая продукция; взаимодействие с внешними организациями; отходы производства; возвращенные кредиты*). Граничные дуги на контекстной диаграмме определяют интерфейсы между промышленным предприятием и внешней средой.

Общая функция системы, представленная на контекстной диаграмме верхнего уровня, разбивается на подфункции, которые представляются отдельными блоками на диаграмме второго уровня. В свою очередь, каждая из подфункций может быть разделена на составные части и представлена на диаграмме более низкого уровня. Процесс декомпозиции продолжается до тех пор, пока моделируемая система не будет описана на уровне детализации, необходимой для достижения целей конкретного проекта. Другими словами, дальнейшая декомпозиция не требуется, если модель достаточно точна, чтобы отвечать на все вопросы, соответствующие ее цели. Например, диаграмма, представленная на рис. 3.32, детализирует общую функцию системы, изображенную на рис. 3.31, указывая на шесть главных функций промышленного предприятия: *управлять предприятием; проводить маркетинговые исследования; проектировать, модернизировать изделия, готовить производство; выпускать и поставлять изделия заказчикам; обслуживать и ремонтировать изделия, анализировать рекламации; формировать структуру, готовить кадры, приобретать оборудование*.

Блок, для которого создается более детальная диаграмма, называется *родительским блоком* (parent box). Диаграмма, содержащая родительский блок, называется *родительской диаграммой* (parent

diagram). Диаграмма, детализирующая родительский блок, является *дочерней диаграммой* (child diagram) по отношению к этому блоку. Дочерняя диаграмма, создаваемая при декомпозиции, охватывает ту же область, что и родительский блок, но описывает ее более подробно.

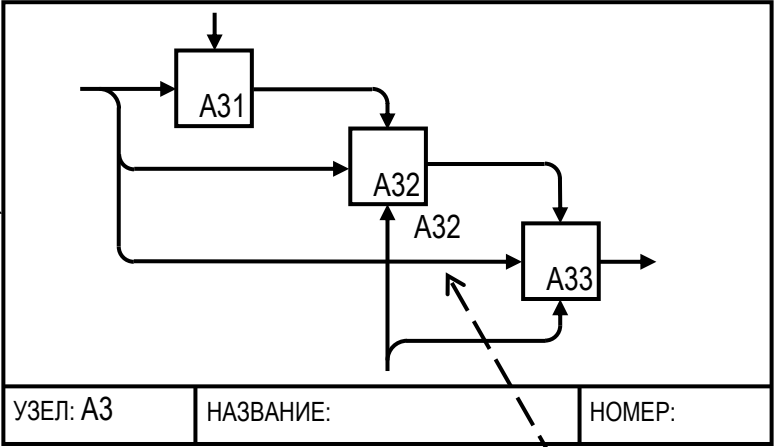


более общее
представление



более детальное
представление

Блок A3 родительский
для этой диаграммы



Ссылка на дочернюю
диаграмму под правым
нижним углом блока
показывает, что этот
блок декомпозирован.

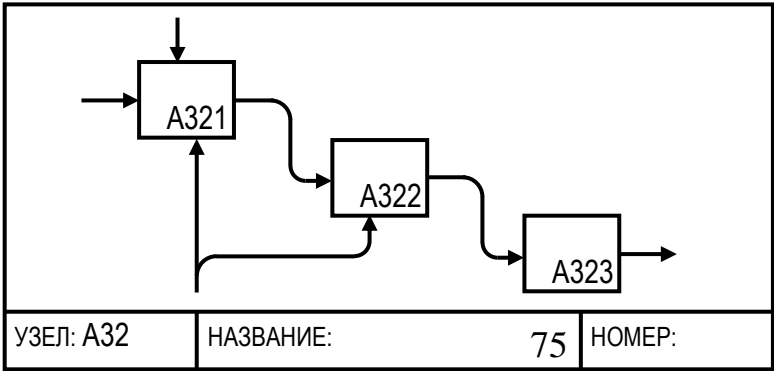


Рис. 3.2. Декомпозиция диаграмм

Блоки, принадлежащие дочерней диаграмме, называются *дочерними блоками* (child box).

Любая диаграмма (за исключением диаграммы самого верхнего уровня) является, по определению, дочерней диаграммой, т.к. она подробно описывает некоторый родительский блок. Таким образом, любая диаграмма может быть как родительской диаграммой (содержать родительские блоки), так и дочерней (подробно описывать определенный родительский блок). Аналогично, блок может быть как родительским (подробно описываться дочерней диаграммой), так и дочерним (принадлежать дочерней диаграмме). Основное иерархическое отношение существует между родительским блоком и дочерней диаграммой, которая его подробно описывает (рис. 3.2).

3.1.2. Синтаксические и семантические правила отображения SADT-моделей

Отличительная особенность SADT как методологии описания систем заключается в том, что она, используя в качестве основы естественный язык экспертов, структурирует этот язык с помощью своих графических средств. Естественный язык позволяет эксперту свободно описывать функционирование системы, пользуясь знакомой и удобной терминологией. В процессе создания диаграмм аналитик снабжает слова эксперта специальной пунктуацией в соответствии с графическим языком SADT. Графические обозначения SADT, также как и пунктуация, обладают высокой степенью точности. Графический язык SADT позволяет устранить из описаний, выполненных экспертом на естественном языке, неоднозначность, повторения и противоречия.

SADT-модели состоят из документов трех типов: графических диаграмм, текста и глоссария. Графическая диаграмма – главный компонент SADT-модели, содержащий блоки (activity box) и дуги (arrow).

Набор структурных компонентов языка, их характеристики и правила, определяющие связи между компонентами, представляют собой *синтаксис* языка. Компоненты синтаксиса SADT – блоки, дуги, диаграммы и правила. Блоки представляют функции моделируемой системы. Дуги представляют материальные или информационные

объекты, связанные с функциями. Правила определяют, как следует применять компоненты. Диаграммы обеспечивают формат графического и словесного описания моделей. Формат образует основу для управления конфигурацией модели.

Семантика определяет содержание (значение) синтаксических компонентов языка и способствует правильности их интерпретации. Интерпретация устанавливает соответствие между блоками и дугами с одной стороны, функциями и их интерфейсами – с другой.

Синтаксис SADT-моделей позволяет аналитику определить границу модели, связать диаграммы в одно целое и обеспечить точное согласование между диаграммами.

3.1.3. Синтаксические и семантические правила SADT, установленные для блоков

Блоки представляют основные функции моделируемой системы. Функциональные блоки изображаются на диаграмме прямоугольниками. Внутри каждого блока помещаются его имя и номер (рис. 3.3).

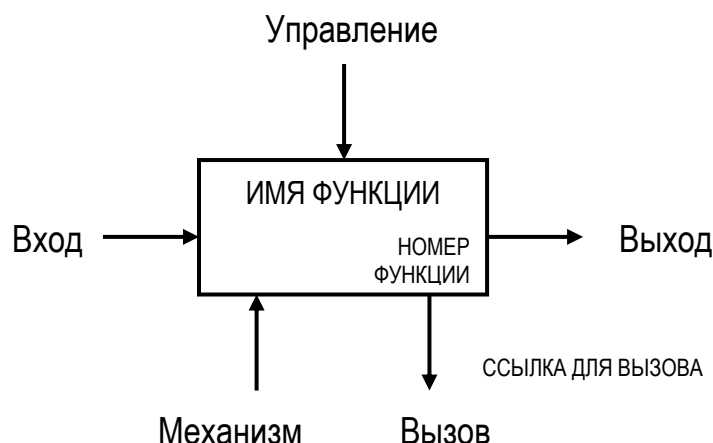


Рис. 3.3. Назначение сторон блока

Имя блока должно быть глаголом или глагольным оборотом, описывающим функцию. Например, блоки диаграммы, изображенной на рис. В2, имеют названия: *управлять предприятием; проводить маркетинговые исследования; проектировать, модернизировать изделия, готовить производство; выпускать и поставлять изделия заказчикам* и др. Имена блоков (функций) должны быть уникальными.

Номер блока размещается в правом нижнем углу. Стандартное расположение номеров позволяет их быстро находить. Номера блоков используются для их однозначной идентификации на диаграмме и для

ссылок на них в соответствующем тексте и глоссарии.

Для блоков установлены следующие синтаксические правила:

1) размеры блоков должны быть достаточными для того, чтобы включить имя и номер блока;

2) блоки изображаются прямоугольниками;

3) блоки должны быть нарисованы сплошными линиями.

Каждая сторона функционального блока имеет стандартное назначение с точки зрения связи блок/дуга (рис. 3.3):

1) с верхней стороной блока связываются управляющие дуги;

2) с левой стороной блока связываются входные дуги;

3) с правой стороной блока связываются выходные дуги;

4) к нижней стороне блока присоединяются и направляются вверх дуги механизмов;

5) к нижней стороне блока присоединяются и направляются вниз дуги вызова, они помечаются ссылкой на вызываемый блок.

Такое назначение отражает определенные системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают: кто, что и как выполняет функцию. Пример функционального блока приведен на рис. 3.4.

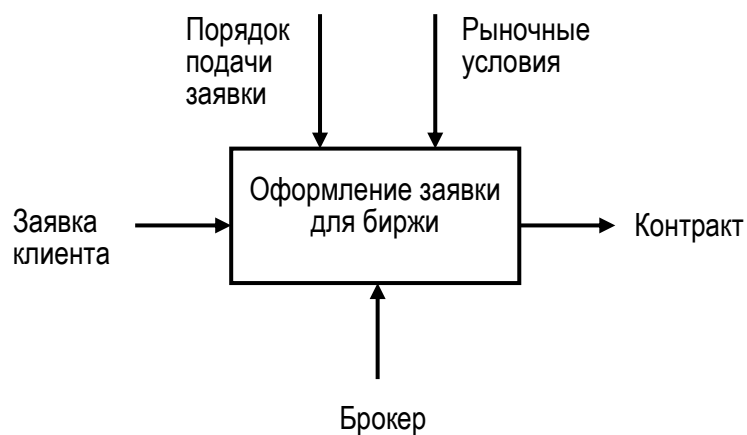


Рис. 3.4. Пример функционального блока

При описании системы в текстовой форме используется стандартная интерпретация графических обозначений SADT. Формальное описание «Блок В, связан с входной дугой I, дугой управления С, выходной дугой О и дугой механизма М» преобразуется в обычное предложение вида: «Функция В преобразует I в О при ограничениях, заданных С, с помощью М». Блок 4 диаграммы, изображенной на рис. 3.5, может быть интерпретирован следующим образом: *детали, сырье и брак обрабатываются на станке и собираются в результа-*


ты обработки с использованием оборудованного рабочего места. Другой пример – блок A12 диаграммы, изображенной на рис. 3.33, может быть интерпретирован следующим образом: при ведении стратегического планирования, а также определении и корректировке целей предприятия служба стратегического планирования на

основе документированных результатов анализа работы предприятия и в соответствии с положениями международного стандарта *ISO 9000* и других стандартов разрабатывает документированные цели и планы стратегического развития предприятия.

У каждого блока должны присутствовать хотя бы одна дуга управления и хотя бы одна выходная дуга, остальные виды дуг могут отсутствовать.

3.1.4. Синтаксические и семантические правила SADT, установленные для дуг

Дуги идентифицируют материальные или информационные объекты, необходимые для выполнения функции или производимые ею. Дуга формируется из одного или нескольких отрезков прямых.

Дуга снабжается *меткой* (arrow label), описывающей материальные или информационные объекты, передаваемые этой дугой. Метка должна быть существительным или оборотом существительного. В метках *не должны* использоваться следующие термины: функция, вход, управление, выход, механизм, вызов. Дуга связывается с меткой посредством ломаной (молниеобразной) выносной, называемой тильда ().

Для дуг установлены следующие синтаксические правила:

- 1) дуги изображаются сплошными линиями, можно использовать линии различной толщины;
- 2) ломаные дуги изменяют направление только под углом 90°;
- 3) дуги могут состоять только из вертикальных или горизонтальных отрезков; не допускается использовать отрезки, направленные по диагонали;
- 4) концы дуг должны касаться внешней границы функционального блока, но не должны пересекать ее;
- 5) дуги должны присоединяться к блоку на его сторонах, присоединение в углах не допускается.

Как уже отмечалось, сторона блока, к которой присоединена дуга, однозначно определяет роль дуги.

Дуги, входящие в блок сверху, – *управления* (control). Управления определяют условия, необходимые функции, чтобы произвести

правильный выход. Обычно дуги управления представляют ограничительную или/и предписывающую информацию, которая управляет действиями функции (планы, правила, инструкции, законы, стандарты и т.п.). Например, на диаграмме, изображенной на рис. 3.37, показано, что *приобретение нового оборудования и замена устаревшего* (блок А63) производятся в соответствии с *программой и графиком обновления оборудования*.

Управления отображают материальные или информационные объекты, которые необходимы для выполнения функции, но не преобразуются ей. Если целью функции является изменение планов, правил, инструкций, законов, стандартов и др., то они должны отображаться входной дугой. У каждого блока должна присутствовать хотя бы одна дуга управления.

Дуги, входящие в левую сторону блока, – *вход(ы)* (input). Входные дуги отображают материальные или информационные объекты системы, которые преобразуются или расходуются функцией, чтобы создать то, что появится на ее выходе. Например, на диаграмме, изображенной на рис. 3.37, показано, что в процессе *создания интегрированной информационной среды* (далее ИИС) *средства на автоматизацию* трансформируются в *ИИС* и *данные о состоянии ИИС* (блок А65). Поскольку могут существовать функции, ничего не преобразующие, то входные дуги могут отсутствовать.

Дуги, покидающие блок справа, – *выход(ы)* (output). Выходы – это материальные или информационные объекты, произведенные функцией. У каждого блока должна присутствовать хотя бы одна выходная дуга.

Дуги, подключенные к нижней стороне блока, – *механизмы* (mechanism). Дуги механизмов представляют материальные или информационные объекты системы, с помощью которых осуществляется преобразование входов в выходы (люди, технические средства, информационные системы и др.). Сами механизмы не преобразуются системой. Дуги, направленные вверх, идентифицируют средства, поддерживающие выполнение функции. Дуги механизма, направленные вниз, являются дугами *вызова*. Дуги вызова обозначают обращение из определенной модели или ее части к блоку, входящему в состав другой модели или другой части модели, обеспечивая их связь. Таким образом, разные модели или разные части одной и той же модели могут совместно использовать один и тот же блок. Дуги механизма и вызова могут отсутствовать. Примеры механизмов: *оборудованное рабочее место* (рис. 3.5, блок 4), *служба стратегического планирования* (рис. 3.33, блок А12), *эксперт-рецензент* (рис. 3.41, блок А14).

Рассмотрим пример функции *выточить деталь* (рис. 3.6). Входом функции является заготовка, из которой вытачивается деталь, она физически преобразуется при выполнении функции. План и чертеж с размерами детали относятся к управлению, они необходимы рабочему для вытачивания правильной детали, но не преобразуются функцией.

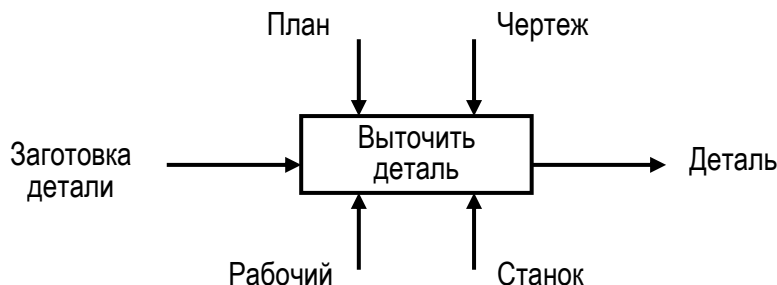


Рис. 3.6.

Рабочий и станок относят к механизмам, рабочий вытачивает деталь, используя станок. Выходом или результатом процесса является деталь.

3.1.5. Синтаксические и семантические правила SADT, установленные для диаграмм

SADT-модели несут в себе сложную и концентрированную информацию. Чтобы ограничить их перегруженность и сделать более читаемыми, приняты соответствующие правила.

Диаграммы SADT-модели организованы в иерархическую структуру. Диаграмма верхнего уровня иерархии обеспечивает общее описание моделируемой системы, другие диаграммы дают более детальное представление о системе.

Каждая модель должна иметь *контекстную диаграмму* верхнего уровня, на которой объект моделирования представлен единственным блоком с граничными дугами. Блок описывает общую функцию, выполняемую системой. Дуги отображают связи объекта моделирования с окружающей средой. Единственный блок контекстной диаграммы верхнего уровня разбивается на составные части и представляется тремя – шестью блоками на диаграмме второго уровня. Аналогичным образом каждый из блоков диаграммы второго уровня может быть детализирован на диаграмме более низкого уровня. Декомпозиция прекращается, когда диаграммы, образующие нижний уровень модели, достаточно детализированы для достижения цели модели. Таким образом, формируется иерархическая структура диаграмм.

На контекстной диаграмме верхнего уровня должны быть ука-

заны *цель* и *точка зрения* модели. На рис. 3.31 показано, что система описывается с позиции *разработчика проекта* (точка зрения) и с целью – *описать бизнес процессы предприятия; показать формирование механизмов (технологического оборудования, кадрового состава, организационно-технической структуры); выявить структуру и функции системы управления*. Иногда в SADT-модели создаются дополнительные контекстные диаграммы, имеющие разные цели и точки зрения.

3.1.5.1. Правила изображения блоков на диаграммах

Правила SADT по размещению блоков на диаграммах способствуют созданию более читаемых диаграмм.

1. Диаграммы (за исключением диаграммы самого верхнего уровня) должны содержать не менее трех и не более шести блоков. Эти ограничения поддерживают сложность диаграмм на уровне, доступном для чтения, понимания и использования.

Диаграммы с количеством блоков менее трех вызывают серьезные сомнения в необходимости декомпозиции родительской функции. Диаграммы с количеством блоков более шести сложны для восприятия читателями и вызывают у автора трудности при внесении в нее всех необходимых графических объектов и меток. Кроме того, диаграмма ограничивается тремя – шестью блоками для того, чтобы детализация осуществлялась постепенно. Вместо одной громоздкой модели используется несколько небольших взаимосвязанных моделей, значения которых взаимно дополняют друг друга, делая понятной структуризацию сложного объекта. Марка Д.А. отмечает: «Мощность краткосрочной памяти человека ограничена восприятием примерно семи категорий, каждая из которых может содержать около семи отдельных единиц информации. Таким образом, SADT-диаграммы создаются так, чтобы не подвергать испытанию пределы краткосрочной памяти человека. Опыт показывает, что диаграммы из четырех – пяти блоков с не более чем пятью дугами, касающимися каждого блока, приближаются к оптимальным по объему информации, которые можно быстро донести до широкой аудитории» [7].

2. Блоки на диаграмме должны располагаться по диагонали – от левого верхнего до правого нижнего угла диаграммы. Диагональное расположение блоков способствует читаемости диаграммы: показывается доминирование блоков, минимизируется число поворотов и пересечений дуг, упрощается изображение обратных связей.

3. На контекстной диаграмме A-0 единственному блоку присва-

ивается номер 0 (ноль). На других диаграммах блоки нумеруются от 1 до 6, начиная с верхнего левого и заканчивая нижним правым блоком (при диагональном размещении блоков). Если некоторые блоки на диаграмме размещены не по диагонали, то сначала нумеруются «диагональные» блоки (также начиная с левого верхнего блока), а затем – «недиагональные» блоки, начиная с нижнего правого угла против часовой стрелки (рис. 3.5).

4. Каждый блок, подвергнутый декомпозиции, должен иметь ссылку на дочернюю диаграмму. Ссылка помещается под правым нижним углом блока. Ссылочный код может формироваться несколькими способами – это может быть узловой номер диаграммы, С-номер диаграммы или номер страницы. Например, на рис. 3.32 показано, что декомпозиция блока А3 выполнена на диаграмме А3 (рис. 3.35).

В методологии SADT описывается шесть типов связей между функциями в пределах одной диаграммы:

- 1) отношение доминирования;
- 2) управление;
- 3) выход–вход;
- 4) обратная связь по управлению;
- 5) обратная связь по входу;
- 6) выход–механизм.

1. *Доминирование* определяется взаимным расположением функций на диаграмме. Функции, расположенные на диаграмме выше и левее, доминируют над функциями, расположенными ниже и правее. Под доминированием понимается влияние, которое одна функция оказывает на другие функции диаграммы. Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные. При параллельном выполнении функций отношение доминирования часто носит формальный характер, и диагональное расположение блоков лишь способствует читаемости диаграммы. Расположение блоков на листе диаграммы отражает авторское понимание доминирования.

