

1. Моделирование систем с использованием стандарта IDEF0

1.1. Концепция стандарта IDEF0

Методология структурно-функционального моделирования (SADT - Structured Analysis and Design Technique -Технология структурного анализа и проектирования) легла в основу стандарта IDEF0 (Integration Definition for Function Modeling).

Стандарт IDEF0 — это технология описания системы в целом как множества взаимозависимых действий, или функций.

Стандарт IDEF0 имеет функциональную направленность, т.е. функции системы исследуются независимо от объектов, которые обеспечивают их выполнение.

«Функциональная» точка зрения на систему позволяет достаточно точно отделить аспекты назначения системы от аспектов ее физической реализации.

Наиболее часто стандарт IDEF0 применяется как технология исследования и проектирования систем на логическом уровне. По этой причине он, как правило, используется на ранних этапах разработки проекта, до IDEF3 (Integration Definition method for Process Description Capture) моделирования для сбора данных и моделирования процесса "как есть".

Результаты IDEF0 анализа могут применяться при проведении проектирования с использованием моделей IDEF3 и диаграмм потоков данных (DFD – Data Flow Diagram).

Методология IDEF0 основана на следующих концептуальных положениях.

Модель — искусственный объект, представляющий собой отображение (образ) системы и ее компонентов.

Считается, что М моделирует А, если М отвечает на вопросы относительно А.

Здесь М — модель, А — моделируемый объект (оригинал).

Модель разрабатывают для понимания, анализа и принятия решений о реконструкции (реинжиниринге) или замене существующей, либо проектировании новой системы.

Система представляет собой совокупность взаимосвязанных и взаимодействующих частей, выполняющих некоторую полезную работу.

Частями (элементами) системы могут быть любые комбинации разнообразных элементов/сущностей, включающие людей, информацию, программное обеспечение, оборудование, изделия, сырье или энергию (энергоносители).

Модель описывает, что происходит в системе, как ею управляют, что она преобразует, какие средства использует для выполнения своих функций и что производит.

Блочное моделирование и его графическое представление. *Основной концептуальный принцип методологии IDEF0* — представление любой изучаемой системы в виде набора взаимодействующих и взаимосвязанных блоков, отображающих процессы, операции, действия (определения — см. п.1.2.1), происходящие в изучаемой системе.

В IDEF0 все, что происходит в системе и ее элементах, принято называть **функциями**. Каждой функции ставится в соответствие блок.

На IDEF0-диаграмме, основном документе при анализе и проектировании систем, блок представляет собой *прямоугольник*.

Интерфейсы, посредством которых блок взаимодействует с другими блоками или с внешней по отношению к моделируемой системе средой, представляются *стрелками*, входящими в блок или выходящими из него.

Входящие стрелки показывают, какие условия должны быть одновременно выполнены, чтобы функция, описываемая блоком, осуществилась.

Лаконичность и точность. Документация, описывающая систему, должна быть точной и лаконичной.

Сведения о свойствах и характеристиках системы в форме традиционных текстов в этом смысле неудовлетворительны, поскольку зачастую содержат избыточную информацию, допускают неоднозначное толкование и т.д. Графический язык стандарта IDEF0 лаконично, однозначно и точно показать все элементы (блоки) системы и все отношения и связи между ними, выявить ошибочные, лишние или дублирующие связи и т.д.

Передача информации. Средства IDEF0 облегчают передачу информации от одного участника разработки модели (отдельного разработчика или рабочей группы) к другому.

К числу таких средств относятся:

- диаграммы, основанные на простой графике блоков и стрелок, легко читаемые и понимаемые;
- метки на естественном языке для описания блоков и стрелок, 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 затем происходит их детализация и уточнение;
- древовидные схемы иерархии диаграмм и блоков, обеспечивающие обзорность модели в целом и входящих в нее деталей, что особенно важно при моделировании больших систем.

Строгость и формализм. Разработка моделей IDEF0 требует

соблюдения ряда строгих формальных правил (синтаксис и семантика моделей IDEF0 приведена в Приложении 2), обеспечивающих преимущества методологии в отношении однозначности, точности и целостности сложных многоуровневых моделей.

Здесь же мы отметим только основное из них: *на всех стадиях и этапах разработки и корректировки модели должны строго, формально соблюдаться синтаксические и семантические правила графического языка, а результаты моделирования — тщательно документироваться с тем, чтобы при эксплуатации модели не возникало вопросов, связанных с неполнотой или некорректностью документации.*

Итеративное моделирование. Разработка модели в IDEF0 представляет собой пошаговую, итеративную процедуру.

На каждом шаге итерации разработчик предлагает вариант модели, который подвергают обсуждению, рецензированию и последующему редактированию, после чего цикл повторяется.

Такая организация работы способствует оптимальному использованию знаний системного аналитика, владеющего методологией и техникой IDEF0, и знаний специалистов — экспертов в предметной области, к которой относится объект моделирования.

Отделение «организации» от «функций». При разработке IDEF0-моделей следует избегать изначальной «привязки» функций исследуемой системы к существующей организационной структуре моделируемого объекта (предприятия, фирмы).

Это помогает избежать субъективной точки зрения, навязанной организацией и ее руководством.

Организационная структура должна явиться результатом использования (применения) модели.

Сравнение результата с существующей структурой позволяет, во-первых, оценить адекватность модели, а во-вторых — предложить решения, направленные на совершенствование этой структуры.

1.2. Методика разработки функциональных моделей в среде IDEF0 []

Объектами функционального моделирования и структурного анализа по методологии IDEF0 являются организационно-экономические и производственно-технические системы.

Функциональный блок - блок A0 (см. Приложение 2), отображающий моделируемую систему в целом, и блоки на любом уровне декомпозиции являются преобразующими блоками.

Преобразующий блок — блок IDEF0-диаграммы, преобразующий входы в выходы под действием управлений при помощи «механизмов» (см. Приложение 2).

Преобразование — цель и результат работы любого блока на диаграмме

любого уровня декомпозиции. Преобразованию в блоке могут подвергаться материальные и информационные объекты, образующие соответствующие потоки.

Материальный поток — непрерывное или дискретное множество материальных объектов, распределенное во времени.

Информационный поток — множество информационных объектов, распределенное во времени.

Информация, участвующая в процессах, операциях, действиях и деятельности в целом, может быть классифицирована на три группы:

- ограничительная;
- описательная;
- предписывающая (управляющая).

Ограничительная информация — сведения о том, что нельзя делать:

а) никогда, ни при каких обстоятельствах (кроме, быть может, форс-мажорных), в любой фазе жизненного цикла и на любом этапе функционирования системы в целом;

б) в рамках функционирования конкретного блока.

Ограничительная информация содержится в законах, подзаконных актах, международных, государственных и отраслевых стандартах, а также в специальных внутренних положениях и документах предприятия, в частности, в технических требованиях, условиях, регламентах и т.д.