2. *Связь управления* возникает, когда выход одной функции служит управляющим воздействием на функцию с меньшим доминированием (рис. 3.7, а).

3. *Связь выход–вход* возникает, когда выход одной функции служит входом другой функции с меньшим доминированием (рис. 3.7, б).

4. *Обратная связь по управлению* возникает, когда выход некоторой функции создает управляющее воздействие на функцию с большим доминированием. Обратная связь по управлению представ-

ляет итерацию (выход функции влияет на будущее выполнение других функций с большим доминированием, что впоследствии влияет на исходную функцию). Например, корректировка графика выполнения некоторого процесса в зависимости от результатов его выполнения. Обратная связь по управлению изображается «вверх и над» (рис. 3.7, в). Такое расположение позволяет показать обратные связи по управлению при минимальном числе дуг и пересечений, а также разместить все дуги управления в верхней правой части диаграммы.

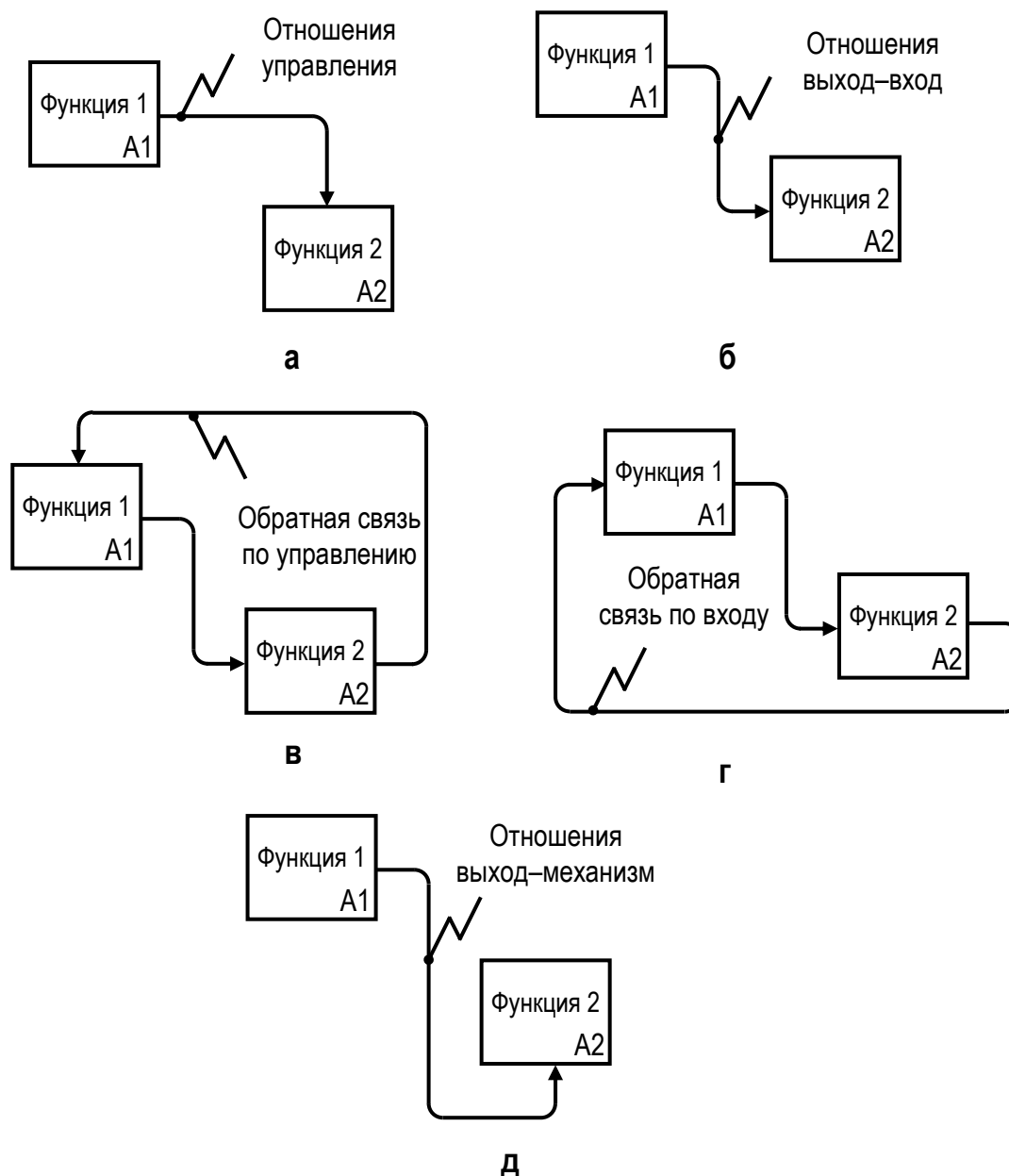


Рис. 3.7. Основные типы связей между функциями

5. *Обратная связь по входу* имеет место, когда выход одной

функции становится входом другой функции с большим доминированием. Обратная связь по входу указывает на циклическую обработку чего-либо (например, периодический сбор сведений, повторная обработка чего-либо с целью достижения определенного уровня качества, повторное использование бракованных деталей в качестве сырья и др.). Обратная связь по входу изображается «вниз и под» (рис. 3.7, г). Такое расположение позволяет показать обратные связи по входам при минимальном числе дуг и пересечений, а также разместить все входные дуги в нижней левой части диаграммы.

б. *Связь выход–механизм* отражают ситуацию, при которой выход одной функции становится средством достижения цели для другой. Связи выход–механизм возникают при отображении в модели процедур пополнения и распределения ресурсов, создания или подготовки средств для выполнения функций системы (например, приобретение или изготовление требуемых инструментов и оборудования, обучение персонала, организация физического пространства, финансирование, закупка материалов и т.д.). Связь выход–механизм изображается «вниз и под» (рис. 3.7, д).

Принятые правила изображения связей функций обеспечивают минимальное число дуг и их пересечений.

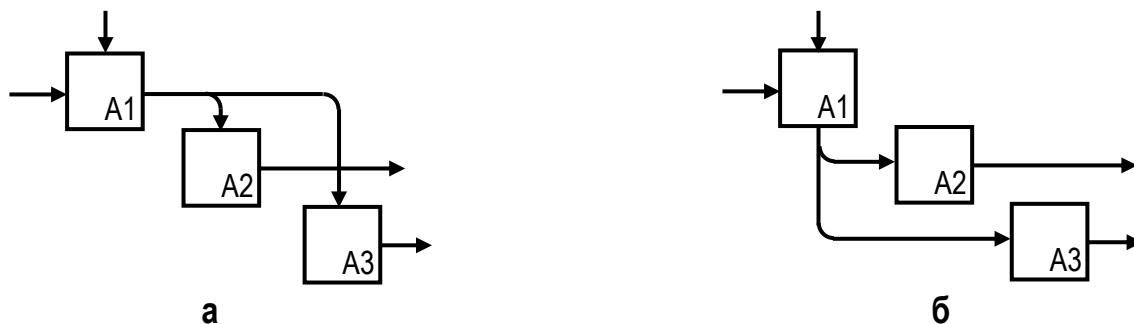


Рис. 3.8. Параллельное выполнение функций

Некоторые функции в модели могут быть выполнены параллельно, если удовлетворяются необходимые ограничения (условия). Как показано на рис. 3.8, один блок может создать управляющую информацию (а) или информационные и материальные объекты (б), необходимые для параллельной работы нескольких блоков.

3.1.5.2. Правила изображения дуг на диаграммах

Дуги, размещаемые на SADT-диаграммах, подразделяются на внутренние и граничные (рис. 3.9). *Внутренняя дуга* – входная, управ-

ляющая или выходная дуга, а также дуга механизма, которые связывают блоки одной диаграммы. *Граничная дуга* – дуга, один из концов которой связан с блоком, а другой не присоединен ни к одному из блоков диаграммы. Граничные дуги отображают связь диаграммы с блоками других диаграмм системы.

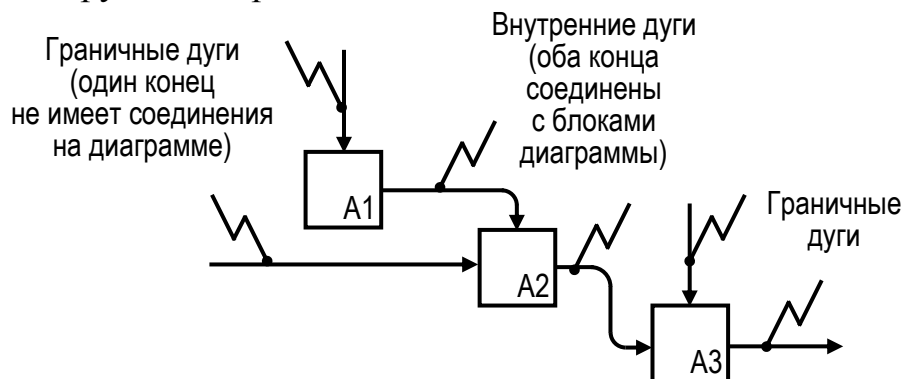


Рис. 3.9. Внутренние и граничные дуги SADT-диаграмм

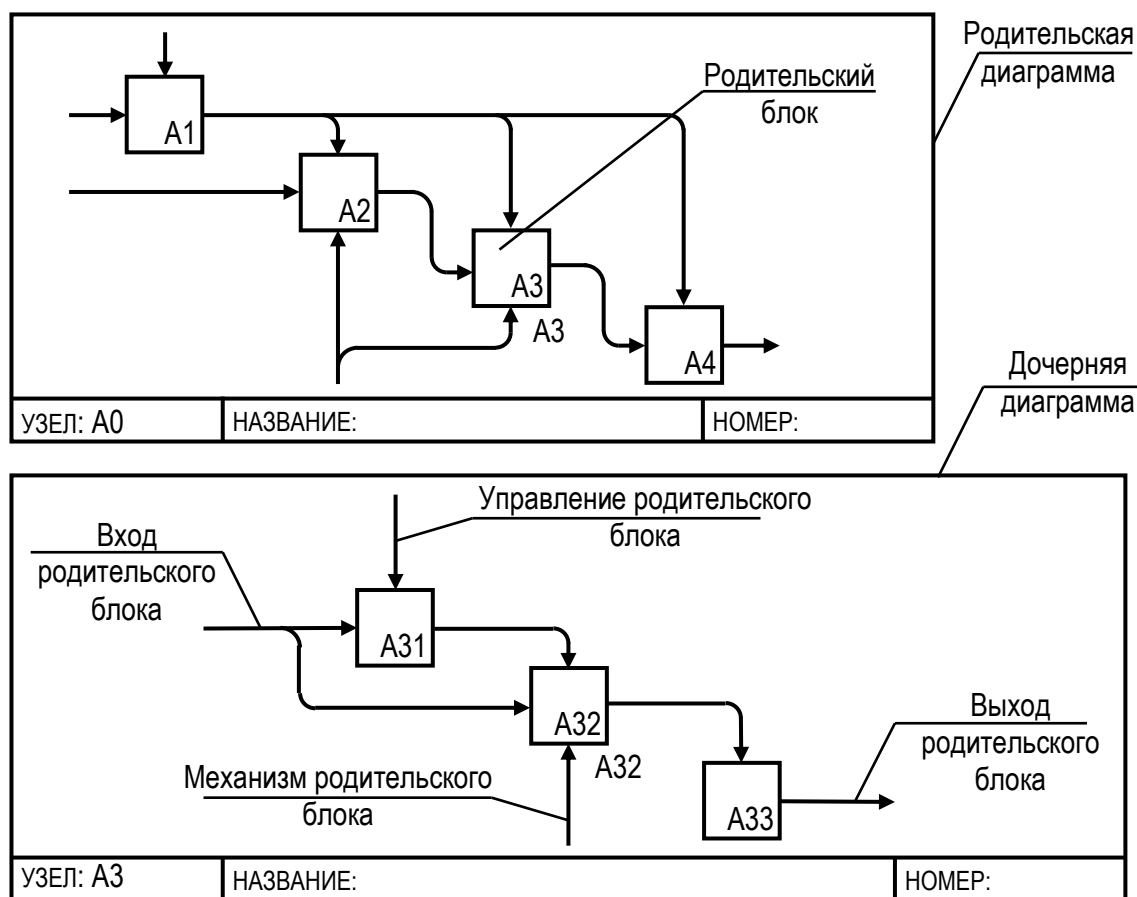


Рис. 3.10. Соответствие дуг родительского блока и дочерней диаграммы

Дуги родительского блока должны быть согласованы с гранич-

ными дугами дочерней диаграммы, детализирующей этот блок (за исключением дуг, помещенных в туннель), т.к. блок и диаграмма представляют одну и ту же часть системы (рис. 3.10). Это означает, что граничные дуги дочерней диаграммы совпадают по числу и наименованию (но не обязательно по расположению) с дугами родительского блока. Например, блок А6 (рис. 3.32) имеет семь дуг: одну входную дугу (*ресурсы на формирование оргструктуры, кадры и оборудование*); две дуги управления (*меры по обеспечению качества продукции; планы реинжиниринга, подготовки кадров, обновления оборудования*); две выходные дуги (*данные о подготовленных кадрах, новом оборудовании; ОТС, кадры*); две дуги механизмов (*внешние участники проекта; руководство предприятия*). Все дуги родительского блока А6 (рис. 3.32) совпадают с граничными дугами дочерней диаграммы (рис. 3.37).

Следующие простые правила позволяют уменьшить сложность диаграмм и облегчить их чтение.

1. Следует максимально увеличить расстояние между параллельными дугами. Это упрощает размещение и чтение меток, а также позволяет легче отслеживать пути дуг (рис. 3.11).

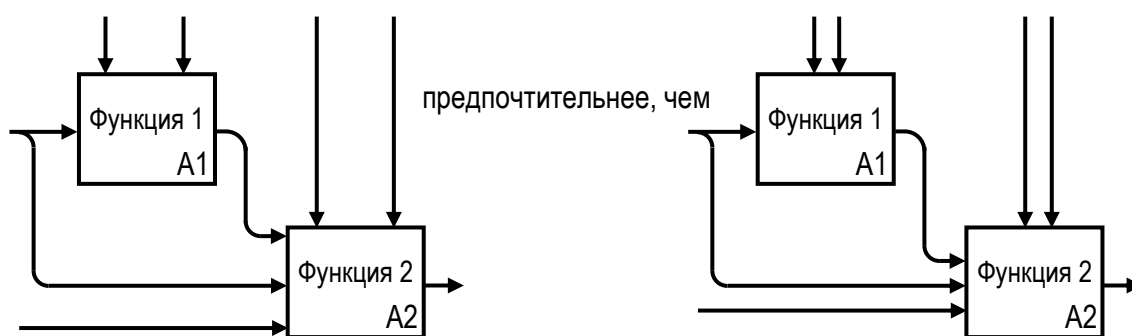


Рис. 3.11. Максимальное расстояние между дугами

2. Следует обеспечить максимальное расстояние между блоками и поворотами дуг, а также между блоками и пересечениями дуг. Это уменьшит вероятность того, что две разные дуги будут перепутаны.

3. При соединении большого числа блоков необходимо избегать необязательных пересечений дуг (рис. 3.12).

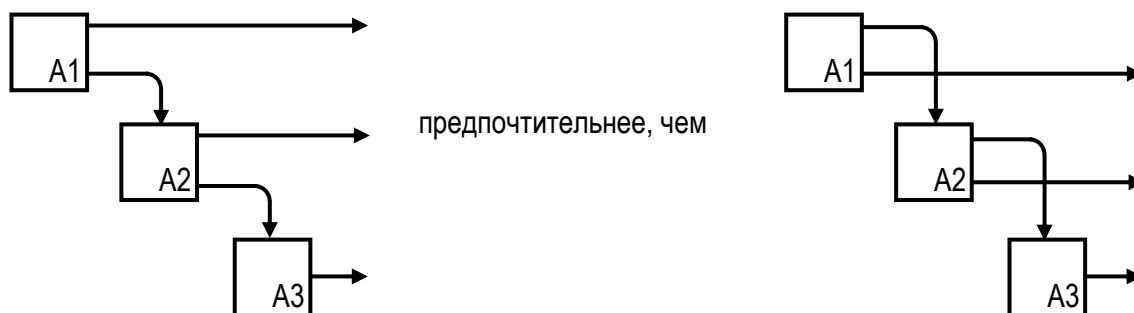


Рис. 3.12. Отсутствие необязательных пересечений дуг

4. Следует минимизировать число петель и поворотов каждой дуги (рис. 3.13).



Рис. 3.13. Минимальное число петель и поворотов дуг

5. По возможности следует присоединять дуги к блокам в одной и той же ICOM позиции. Тогда соединения дуг конкретного типа с блоками будут согласованными, что упростит чтение диаграммы (рис. 3.14).



Рис. 3.14. Присоединение дуг в одной и той же ICOM-позиции

6. Если дуги представляют связанные данные и имеют общий источник или общий приемник, их следует объединить (рис. 3.15).

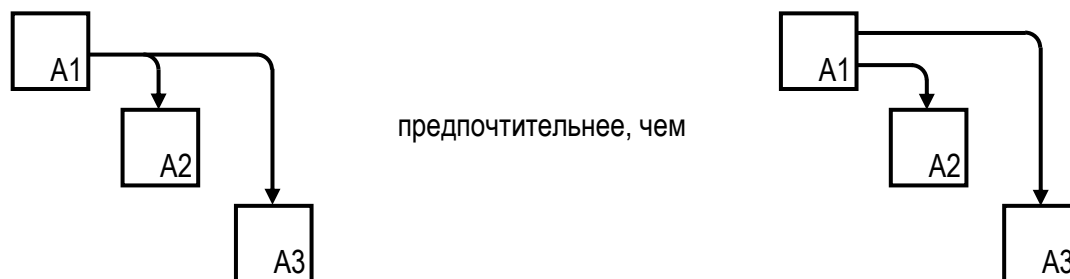


Рис. 3.15. Объединение дуг с общим источником

7. Следует минимизировать число дуг, касающихся каждой стороны блока, если, конечно, природа данных не слишком разнородна (рис. 3.15).

8. Следует объединить дуги, если они представляют сходные данные, и их источник не указан на диаграмме (рис. 3.16). Так графически отображается единый источник сходных данных.

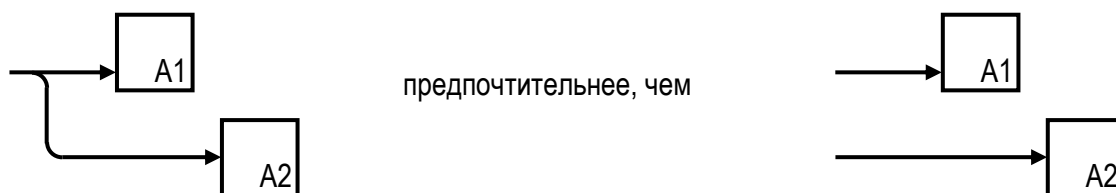


Рис. 3.16. Объединение дуг со сходными данными

9. Если одни и те же данные служат и для управления, и для входа, вычерчивается только дуга управления. Этим подчеркивается управляющий характер данных и уменьшается сложность диаграммы

10. В некоторых случаях требуется выделить циклические обратные связи для одного блока. Обычно обратную связь изображают на диаграмме, декомпозирующей блок, однако иногда требуется выделить повторно используемые объекты на текущей диаграмме (рис. 3.17).



Рис. 3.17. Циклические обратные связи

11. Метки дуг должны быть уникальными. Если метки дуг совпадают, это значит, что дуги отображают тождественные данные.

12. При наличии дуг со сложной топологией целесообразно повторить метку для удобства ее идентификации.

Туннельные дуги – дополнительное средство графического языка, облегчающее чтение и понимание диаграмм. *Туннель* – круглые скобки в начале или в конце дуги. Помещение дуг в туннель освобождает родительскую и дочернюю диаграммы от несущественных для их уровня дуг. Туннельные дуги используются в двух особых случаях связи между диаграммами.

Первый случай, когда объекты, представленные граничными дугами отсутствуют на родительской диаграмме, т.е. имеют скрытый источник. Туннельные дуги от скрытого источника начинаются скоб-

ками на свободном конце дуги (рис. 3.18, а). Это указывает на то, что такие дуги идут из какой-то другой части модели или прямо извне модели. Например, граничные дуги с метками *неподготовленный персонал* и *сведения о существующей оргструктуре* (рис. 3.37) начинаются скобками. Это указывает на то, что объекты, передаваемые этими дугами, отсутствуют на родительской диаграмме (рис. 3.32).