Описательная информация — сведения об атрибутах объекта (потока), преобразуемого функциональным блоком. Эта информация содержится в чертежах, технических и иных описаниях, реквизитах и других документах, являясь неотъемлемым компонентом объекта в течение всего жизненного цикла. Данная информация сама преобразуется (изменяется) в результате выполнения функции.

Предписывающая (управляющая) информация — сведения о том, как, при каких условиях и по каким правилам следует преобразовать объект (поток) на входе в объект (поток) на выходе блока. Эта информация в технологических (в широком смысле) инструкциях, руководствах, документах, определяющих «настройки» и характеристики блока.

Схематическое изображение связей преобразующего блока в соответствии с соглашениями системы IDEF0 показано на рис. 1.1.

Ограничительная и предписывающая информация изображается стрелками, присоединяемыми к блоку на стороне управления, а описательная информация поступает на вход блока и формируется на его выходе, отображаясь стрелками входа и выхода соответственно.

Материальный поток и описывающий его *информационный поток* везде, где это не вызывает недоразумений, можно изображать одной стрелкой.



Рис. 1.1

1.2.1. Классификация функций, моделируемых блоками IDEF0

Практика построения функциональных моделей требует введения классификации явлений и событий, отображаемых в моделях.

Такая классификация облегчает выбор глубины декомпозиции моделируемых систем и способствует выработке единообразных подходов и приемов моделирования в конкретных предметных областях.

Здесь предлагается классификация, ориентированная на достаточно широкий круг организационно-экономических и производственно-технических систем.

Классификация делит все функции таких систем на четыре основных и два дополнительных вида.

Каждая рубрика в классификации представляет собой класс преобразующих блоков, экземпляры которого возникают и используются при моделировании конкретной системы.

А. Основные виды функций, моделируемых блоками IDEF0

1. Деятельность (синонимы: дело, бизнес) — совокупность **процессов**, выполняемых (протекающих) последовательно или/и 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 IDEF0 деятельность описывается блоком А0 на основной контекстной диаграмме А—0 (см. Приложение 2).

При моделировании крупных, многопрофильных структур (фирм, организаций, предприятий), которые по своему статусу занимаются различными видами деятельности, последние представляют собой различные

экземпляры класса «деятельность» и могут найти отражение в дополнительной контекстной диаграмме А—1 (см. Приложение 2).

В этом случае общая модель сложной структуры будет состоять из ряда частных моделей, каждая из которых относится к конкретному виду деятельности.

2. Процесс (синоним: бизнес-процесс) — совокупность последовательно или/и параллельно выполняемых операций, преобразующая 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.

3. Операция — совокупность последовательно или/и параллельно выполняемых действий, преобразующих объекты, входящие в состав материального или/и информационного потока, в соответствующие объекты с другими свойствами.

Операция выполняется:

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

4. 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.

Б. Дополнительные виды функций, моделируемых блоками IDEF0

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

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

Понятия группы А образуют естественную иерархию блоков на IDEF0-диаграммах при декомпозиции, предусматривая четыре уровня последней.

При анализе сложных видов деятельности могут потребоваться промежуточные уровни декомпозиции, основанные на применении функций

группы Б.

Уровни декомпозиции, детализирующие действия, естественно считать состоящими из элементарных или простых функций.

1.2.2 Организационно-технические структуры и механизмы IDEF0-моделей

Все функции, входящие в приведенную выше классификацию, находятся между собой в отношениях иерархической подчиненности по принципу «сверху вниз»: деятельность — субдеятельность — процесс — подпроцесс — операция — действие.

Согласно методологии IDEF0 каждая функция выполняется посредством механизма.

В большинстве систем, анализируемых при помощи функциональных моделей, такими механизмами служат организационно-технические структуры.

Одним из концептуальных принципов функционального моделирования является «отделение «организации» от функций».

Вместе с тем анализ показывает, что между иерархией функций (преобразований) и иерархией механизмов существует определенное соответствие, иллюстрируемое рис. 1.2.

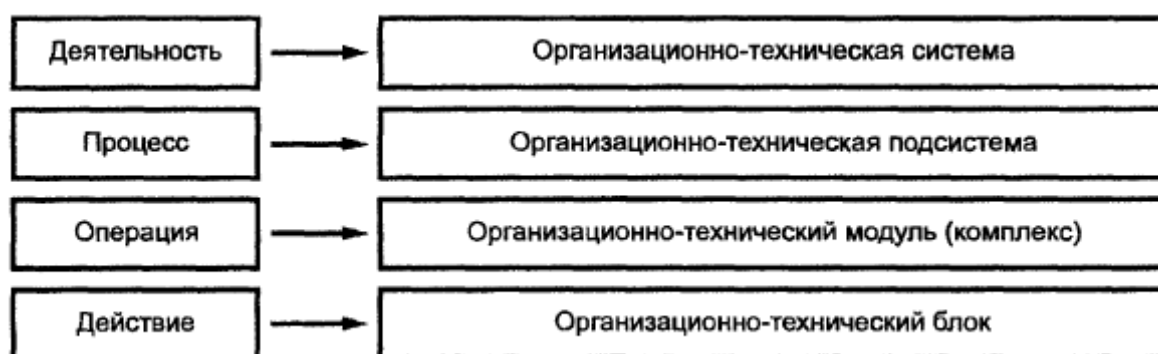


Рис. 1.2

Элементы иерархии механизмов определяются следующим образом.

Организационно-техническая система — организационная структура, персонал и комплекс технических средств (оборудование), необходимые для осуществления деятельности.

Организационно-техническая подсистема — часть организационно-технической системы, обеспечивающая протекание процесса (субдеятельности).

Организационно-технический комплекс (модуль) — часть организационно-технической подсистемы, предназначенная для выполнения операции.

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

Таким образом, при корректном построении модели в стандарте IDEF0 (без априорной привязки к «организации») появляется возможность связать ее блоки на разных уровнях декомпозиции с объектами организационно-технической структуры, выступающими в качестве механизмов. В этом случае организационно-техническая структура становится результатом функционального моделирования.

Во многих IDEF0-моделях находит или должно находить отражение явление, состоящее в формировании или специфической настройке (перестройке) механизмов в ходе деятельности. Это явление часто именуется реинжинирингом производства и/или бизнес-процессов на предприятии (в организации).

Это явление отражается в модели как субъектность, поскольку почти всегда состоит из нескольких процессов.

Укрупненная схема этой субъектности приведена на рис.1.3.



Э — энергия, П — персонал, О — оборудование,
Ф — финансы

Рис. 1.3

Согласно схеме входом и одновременно потребляемым ресурсом субъектности являются финансы, преобразуемые в другие виды ресурсов — энергетические, трудовые, материальные (оборудование, вспомогательные материалы и т.д.).

Механизм любого уровня обеспечивает выполнение деятельности (процесса, операции, действия), потребляя ресурсы: финансовые, энергетические, трудовые, непосредственно или с помощью промежуточных преобразований (рис.1.3), т.е. специфических процессов, которые можно назвать поддерживающими, обеспечивающими или вспомогательными (по аналогии с вспомогательными производствами, цехами, участками на машиностроительном предприятии) по отношению к основным процессам, где происходят преобразования, однозначно обусловленные целью

деятельности.

Существенным признаком вспомогательного процесса является то, что этот процесс не создает конечного продукта деятельности и, следовательно, прибыли.

Один из общих принципов методологии IDEF0 требует, чтобы к каждому блоку на диаграмме была присоединена хотя бы одна *управляющая стрелка*, отображающая условия правильного функционирования блока (см. Приложение 2).

Ниже сформулирован ряд определений и методических положений, которыми следует руководствоваться при отражении управлений на функциональных моделях.

Управление деятельностью — это процесс, состоящий как минимум из следующих операций:

- формулирование целей деятельности;
- оценивание ресурсов, необходимых для осуществления деятельности и их сопоставление с имеющимися ресурсами;
- сбор информации об условиях протекания и фактическом состоянии деятельности («глобальная» обратная связь);
- 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 решений (исполнение директив) и оценка их результатов («локальная обратная связь»);
- корректировка (в случае необходимости, например при нехватке ресурсов) ранее сформулированных целей (самонастройка, адаптация).

Управление процессом — это операция, состоящая как минимум из следующих действий:

- анализ директивы на управление процессом, ее декомпозиция на директивы управления операциями;
- сбор (прием по каналам связи) 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 данными директив и выработка локальных решений, направленных на устранение отклонений;
- корректировка (в случае необходимости) директив на выполнение операций.

Управление операцией — это действие, состоящее в выработке на основании директивы на управление операцией команд на управление действиями, в реализации этих команд, оценке результатов выполнения, передаче необходимой информации в комплекс управления процессом, корректировке команд в случае необходимости.

Блоки управления должны присутствовать на каждой IDEF0-диаграмме

(кроме тех, которые являются декомпозициями самих таких блоков). Через них осуществляются управляющие воздействия на остальные блоки диаграммы. Именно эти блоки воспринимают ограничивающую и предписывающую информацию и преобразуют ее в соответствующие директивы и команды. Имена таких блоков управления, как правило, содержат глагол «*Управлять ...*».

Стрелки, исходящие из блока с именем «*Управлять ...*», описывают централизованную схему управления - управленческую «вертикаль».

Возможны варианты структур, в которых выходная информация одного из блоков является управляющей для другого. Это отображает децентрализацию управления, т.е. «горизонтальные» связи присутствующие в моделируемой системе.

1.2.3. Типизация функциональных моделей и IDEF0-диаграмм

Эффективность и производительность труда разработчиков функциональных моделей могут быть повышены за счет применения типовых моделей и отдельных диаграмм, ориентированных на применение в конкретных предметных областях. Так, например, на основе представлений о жизненном цикле продукции (изделия) можно предложить типовую диаграмму уровня А0 для промышленного предприятия, которая может иметь вид, схематически показанный на рис.1.4.

Аналогичные типовые модели могут быть разработаны для других видов бизнеса (оказание услуг, транспорт, банковское дело, финансовая деятельность и т.д.).

1.2.4. Последовательность построения IDEF0-моделей

Основные этапы. Прежде чем начать моделирование, *аналитик* проводит

1. подготовку к моделированию;
2. собирает информацию по моделируемому объекту;
3. декомпозирует моделируемый объект;
4. обобщает полученную декомпозицию моделируемого объекта.

Подготовка к моделированию включает в себя:

- выбор цели модели (например, описание того, как механический цех производит детали);
- выбор точки зрения, с которой будет представлена модель (например, мастер, рабочий);
- тип создаваемой модели (например, модель "потокowego" процесса);
- предполагаемое использование построенной и проверенной модели (например, подготовить нового оператора).

Таким образом, подготовка к моделированию должна максимально облегчить сбор информации.

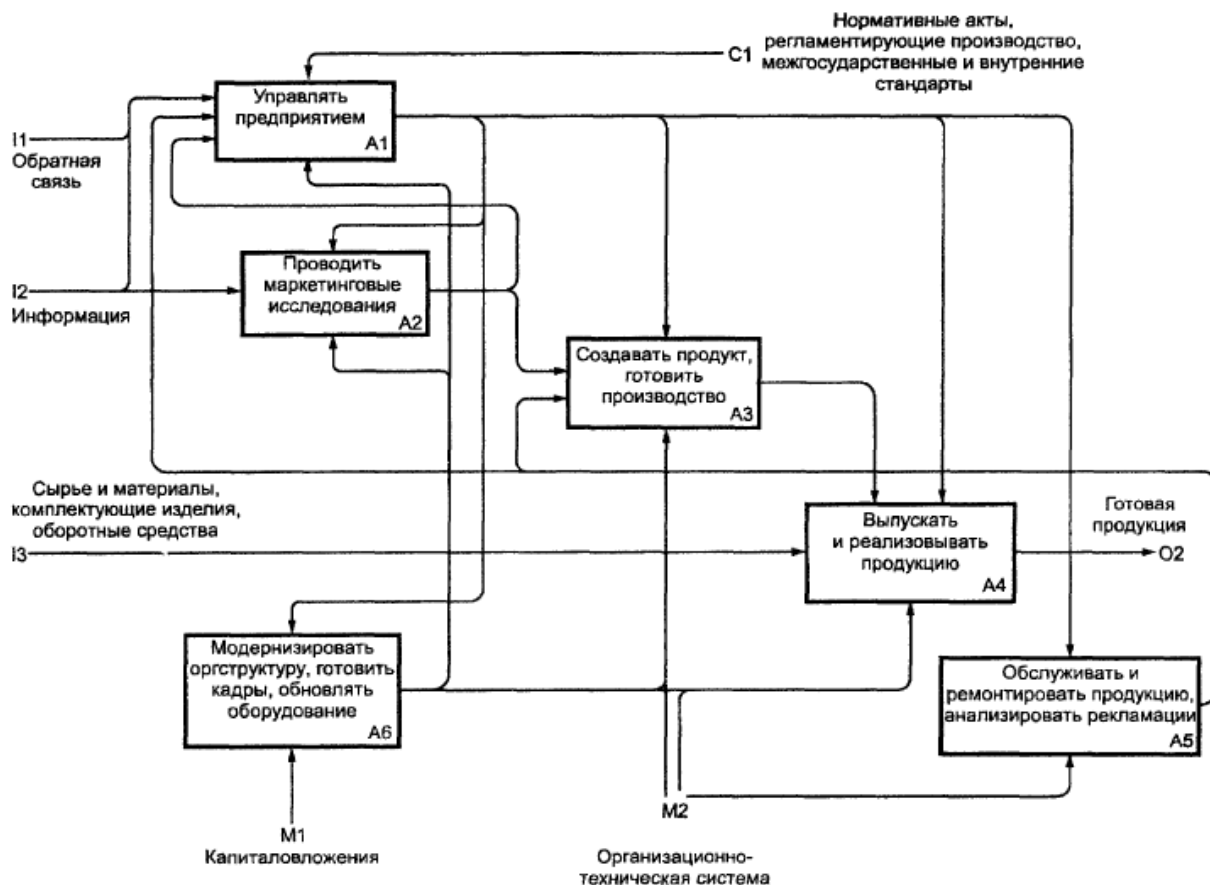


Рис.1.4

Сбор информации по моделируемому объекту может включать любую комбинацию следующих видов деятельности:

- чтение документов;
- наблюдение за существующими операциями;
- анкетирование группы экспертов;
- опрос одного или нескольких экспертов;
- использование собственных знаний и придуманного описания работы системы, которое впоследствии может быть откорректировано.