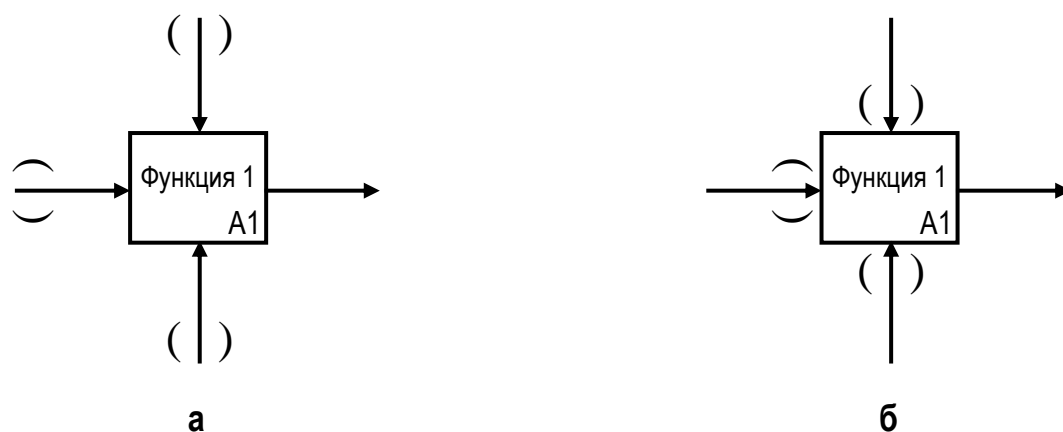


Рис. 3.18. Способы изображения туннельных дуг

Второй случай, когда объекты, выраженные граничной дугой, не будут показаны на следующем уровне декомпозиции, т.е. имеют скрытый приемник. Туннельные дуги, имеющие скрытый приемник, заканчиваются скобками там, где они присоединяются к блоку (рис. 3.18, б). Это отражает тот факт, что такая дуга идет к какой-то другой части модели или выходит из нее, или что она не будет более в этой модели рассматриваться.

Дуги могут разветвляться и соединяться. *Ветвление* – разделение дуги на несколько сегментов. *Слияние* – объединение двух или большего числа сегментов дуг в один сегмент. Дуги объединяются (сливаются), если они имеют общий источник или приемник, или они представляют связанные данные. Ветвление и слияние дуг призвано уменьшить загроуженность диаграмм графическими элементами (линиями).

С процессами ветвления и слияния дуг связаны процессы развязывания и связывания значений дуг. *Развязывание* – разделение значений дуг. *Связывание* – объединение значений дуг в составное значение. Чтобы правильно описать объекты, которые передают сегменты дуг при их ветвлении или слиянии, используются метки.

В SADT приняты определенные соглашения для размещения меток на сегментах дуг в случае их ветвления или слияния.

1. Непомеченные сегменты содержат все объекты, указанные в

метке дуги перед ветвлением, т.е. все объекты принадлежат каждому из сегментов (рис. 3.19). Например, функция *управлять предприятием* имеет выход *меры по обеспечению качества продукции*, который является управляющей информацией для всех других функций диаграммы (рис. 3.32). Метка *меры по обеспечению качества продукции* находится перед ветвлением дуги, поэтому непомеченные сегменты, выходящие из этой дуги, содержат все объекты, указанные данной меткой. Рассмотрим еще один пример. Функция *эксплуатировать ИИС* имеет выход *данные об изделии*, который является входом для трех других функций диаграммы (рис. 3.35). Метка *данные об изделии* также находится перед ветвлением дуги, следовательно, непомеченные сегменты, выходящие из этой дуги, содержат все объекты, указанные этой меткой.

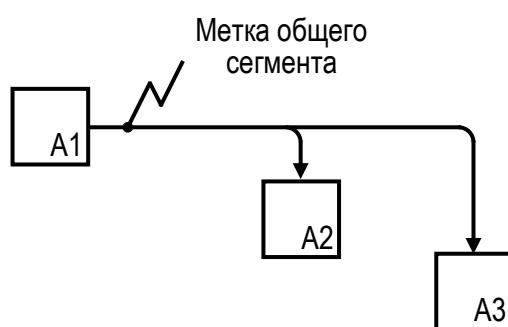


Рис. 3.19. Непомеченные сегменты содержат все объекты, указанные общей меткой

2. Сегменты, помеченные после точки ветвления, содержат все объекты, указанные в метке дуги перед ветвлением, или их часть, описываемую меткой каждого конкретного сегмента (рис. 3.20). Например, функция *управлять предприятием* имеет выход *распределенные ресурсы*, который является входом для всех других функций диаграммы (рис. 3.32). Четыре сегмента, выходящие из этой дуги, имеют собственные метки. Следовательно, они содержат только часть объектов, описанных меткой конкретного сегмента.

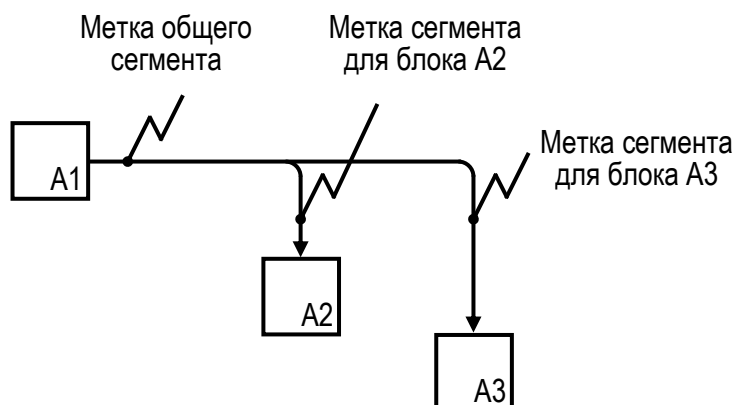


Рис. 3.20. Сегменты содержат только часть объектов, описанных меткой конкретного сегмента

3. При слиянии непомеченных сегментов объединенный сегмент дуги содержит все объекты, принадлежащие сливаемым сегментам и указанные в общей метке дуги после слияния (рис. 3.21). В качестве примера рассмотрим функцию *выпускать и поставлять изделия заказчикам* и функцию *обслуживать и ремонтировать изделия, анализировать рекламации* (рис. 3.32). Данные функции имеют выходные дуги, которые сливаются в общую дугу. Сегменты дуг до слияния не помечены, поэтому объединенный сегмент дуги, имеющий метку *отходы производства*, содержит все объекты, принадлежащие сливаемым сегментам двух функций.

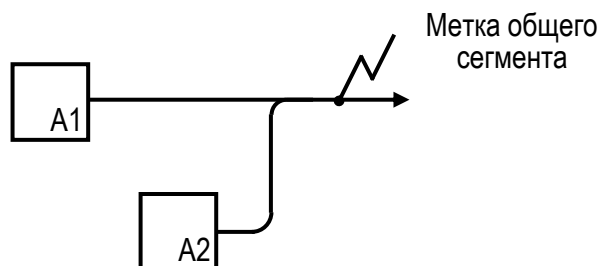


Рис. 3.21. Объединенный сегмент дуги содержит все объекты, принадлежащие непомеченным до слияния сегментам дуг

4. При слиянии помеченных сегментов объединенный сегмент содержит все или часть объектов, принадлежащие сливаемым сегментам и перечисленные в общей метке после слияния (рис. 3.22, а). Если общая метка после слияния отсутствует, это означает, что общий сегмент передает все объекты, принадлежащие сливаемым сегментам (рис. 3.22, б).

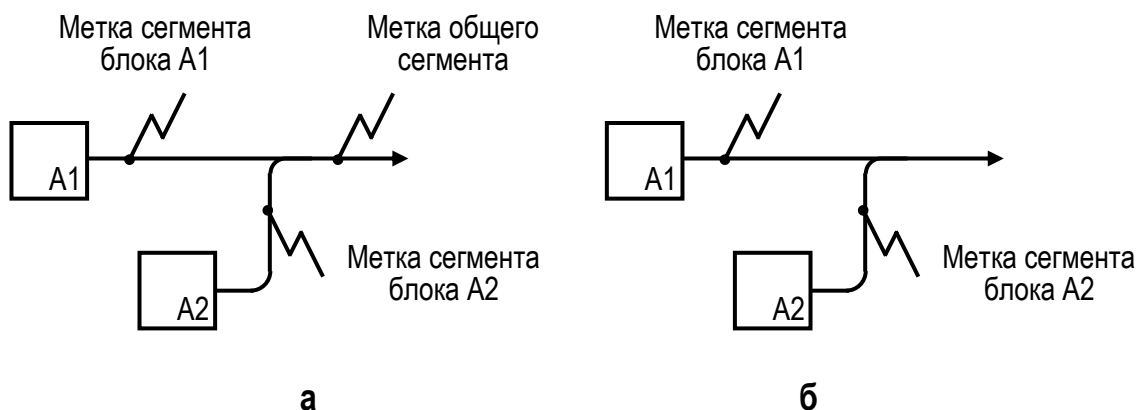


Рис. 3.22. Способы передачи объектов, принадлежащих сливаемым сегментам

В качестве примера рассмотрим функцию *выпускать и поставлять изделия заказчикам* и функцию *обслуживать и ремонтировать изделия, анализировать рекламации* (блоки A4 и A5, рис. 3.32). Данные функции имеют выходы, дуги которых сливаются в общую дугу. Сегменты дуг до слияния имеют соответственно метки *готовые изделия* и *изделия после обслуживания и ремонта*, объединенный сегмент дуги имеет метку *готовая продукция*. В этом случае объединенный сегмент содержит те объекты, которые принадлежат сливаемым сегментам и перечислены в общей метке после слияния. Рассмотрим еще один пример. Следующие функции, изображенные на рис. 3.35, имеют выходные дуги, которые сливаются в общую дугу:

1) *проектировать новые виды продукции, модернизировать выпускаемые изделия* имеет выход *сведения о ходе проектно-конструкторских работ*;

2) *проектировать технологию изготовления продукции, инструмент и технологическую оснастку* имеет выход *сведения о ходе технологического проектирования*;

3) *разрабатывать оргпроект, планы технического переоснащения, потребность в кадрах, планы запуска и оснащения продукции в производстве* имеет выход *сведения о ходе организационно-экономического проектирования*.

Сегменты дуг до слияния имеют соответствующие метки. Объединенный сегмент не помечен. В этом случае общий сегмент передает все объекты, принадлежащие сливаемым сегментам.

3.1.5.3. Узловые номера, ICOM-коды и С-номера

Для правильного соединения диаграмм и построения согласованной SADT-модели используются специальные ссылочные выражения: узловой номер, ICOM-код, С-номер. Ссылочные выражения присваиваются всем элементам модели: диаграммам, блокам, дугам и примечаниям. Ссылочные выражения используются в различных контекстах для точного указания на нужный элемент модели. Узловой номер однозначно идентифицирует диаграммы и блоки в иерархии SADT-модели. ICOM-коды используются для связи граничных дуг дочерней диаграммы с дугами родительского блока. С-номера используются для идентификации различных версий диаграмм. Помимо это-

го узловые номера и С-номера применяются для связывания родительских и дочерних диаграмм.

Номер узла (узловой номер) используется для однозначной идентификации диаграмм и блоков SADT-модели.

Все узловые номера SADT обычно начинаются с заглавной буквы А – первой буквы в слове activity (деятельность или функция системы). Узловой номер блока формируется добавлением его номера к номеру диаграммы, на которой он находится. Например, узловым номером блока 2 на диаграмме А25 будет А252.

Контекстная диаграмма верхнего уровня имеет узловой номер А-0 (А дефис ноль). Эта диаграмма содержит единственный блок, который имеет уникальный номер 0 (ноль) и узловой номер А0. Если в SADT-модели созданы дополнительные контекстные диаграммы, они будут иметь узловые номера А-1, А-2 и т.д.

Диаграмма, детализирующая контекстную диаграмму, имеет номер узла без дефиса – А0 (рис. 3.2). Блокам диаграммы А0 присваиваются узловые номера А1, А2, ..., А6. Номера узлов всех диаграмм, за исключением А-0, совпадают с узловыми номерами их родительских блоков (табл. 3.1). При декомпозиции блока узловой номер дочерней диаграммы, совпадающий с узловым номером самого блока, помещается под правым нижним углом блока (рис. 3.2).

Таблица 3.1

Примеры узловых номеров диаграмм и блоков SADT-модели

Узловой номер	Диаграмма и/или блок
А-0	Обязательная контекстная диаграмма верхнего уровня (содержащая блок А0)
А-1, А-2, ...	Дополнительные контекстные диаграммы
А0	Единственный блок на диаграмме А-0 и дочерняя диаграмма этого блока
А1, А2, ..., А6	Блоки на диаграмме А0 и их дочерние диаграммы
А11, А12, ..., А16	Блоки на диаграмме А1 и их дочерние диаграммы
А111, А112, ..., А116	Блоки на диаграмме А11 и их дочерние диаграммы

На рис. 3.2 показано, что диаграмма с номером узла А0 детализирует контекстную диаграмму А-0, диаграмма с номером узла А3 детализирует блок 3 диаграммы А0 (первый ноль при образовании номера узла принято опускать, поэтому вместо А03 пишется А3), диа-

грамма с номером узла A32 детализирует блок 2 диаграммы A3.

Для связи граничных дуг дочерней диаграммы с дугами родительского блока используются *ICOM*-коды. Нотация, названная *ICOM*-кодированием, получила название по первым буквам английских слов вход (Input), управление (Control), выход (Output), механизм (Mechanism). *ICOM*-код помещается около свободного конца граничной дуги на дочерней диаграмме.

ICOM-код состоит из двух частей. Первая часть – буква, определяющая вид дуги родительского блока: *I* – входная дуга, *C* – дуга управления, *O* – выходная дуга, *M* – дуга механизма. Вторая часть – число, указывающее на относительное положение точки подключения дуги к родительскому блоку; это положение определяется слева направо или сверху вниз. Например, граничная дуга *выручка от реализации* с кодом *I4* (рис. 3.33) соответствует четвертой сверху входной дуге родительского блока *A1* (рис. 3.32). Дуга механизма *консалтинг по управлению* с кодом *M2* (рис. 3.33) соответствует второй слева дуге механизма родительского блока *A1* (рис. 3.32).

ICOM-кодирование связывает дочернюю диаграмму со своим родительским блоком. Если блоки на дочерней диаграмме подвергаются дальнейшей декомпозиции и подробно описываются на дочерних диаграммах следующего уровня, то на каждую новую диаграмму назначаются новые *ICOM*-коды, связывающие граничные дуги этих диаграмм с дугами их родительских блоков.

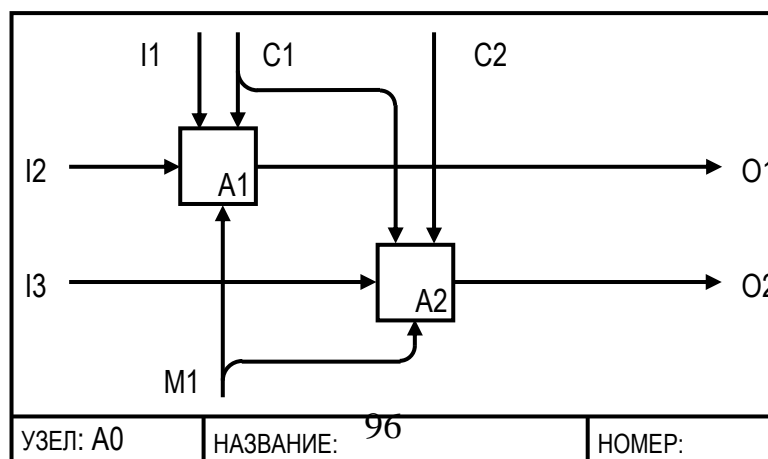
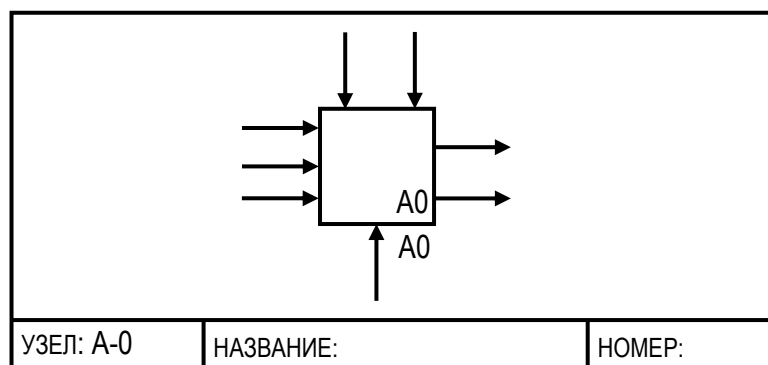


Рис. 3.23. Изменение ролей дуг

Иногда роли граничных дуг (вход, управление, механизм) могут меняться при переходе от родительского блока к дочерней диаграмме. Например, управляющая дуга в родительском блоке может быть входом на дочерней диаграмме. Аналогично, вход родительского блока может быть управлением для дочерних блоков. Пример изменения ролей дуг приведен на рис. 3.23: первая входная дуга на родительской диаграмме стала дугой управления на дочерней диаграмме.

При создании SADT-модели одну и ту же диаграмму вместе с ее блоками и дугами перечерчивают несколько раз, что приводит к появлению различных ее вариантов. Чтобы различать разные версии одной и той же диаграммы, в SADT используется схема контроля конфигурации диаграмм, основанная на хронологических номерах, или С-номерах. С-номера образуются из инициалов автора и последовательных номеров. Эти коды ставятся в нижнем правом углу SADT-бланка. Если диаграмма заменяет более старый вариант, то автор помещает предыдущий С-номер в скобках, чтобы указать на связь с предыдущей работой. Каждый автор проекта SADT ведет реестр всех версий созданных им диаграмм, нумеруя их последовательными целыми числами. Для этого используется бланк реестра С-номеров SADT.

Помимо использования для идентификации версий диаграмм, С-номера применяются для согласования родительских и дочерних диаграмм. С-номер дочерней диаграммы можно использовать вместо ее узлового номера, помещая его под родительским блоком. Связь дочерней диаграммы с родительской можно показать в области контекста SADT-бланка (правый верхний угол). В этой области изображают все блоки родительской диаграммы маленькими квадратиками. Квадратик родительского блока заштриховывается и возле заштрихованного квадрата размещается С-номер родительской диаграммы. Соединение диаграмм посредством С-номеров, в отличие от узловых номеров, позволяет указать не только номер, но и нужную версию диаграммы.

3.1.6. Перечень и дерево узлов

Перечень узлов представляет информацию о входящих в модель

узлах в форме списка, напоминающего обычное оглавление и отражающего иерархическую структуру модели, как показано на рис. 3.24.

Разработанная модель SADT со всеми уровнями структурной декомпозиции может быть представлена на единственной диаграмме в виде *дерева узлов*, дополняющего перечень узлов.

A0 Производить продукт

A1 Планировать производство

A11 Выбрать технологию производства

A12 Оценить требуемое время и затраты на производство

A13 Разработать производственные планы

A14 Разработать план вспомогательных действий

A2 Разрабатывать и управлять графиком выпуска и ресурсами

A21 Разработать основной график

A22 Разработать график координации работ

A23 Оценивать затраты и приобретать ресурсы

A24 Следить за выполнением графика и расходом ресурсов

A3 Планировать выпуск продукции

Рис. 3.24. Перечень узлов

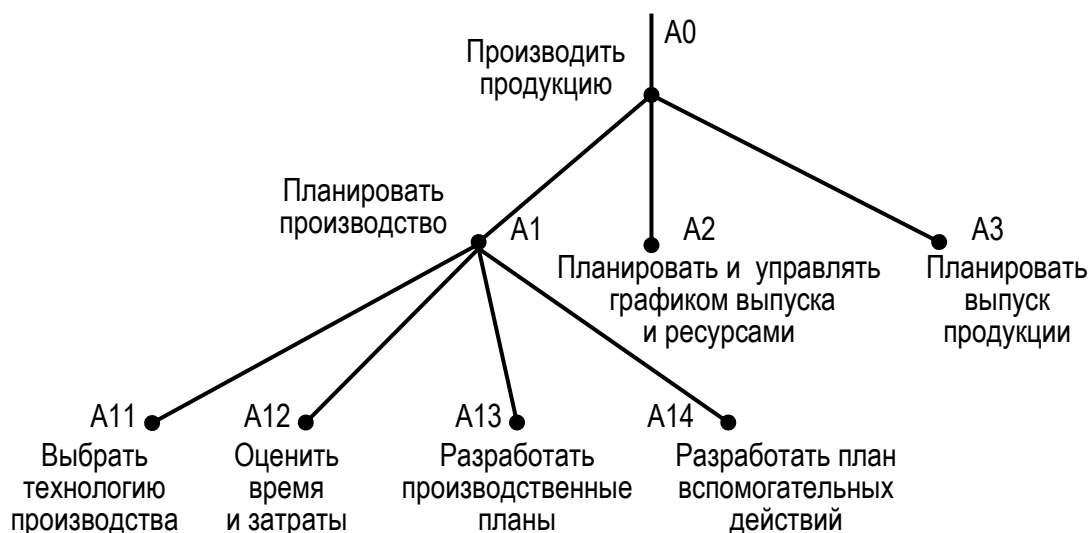


Рис. 3.25. Дерево узлов

Для изображения дерева узлов нет стандартного формата. Единственное требование состоит в том, что вся иерархия узлов модели должна быть представлена наглядно и понятно. Пример дерева узлов приведен на рис. 3.25.