Декомпозируя моделируемый объект, нужно, прежде всего, обратить внимание на входные и выходные данные для всей системы.

Декомпозиция всей системы/объекта начинается с составления списка основных типов данных и основных функций. При этом необходимо выявить основные функции системы/объекта, рассматривая все нормальные и аномальные ситуации, обратные связи и случаи потенциальных ошибок. Далее сформированные списки основных функций системы/объекта снабжаются комментариями для указания основных типов, как данных, так и функций системы или их различных сочетаний. Наконец, сформированные списки с комментариями используются для создания диаграммы A0, которая затем обобщается с помощью диаграммы A-0.

Выбор цели и точки зрения. Цель и точка зрения модели определяются на самой ранней стадии создания модели.

Выбор цели осуществляется с учетом вопросов, на которые должна ответить модель, а выбор точки зрения - в соответствии с выбором позиции, с которой описывается система.

Иногда цель и точку зрения можно выбрать до того, как будет сделана первая диаграмма.

Например, цель модели экспериментального механического цеха можно определить заранее, потому что она очевидна в постановке задачи: "понять обязанности всех работающих в цехе так, чтобы объяснить их новому персоналу".

Настоятельно рекомендуется, как можно раньше определять цель и выбирать точку зрения новой модели.

На начальном этапе следует записать ряд специфических вопросов, на которые модель должна ответить, чтобы убедиться, что цель сформулирована точно, и постараться рассмотреть систему с нескольких различных точек зрения, прежде чем выбрать одну из них.

На рис. 1.5 показано, как это делается для задачи, связанной с экспериментальным механическим цехом.

ИСПОЛЬЗУЕТСЯ В:	АВТОР: Марса ПРОЕКТ: ЭМЦ ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10	ДАТА: 02/20/93 ПЕРЕСМОТР:	<input checked="" type="checkbox"/> РАБОЧАЯ ВЕРСИЯ <input type="checkbox"/> ЭСКИЗ <input type="checkbox"/> РЕКОМЕНДОВАНО <input type="checkbox"/> ПУБЛИКАЦИЯ	ЧИТАТЕЛЬ	ДАТА	КОНТЕКСТ: Тор
Вопросы:		Цель:				
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 деталями ?		Определить обязанности каждого работника экспериментального механического цеха и понять, как эти обязанности <u>взаимосвязаны</u> между собой с тем, чтобы написать учебное пособие.				
Претенденты :		2 Процесс обучения для различных типов работников требует декомпозиции в зависимости от обязанностей, которые выполняют эти работники в цехе (см. замечание N5 на диаграмме DAM001).				
Мастер Механик Контролер Начальник		Точка зрения: Начальника цеха 1 Только с этой точки зрения можно показать <u>взаимосвязи</u> между отдельными работами и обязанностями персонала.				
УЗЕЛ: ЭМЦ/А-0	НАЗВАНИЕ: Цель и точка зрения модели ЭМЦ	НОМЕР: DAM002				

Рис. 1.5. Определение целей и точки зрения

Достаточно часто складывается ситуация, когда определить цель и

точку зрения в самом начале моделирования чрезвычайно трудно.

В таком случае целесообразно составить списки данных и функций и, может быть, нарисовать диаграмму A0.

Сделав это, можно «почувствовать» систему и установить, описывает ли ее диаграмма A0 с нужной точки зрения.

Возможно, что придется нарисовать несколько альтернативных A0-диаграмм, прежде чем появится достаточная уверенность для того, чтобы осуществить выбор правильной цели и точки зрения.

Составление списка данных. Списки объектов системы, создаваемые в ходе моделирования, в стандарте IDEF0 принято называть «списками данных».

Термин «данное» здесь употребляется как синоним слова «объект». Поэтому, при обсуждении различных аспектов IDEF0 моделирования, как правило, применяется термин «список данных».

Составление списка данных является начальным этапом создания каждой диаграммы функциональной IDEF0-модели.

Правило заключается в том, чтобы вначале составить список данных, а потом список функций.

Формирование диаграммы начинается с выделения всех основных групп и категорий данных, используемых и генерируемых системой. При этом записывают все разумные возможности. При возникновении сомнения рекомендуется записывать все, что приходит на ум, потому что лучше записать слишком много, чем провести неполный анализ.

Так на рис. 1.6 в список вошло много деталей, хотя аналитик пытался создать диаграмму цеха как единого целого.

В современных аналитических методах слишком часто уделяется повышенное внимание функциям в ущерб данным.

Начиная с составления списка данных, можно избежать перехода к немедленной функциональной декомпозиции.

Списки данных помогают выполнить более глубокий анализ и при этом не концентрироваться на функциях системы, избегая пробелов, которые часто возникают из предвзятых представлений о функциональных декомпозициях.

IDEF0-диаграммы представляют *границы функций и ограничения, накладываемые на них, причем ограничения должны присутствовать во всех системах.*

Указывая вначале ограничения, выявляют естественную структуру анализируемой системы/объекта.

Без ограничений функциональная IDEF0-диаграмма представляет собой не более чем схему потоков данных.

Без ограничительных дуг диаграммы не смогут рассказать ее читателю, почему аналитик выбрал именно данную декомпозицию.

Благодаря тому, что в IDEF0 различаются входные дуги и дуги управления (информация, необходимая для пояснения декомпозиции),

IDEF0-диаграммы ясно объясняют изучаемую систему и причину такой декомпозиции.

Составление списка функций. После формирования списка данных, приступают с его помощью к составлению списка функций.