3.1.7. Текст, глоссарий и диаграммы-иллюстрации (FEO)

Диаграмме может быть поставлен в соответствие структуриро-

ванный *текст*, представляющий собой краткий комментарий к содержанию диаграммы.

Текст используется для объяснений и уточнений характеристик, потоков, внутриблочных соединений и т.д. Текст не должен использоваться для описания и без того понятных блоков и дуг на диаграммах. При большом объеме текст располагается на отдельном листе модели (текстовой странице).

Глоссарий (glossary) предназначен для определения аббревиатур, ключевых слов и фраз, используемых в качестве имен и меток на диаграммах. Глоссарий определяет понятия и термины, которые должны быть одинаково понятны для всех участников разработки и пользователей модели, чтобы правильно интерпретировать ее содержание. Глоссарий составляется с любой необходимой степенью подробности.

Диаграммы FEO (For Exposition Only diagrams, диаграмма только для чтения) используются в качестве дополнений, поясняющих специфику содержания основных диаграмм в случаях, когда это необходимо. Диаграмма FEO не должна подчиняться синтаксическим правилам SADT.

3.1.8. Мастер-страница

Каждая диаграмма SADT изображается на стандартном бланке, именуемом мастер-страницей (Master Page). Бланк снабжен верхним и нижним штампами, содержащими информацию о конкретной диаграмме и о проекте, в состав которого входит диаграмма. Стандартный бланк обеспечивает единообразное оформление и хранение всех документов, относящихся к модели. Бланк имеет формат A4.

Мастер-страница содержит три области (рис. 3.5):

- 1) область проектной информации – в верхней части страницы;
- 2) рабочая область, в которой располагается диаграмма и другая информация о модели системы – в центральной части страницы;
- 3) область идентификации – вдоль нижнего края страницы.

Рабочее поле содержит блоки, дуги и метки. Оно служит также для размещения глоссария, текстовых комментариев, примечаний, замечаний, иллюстраций, эскизов и др.

Поля, которые содержат области проектной документации и идентификации описаны в табл. 3.2.

Таблица 3.2

Основные поля областей проектной документации и идентификации

Поле	Назначение
<i>Поля области проектной информации</i>	
Используется в (Used at)	Содержит название общей модели, составной частью которой является настоящая модель, или область ее применения

Поле	Назначение
Автор (Author)	Указывают имя автора диаграммы или наименование организации-разработчика проекта
Проект (Project)	Помещается название проекта или его аббревиатура
Замечания (Notes)	Используется для указания количества замечаний на текущей странице (для этого соответствующее число перечеркивают крестом или обводят эллипсом)
Пересмотр или версия (Revision)	Используется для указания информации о версии диаграммы
Дата и время (Date)	Дата и время либо создания страницы, либо внесения последнего изменения на странице, либо текущая дата и время
Поля для установления статуса страницы (Status)	В первом поле в зависимости от статуса ставится символ «X»; во втором поле указывается наименование статуса: рабочая версия, проект, рекомендовано, публикация
Читатель (Reader)	Поля для подписей читателей (экспертов) диаграммы
Дата (Date)	Дата прочтения диаграммы экспертом
Контекст (Context)	Содержит миниатюрную картинку соответствующую родительской диаграмме (без дуг), родительский блок на этой картинке затемнен (это позволяет иметь перед глазами структуру родительской диаграммы и место на ней декомпозируемого блока)
<i>Поля области идентификации</i>	
Узел (Node)	Содержит номер родительского блока диаграммы
Заголовок (Title)	Помещается имя родительского блока, в результате декомпозиции которого создана текущая диаграмма
Номер (C-Number)	Содержит С-номер диаграммы или номер ее страницы, которые указаны в качестве ссылки на родительской диаграмме под родительским блоком; здесь же указывается С-номер предыдущей версии диаграммы

3.1.9. Виды информационных потоков

Объектами структурного анализа и функционального моделиро-

вания методологии SADT являются организационно-экономические и производственно-технические системы. Такие системы выполняют преобразование материальных и информационных объектов, образующих соответствующие потоки. *Материальный поток* – непрерывное или дискретное множество материальных объектов, распределенное во времени. *Информационный поток* – множество информационных объектов, распределенное во времени.

Согласно методологии SADT информация подразделяется на три вида: ограничительная, описательная и предписывающая (управляющая).

Ограничительная информация – сведения о том, что запрещается делать:

а) на любом этапе жизненного цикла системы в целом и при всех обстоятельствах;

б) в рамках конкретного функционального блока.

Ограничительная информация содержится в регламентирующих документах: законах, подзаконных актах, международных, государственных и отраслевых стандартах, а также в специальных внутренних документах предприятия (например, в технических требованиях, условиях, регламентах и др.).

Описательная информация – сведения о свойствах объекта, преобразуемого функциональным блоком. Содержится в чертежах, технических описаниях и других документах, являясь неотъемлемой частью объекта в течение всего жизненного цикла. При выполнении функции данный вид информации изменяется.

Предписывающая (управляющая) информация – сведения о том, как, при каких условиях и по каким правилам следует преобразовать объект на входе в объект на выходе функционального блока. Содержится в технологических инструкциях, руководствах, документах, определяющих параметры и характеристики функционального блока.

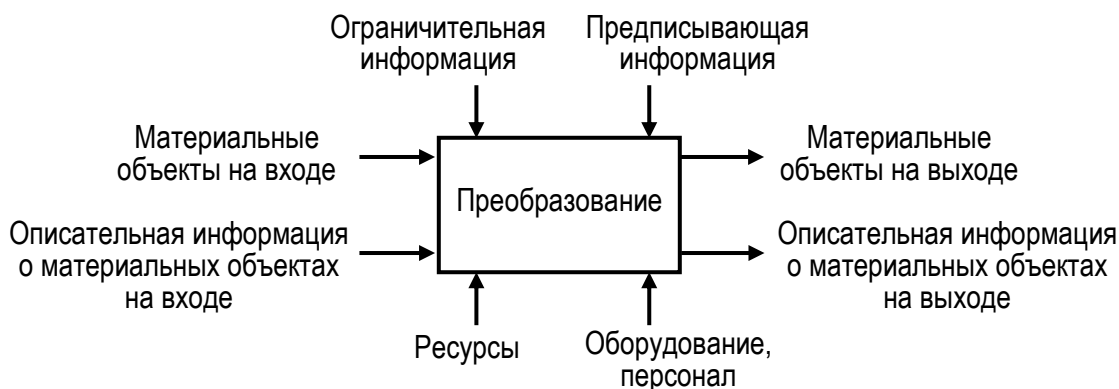


Рис. 3.26. Виды информации согласно методологии SADT

Ограничительная и предписывающая информация изображаются дугами управления. Описательная информация поступает на вход блока и формируется на его выходе, отображаясь дугами входа и выхода соответственно (рис. 3.26).

3.1.10. Классификация функций

Классификация функций, принятая в SADT, позволяет облегчить выбор глубины декомпозиции моделируемых систем и способствует выработке единообразных подходов моделирования в конкретных предметных областях.

Согласно методологии SADT функции подразделяются на четыре основных и два дополнительных вида. К основным видам функций относятся: деятельность, процесс, операция, действие. Дополнительными видами функций являются субдеятельность и подпроцесс.

Деятельность – совокупность процессов, выполняемых последовательно или параллельно, преобразующих множество материальных или/и информационных потоков во множество материальных или/и информационных потоков с другими свойствами. Деятельность осуществляется в соответствии с заранее определенной и постоянно корректируемой целью, с потреблением финансовых, энергетических, трудовых и материальных ресурсов, при выполнении ограничений со стороны внешней среды.

В модели SADT деятельность описывается блоком A0 на основной контекстной диаграмме A-0. Крупные многопрофильные структуры (предприятия, организации, фирмы) обычно занимаются различными видами деятельности. Поэтому общая модель таких сложных структур состоит из ряда частных моделей, каждая из которых относится к конкретному виду деятельности. В этом случае для разных видов деятельности создаются дополнительные контекстные диаграммы A-1, A-2,

Процесс – совокупность операций, выполняемых последовательно или параллельно, преобразующих материальный или/и информационный потоки в соответствующие потоки с другими свойствами. Процесс протекает в соответствии с управляющими директивами, вырабатываемыми на основе целей деятельности. В ходе процесса потребляются финансовые, энергетические, трудовые и материальные ресурсы и выполняются ограничения со стороны других процессов и внешней среды.

Операция – совокупность последовательно или параллельно выполняемых действий, преобразующих объекты, входящие в состав ма-

териального или/и информационного потока, в соответствующие объекты с другими свойствами. Операция выполняется:

- в соответствии с директивами, вырабатываемыми на основе директив, определяющих протекание процесса, в состав которого входит операция;
- с потреблением всех видов необходимых ресурсов;
- с соблюдением ограничений со стороны других операций и внешней среды.

Действие – преобразование какого-либо свойства материального или информационного объекта в другое свойство. Действие выполняется в соответствии с командой, являющейся частью директивы на выполнение операции, с потреблением необходимых ресурсов и с соблюдением ограничений, налагаемых на осуществление операции. Уровни декомпозиции, детализирующие действия, считаются состоящими из элементарных или простых функций.

Субдеятельность – совокупность нескольких процессов в составе деятельности, объединенная некоторой частной целью (являющейся «подцелью» деятельности).

Подпроцесс – группа операций в составе процесса, объединенная технологически или организационно.

Основные виды деятельности образуют естественную иерархию блоков SADT-диаграмм, предусматривая четыре уровня иерархии. Дополнительные виды функций требуются при анализе сложных форм деятельности с промежуточными уровнями декомпозиции.

На рис. 3.27 – рис. 3.30 приведены SADT-диаграммы, представляющие иерархию функций в виде абстрактной метамодели. Из метамодели видно, как функции взаимодействуют между собой на разных уровнях декомпозиции. Метамодель служит шаблоном, применение которого может облегчить создание реальной модели в конкретной предметной области.

SADT методология особое внимание уделяет процессу управления. Для каждого вида функций управление имеет свои отличительные особенности.

Управление деятельностью – процесс, состоящий из следующих операций:

- формулировка целей деятельности;
- оценка ресурсов, необходимых для осуществления деятельности и их сопоставление с имеющимися ресурсами;
- сбор информации об условиях протекания и фактическом состоянии деятельности;
- выработка и принятие решений, направленных на достижение

целей, в частности, решений о распределении ресурсов по процессам, входящим в состав деятельности; оформление решений в виде директив на управление процессами;

- реализация (исполнение) решений и оценка их результатов;
- корректировка (адаптация) ранее сформулированных целей (в случае необходимости, например, при нехватке ресурсов).

Управление процессом – операция, состоящая из следующих действий:

- анализ директивы на управление процессом, ее декомпозиция на директивы управления операциями;

- сбор информации о ходе выполнения операций, ее обобщение и формирование сведений о состоянии процесса; передача данных в подсистему управления деятельностью;

- сопоставление информации о ходе операций с директивами и выработка локальных решений, направленных на устранение отклонений;

- корректировка (в случае необходимости) директив на выполнение операций.

Управление операцией – действие, состоящее в выработке на основании директивы на управление операцией команд на управление действиями; реализации команд на управление действиями, оценке результатов выполнения, передаче необходимой информации в комплекс управления процессом; корректировке (в случае необходимости) команд на управление действиями.

Блоки управления должны присутствовать на каждой диаграмме. Через них осуществляются управляющие воздействия на остальные блоки диаграммы. В блоки управления поступает ограничивающая и предписывающая информация, которую они преобразуют в соответствующие директивы и команды. Имена блоков управления, как правило, содержат глагол «Управлять ... ».

3.1.11. Типизация функциональных моделей и диаграмм

Использование типовых моделей и отдельных диаграмм, ориентированных на применение в конкретных предметных областях, повышает эффективность процесса разработки функциональных моделей. Пример типовой SADT-модели промышленного предприятия приведен на рис. 3.31 – рис. 3.37. Аналогичные типовые модели могут быть разработаны для других видов бизнеса (оказание услуг, транспорт, банковское дело, финансовая деятельность и т.д.).

3.1.12. Организационно-технические структуры и механизмы SADT-моделей

Между различными видами функций существуют следующие отношения иерархической подчиненности: деятельность – субдеятельность – процесс – подпроцесс – операция – действие. Согласно методологии SADT каждая функция выполняется посредством механизма. В большинстве систем механизмами являются организационно-технические структуры. Одним из концептуальных принципов функционального моделирования является *отделение функций системы от ее организационной структуры*. Вместе с тем анализ показывает, что между иерархией функций и иерархией механизмов существует соответствие, иллюстрируемое рис. 3.38.

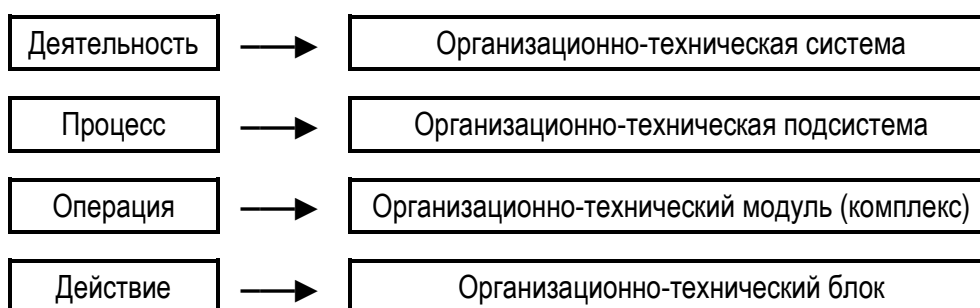


Рис. 3.38. Соответствие иерархий функций и механизмов

Выделяют следующие элементы иерархии механизмов.

Организационно-техническая система – организационная структура, персонал и комплекс технических средств, необходимые для осуществления деятельности.

Организационно-техническая подсистема – часть организационно-технической системы, обеспечивающая протекание процесса или субдеятельности.

Организационно-технический комплекс – часть организационно-технической подсистемы, предназначенная для выполнения операции.

Организационно-технический блок – часть организационно-технического комплекса, обеспечивающая выполнение действия.

При корректном построении модели ее блоки на разных уровнях декомпозиции связываются с определенными объектами организационно-технической структуры, выступающими в качестве механизмов. Таким образом, в результате функционального моделирования формируется организационно-техническая структура.

Во многих моделях механизмы создаются в результате выполнения нескольких процессов (рис. 3.39). Потребляемым ресурсом

(входом) этих процессов являются финансы, преобразуемые в другие виды ресурсов – энергетические, трудовые, материальные (оборудование, вспомогательные материалы и др.).

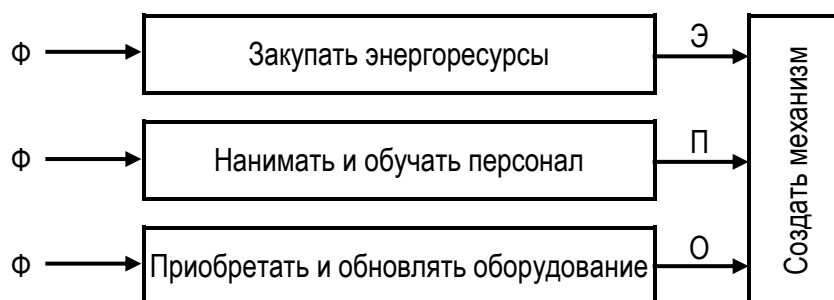


Рис. 3.39. Формирование механизма:
 Ф – финансы; Э – энергия; П – персонал; О – оборудование

Механизм любого уровня обеспечивает выполнение деятельности (процесса, операции, действия), потребляя ресурсы: финансовые, энергетические, трудовые, непосредственно или с помощью промежуточных преобразований, называемых вспомогательными процессами (рис. 3.39). В отличие от основных, вспомогательные процессы не создают конечного продукта деятельности и, следовательно, прибыли.

3.1.13. Организация процесса моделирования и управление проектом

SADT-модель является результатом скоординированной совместной работы большой группы участников проекта. В коллектив, занимающийся разработкой моделей, должны входить:

- руководитель проекта;
- авторы (разработчики) модели;
- технический совет;
- эксперты в предметной области;
- библиотекарь.

«Источники информации» считаются дополнительным участником проекта.

Руководитель проекта – лицо, осуществляющее административное управление проектом. Руководитель проекта должен выполнять следующие функции:

1. Выбирать разработчиков модели (авторов).
2. Определять обязательные источники информации, на которые разработчик модели будет опираться при построении модели. Этими источниками могут быть либо люди, осведомленные о различных ас-

пектах рассматриваемой сферы деятельности, либо документы, освещающие предметную область моделирования. Эти источники определяются на начальной стадии моделирования, но их состав должен пересматриваться и уточняться в процессе моделирования, т.к. по мере построения модели потребность в информации меняется.

3. Выбирать экспертов, чьи знания будут использованы разработчиком для получения оценки моделей. Список экспертов составляется на начальной стадии проекта и уточняется о мере необходимости.

4. Формировать технический совет. Руководитель проекта по должности является председателем технического совета.

5. Утверждать статус модели или ее частей, рассматриваемых техническим советом (статусы: рабочая версия, проект, рекомендовано и публикация).

Разработчики модели (авторы) – лица, создающие SADT-модели. Разработчик должен выполнять следующие функции:

1. Осуществлять сбор информации об исследуемой области, используя обязательные источники информации, определенные руководителем проекта. При недостаточности собранных сведений автор может использовать другие источники информации (людей или документы) с обязательным указанием ссылок на них.

2. Обучать (при необходимости) основам SADT-моделирования других участников проекта (руководителя проекта, экспертов и членов технического совета) для обеспечения правильного понимания ими моделей. Обучение заключается в ознакомлении с правилами чтения модели и техникой моделирования.

3. Создавать SADT-модели: строить SADT-диаграммы, составлять глоссарий, снабжать модель иллюстрациями и текстовыми примечаниями. Разработчик составляет SADT-папки с рабочим материалом и передает их другим участникам проекта для рецензирования. В процессе итеративного рецензирования модели разработчик корректирует модель в соответствии с замечаниями экспертов и технического совета.

Технический совет следит за соответствием выполняемых работ конечным целям проекта. Технический совет принимает и утверждает модели. Модели, достигшие достаточного уровня детализации и точности, направляются в технический совет. Технический совет представляет автору письменные замечания и рекомендации для доработки модели. В спорных ситуациях, которые могут возникнуть в процессе моделирования, технический совет отвечает за подготовку компромиссных решений. Для руководителя проекта совет формирует предложения по установлению статуса моделей.

Технический совет формируется из специалистов с высоким уровнем компетентности: экспертов и профессионалов, знакомых с предметной областью моделирования. В совете должны быть представлены все службы, имеющие отношение к рассматриваемой в проекте предметной области. Желательно включать в совет экспертов из смежных областей, не входящих в исследуемую область, но связанных с ней. Эти эксперты помогут адекватно оценить влияние окружающей среды на объект моделирования. Иногда в совет могут входить лица, играющие роль источников информации. Авторы в технический совет не входят.

Эксперт – выбираемое руководителем проекта лицо, обладающее специальными знаниями определенных аспектов моделируемой области. Его опыт в предметной области, к которой относится моделируемый объект, позволяет делать полезные критические замечания в процессе создания модели.

Эксперты призваны критически оценивать создаваемую модель. Это осуществляется в процессе итеративного рецензирования моделей (цикл автор/читатель). Главной задачей эксперта является оценка адекватности модели определенной предметной области. Иногда от эксперта требуется заполнить пробелы или завершить изложение материала, представленного в SADT-папке. Комментарии экспертов служат ценным материалом для уточнения модели. Эксперты подразделяются на две группы: эксперты-рецензенты и эксперты-читатели.

Эксперт-рецензент – член коллектива разработчиков, знающий предметную область моделирования, специализирующийся на конкретной функции предприятия и ответственный за обеспечение критических комментариев относительно разрабатываемой модели. Эксперт-рецензент должен знать SADT-методологию и уметь делать письменные структурированные замечания в рассылаемых папках. Он является постоянным и активным участником цикла автор/читатель.