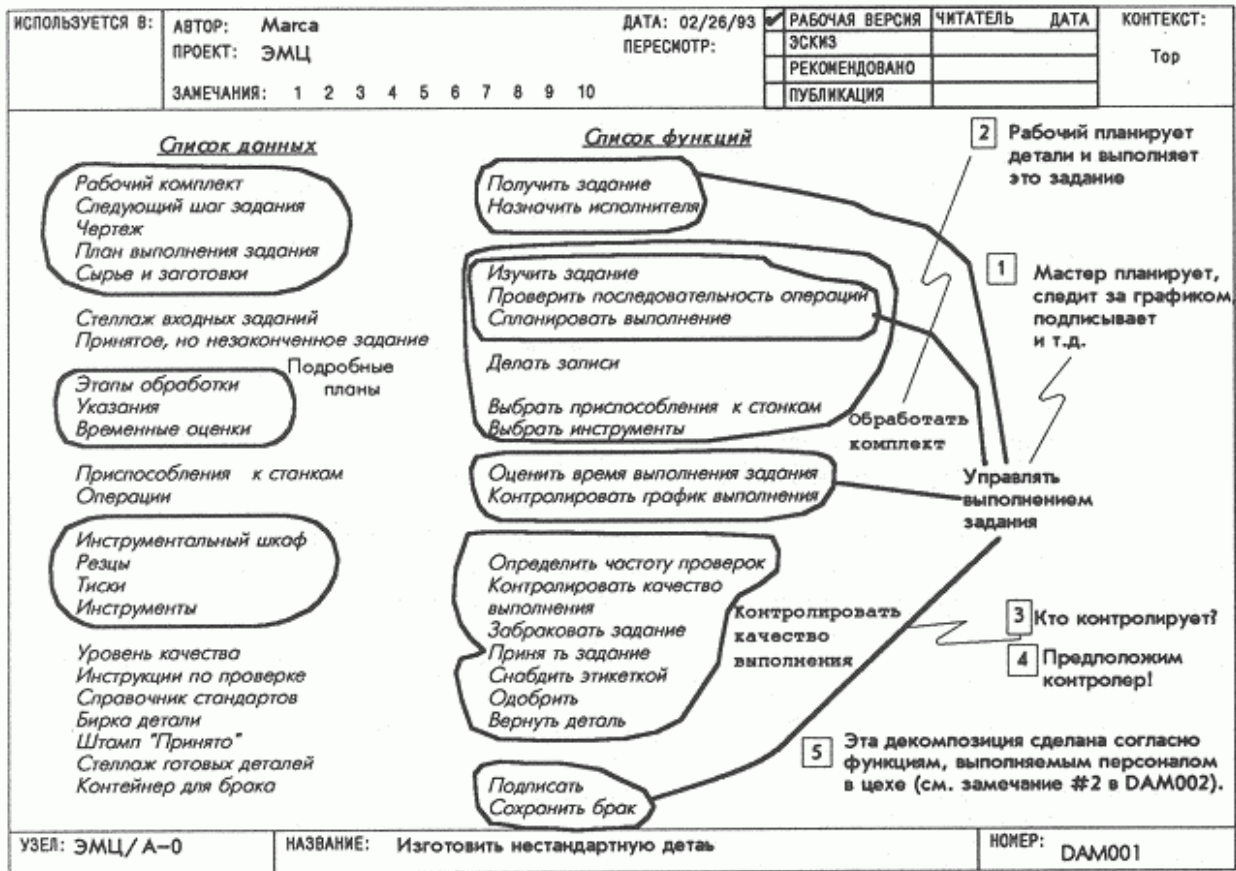


Рис. 1.6. Подготовка списка функций и списка данных

Для этого представляют себе функции системы, использующие тот или иной класс (тип) или набор данных.

При этом необходимо помнить, что несколько различных типов данных может использоваться одной функцией.

Следует обозначить, какие типы или наборы данных необходимы для каждой конкретной функции. Это позволит выделить *данные сходных типов, которые затем можно объединить в метатипы.*

По мере продвижения по списку, проверяют, верны ли первоначальные представления, которые часто могут не совпадать с выбранной целью и точкой зрения модели.

С другой стороны, не следует автоматически отвергать первоначальные идеи, если они кажутся неверными. Дальнейшие размышления могут прояснить внутренние аспекты работы системы, не очевидные при первом взгляде, и вы, возможно, восстановите исходные идеи после построения нескольких других диаграмм.

Список функций должен находиться на одной странице со списком данных.

При составлении исходного списка не стоит пытаться объединять функции между собой.

Вместо этого следует вначале сосредоточиться на каждой конкретной функции и ее отношении к группам данных. Кроме того, необходимо подбирать такие функции, которые могли бы работать с наиболее общими типами данных из сформированного списка. Так на рис. 1.6 сделано достаточно много объединений данных для того, чтобы собрать несколько детальных функций в одну более общую

Не следует слишком досконально записывать функции, даже если возникают сомнения в том, что их выполняет рассматриваемая система. Применительно к пограничным функциям (функциям, которые могут выполняться либо системой, либо ее окружением), то сначала достаточно сложно определить, входят они в модель или нет.

Далее функции объединяют в «агрегаты».

Следует стремиться к организации 3-6 функциональных группировок одного и того же уровня сложности, содержащие примерно одинаковый «объем» функциональности, при этом функции в каждой из них должны иметь сходные операции и цели.

На рис. 1.6 видно, что *исходный список функций сгруппирован в три функции более высокого уровня.*

Объединение подобного вида не всегда легко осуществить. Возможно обнаружится, что на каком-то уровне модели трудно выбрать «наилучший» способ объединения функций. Но неудачная группировка функций проявится на этапе декомпозиции предварительно сформированной модели и тогда необходимо вернуться назад и попробовать сформировать другой вариант объединения выявленных функций.

Построение диаграммы А0. Исходное содержание диаграммы А0 обеспечивают списки данных и функций.

Для правильного описания системы содержанию надо придать определенную форму. В стандарте IDEF0 это делается посредством построения диаграммы.

При построении диаграммы следует придерживаться определенного порядка:

- расположить блоки на странице;
- нарисовать основные дуги, представляющие ограничения;
- нарисовать внешние дуги;
- нарисовать все оставшиеся дуги.

1. Правильное расположение блоков является ключевым этапом построения диаграммы.

Блоки располагаются в соответствии с их доминированием, т.е. по

степени важности или по порядку следования.

Самый доминантный блок обычно располагается в верхнем левом углу, а наименее доминантный - в нижнем правом.

Это приводит к расположению, при котором более доминантные блоки ограничивают менее доминантные, образуя "ступенчатую" схему.

Доминирование имеет важнейшее значение для ясного представления процесса. Например, не имеет смысла говорить о контроле за выполнением задания до изготовления детали.

2. *Изображение основных дуг, представляющих ограничения, является второй важной частью построения диаграммы АО. Они (дуги) дают основание для разбиения объекта диаграммы на 3 - 6 системных функций, изображаемых блоками. Например, справочник стандартов качества оказывает решающее влияние на то, как контролируются незаконченные детали.*

Изображая эти дуги, проверяют, действительно ли каждая из них оказывает влияние, соответствующее декомпозиции объекта.

Необходимо проследить по списку данных, не отсутствуют ли какие-то дуги, представляющие ограничения. Если это так, то, возможно, необходимо проверить правильность проводимой декомпозиции.

Основными дугами, представляющими ограничения, всегда являются внешние дуги, т.е. дуги, представляющие данные, поступающие из непосредственного окружения диаграммы.

3. *Размещение остальных внешних дуг и назначение им соответствующих ИСОМ-кодов.*

В итоге, все данные, входящие в систему или выходящие из нее, оказываются учтенными на рисунке.

Потеря внешней дуги - это ошибка интерфейса, одна из самых распространенных в системном анализе.