Эксперт-читатель – член коллектива разработчиков, профессионально знающий предметную область моделирования, понимающий SADT-методологию и умеющий читать SADT-диаграммы. Эксперт-читатель знакомится с документацией (SADT-папкой), не делая письменных комментариев. От экспертов-читателей авторы получают замечания с помощью опроса.

Библиотекарь – лицо, ответственное за хранение документации, изготовление копий и координацию обмена письменной или электронной информацией (регистрация и рассылка SADT-папок, получение рецензий). Кроме того, библиотекарь печатает законченные модели после того, как они одобрены и приняты к использованию.

Источники информации – различные источники (люди, документы), из которых исходная информация для SADT-модели поступает к разработчику. По мере развития процесса моделирования потребности в информации изменяются, и руководитель проекта пересматривает состав источников информации.

Люди, являющиеся источниками информации, обладают конкретными знаниями о частных свойствах объекта моделирования, их участие в моделировании может быть ограничено несколькими минутами опроса. Однако именно эти источники обеспечивают основу для моделирования. Информация, предоставляемая ими, помогает автору лучше разобраться в различных аспектах предметной области, что способствует построению более точной модели.

Каждый источник информации представляет предметную область по-своему, и на разработчике лежит ответственность за правильный отбор информации. Особенно это относится к источникам-документам. *Источники-документы* отражают состояние объекта моделирования в некоторый момент времени, и их роль крайне важна, но для их эффективного использования необходима дополнительная работа, связанная с интерпретацией, пониманием и подтверждением содержащихся в них сведений.

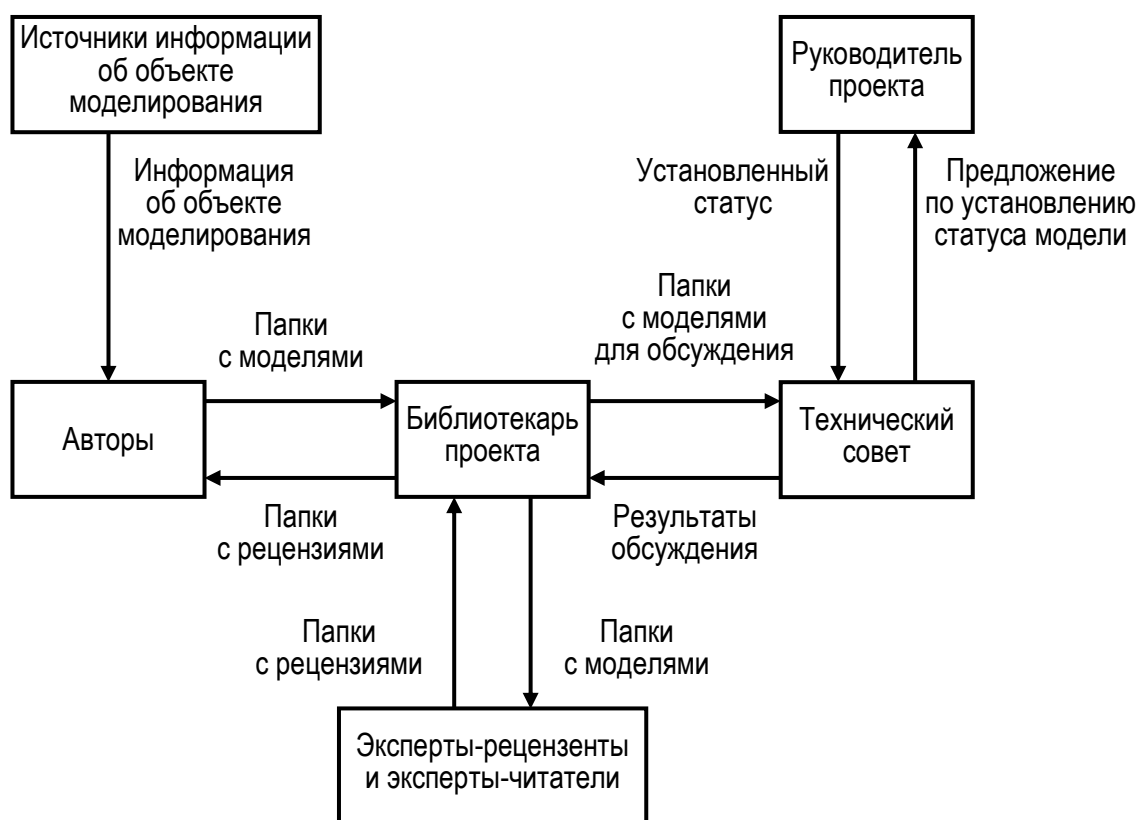


Рис. 3.40. Структура взаимодействия участников проекта

Структура взаимодействия участников проекта приведена на рис. 3.40. Руководитель проекта и разработчики модели (авторы) являются главными исполнителями. Хотя конечной целью разработчика является получение одобрения модели техническим советом, окончательно утверждает результаты руководитель проекта. Таким образом, обеспечивается согласованность интересов авторов, рецензентов, технического совета и руководителя проекта.

Процесс моделирования в соответствии с методологией SADT включает следующие основные виды деятельности (рис. 3.41):

- сбор информации об объекте моделирования;
- создание модели в виде SADT-диаграмм и сопутствующих материалов;
- хранение информации и координация обмена информацией между участниками проекта, с использованием SADT-папок;
- рецензирование модели;
- принятие и утверждение модели.

Процесс SADT-моделирования, как показано на рис. 3.41, имеет итерационный характер. Постепенное уточнение модели обеспечивает создание точного описания объекта моделирования.

В период подготовки проекта изучается и устанавливается область действия модели, определяются цели проекта, разрабатывается детальный план проекта, выбираются авторы, эксперты, вспомогательный персонал (библиотекарь, делопроизводители и технические оформители) и формируется технический совет.

На начальном этапе моделирования автор создает первоначальную модель системы (model draft – черновик модели). Для этого он осуществляет сбор информации об исследуемой области, документирует полученную информацию и представляет ее в виде начальных SADT-диаграмм, глоссария и других элементов модели.

Существует несколько стратегий получения информации: опрос экспертов, анкетирование группы экспертов, чтение документов, наблюдение за работой системы и исследование аналогов.

Лучшим способом получения информации считается *опрос экспертов*. Экспертам известны различные нюансы и аспекты системы, которые не отражены в документах или которые трудно объяснить. Цель опроса – получение информации, необходимой для построения определенной части модели.

Организация эффективного опроса экспертов требует дополнительной подготовки. Автор модели заранее определяет круг проблем и формулирует конкретные вопросы, на которые необходимо получить ответы экспертов. Для этого он использует такие стратегии сбора ин-

формации как чтение документов, наблюдение за работой системы и анализ аналогов. Материалы опроса оформляются сразу же после встречи с экспертом, что уменьшает потери важной информации. На основе полученных сведений составляются SADT-диаграммы, глоссарий и новый список вопросов к экспертам. Обычно проводится несколько опросов для выяснения все более конкретных деталей рассматриваемой области.

Анкетирование проводится для того, чтобы опросить большие группы экспертов в сжатые сроки. Его используют, например, когда необходимо быстро получить сведения о работе какой-либо определенной части системы с разных позиций. Анкетирование при опросе экспертов позволяет выявить, какие части системы более всего нуждаются в улучшении. Рекомендуется использовать анкеты только после достаточного знакомства с системой, когда можно задать конкретные, четко сформулированные вопросы.

Чтение документов – один из способов получить первоначальное представление о системе и сформулировать вопросы к экспертам. Документы – хороший источник информации, т.к. они всегда доступны, и с ними можно работать в удобном темпе. В SADT-проектах документы обычно хранятся в библиотеке проекта.

Еще одна стратегия получения информации – *наблюдение за работой моделируемой системы* или даже участие в ее работе. Во время наблюдения за работой системы часто возникают вопросы, которые не возникли бы, если бы автор только читал документы или разговаривал с экспертами.

Иногда авторы *исследуют аналоги* определенного класса систем (например, телефонные сети, бухгалтерия), таким образом, приобретая фундаментальные знания в соответствующей предметной области, и сами становятся источниками информации.

Каждый источник информации представляет информацию со своей точки зрения. Синтезируя различные точки зрения в процессе сравнения, разработчик создает адекватное описание объекта моделирования.

Методология SADT предусматривает необходимость сохранения всей информации, собранной в процессе моделирования.

На основе полученной информации определяется цель и точка зрения модели, составляется список объектов и список функций системы, осуществляется построение диаграммы A0 и ее обобщение на диаграмме A-0.

Цель модели определяется исходя из вопросов, на которые должна ответить модель. Точка зрения модели должна соответство-

вать той позиции, с которой описывается система. Не всегда можно определить цель и точку зрения в начале проекта, они уточняются после создания диаграммы A0.

Разработка диаграммы A0 начинается с составления списка информационных и материальных объектов системы. Затем составляется список функций системы, при этом учитывается работа системы в нормальных условиях и в случае ошибок. На диаграмму A0 помещаются 3 – 6 основных функций системы, изображаемых блоками, а также основные объекты системы, изображаемые граничными и внутренними дугами. Размещение блоков и дуг на диаграмме производится в соответствии с синтаксическими и семантическими правилами SADT.

Информация, зафиксированная на диаграмме A0, обобщается на диаграмме A-0. Имя единственного блока диаграммы A-0 совпадает с названием диаграммы A0 и должно точно отражать то, что делает система. Все граничные дуги диаграммы A-0 соответствуют граничным дугам диаграммы A0 и описывают то, чем система обменивается с внешней средой. На диаграмме A-0 также размещаются цель и точка зрения модели.

Диаграммы A-0 и A0 представляют общий вид системы, описывая основную деятельность системы и показывая ее связь с внешней средой. Создавая диаграммы A0 и A-0 автор выполняет начальную декомпозицию системы, а затем обобщает эту декомпозицию. Декомпозиция системы, представленная на диаграмме A0, отражает наиболее важные функции и объекты системы. Обобщение модели, представленное на диаграмме A-0, отражает основную функцию системы, наиболее важные входы, управления, выходы и механизмы системы, а также определяет цель и точку зрения модели.

Начальные диаграммы и связанный с ними глоссарий составляют первую папку, которая проходит несколько циклов автор/читатель, пока не будут созданы стабильные диаграммы A0 и A-0.

Блоки диаграммы A0 детализируются на диаграммах следующего уровня. Новые диаграммы строятся аналогично диаграмме A0. Цель и точка зрения диаграмм известны и определены на диаграмме A-0. В процессе декомпозиции для каждого блока составляется свой список объектов и список функций, которые затем отображаются на более детальных диаграммах в виде дуг и блоков. Аналогично можно выполнить декомпозицию блоков диаграмм второго уровня на диаграммах более низкого уровня. Декомпозиция прекращается, когда диаграммы, образующие нижний уровень модели, достаточно детализированы для достижения цели модели.

Обычно рассматривается несколько альтернативных декомпозиций системы, что улучшает понимание работы системы и способствует выявлению скрытых фактов ее функционирования.

Один из способов оценки точности диаграммы заключается в рассмотрении сценариев ее работы. Автор рассматривает одну из возможных ситуаций и оценивает, насколько корректно работает диаграмма в заданных условиях. По мере развития сценария (при необходимости) в диаграмму вносятся изменения.

Дальнейшее совершенствование первоначальных SADT-моделей выполняется в процессе итеративного рецензирования, называемого циклом автор/читатель.

Итеративный процесс рецензирования гарантирует, что окончательная версия SADT-модели будет точно и полно описывать моделируемую систему.

Завершив построение первоначальной модели, автор оформляет результаты работы в виде SADT-папки и передает другим участникам проекта для рецензирования.

SADT-папка представляет собой небольшой набор взаимосвязанных SADT-диаграмм и других материалов, отобранных автором для рецензирования читательской аудиторией.

Титульный лист папки содержит название папки, краткое содержание рабочих материалов, фамилии читателей, которые должны прочитать папку, и дату возврата автору. Диаграммы располагаются после титульного листа в порядке возрастания номеров узлов. Листы с глоссарием и иллюстрациями, которые дополняют диаграммы, располагаются сразу же после тех диаграмм, к которым они относятся.

Размеры папок должны быть небольшими, чтобы обеспечить быстрое рецензирование. Опыт показывает, что папка не должна содержать более одной диаграммы и ее прямых потомков – в общей сложности не более шести диаграмм. Если в папку включен дополнительный материал, количество диаграмм следует уменьшить. Папка с более чем 6 страницами информации может перегрузить читателей и получить слабый отклик. Единственным исключением является публикация всей модели в виде папки в конце аналитического проекта. Читатели готовы к тому, что такие папки должны быть объемными, но в то же время они знают, что их будет немного. Типичная для большинства SADT-проектов рабочая папка включает одну диаграмму, декомпозицию хотя бы одного из ее блоков, лист глоссария и, возможно, иллюстрацию.

В информационных папках перед экспертом должны ставиться конкретные, четко сформулированные вопросы, связанные с модели-

рованием.

Папки составляются, как только автор чувствует, что накопилось достаточно новой информации для рецензирования. Папка также составляется, когда автору не хватает информации для продолжения работы или он сомневается в верности сделанного.

Автор посылает папки библиотекарю для распространения материалов среди участников проекта. Полученные библиотекарем папки регистрируются, копируются и рассылаются читательской аудитории в соответствии со списком адресатов на титульном листе папки. Записываются дата рассылки папки и срок ответа автору. Таким образом, библиотекарь знает, кто и когда получил папку и дату ее возврата. Читатели знакомятся с материалами, включенными в папку, записывают свои комментарии и возвращают ее библиотекарю. Папки с комментариями, полученные библиотекарем, вновь регистрируются и отсылаются автору. Автор просматривает папки с комментариями и пишет ответы на них. Папки с ответами передаются библиотекарю для регистрации и возврата читателям. Опаздывающим читателям библиотекарь напоминает о необходимости вернуть задержанные папки. Таким образом обеспечивается своевременная обратная связь между авторами и читательской аудиторией.

Библиотекарь является главным координатором процесса итеративного рецензирования моделей, обеспечивая своевременное и согласованное распространение рабочих материалов.

Читатели (эксперты или другие авторы модели) знакомятся с рабочими материалами, находящимися в папке, и критически оценивают их. Те читатели, у которых есть замечания или предложения, размещают их непосредственно на копиях диаграмм в виде пронумерованных комментариев. Письменная форма рецензирования, принятая в SADT, предусматривает документирование всех идей, возникающих в процессе проектирования, что дает возможность обращаться к ним при повторном пересмотре материала.

Обычно отдельная папка рецензируется одновременно несколькими читателями. Критическая оценка модели несколькими экспертами позволяет отразить различные аспекты и составить более точное описание моделируемой системы.

Автор дает письменный ответ на каждое замечание читателей, помещая его рядом с соответствующим комментарием, и возвращает папку для дальнейшего рассмотрения. Вопросы, которые не удастся согласовать в процессе письменного рецензирования, обсуждаются во время устной беседы автор/читатель. Этот цикл продолжается до тех пор, пока авторы и читатели не придут к единому мнению.

В процессе ответов на комментарии читателей, автор вносит изменения в свой экземпляр папки. Это позволяет обобщить различные, иногда противоречивые позиции читателей. Во всех спорных ситуациях должны быть найдены компромиссные решения.

Автор переделывает диаграмму и другие элементы модели в соответствии с комментариями читателей и формирует новую папку. После внесения изменений старые варианты диаграмм остаются в архиве проекта. На этом текущий цикл рецензирования считается завершенным, и автор может начинать новый цикл на основе переработанного материала.

После нескольких циклов автор/читатель читательская аудитория обычно приходит к согласию относительно содержания конкретной модели. Появляется общее понимание системы.

Как только автор сочтет, что набор диаграмм и сопутствующих материалов достигли достаточного уровня детализации и точности, он формирует на их основе SADT-папку и посылает на утверждение в технический совет. Члены совета обсуждают и оценивают, насколько точно и полно модель описывает моделируемую систему, после чего автору направляются рекомендации и замечания для доработки модели. Автор переделывает модель в соответствии с этими рекомендациями и замечаниями. Технический совет утверждает результаты.

В SADT методологии приняты следующие виды статусов (status), которые постепенно присваиваются моделям, в зависимости от уровня их готовности:

1) рабочая версия (working) – диаграмма с большим числом изменений (все новые диаграммы являются рабочими версиями);

2) проект (draft) – диаграмма, прошедшая несколько циклов автор/читатель; имеет меньше изменений по сравнению с предыдущими версиями, что свидетельствует о достижении некоторого согласия между автором и читателями; нуждается в уточнении;

3) рекомендовано (recommended) – диаграмма и сопутствующие ей тексты прорецензированы и утверждены техническим советом, в нее не предполагается внесение существенных изменений;

4) публикация (publication) – материал, который может быть представлен заказчику модели, готовый для окончательной печати и публикации.

Когда с помощью модели можно ответить на вопросы, определяющие цель моделирования, модель считается завершенной. Руководитель проекта производит официальное утверждение окончательной версии модели. В ней должны быть устранены все разногласия. Окончательная версия модели является согласованным представлением о

системе с определенной точки зрения и может применяться для достижения поставленной цели.

3.2. Методология моделирования потоков данных DFD

Методология моделирования потоков данных DFD (Data Flow Diagrams) применяется для построения функциональной модели системы. В соответствии с этой методологией, модель системы представляется как иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от момента ввода ее в систему до выдачи конечному пользователю. Методология моделирования потоков данных показывает функции верхнего уровня и их взаимодействия через материальные или информационные потоки. Методология DFD не делает акцента на последовательности выполнения функций во времени.

Существуют несколько нотаций для представления диаграмм потоков данных: Гейна-Карсона (Chris Gane, Trish Sarson), Йордана-ДеМарко (Edward Yourdon, Tom DeMarko), SSADM и др. Далее будет использоваться нотация Гейна-Карсона.

3.2.1. Состав диаграмм потоков данных

Основными компонентами диаграмм потоков данных являются:

- 1) внешние сущности;
- 2) процессы;
- 3) потоки данных;
- 4) накопители данных.



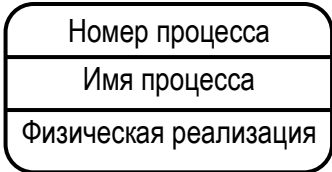

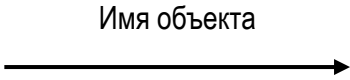
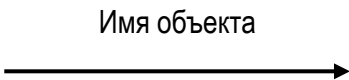
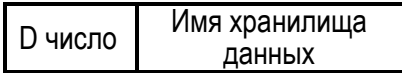

1. *Внешняя сущность* (external entity) – это материальный объект или физическое лицо, выступающее в качестве источника или приемника материальных или информационных объектов (например, *заказчики, поставщики, банк* и др.). Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой системы. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой системы, или, наоборот, часть процессов системы может быть вынесена за пределы диаграммы и представлена как внешняя сущность.

Внешняя сущность изображается на диаграмме прямоугольником, у которого с верхней левой стороны имеется тень (табл. 3.3). Прямоугольник расположен как бы над диаграммой и бросает на нее

тень. Внешние сущности обычно размещаются у краев диаграммы. Одна и та же внешняя сущность может присутствовать на диаграмме в нескольких экземплярах. Данный прием позволяет сократить количество линий, соединяющих элементы диаграммы.

Таблица 3.3

Элементы диаграммы потоков данных

Элемент	Нотация Гейна-Сарсона	Нотация Йордана-ДеМарко
Внешняя сущность		
Процесс		
Поток данных		
Хранилище данных		

2. *Процесс* (activity) – это преобразование входных материальных или информационных потоков в выходные в соответствии с определенным алгоритмом. Каждый процесс в системе связан с исполнителем, который осуществляет данное преобразование. Например, процесс может выполняться *программой, подразделением предприятия, сотрудником предприятия, аппаратным устройством* и др.

Процесс изображается на диаграмме прямоугольником со скругленными углами, имеющим три поля: номер, имя и физическая реализация (табл. 3.3).

Номер процесса служит для его идентификации.

Имя процесса рекомендуется формировать в следующем порядке. В начале указывается действие, выполняемое процессом. Действие описывается активным недвусмысленным глаголом в неопределенной

форме (например, *вычислить, рассчитать, проверить, определить, создать, получить* и др.). За действием указывается объект, над которым действие осуществляется. Объект описывается существительным в винительном падеже. Например, *ввести сведения о клиентах, выдать информацию о текущих расходах*. Использование таких глаголов, как «обработать», «модернизировать» или «отредактировать» означает, как правило, недостаточно глубокое понимание данного процесса и требует дальнейшего анализа.

Информация в поле *физической реализации* показывает, кто или что (программа, подразделение организации, аппаратное устройство или др.) выполняет данный процесс, например, *бухгалтерия* (рис. 3.42).

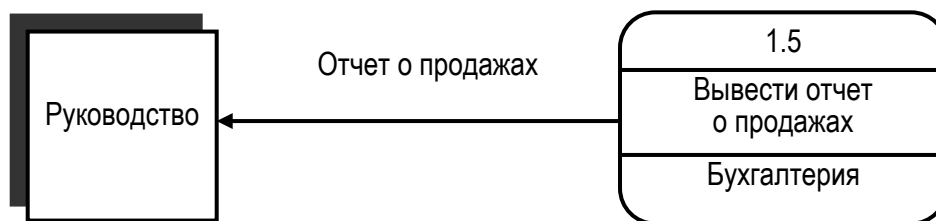


Рис. 3.42. Пример потока данных

3. *Поток данных* (data flow) определяет материальные или информационные объекты, передаваемые от источника к приемнику. Например, поток данных может быть *информацией, передаваемой по кабелю между двумя устройствами; письмами, пересылаемыми по почте; дисками, переносимыми между компьютерами; товарами, поступающими от поставщика на склад*, и др.

Поток данных изображается на диаграмме стрелкой, которая показывает направление потока. Над линией указывают имя потока, наиболее точно отражающее его содержание (табл. 3.3). Имя потока данных рекомендуется формировать в следующем порядке: вначале указывается материальный или информационный объект, определяемый потоком данных, затем указывается статус объекта. Например, *отчет о продажах* (рис. 3.42); *продукция, отгруженная клиенту; товары, полученные от поставщика*, и др.

Стрелки могут присоединяться к любой стороне элементов диаграммы потоков данных. Для отображения обмена объектами между элементами диаграммы в обоих направлениях используются двунаправленные стрелки. На рис. 3.49 показан взаимный обмен между *управлением предприятия* и *поставщиками*, при заключении договоров на поставку товаров.

Стрелки на DFD-диаграммах могут разветвляться и соединять-

ся. Каждому сегменту стрелки после точки ветвления присваивается собственное имя, чтобы показать декомпозицию объектов, переносимых потоком (рис. 3.43).

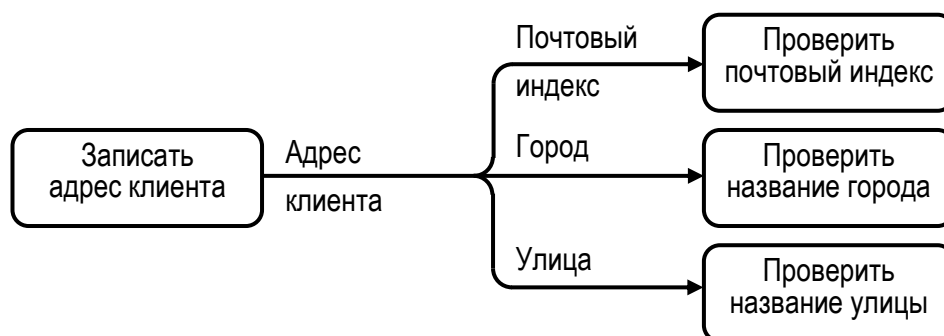


Рис. 3.43. Разветвление стрелки, иллюстрирующее декомпозицию данных

При объединении стрелок формируются так называемые комплексные объекты. Пример объединения приведен на рис. 3.44.



Рис. 3.44. Объединение потока в один

4. *Хранилище данных* (или накопитель данных, data store) – это абстрактное устройство для хранения материальных или информационных объектов. Объекты в любой момент можно поместить или извлечь из хранилища данных, при этом допускаются разнообразные способы помещения или извлечения объектов. Примерами хранилищ данных могут быть: *склад для хранения продукции; сейф для хранения трудовых книжек сотрудников; ящик в картотеке; архив, в котором хранятся документы; база данных или любой файл на диске, в котором хранится информация; таблица в оперативной памяти* и др.

Хранилище данных на диаграмме изображается, как показано в табл. 3.3. В первом поле указывается идентификационный номер хранилища данных, во втором – его имя. Идентификационный номер формируется из буквы «D» и числа. Имя должно наиболее точного

описывать сохраняемые объекты. На рис. 3.45 приведен пример хранилища данных.



Рис. 3.45. Пример хранилища данных

Хранилище данных является прообразом будущей базы данных.

3.2.2. Построение иерархии диаграмм потоков данных

На первом этапе построения иерархии диаграмм потоков данных создается *контекстная диаграмма*, которая определяет границу системы. На контекстной диаграмме изображаются: основная функция системы, внешние сущности, материальные или информационные потоки, которыми обмениваются система и внешние сущности. Контекстные диаграммы детализируются на диаграммах потоков данных следующего уровня.

При проектировании простых систем строится единственная контекстная диаграмма с топологией в форме звезды. В центре такой диаграммы находится главный процесс (основная функция системы), соединенный с внешними сущностями материальными или информационными потоками.

Система считается сложной, если содержит большое количество внешних сущностей (более десяти), имеет распределенную природу или состоит из нескольких подсистем. Для сложных систем строится несколько иерархически упорядоченных контекстных диаграмм. Контекстная диаграмма верхнего уровня содержит набор подсистем, соединенных материальными или информационными потоками. Контекстные диаграммы следующего уровня детализируют подсистемы.

Подсистема, входящая в состав сложной системы, изображается на контекстной диаграмме, как показано на рис. 3.46. Поле *физической реализации* для подсистем не заполняется.

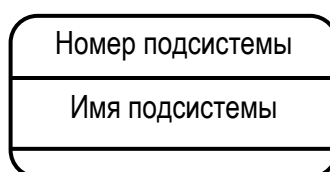


Рис. 3.46. Подсистема

Номер подсистемы служит для ее идентификации. В поле *име-*

ни подсистемы указывается наименование подсистемы в форме подлежащего, за которым следуют определения и дополнения, например, *регистрация налогоплательщиков; проверка документов; подсистема обслуживания клиентов* и др.

Контекстные диаграммы проверяют на полноту информации об объектах системы и отсутствие информационных связей между объектами системы. Воздействия внешних сущностей на систему называют *событиями*. Для проверки контекстной диаграммы на полноту составляют список событий и реакций системы на эти события. Входные потоки данных отображают события, выходные потоки – реакцию системы на события. Каждое событие должно быть представлено хотя бы одним входным потоком данных и соответствующими выходными потоками.

На следующем этапе каждую подсистему контекстной диаграммы, детализируют при помощи диаграммы потоков данных. В свою очередь, каждый процесс на диаграмме потоков данных может быть детализирован на диаграмме потоков данных более низкого уровня. Решение о завершении детализации процесса, и признании его *элементарным*, принимают в следующих случаях:

- наличие у процесса небольшого количества входных и выходных потоков данных (2 – 3 потока);
- процесс выполняет единственную логическую функцию преобразования входной информации в выходную;
- возможность описания логики процесса в виде последовательного алгоритма;
- возможность описания логики процесса при помощи мини-спецификации небольшого объема (не более 20 – 30 строк).

Для элементарного процесса диаграмма потоков данных не строится, он описывается посредством мини-спецификации процесса.

Мини-спецификация процесса содержит описание логики (функций) процесса. Мини-спецификация содержит номер и/или имя процесса, списки входных и выходных данных, а также тело (описание) процесса, являющееся спецификацией алгоритма или операции, преобразующей входные потоки данных в выходные. Существует большое количество разнообразных методов, позволяющих описать тело процесса. Языки, соответствующие этим методам, варьируются от псевдокода до визуальных языков проектирования. Мини-спецификация процесса составляется на самом нижнем уровне иерархии диаграммы потоков данных. У процесса должна существовать только одна мини-спецификация.

При детализации должны выполняться следующие правила:

– правило *балансировки* – детализирующая диаграмма в качестве внешних источников или приемников объектов может иметь только те компоненты (подсистемы, процессы, внешние сущности, хранилища данных), с которыми связана детализируемая подсистема или процесс;

– правило *нумерации* – при детализации процессов должна поддерживаться иерархия номеров, например, процессы, детализирующие процесс с номером 12, получают номера 12.1, 12.2, 12.3 и т.д.

Полная спецификация процессов включает дополнительно описание *структур данных*, используемых как при передаче информации в потоке данных, так и при хранении в накопителе данных. Структуры данных могут содержать альтернативы, условные вхождения и итерации. *Альтернатива* означает, что в структуру данных может входить только один из перечисленных элементов. *Условное вхождение* означает, что соответствующие элементы в структуре данных могут отсутствовать. *Итерация* означает, что элемент может повторяться определенное количество раз.

Для данных может быть указан тип: непрерывное или дискретное значение. Для непрерывных данных может указываться единица измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования. Для дискретных данных может указываться таблица допустимых значений.

Окончательную версию модели системы проверяют на полноту и согласованность. Для *полноты* модели все ее объекты (подсистемы, процессы, потоки данных) должны быть подробно описаны и детализированы. Для *согласованности* модели необходимо исследовать потоки данных системы. Если не все результаты выполнения процессов используются в системе, то это означает следующее: либо результаты являются лишними или ошибочными, либо необходимо выявить процесс, для которого эти результаты являются входом. Кроме того, для всех потоков и накопителей данных должно выполняться *правило сохранения информации*: все поступающие от куда-либо данные должны быть считаны и записаны.

3.2.3. Уточнение диаграмм потоков данных DFD на диаграммах потоков работ WFD

Если процесс невозможно представить в виде временной последовательности более мелких работ, тогда используется методология DFD. В противном случае процесс целесообразно описать посредством WFD-модели. Диаграммы потоков работ WFD (Work Flow

Diagram) используются при описании процессов нижнего уровня. На этих диаграммах имеются дополнительные объекты для описания процессов: логические операторы, события начала и окончания процесса, а также элементы, показывающие временные задержки (рис. 3.47).

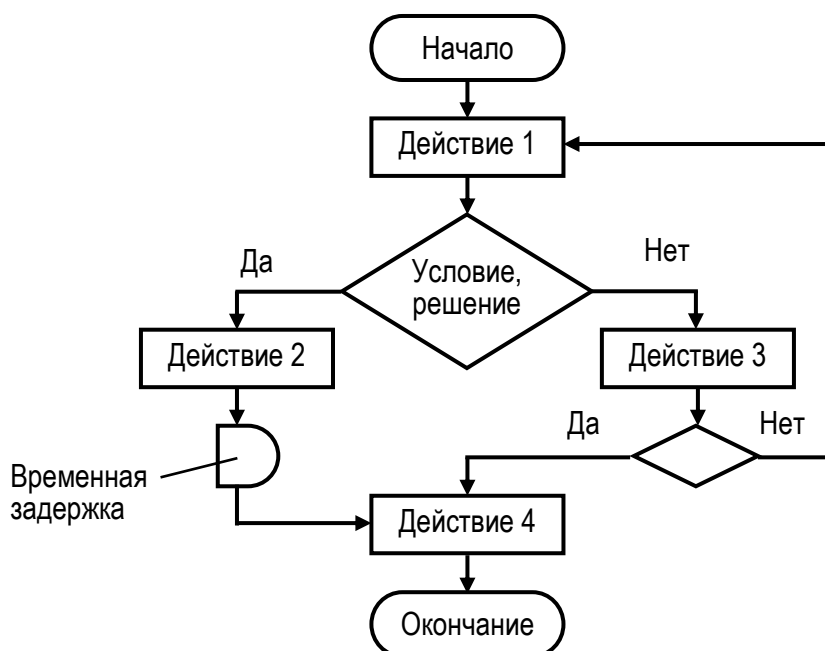


Рис. 3.47. Пример диаграммы потоков работ WFD

С помощью логических операторов, которые еще называют блоками принятия решений, показывают альтернативы, которые происходят в процессе, показывается в каких случаях процесс протекает по одной технологии, а в каких случаях по другой. Например, с помощью данных элементов можно описать ситуацию, когда договор, стоимость которого меньше определенной суммы согласуется одной группой сотрудников, а договор с большей стоимостью согласуется по более сложной технологии, в которой участвуют большее количество сотрудников.

С помощью событий начала и окончания процесса показывается, когда процесс начинается и когда заканчивается. Для жестко формализованных процессов таких, как бюджетирование, в качестве событий может выступать время.

В случаях, когда описание процесса проводится с целью его дальнейшей временной оптимизации, используют элементы задержки времени, которые указывают, между какими работами имеется временной разрыв. В данном случае последующая работа начинается

только через определенное время после завершения предшествующей.

На диаграмме WFD не показывают документы, т.к. эти схемы используются для описания процессов нижнего уровня, которые содержат детальные работы, и по названию которых понятно, что является входом и что является выходом.

Отличительной особенностью диаграмм WFD является то, что стрелки между операциями процесса обозначают не потоки объектов (материальных или информационных), а потоки или временную последовательность выполнения работ.

Таким образом, с помощью двух классических схем DFD и WFD можно описать подробно все процессы системы.

Современные методологии бизнес-моделирования поддерживают множество модификаций классической методологии WFD, например, eEPC-модель в ARIS.

3.3. Моделирование данных. Методология IDEF1X

IDEF1X (IDEF1 eXtended) – методология для построения модели данных. Базовыми понятиями методологии IDEF1X являются: объекты предметной области (сущности); свойства сущностей (атрибуты); отношения между сущностями (связи).

Сущность (entity) в IDEF1X описывает множество объектов, обладающих схожими свойствами, но однозначно отличаемых друг от друга по одному или нескольким признакам. Конкретный объект называют экземпляром сущности. Например, сущность *сотрудник* представляет собой всех сотрудников предприятия, а *Иванов Иван Иванович* – один из сотрудников – является конкретным экземпляром этой сущности.

Каждая сущность обладает уникальным именем. Имя сущности должно описывать множество объектов, а не конкретный экземпляр этого множества (*аэропорт*, а не *внуково*). Имя сущности указывается в единственном числе.

Каждая сущность обладает одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются через связь. *Атрибут* (attribute) – любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности.

Атрибуты отражают характерные свойства и признаки объектов, относящихся к определенной сущности. Атрибут имеет уникальное

имя, которое описывает представляемую атрибутом характеристику объекта. Имя атрибута указывается в единственном числе.

Каждый экземпляр сущности имеет свои собственные значения атрибутов.

Атрибуты делятся на *ключевые* и *описательные*. Каждая сущность обладает одним или несколькими ключевыми атрибутами.

Ключевые атрибуты – это атрибуты, предназначенные для однозначной идентификации каждого экземпляра сущности и позволяющие отличить один экземпляр сущности от другого (совокупность признаков, позволяющих однозначно идентифицировать объект). Если сущность имеет несколько ключевых атрибутов, то один из них обозначается в качестве первичного ключа, а остальные – как альтернативные ключи.

Первичный ключ (Primary Key) сущности – это набор атрибутов, выбранных для идентификации уникальных экземпляров сущности. Экземпляр сущности однозначно идентифицируется с помощью первичного ключа, при изменении ключа, соответственно меняется экземпляр сущности.

Ни одна из частей ключа не может быть NULL, не заполненной или отсутствующей. Установлены следующие правила для атрибутов, входящих в первичный ключ:

- 1) атрибут должен уникальным образом идентифицировать экземпляр сущности;
- 2) атрибут не должен использовать NULL значений;
- 3) атрибут не должен изменяться со временем;
- 4) имя атрибута должно быть как можно более коротким.

Описательные или *неключевые атрибуты* – это атрибуты, которые не выбраны ключевыми. Описательные атрибуты бывают обязательными или необязательными. *Обязательные атрибуты* для каждой сущности всегда имеют конкретное значение, *необязательные* – могут быть не определены.

На диаграмме сущность изображается прямоугольником, содержащим список атрибутов сущности. Над прямоугольником помещаются уникальное имя сущности и ее номер, разделяемые косой чертой «/».

Прямоугольник, отображающий сущность, разделяется горизонтальной линией на две части. В верхней части прямоугольника, называемой *ключевой областью*, расположены атрибуты, составляющие первичный ключ. В нижней части прямоугольника, называемой *областью данных*, расположены прочие, неключевые атрибуты (не входящие в первичный ключ).

В конце атрибутов, составляющих внешний ключ, добавляется «(FK)» (Foreing Key, внешний ключ).

Связи (relationship) представляют собой ссылки, соединения и ассоциации между сущностями. Связи присваивается имя в форме глагола, который описывает, каким образом сущности связаны между собой. Например, отдел *состоит из* сотрудников; самолет *перевозит* пассажиров; сотрудник *создает* отчеты.

Связи в IDEF1X отображаются в виде линии между двумя сущностями с жирной точкой на одном или обоих концах. Над линией связи указывается имя сущности. Связь позволяет отобразить следующую информацию: тип связи (идентифицирующая, неидентифицирующая), родительский конец связи, дочерний конец связи, знак обязательности связи, мощность связи, имя связи.

Если любой экземпляр одной сущности связан хотя бы с одним экземпляром другой сущности, то связь является *обязательной*. *Необязательная* связь представляет собой условное отношение между сущностями.

Сущности бывают зависимыми и независимыми.

Если между сущностями существует связь, тогда одной сущности, называемой *дочерней*, передается один или несколько ключевых атрибутов другой сущности, называемой *родительской*. Передаваемые атрибуты называются внешними ключами. *Внешние ключи* (Foreing Key, FK) – это атрибуты первичных ключей родительской сущности, переданные дочерней сущности через их связь. Передаваемые атрибуты называются *мигрирующими*.

Дочерняя сущность, уникальность которой зависит от значений атрибута внешнего ключа, называется *зависимой сущностью*. Для таких сущностей внешний ключ является частью их первичного ключа. Например, сущность *строка приходной накладной* является зависимой, т.к. ее идентификация зависит от сущностей *приходная накладная* и *товар* (рис. 3.51). Зависимая сущность в IDEF1X отображается прямоугольником с закругленными углами.

Зависимые сущности далее классифицируются на сущности, которые не могут существовать без родительской сущности, и сущности, которые не могут быть идентифицированы без использования ключа родительской сущности. Примером зависимости первого типа является отношение между двумя сущностями: *заказ*, используемый для сохранения заказов покупателей, и *элемент заказа*, который используется для сохранения отдельного элемента заказа. Связь между этими сущностями можно выразить в виде *заказ содержит один или несколько элементов заказа*. В этом случае сущность *элемент заказа*

зависит от существования *заказа*. Примером зависимости второго типа является отношение между сущностями *сотрудник* и *отдел*. В этом случае сущность *сотрудник* может существовать без сущности *отдел*.

Если существование дочерней сущности зависит от родительской, такая связь называется *обязательной*.

Сущности, независящие при идентификации от других сущностей, называются *независимыми сущностями*. В примере, представленном на рис. 3.51, сущность *поставщик* является независимой. Каждый экземпляр независимой сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями. В IDEF1X независимые сущности изображаются прямоугольниками.

Концепция зависимых и независимых сущностей усиливается типом взаимосвязи между двумя сущностями. Если экземпляр дочерней сущности однозначно определяется своей связью с родительской сущностью, то связь называется идентифицирующей, в противном случае – неидентифицирующей.

При *идентифицирующей связи* атрибуты первичного ключа родительской сущности мигрируют в ключевую область дочерней сущности. Такая связь обозначается сплошной линией с жирной точкой на конце, который соответствует дочерней сущности. Например, когда атрибут *код товара* мигрирует из сущности *товар* в сущность *строка приходной накладной* (рис. 3.51), то каждый экземпляр сущности *строка приходной накладной* зависит от *код товара*, который определяет его уникальным образом.

Связь, которая не ставит дочернюю сущность в зависимость от родительской называется *неидентифицирующей связью*. При неидентифицирующей связи атрибуты первичного ключа родительской сущности мигрируют в неключевую область дочерней сущности. Такая связь обозначается пунктирной линией с жирной точкой на конце возле дочерней сущности. Пример неидентифицирующей связи приведен на рис. 3.51, когда атрибут *код поставщика* мигрирует из сущности *поставщик* в неключевую область сущности *приходная накладная*.


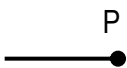



Если атрибуты, которые мигрировали в неключевую область дочерней сущности, не являются обязательными в этой сущности, то связь называется *необязательной неидентифицирующей связью*. В этом случае мигрировавшие атрибуты не нужны для идентификации дочерней сущности, она может существовать без этих атрибутов. Необязательная неидентифицирующая связь обозначается пунктирной линией с жирной точкой на конце возле дочерней сущности и ромбом на конце возле родительской сущности.

Поскольку переданные ключи в неидентифицирующей связи не являются составной частью первичного ключа дочерней сущности, то этот вид связи не проявляется ни в одной идентифицирующей зависимости.