Занимаясь декомпозицией объекта, можно забыть об интерфейсных данных, потому что очень легко сосредоточиться на деталях формируемой диаграммы. Если начинать построение диаграммы с изображения всех внешних дуг, то это повышает точность диаграммы, за счет включения в нее всех интерфейсных данных.

4. *Изобразите все остальные дуги, отражающие детали работы системы в целом.*

Сначала, следует изобразить все оставшиеся ограничения, действующие между блоками. Например, в рассматриваемом примере моделирования функционирования экспериментального механического цеха *чертеж влияет на проверку детали.*

Затем, изобразите основной поток данных.

На рис. 1.7 показана обработка сырья и заготовок в соответствии с *планом выполнения задания и контроль качества выполнения задания* (иногда неоднократный) и в соответствии с *чертежом*.

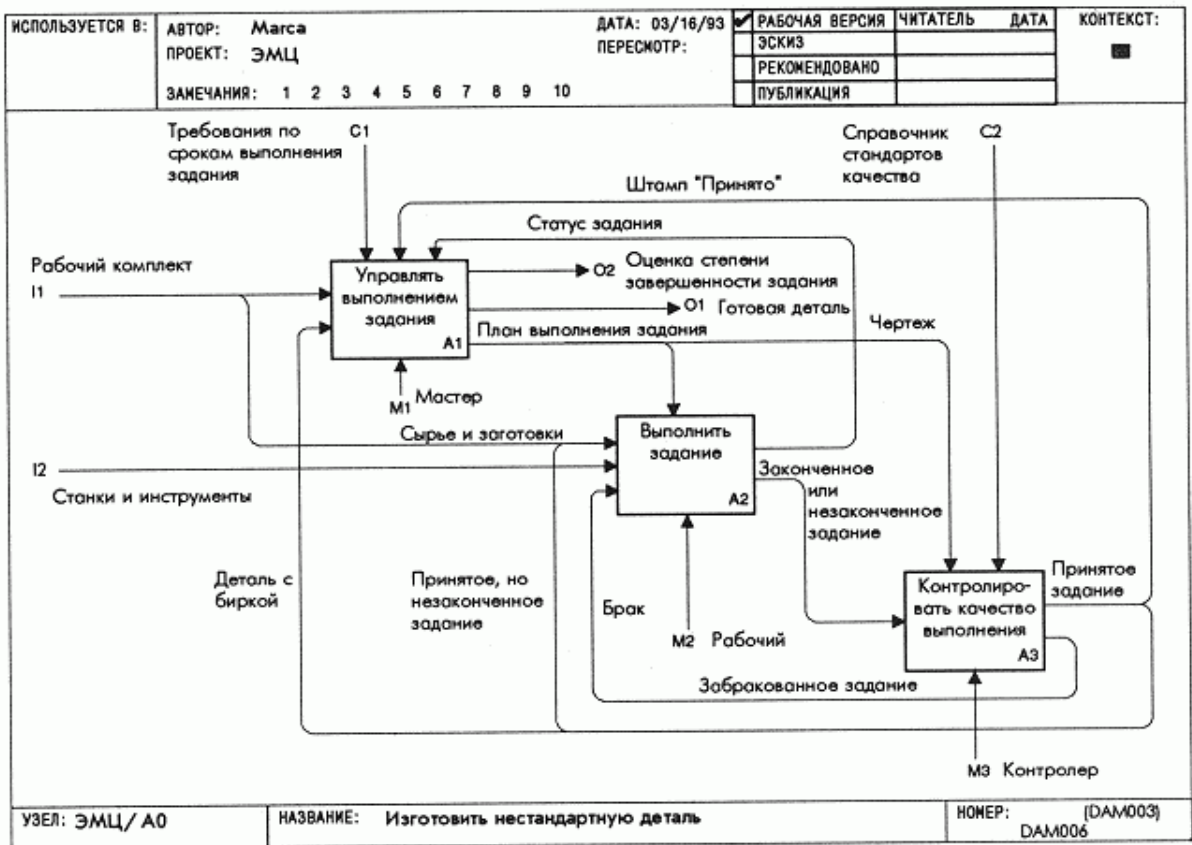


Рис. 1.7. Диаграмма А0

Далее необходимо рассмотреть все «патологические» потоки данных, которые могут привести к возникновению ошибок.

И наконец, следует уточнить существование обратных связей в выявленных потоках данных. Например, *забракованное задание* снова попадает в цикл в качестве *брака*.

В заключение следует изобразить все обратные связи, вызываемые ошибочными ситуациями.

Здесь следует обратиться к одному очень важному моменту моделирования.

На практике оказывается невозможным нарисовать диаграмму сразу набело. Для того чтобы придать некоторую форму данным и функциям, лучше всего сделать набросок (черновик). В процессе работы с черновиком, ситуация начинает проясняться. При этом достаточно часто приходится переименовывать дуги и блоки, зачеркивать дуги, перемещать блоки.

Поэтому рекомендуется вначале делать набросок диаграммы, а потом перерисовывать диаграмму набело, чтобы уточнить свое понимание, прояснить ситуацию и создать описание, которое могут посмотреть другие.

Обобщение диаграммы А0. *Обобщение является последним важным шагом начального этапа моделирования.*

Следует вспомнить, что для любой IDEF-диаграммы есть родительская

диаграмма, содержащая ее контекст, где под контекстом понимается блок с набором входных дуг, дуг управления и выходных дуг.

Верхняя диаграмма модели (т.е. диаграмма А0) не является исключением.

Контекстом для нее служит диаграмма А-0, представляющая собой обобщение всей модели.

Диаграмма А-0 имеет несколько предназначений.

Во-первых, она объявляет общую функцию всей системы.

Например, блок на рис. 1.8 с названием «Изготовить нестандартную деталь» ясно указывает, что делает цех.

Во-вторых, она (диаграмма А-0) дает множество основных типов или наборов данных, которые использует или производит система. Например, «справочник стандартов качества» позволяет осуществлять «контроль качества» при выполнении задания.

В-третьих, диаграмма А-0 указывает взаимоотношения между основными типами данных, проводя их разграничение. Например, «рабочий комплект» рассматривается как входное данное, нечто, изменяемое процессом, в то время как «справочник стандартов качества» контролирует «выполнение цехом заданий».