Связи между сущностями могут иметь разную мощность. *Кардинальность (мощность) связи* определяет количество экземпляров дочерней сущности для каждого экземпляра родительской сущности. В методологии IDEF1X выделяют пять основных значений кардинальности (табл. 3.4).

Таблица 3.4

Мощность связи

Мощность связи	Нотация IDEF1X	Описание
Ноль, один или более	 Не выделяется на диаграмме	Каждая родительская сущность связана с ноль, одним или более экземпляров дочерней сущности
Один или более	 Обозначается буквой «Р»	Каждая родительская сущность связана с одним или более экземпляров дочерней сущности
Ноль или один	 Обозначается буквой «Z»	Каждая родительская сущность связана не более чем с одним экземпляром дочерней сущности
Точно n, где n – число	 n	Каждая родительская сущность связана с фиксированным числом экземпляров дочерней сущности
От n до m, где n и m – числа (n < m)	 n-m	Каждая родительская сущность связана не менее чем с n и не более чем с m экземплярами дочерней сущности

Неопределенная связь или связь «многие-ко-многим» означает, что один экземпляр первой сущности связан с несколькими экземплярами второй сущности и, наоборот, один экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Такая связь отображается сплошной линией с жирными точками на обоих концах связи. Для того чтобы избежать конфликтов, создаваемых связями «многие-ко-многим», в модель данных добавляют про-

межуточные сущности, которые служат посредниками между двумя сущностями, связанными неопределенной связью. При этом неопределенная связь преобразуется в две последовательные связи «один-ко-многим».

Ассоциированная сущность представляет данные, которые ассоциируются с отношениями между двумя и более сущностями. Обычно данный вид сущностей используется в модели для разрешения отношения «многие-ко-многим».

Кроме этого, модель включает понятия взаимно исключающих, рекурсивных и неперемещаемых связей. При наличии взаимно исключающей связи экземпляр сущности участвует только в одной связи из некоторой группы связей. Рекурсивная связь предполагает, что сущность может быть связана сама с собой. Неперемещаемая связь означает, что экземпляр сущности не может быть перенесен из одного экземпляра связи в другой.

Рекурсивная связь – такая связь, при которой одна и та же сущность является и родительской, и дочерней. Такая ситуация часто встречается в реальном мире. *Люди* – родители *людей*. *Руководители* управляют *руководителями*. *Компании* сотрудничают с *компаниями*. Существует два типа рекурсии. Первый тип называется *иерархической рекурсией* (другое название – однотабличная рекурсия) и задает иерархию связей между родительской и дочерней сущностями, при которой родитель может породить любое количество дочерних сущностей, но дочерняя сущность может иметь только одного родителя. Второй тип – *сетевая рекурсия* (другое название – двухтабличная рекурсия). Она задает «сеть» отношений между родительскими и дочерними сущностями, когда родитель может иметь любое количество дочерних сущностей, и дочерняя сущность может иметь любое количество родительских сущностей. В обоих случаях все связи представляют собой пары первичных ключей в одной и той же таблице, но каждый из двух случаев имеет свое собственное значение. В обоих случаях необходимо присвоить имена ролей мигрировавшим внешним ключам для того, чтобы передать значение рекурсивной связи.

Когда возникает проблема рекурсии «многие-ко-многим», т.е. сетевой рекурсии, можно прояснить ситуацию, создав промежуточную сущность и преобразуя связь «многие-ко-многим» в две связи «один-ко-многим».

Группу подобных сущностей можно объединить в одну категорию. Такая связь называется *Subtype Relationship Connector* – объединение подтипов. Сущность, описывающая категорию, называется

надтипом, а ее элементы – подтипами (рис. 3.48).



Рис. 3.48. Пример связи Subtype Relationship Connector

Нотация методологии IDEF1X лежит в основе многих современных Case-средств для проектирования информационных систем (ERStudio, Erwin, Designer и д.р.).

3.4. Пример использования структурного подхода

3.4.1. Описание предметной области

В качестве примера рассмотрим упрощенную схему деятельности предприятия розничной торговли. Торговое предприятие – это первичное звено торговли, ее самостоятельный хозяйствующий субъект с правом юридического лица, созданный для закупки, хранения, реализации товаров и предоставления разного рода сопутствующих услуг в целях удовлетворения потребностей рынка и получения прибыли. Розничная торговля включает предпринимательскую деятельность, связанную с продажей товаров конечным потребителям для их личного некоммерческого пользования. Структура торговотехнологического процесса предприятия розничной торговли зависит от его функциональных особенностей (стационарное, передвижное, сезонное, посылочное и др.), особенностей его устройства (магазин, магазин-склад, палатка и др.), размера торговой площади (гипермаркет (от 5000 м²), универсам (от 400 м²), гастроном (от 18 м²) и др.), применяемых форм продажи товаров и методов обслуживания (самообслуживание, продажа товаров через прилавки) и других факторов.

Основными бизнес-процессами предприятия розничной торговли будем считать: закупку товаров, хранение запасов товаров и продажу товаров. К бизнес-процессу управления предприятием розничной торговли отнесем: распределение ресурсов; планирование; ценообразование; управление основными бизнес-процессами (закупкой, хранением, продажей и др.); маркетинг (формирование ассортимента товаров); предоставление покупателям информации о товарах (через продавцов, рекламу и т.п.); бухгалтерский и финансовый учет. Ин-

формационная система применяется для поддержки всех процессов розничной торговли, поэтому будем считать процесс ее использования обеспечивающим процессом. Бухгалтерский и финансовый учет также используется всеми процессами, для удобства отображения информации на диаграммах будем изображать этот процесс отдельно от процесса управления предприятием.

Задачей бизнес-процесса *закупка товаров* является обеспечение наличия на складе необходимого товарного запаса под продажи. Данный бизнес-процесс включает в себя следующие процессы: управление закупками товаров; формирование заказов поставщикам; обеспечение доставки товаров; приемка товаров (по количеству и качеству) от поставщиков на склад; оприходование товаров в базу данных.

Процесс *управления закупками товаров* обычно осуществляется начальником отдела закупок, который руководствуется ассортиментом товаров, полученным от менеджера по ассортименту, заявками на закупку товаров, полученными со склада и торгового зала, и ценовой политикой предприятия. Управление закупками, как правило, включает выполнение следующих функций:

- 1) определение товарных запасов на складе и в торговом зале;
- 2) анализ ассортимента товаров на складе и в торговом зале;
- 3) формирование списка требуемых товаров;
- 4) прогноз объема спроса каждого товара;
- 5) определение страхового запаса для каждого товара (минимального количества товара на предприятии);
- 6) определение периода заказа каждого товара;
- 7) определение оптимального количества каждого товара для заказа;
- 8) осуществление поиска и выбора поставщиков товаров;
- 9) финансовое планирование сделки (определение оптимальных закупочных цен);
- 10) согласование с поставщиками заявок на закупку товаров (связь с поставщиками, проработка закупочных цен, выписка счетов);
- 11) подготовка и согласование с руководством предприятия плана закупок товаров;
- 12) подготовка/заключение договоров на поставку товаров с поставщиками;
- 13) подготовка/оформление отчетов руководству о закупках товаров.

После согласования заявок на поставку товаров с поставщиками и получения от них подтверждения на отгрузку товаров, начальник отдела закупок дает менеджеру отдела закупок задание на *формиро-*

вание заказов на закупку товаров у поставщиков. Сформированные заказы направляются поставщикам, от которых ожидается уведомление на отгрузку товаров. По требованию поставщиков может осуществляться предоплата поставщиков. Менеджеры отдела закупки совместно с менеджерами отдела логистики контролируют и организуют *доставку товаров*. Они формируют план поставок товаров, и передают его на склад для подготовки к приемке товаров.

Поступившие на предприятие розничной торговли товары доставляются в зону приемки. Бизнес-процесс *приемка товаров* является важнейшей операцией и проводится материально-ответственными лицами (товароведами, заведующими отделами и др.). Приемка товаров по количеству заключается в сверке массы, числа, единиц поступивших товаров с показателями счетов-фактур, товарно-транспортных накладных и других сопроводительных документов. Приемка товаров по качеству проводится в соответствии с требованиями, предусмотренными стандартами, по сопроводительным документам (сертификатам соответствия, сертификатам качества, гигиеническим сертификатам, выданным уполномоченными органами по стандартизации). Одновременно производится приемка тары по количеству, проверяется комплектность товаров, маркировка. При несоответствии заказа на закупку товаров поставке, получении товаров ненадлежащего качества или возврате товаров поставщику составляются соответствующие акты.

Сверенные накладные и счета-фактуры поставщиков, счета на предоплату товаров и оплату доставленных товаров от поставщиков, счета на оплату транспортных расходов направляются в бухгалтерию. Бухгалтерия после согласования с руководством предприятия производит оплату поступивших счетов. Информация о поступлении (приходе) товаров заносится в информационную систему.

После приемки товары доставляют в помещение для хранения. Бизнес-процесс *хранение товаров и управление товародвижением на складе* включает в себя следующие процессы: управление хранением и перемещением товаров на складе; формирование розничных цен на товары; хранение и размещение товаров на складе; маркировка товаров; проставление цены товаров; перемещение товаров в торговый зал; проведение инвентаризации товарных запасов на складе; списание товаров с истекшим сроком годности или испортившихся товаров.

Процесс хранения товаров предусматривает создание оптимального режима хранения товаров, рациональное размещение и укладку товаров, проведение санитарно-гигиенических мероприятий, предупреждающих ухудшение качества товаров, обеспечение сохран-

ности товарно-материальных ценностей и др. Сохранность количества и качества товаров в процессе хранения должны обеспечивать материально-ответственные лица (товаровед, работники склада, охранники).

Перед подачей в торговый зал товары должны быть полностью подготовлены к продаже. Количество подготовительных операций зависит от степени подготовленности товаров к продаже в момент поступления на предприятие розничной торговли, сложности ассортимента и других факторов. *Подготовка товаров к продаже* включает распаковку, сортировку, очистку, фасовку, упаковку, маркировку, проставление цены и др. *Маркировка товаров* предусматривает выполнение следующих операций: получение штрих-кодов и ценников на товары с помощью информационной системы; наклейка на товары штрих-кодов и проставление цены товаров.

Инвентаризация является обязательным процессом для торгового предприятия, она выполняется в соответствии с приказом руководства предприятия на инвентаризацию. Инвентаризация позволяет проверить соблюдение правил и условий хранения товаров, ведения складского хозяйства, фактическое соответствие остатков товарно-материальных ценностей данным бухгалтерского учета. В процессе инвентаризации выявляются товары с истекшим сроком годности или испорченные товары и передаются на списание. Инвентаризационная опись товаров, сличительная ведомость, а также акт списания товаров, полученные в процессе инвентаризации, направляются в бухгалтерию.

Информационная система на данном этапе используется для получения сведений о наличии товаров, формирования розничных цен на товары, получения штрих-кодов и ценников на товары, создания схемы размещения товаров на складе, формирования накладных на внутреннее перемещение товаров. В информационную систему вносятся результаты внутреннего перемещения, инвентаризации и списания товаров.

По заявкам менеджеров торгового зала товары со склада перемещаются в торговый зал для продажи. Бизнес-процесс *продажа товаров* включает в себя следующие процессы: управление продажей товаров; приемка товаров со склада; размещение и выкладка товаров в торговом зале; осуществление продажи товаров покупателям; подготовка кассового отчета; проведение инвентаризации товарных запасов на складе; перемещение товаров по результатам инвентаризации обратно на склад.

Сверенные накладные на внутреннее перемещение товаров со склада в торговый зал или обратно передаются в бухгалтерию.

Процесс *продажи товаров покупателям* состоит из: выбора по-

купателями товаров, получения от покупателей оплаты стоимости товаров, выдача покупателям чеков об оплате товаров, внесение сведений о продаже товаров в информационную систему.

В конце смены кассир-операционист составляет и сдает старшему кассиру справку-отчет, приходные и расходные кассовые ордера, денежные средства, полученные от продажи товаров и по приходным кассовым ордерам. Старший кассир в присутствии кассир-операциониста снимает показания секционных и контрольных счетчиков, получает распечатку или вынимает из контрольно-кассовой машины (далее ККМ) использованную в течение дня контрольную ленту. После снятия показаний счетчиков и проверки фактической суммы выручки кассир делает запись в кассовой книге. На основе полученных документов старший кассир составляет сводный отчет и вместе с актами, приходными и расходными кассовыми ордерами, а также контрольной лентой из ККМ сдает в бухгалтерию до начала работы следующей смены.

Бухгалтерия предоставляет руководству предприятия отчеты о продажах товаров и финансовые отчеты.

3.4.2. Построение моделей

Согласно методологии моделирования потоков данных DFD проект можно условно разделить на четыре фазы: анализ; глобальное проектирование (проектирование архитектуры системы); детальное проектирование; реализация (программирование).

На *фазе анализа* строится модель окружения (Environment Model). На данной фазе выполняют следующие действия:

1) анализ поведения системы, включающий определение назначения системы, построение начальной контекстной диаграммы, формирование матрицы списка событий (ELM), построение подробной контекстной диаграммы;

2) анализ данных, включающий определение состава потоков данных и построение диаграмм структур данных (DSD), конструирование концептуальной модели данных в виде диаграммы «сущность-связь» (ER-диаграммы).

Назначение системы определяет соглашение между проектировщиками и заказчиками относительно цели системы, а также общее описание системы и ее границ. Назначение системы оформляется в виде текстового комментария к контекстной диаграмме.

В примере для *предприятия розничной торговли* назначение системы можно сформулировать следующим образом: закупать товары

заданного ассортимента у поставщиков и реализовывать их покупателям с получением максимально возможной прибыли. При этом предприятие в качестве источника денежных средств может использовать кредиты банка.

Для построения начальной контекстной диаграммы необходимо определить внешние сущности, оказывающие влияние на функционирование системы. Внешние сущности взаимодействуют с системой, обмениваясь с ней материальными или информационными объектами. В процессе деятельности предприятия розничной торговли участвуют следующие внешние сущности: *поставщики*, *покупатели* и *банк*. На рис. 3.49 изображена начальная контекстная диаграмма для примера предприятия розничной торговли.

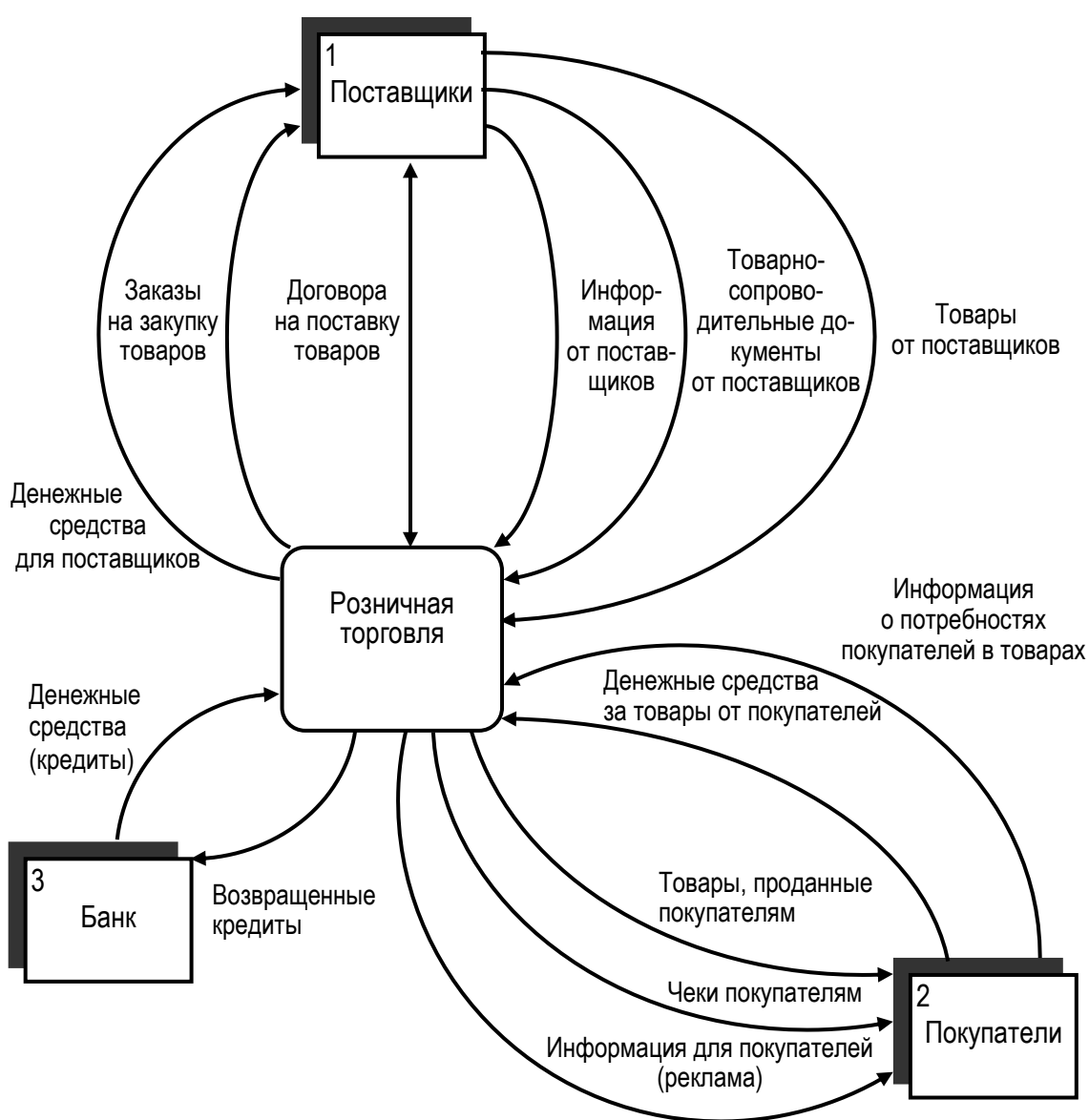


Рис. 3.49. Начальная контекстная диаграмма

Матрица списка событий (ELM) описывает внешние события, воздействующие на систему, и реакцию системы на эти события. Различают шесть типов событий:

- 1) нормальное управление (NC),
- 2) нормальные данные (ND),
- 3) нормальное управление/данные (NCD),
- 4) временное управление (TC),
- 5) временные данные (TD),
- 6) временное управление/данные (TCD).

Нормальные данные соответствуют событиям, которые система воспринимает непосредственно, например, *приход товаров от поставщиков на склад* – поступившие товары должны быть размещены на складе, а сведения о них должны быть занесены в базу данных. Нормальные данные отображаются на диаграммах в виде потоков данных. В табл. 3.5 приведены некоторые события из матрицы списка событий для примера *предприятия розничной торговли*.

Таблица 3.5

Примеры событий из матрицы списка событий

№ п/п	Событие	Тип	Реакция системы
1	Поставщик сообщает об изменении адреса	ND	Регистрация измененного адреса поставщика
2	Поставщик подтверждает поставку товаров	ND	Сформировать и направить поставщику заказ на закупку товаров
3	Поставщик осуществил доставку товаров с товарно-сопроводительными документами на склад	ND	Осуществить приемку товаров: сверить накладные и счета-фактуры от поставщиков с заказами на закупку товаров; сверить количество фактически поступивших товаров с показателями счетов-фактур, накладных и другой сопроводительной документацией; проверить качество поступивших товаров; внести сведения о поставке товаров в базу данных; сформировать отчеты о приемке товаров

			для бухгалтерии и руководства
--	--	--	-------------------------------

Окончание таблицы 3.5

№ п/п	Событие	Тип	Реакция системы
4	Покупатель выбрал товар и желает его приобрести	ND	Осуществить продажу товара покупателю: получить от покупателя оплату за товар; выдать покупателю чек об оплате товара; внести сведения о продаже товара в базу данных; сформировать сводный отчет о продажах товаров за текущий день (или смену) для бухгалтерии и руководства

Анализ функционального аспекта поведения системы завершается построением полной контекстной диаграммы. Процесс *розничная торговля*, представленный на начальной контекстной диаграмме, декомпозируется на несколько процессов, отражающих основные виды деятельности предприятия розничной торговли (рис. 3.50). Абстрактные потоки данных между внешними сущностями и процессами на начальной контекстной диаграмме преобразуются в потоки, представляющие обмен данными на более конкретном уровне. Один абстрактный поток может быть разделен на несколько конкретных потоков (табл. 3.6). Каждое внешнее событие из матрицы списка событий должно формировать некоторый поток данных (событие формирует входной поток, а реакция системы на событие – выходной поток).