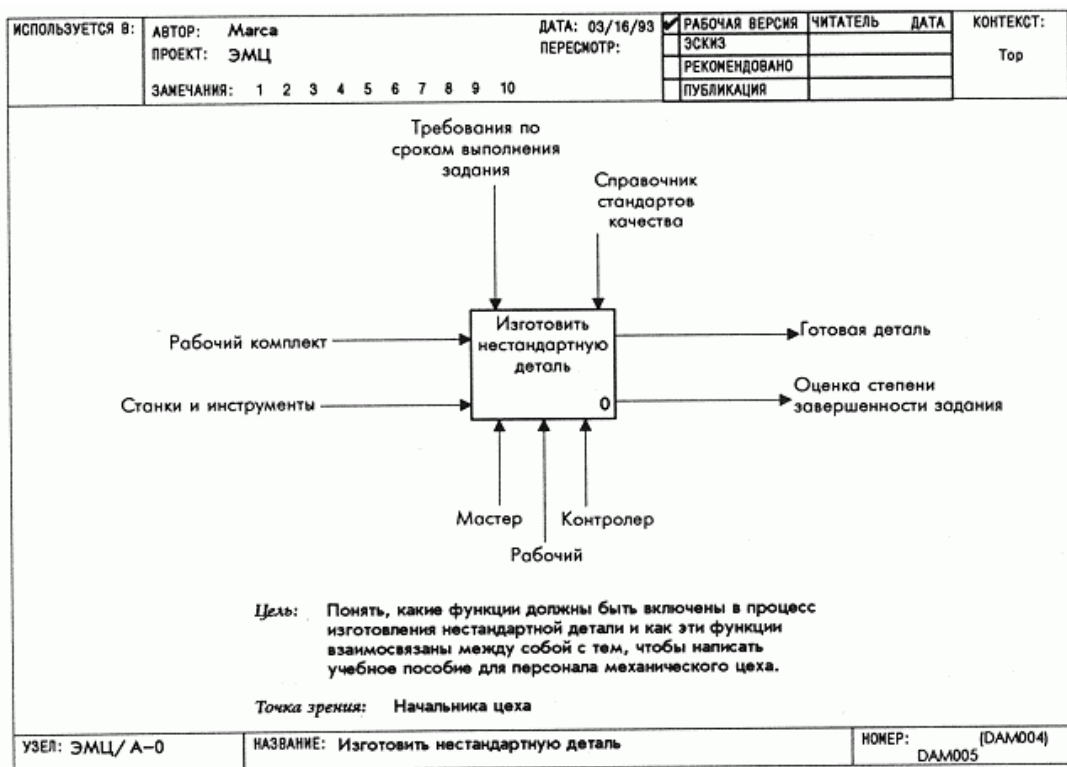


Рис. 1.8. Диаграмма А-0

Таким образом, диаграмма А-0 представляет собой общий вид изучаемой системы.

При создании диаграммы А-0 используется информация, уже зафиксированная на диаграмме А0.

Вначале в центре IDEF-бланка рисуют один большой блок, название которого совпадает с названием диаграммы A0. В этот момент следует проверить, адекватно ли название отражает то, что делает система. Все внешние дуги диаграммы A0 изображаются на диаграмме A-0 входящими в соответствующую сторону блока. При этом необходимо проверить, чтобы название каждой дуги описывает именно то, чем обмениваются моделируемая система и ее внешняя среда.

После того как изображены входные и выходные дуги следует проверить точность описания потока данных. После изображения дуг управления следует убедиться, что именно они управляют тем, как система преобразует входные данные в выходные.

В заключение, необходимо написать цель и точку зрения модели под основным блоком и сверить их с тем, что представляется блоком и его дугами.

В процессе обобщения можно убедиться в том, что он помогает прояснить описание системы, потому что при обобщении просматривают метки дуг для более точного наименования данных, которыми обмениваются система и ее внешняя среда.

Кроме того, во время обобщения дуги часто объединяются для упрощения изображения модели. В этом случае дуги разветвляются на свои составляющие на диаграмме A0. Например, на диаграмме A0, приведенной на рис. 1.8, указано, что сырье и заготовки входят в состав рабочего комплекта.

Построение диаграммы A-0 свидетельствует об окончании начального этапа моделирования.

К этому моменту сделана первая попытка обобщить и описать основную деятельность системы и показать связь системы с ее средой.

Несмотря на ограниченное число описанных деталей, диаграммы A-0 и A0 представляют законченную картину, потому что они отражают все основные входы, управления, выходы и функции системы.

Общий вид системы, полученный с помощью диаграмм A-0 и A0, - основная цель аналитика на начальном этапе построения IDEF-модели.

Начальный этап моделирования включает определение объекта, цели и точки зрения модели, ограничения, накладываемые на объект, построение диаграммы верхнего уровня и ее обобщение, составление списков данных и функций, объединение функций в блоки, формирование с использованием списка данных взаимоотношений между блоками.

Продолжение моделирования основывается на тех же методах и выводит модель на следующий уровень детализации.

Для этого требуется создать отдельную диаграмму для, возможно, каждого блока диаграммы верхнего уровня, затем построить диаграммы для всех блоков новых диаграмм, и так до тех пор, пока модель не будет описывать объект с нужной для достижения цели степенью детализации.

Таким образом, продолжение моделирования является рекурсивным

процессом.

Декомпозиция ограниченного объекта. *Декомпозиция модели похожа на начальный этап моделирования, но несколько проще его.*

Это обусловлено тем, что при декомпозиции модели аналитик всегда находится в контексте, определенном блоком со своими дугами одной из диаграмм.

Эта граница, называемая границей объекта, определена двумя способами.

Во-первых, объект, цель и точка зрения каждой новой диаграммы уже определены на диаграмме А0.

Во-вторых, каждый блок, декомпозируемый в новую диаграмму, уже является ограниченным объектом.

Другими словами, он идентифицирует конкретную функцию и все данные, которые для нее требуются или ею порождаются.

Строить диаграмму, исходя из этой информации, проще, потому что список данных создается на основе дуг, входящих в блок и выходящих из него, а также потому, что список функций подробно раскрывает название блока.

Процесс декомпозиции ограниченного объекта состоит из следующих шагов:

1. выбор блока диаграммы;
2. рассмотрение объекта, определенного этим блоком;
3. создание новой диаграммы;
4. выявление недостатков новой диаграммы;
5. создание альтернативных декомпозиции;
6. корректировка новой диаграммы;
7. корректировка всех связанных с ней диаграмм.

Шаги 1-3 представляют созидательную часть процесса. Выполняя их, аналитик концентрирует свои усилия, связанные с выявлением новой информации об объекте, на более высоком уровне детализации, чтобы достичь ясности изложения.

Шаги 4-7 составляют этап саморецензирования, в ходе которого аналитик, создав новую диаграмму, проверяет, какую она несет информацию и в каком она находится отношении с родительской диаграммой. Затем в созданную диаграмму и соответственно в связанные с ней диаграммы вносятся изменения, чтобы достичь ясности для других.

Рассмотрим более подробно шаги 1-3.

Шаг 1. Выбор блока. Декомпозиция начинается с чтения диаграммы А0 и определения самого *содержательного* блока.

Это такой блок, декомпозиция которого выявит многие аспекты диаграммы А0 и будет оказывать большое влияние на будущие декомпозиции других блоков этой диаграммы.

При выборе самого *содержательного* блока следует учитывать

доминирование, функциональную сложность и понятность.

Лучшим не обязательно будет блок, наиболее трудный для понимания.

Лучшим блоком для первой декомпозиции будет тот, который позволит наиболее глубоко проникнуть в суть рассматриваемой системы.

Рассмотрим диаграмму А0 для модели экспериментального механического цеха (рис. 1.9).

На ней три блока: «управлять выполнением задания», «выполнить задание» и «контролировать качество выполнения».