Таблица 3.6

Соответствие потоков данных начальной и полной контекстных диаграмм

Потоки данных начальной контекстной диаграммы	Потоки данных полной контекстной диаграммы
Информация от поставщиков	Информация от поставщиков, необходимая для формирования заказов (данные о поставщиках, ассортимент товаров, цены на товары и др.)
Товарно-сопроводительные документы от поставщиков	Приходные накладные, счета-фактуры, сертификаты качества товаров; счета от поставщиков на предоплату товаров и оплату достав-

	ленных товаров
--	----------------

Анализ функционального аспекта поведения системы дает представление об обмене и преобразовании данных в системе.

Взаимосвязь между абстрактными потоками данных и конкретными потоками данных, представленными на контекстных диаграммах, отображается на диаграммах структур данных.

На фазе анализа строится концептуальная модель данных, представляемая в виде ER-диаграмм. Данная модель строится исходя из следующих соображений:

1) сущности могут быть распознаны как структуры данных на DFD диаграммах; чтобы рассматривать объект в качестве сущности, он должен обладать более чем одним атрибутом;

2) связи должны отражать наличие взаимодействия между сущностями, причем в системе должна сохраняться информация об этом взаимодействии.

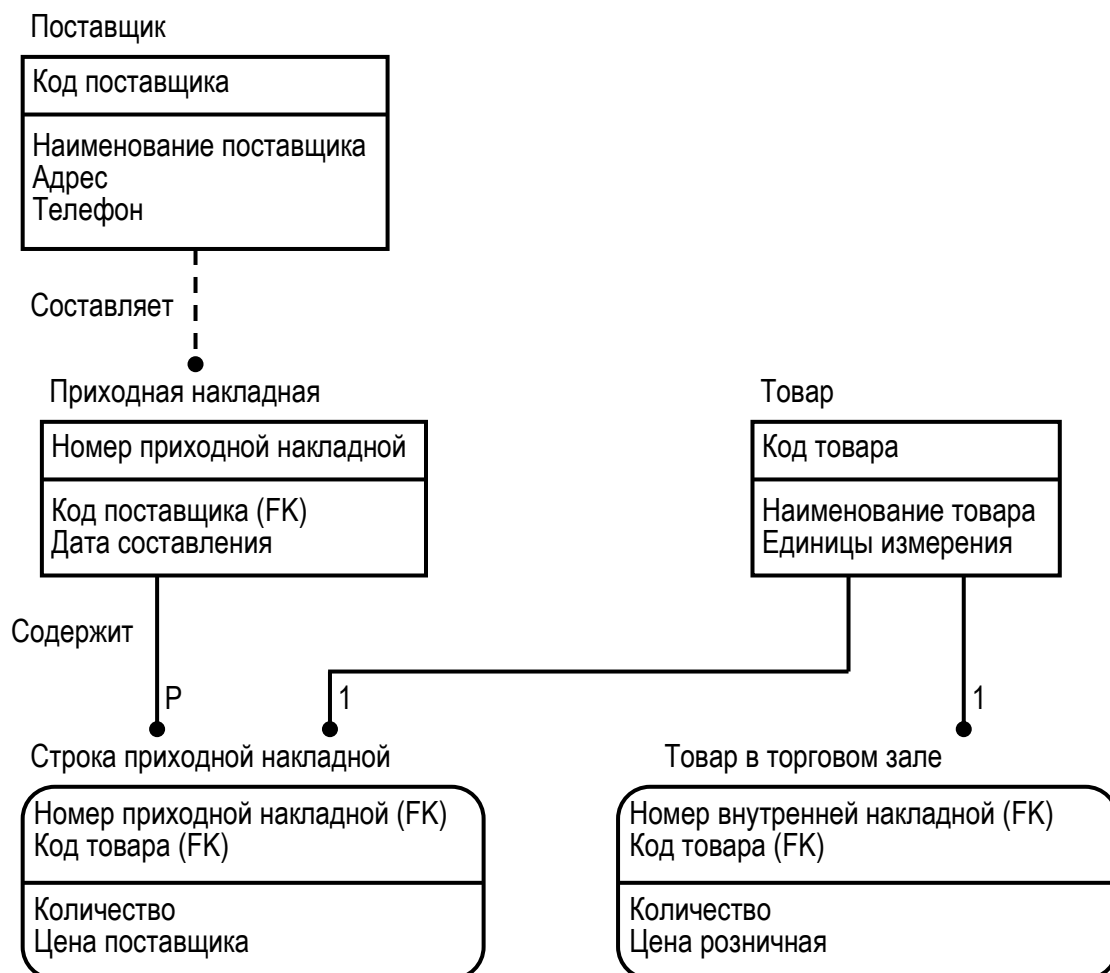


Рис. 3.51. Фрагмент ER-диаграммы

На рис. 3.51 приведен фрагмент ER-диаграммы для примера *предприятия розничной торговли*. На диаграмме представлены сущности: *поставщик*, *приходная накладная*, *товар*, *строка приходной накладной*. Сущность *поставщик* содержит информацию о поставщиках товаров, необходимую для составления приходной накладной. Сущность *приходная накладная* содержит информацию о количестве и ценах поступивших товаров. Поскольку общая информация о товарах применяется не только при составлении приходной накладной, то она сохраняется в отдельной сущности *товар*. Дополнительная сущность *строка приходной накладной* позволяет хранить сведения о количестве и ценах поступивших товаров отдельно от общей информации о товарах.

Между различными типами диаграмм существуют следующие взаимосвязи:

- 1) ELM – DFD диаграммы: внешние события – входные потоки данных, реакция системы на внешние события – выходные потоки данных;
- 2) DFD – DSD диаграммы: потоки данных – структуры данных верхнего уровня;
- 3) DFD – ER-диаграммы: накопители данных – ER-диаграммы;
- 4) DSD – ER-диаграммы: структуры данных нижнего уровня – атрибуты сущностей.

На *фазе проектирования архитектуры системы* (глобального проектирования) выполняются следующие действия:

- 1) детальное описание функционирования системы;
- 2) дальнейший анализ используемых данных и построение логической модели данных для последующего проектирования базы данных;
- 3) определение структуры пользовательского интерфейса, спецификации форм и порядка их появления;
- 4) уточнение диаграмм потоков данных и списка событий, выделение среди процессов нижнего уровня интерактивных и неинтерактивных, составление для них миниспецификаций.

Результатами проектирования архитектуры системы являются:

- 1) модель процессов (диаграммы архитектуры системы (SAD) и миниспецификации);
- 2) модель данных (ER-диаграммы);
- 3) модель пользовательского интерфейса (классификация процессов на интерактивные и неинтерактивные функции, диаграммы последовательности форм (FSD)).

Диаграмма последовательности форм FSD (Form Sequence

Diagram), показывает: основные формы приложения и порядок их появления; взаимосвязь между каждой формой и определенной структурной схемой; взаимосвязь между каждой формой и одной или более сущностями на ER-диаграммах. На диаграммах FSD фиксируется набор и структура вызовов экранных форм. Диаграммы FSD образуют иерархию, на вершине которой находится главная форма приложения, реализующего подсистему. На втором уровне находятся формы, реализующие процессы нижнего уровня функциональной структуры, зафиксированной на диаграммах SAD.

На *фазе детального проектирования системы* выполняются следующие действия:

- 1) уточнение модели базы данных для последующей генерации SQL-предложений;
- 2) уточнение структуры пользовательского интерфейса;
- 3) построение структурных схем, отражающих логику работы пользовательского интерфейса и модель бизнес-логики (Structure Charts Diagram – SCD) и привязка их к формам.

Результатами детального проектирования являются:

- 1) модель процессов (структурные схемы интерактивных и не-интерактивных функций);
- 2) модель данных (определение на ER-диаграммах всех необходимых параметров для приложений);
- 3) модель пользовательского интерфейса (диаграммы последовательности форм (FSD)).

На *фазе реализации* выполняются следующие действия:

- 1) генерация SQL-предложений, определяющих структуру целевой БД (таблицы, индексы, ограничения целостности);
- 2) уточнение структурных схем (SCD) и диаграмм последовательности форм (FSD);
- 3) генерация кода приложений.

На основе анализа потоков данных и взаимодействия процессов с хранилищами данных осуществляется окончательное выделение подсистем (предварительное должно было быть сделано и зафиксировано на этапе формулировки требований в техническом задании). При выделении подсистем необходимо руководствоваться принципом функциональной связанности и принципом минимизации информационной зависимости. Необходимо учитывать, что на основании таких элементов подсистемы как процессы и данные на этапе разработки должно быть создано приложение, способное функционировать самостоятельно. С другой стороны при группировке процессов и данных в подсистемы необходимо учитывать требования к конфигурированию

продукта, если они были сформулированы на этапе анализа.

Контрольные вопросы

1. Каково назначение методологии функционального моделирования IDEF0?
2. Дайте определение методологии SADT. На каких концептуальных положениях основана эта методология?
3. Дайте определение понятия система. Из каких компонентов могут состоять системы? Приведите примеры.
4. Дайте определение понятия моделирование. Что описывает модель? Для чего разрабатываются модели?
5. Приведите формальное определение модели с позиции SADT.
6. С каких позиций представляют систему функциональная модель и модель данных?
7. Дайте определения следующих понятий: SADT-модель, цель модели и точка зрения модели. На основе чего формируется цель модели? Почему у модели должна быть единственная точка зрения?
8. Дайте определения следующих понятий: декомпозиция системы, родительский блок, родительская диаграмма, дочерняя диаграмма, дочерний блок. Опишите процесс декомпозиции системы и построения иерархии диаграмм в соответствии с методологией SADT.
9. Дайте определения синтаксиса и семантики языка в соответствии с методологией SADT. Каково назначение основных компонентов синтаксиса SADT (блоков, дуг, диаграмм и правил)?
10. Что отображают на диаграмме блоки? Приведите основные синтаксические и семантические правила SADT, установленные для блоков.
11. Что отображают на диаграмме дуги? Приведите основные синтаксические и семантические правила SADT, установленные для дуг.
12. Дайте краткую характеристику следующих дуг: дуга управления, входная дуга, выходная дуга и дуга механизма.
13. Приведите основные правила, установленные для изображения блоков на SADT-диаграммах.
14. Дайте краткую характеристику следующих связей между функциями: доминирование, управление, выход-вход, обратная связь по управлению, обратная связь по входу, выход-механизм.
15. Дайте определения внутренней и граничной дуги. Как согласуются граничные дуги родительского блока и дочерней диаграммы?
16. Приведите основные правила, установленные для изображения дуг на SADT-диаграммах.

17. Как изображаются на диаграммах туннельные дуги? В каких случаях они используются?

18. Дайте определения следующих понятий: ветвление, слияние, развязывание и связывание. Приведите соглашения SADT, принятые для размещения меток на сегментах дуг в случае их ветвления или слияния.

19. Каково назначение узловых номеров, ICOM-кодов и C-номеров? Опишите правила их формирования.

20. Что и в каком виде представляют перечень и дерево узлов?

21. Каково назначение следующих элементов SADT-модели: текст, глоссарий и диаграммы FEO.

22. Опишите назначение основных областей и полей мастер-страницы.

23. Дайте определения материального и информационного потока. Дайте краткую характеристику следующих видов информации: ограничительная, описательная и предписывающая.

24. Дайте краткую характеристику основных (деятельность, процесс, операция, действие) и дополнительных (субдеятельность, подпроцесс) видов функций.

25. Какова связь организационно-технической структуры и механизмов SADT-модели?

26. Опишите основные функции участников проекта в процессе SADT-моделирования (руководитель проекта, разработчик модели (автор), эксперты, библиотекарь, технический совет, источники информации).

27. Опишите (используя диаграмму) процесс моделирования в соответствии с методологией SADT. Дайте краткую характеристику основных видов деятельности процесса SADT-моделирования (сбор информации; создание модели; хранение информации и координация обмена информацией; рецензирование модели; принятие и утверждение модели). Каково назначение и состав SADT-папки?

28. Дайте краткую характеристику основных статусов, присваиваемых моделям, в зависимости от уровня их готовности (рабочая версия, проект, рекомендовано и публикация).

29. Как определяется система в соответствии и методологией моделирования потоков данных?

30. Дайте краткую характеристику основных элементов диаграммы потоков данных: внешние сущности, процессы, потоки данных, хранилища данных. Приведите примеры.

31. Опишите процесс построения иерархии диаграмм потоков данных.

32. Опишите основные положения методологии моделирования данных IDEF1.

33. Опишите основные элементы диаграмм данных (в методологии моделирования данных IDEF1).

34. На примере рассмотрите основные этапы проектирования программного обеспечения при использовании структурного подхода.

Рекомендуемая литература

1. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: учебник для вузов. – М.: Финансы и статистика, 2000. – 352 с.

2. Иванова Г.С. Технология программирования: учебник для вузов. – М.: изд-во МГТУ им. Н.Э. Баумана, 2006. – 336 с.

3. Йордан Э. Структурное проектирование и конструирование программ. – М.: Мир, 1979.

4. Марка Д.А., МакГоуэн К.Л. Методология структурного анализа и проектирования SADT. – М.: МетаТехнология, 1993. – 240 с.

5. Черемных С.В., Семенов И.О., Ручкин В.С. Моделирование и анализ систем. IDEF-технологии: практикум. – М.: Финансы и статистика, 2006. – 192 с.

6. Рекомендации по стандартизации. Р 50.1.028-2001 Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования. – Введ. 01.07.2002. – М.: ИПК Изд-во стандартов, 2001. – 49 с.

7. Integration Definition for Function Modeling (IDEF0). Software Standard, Modeling Techniques. – National Institute of Standards and Technology, 1993. (Draft Federal Information Processing Standards Publication 183, 1993, Desember 21, 79 p.)

8. Integration Definition for Information Modeling (IDEF1X). Federal Information Processing Standards Publication 184, 1993, 87 p.

9. R.J. Mayer, C.P. Menzel, M.K. Painter, P.S. deWitte. Information Integration For Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report, September 1995. – Wright-Patterson AFB.

Глоссарий

Методология моделирования потоков данных DFD применяется для построения функциональной модели системы. Данная методология определяет модель системы как иерархию диаграмм потоков данных, описывающих асинхронный процесс преобразования информации с момента ввода ее в систему до выдачи пользователю.

Методология IDEF1X (IDEF1 eXtended) применяется для построения модели данных (в виде ER-диаграмм).

Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Под **модульным подходом** понимается разделение программы на части (модули) по некоторым установленным правилам.

Модуль – это отдельная функционально завершенная программная единица, которая может применяться самостоятельно, либо быть частью программы. Модуль обладает тремя основными признаками: реализует одну или несколько функций, имеет определенную логическую структуру, используется в одном или нескольких контекстах.

Список сокращений

ERD (Entity-Relationship Diagram) – диаграмма «сущность-связь»

DFD (Data Flow Diagrams) – диаграмма потоков данных

ICAM (Integrated Computer Aided Manufacturing) – программа интегрированной компьютеризации производства, направленная на увеличение эффективности промышленных предприятий посредством широкого внедрения компьютерных (информационных) технологий. Реализована Министерством обороны США в конце 70-х гг.

IDEF0 (ICAM DEFinition) – методология функционального моделирования, позволяющая исследовать структуру, параметры и характеристики производственно-технических и организационно-экономических систем; методология IDEF0 основана на методологии SADT

IDEF1X (IDEF1 eXtended) – методология для построения модели данных

SADT (Structured Analysis and Design Technique) – метод структурного анализа и проектирования

WFD (Work Flow Diagrams) – диаграмма потоков работ

Библиографический список

1. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: учебник для вузов. – М.: Финансы и статистика, 2000. – 352 с.
2. Дейкстра Э.В. Заметки по структурному программированию / в книге Дал У., Дейкстра Э.В., Хоар К. Структурное программирование. – М.: Мир, 1975. – С. 7 – 97.
3. Зиглер К. Методы проектирования программных систем. – М.: Мир, 1985. – С. 111 – 117.
4. Иванова Г.С. Технология программирования: учебник для вузов. – М.: изд-во МГТУ им. Н.Э. Баумана, 2006. – 336 с.
5. Йордан Э. Структурное проектирование и конструирование программ. – М.: Мир, 1979.
6. Майерс Г.Дж. Надежность программного обеспечения. – М.: Мир, 1980. – С. 92 – 100.
7. Марка Д.А., МакГоуэн К.Л. Методология структурного анализа и проектирования SADT. – М.: МетаТехнология, 1993. – 240 с.
8. Мейер Б. Основы объектно-ориентированного программирования. Лекция № 3. Модульность [Электронный ресурс] // Интернет-университет информационных технологий. – Электрон. дан. – 2005. – Режим доступа:
http://www.intuit.ru/department/se/oopbases/3/oopbases_3.html. – Загл. с экрана.
9. Хьюз Дж.К., Мичтом Дж. Структурный подход к программированию. – М.: Мир, 1980. – 276 с.
10. Черемных С.В., Семенов И.О., Ручкин В.С. Моделирование и анализ систем. IDEF-технологии: практикум. – М.: Финансы и статистика, 2006. – 192 с.
11. Рекомендации по стандартизации. Р 50.1.028-2001 Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования. – Введ. 01.07.2002. – М.: ИПК Изд-во стандартов, 2001. – 49 с.
12. Integration Definition for Function Modeling (IDEF0). Software Standard, Modeling Techniques. – National Institute of Standards and Technology, 1993.
13. Integration Definition for Information Modeling (IDEF1X). Federal Information Processing Standards Publication 184, 1993, 87 p.
14. R.J. Mayer, C.P. Menzel, M.K. Painter, P.S. deWitte. Information Integration For Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report, September 1995. – Wright-Patterson AFB.

Оглавление

Предисловие	3
Введение	4
Глава 1. Развитие модульного и структурного подходов	
разработки ПС.....	5
1.1. Основные подходы разработки архитектуры	
программных систем.....	6
1.1.1. Уровни абстракции.....	6
1.1.2. Метод портов.....	11
1.2. Методы проектирования программ, основанные	
на модульном принципе.....	17
1.2.1. Методы нисходящего проектирования.....	18
1.2.1.1. Пошаговое уточнение.....	18
1.2.1.2. Анализ сообщений.....	25
1.2.2. Методы расширения ядра.....	30
1.2.2.1. Метод иерархического проектирования	
модулей, разработанный Джексоном.....	31
1.2.2.2. Метод Парнаса.....	43
1.2.3. Метод восходящего проектирования.....	43
1.3. Структурное программирование.....	44
Глава 2. Модульный подход к разработке ПС	49
2.1. Понятие модуля.....	49
2.2. Внутренняя связность и внешнее сцепление	
модулей.....	50
2.3. Дополнительные характеристики,	
оказывающие влияние на независимость модулей...	56
2.4. Проектирование и программирование	
отдельных модулей.....	57
2.5. Модульная структура программы.....	60
Глава 3. Структурный подход к разработке ПС	62
3.1. Методология структурного анализа	
и проектирования SADT.....	62
3.1.1. Общие сведения.....	62
3.1.1.1. Основные термины, используемые SADT.....	65
3.1.1.2. Основные концептуальные положения SADT.....	69
3.1.1.3. SADT-модели.....	70
3.1.1.4. Принцип декомпозиции. Иерархия диаграмм.....	73
3.1.2. Синтаксические и семантические правила	
отображения SADT-моделей.....	76

3.1.3. Синтаксические и семантические правила SADT, установленные для блоков.....	77
3.1.4. Синтаксические и семантические правила SADT, установленные для дуг.....	80
3.1.5. Синтаксические и семантические правила SADT, установленные для диаграмм.....	82
3.1.5.1. Правила изображения блоков на диаграммах.....	83
3.1.5.2. Правила изображения дуг на диаграммах.....	86
3.1.5.3. Узловые номера, ICOM-коды и C-номера.....	94
3.1.6. Перечень и дерево узлов.....	97
3.1.7. Текст, глоссарий и диаграммы-иллюстрации (FEO).....	98
3.1.8. Мастер-страница.....	99
3.1.9. Виды информационных потоков.....	100
3.1.10. Классификация функций.....	102
3.1.11. Типизация функциональных моделей и диаграмм.....	108
3.1.12. Организационно-технические структуры и механизмы SADT-моделей.....	116
3.1.13. Организация процесса моделирования и управления проектом.....	117
3.2. Методология моделирования потоков данных DFD..	128
3.2.1. Состав диаграмм потоков данных.....	128
3.2.2. Построение иерархии диаграмм потоков данных...	132
3.2.3. Уточнение диаграмм потоков данных DFD на диаграммах потоков работ WFD.....	134
3.3. Моделирование данных. Методология IDEF1X.....	136
3.4. Пример использования структурного подхода.....	142
3.4.1. Описание предметной области.....	142
3.4.2. Построение моделей.....	146
Глоссарий.....	157
Библиографический список.....	158