На первый взгляд, кажется, что блок «управлять выполнением задания» является самым сложным блоком, потому что он самый доминантный и с ним связано самое большое количество дуг.

Но при внимательном рассмотрении можно заметить, что многие из его дуг являются просто обратными связями или выходами.

Блок «контролировать качество выполнения», видимо, является простейшей функцией, так что ее декомпозиция не дает много новой информации.

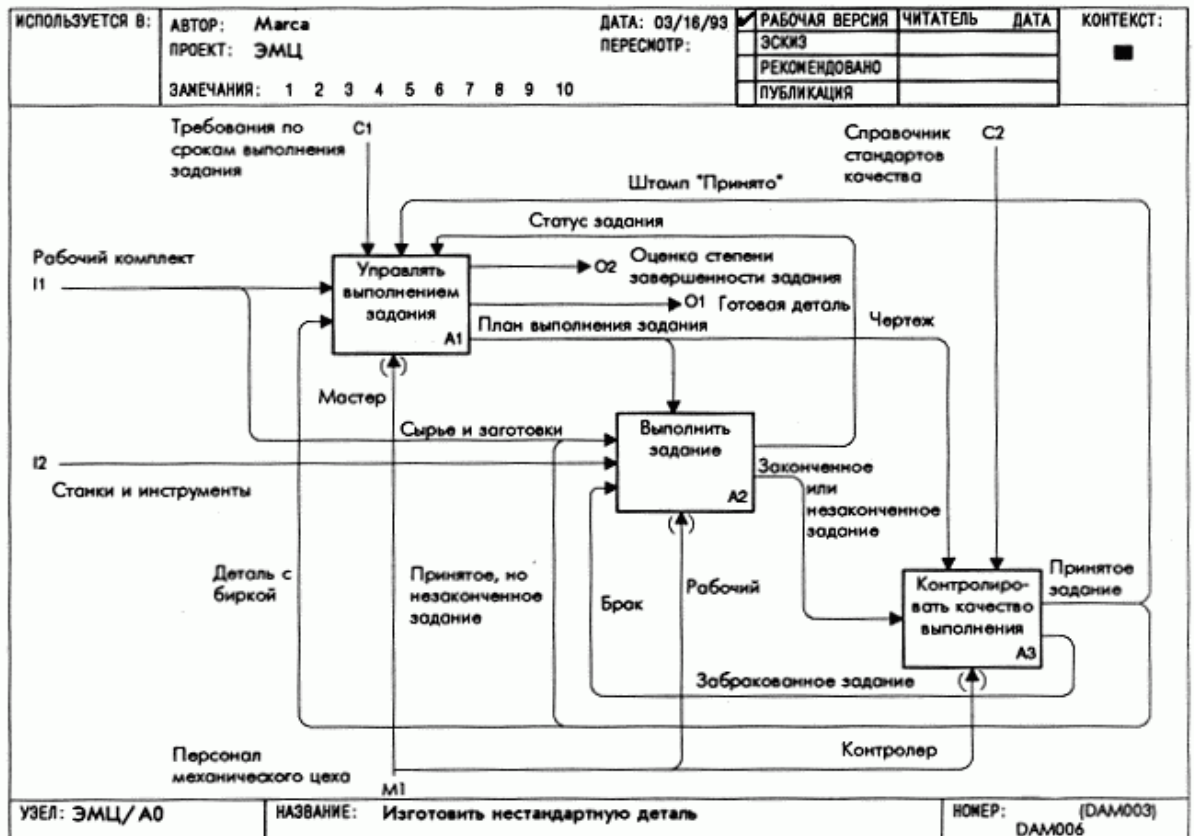


Рис. 1.9.

Поэтому эти два блока - плохие кандидаты для первой декомпозиции.

Блок «выполнить задание» выглядит более интересным, потому что этот блок участвует во многих циклах и имеет широкий спектр входов, управлений и выходов.

Поняв, как рабочий выполняет задание, аналитик, видимо, окажется в лучшем положении для дальнейшей декомпозиции остальной части модели.

Блок А2 «выполнить задание» становится теперь самостоятельным объектом декомпозиции.

Для выполнения этой декомпозиции вначале следует бегло осмотреть обобщающую диаграмму - диаграмму А-0 на рис. 1.8 и вспомнить цель и точку зрения модели.

Сделав это, можно увидеть, что необходимо описать блок «выполнить задание» с точки зрения начальника цеха, чтобы можно было разработать инструкции для обучения нового персонала цеха.

Кроме того, после изучения блока А2 диаграммы А0 и соединенные с ним дуги, можно выявить его особенности. Например, дуга механизма с названием «рабочий» указывает, что при декомпозиции этого блока можно выявить, чем занимаются рабочие.

Затем составляется список данных со всех дуг, касающихся этого блока, используя ИСОМ-кодирование для того, чтобы не потерять какие-либо интерфейсные данные.

Например, «план выполнения задания», «станки и инструменты», а также «брак» входят в начальный список данных (см. верхнюю левую колонку на рис. 1.10).

Этот список теперь улучшается благодаря декомпозиции первоначальных данных или введению новых, тесно связанных данных.

Например, при дальнейшем рассмотрении «плана выполнения задания» возникает мысль об «указаниях».

Шаг 2. Рассмотрение объекта, определенного этим блоком. Далее, на основе списка данных составляют список функций, придерживаясь функции, соответствующей блоку верхней диаграммы.

Следует обратить внимание на то, что «выбрать инструменты», «подготовить рабочее место», «обработать на станке» и «собрать и определить степень выполнения задания», по-видимому, действительно являются функциями, выполняемыми рабочим при выполнении задания.

Необходимо стремиться ограничиваться разумным уровнем сложности при объединении функций и данных: четыре-пять функциональных блоков, как правило, лучше всего.

Слишком много данных и функций часто содержат слишком много информации. Это приводит к запутанным диаграммам.

Наоборот, небольшое число блоков дает слишком мало, и диаграмма становится почти бесполезной.

Если аналитик уверен, что достиг баланса, следует проверить, во всех ли отношениях написанные слова адекватны объекту, определенному блоком и его граничными дугами на родительской диаграмме.

Теперь имеется всё необходимое для построения диаграммы.

Шаг 3. Создание новой диаграммы. Новая диаграмма строится аналогично диаграммам А0 и А-0.

Блоки размещаются в соответствии с доминированием (т.е. согласно взаимным ограничениям блоков), затем создаются основные дуги, представляющие ограничения, потом внешние и, наконец, внутренние дуги.

Вообще говоря, IDEF-диаграммы строятся в соответствии с той информацией, которую они несут. На рис. 1.10 показан результат работы аналитика, который сделал набросок диаграммы ограниченного объекта.

Вот некоторые важные моменты этой диаграммы: дуга следующего шага задания является носителем очень существенной информации - она определяет, что нужно сделать на следующем шаге задания.

Следовательно, поскольку она ограничивает все остальные функции, блок «определить степень выполнения», порождающий дугу следующий шаг задания, является самым доминантным на этой диаграмме.

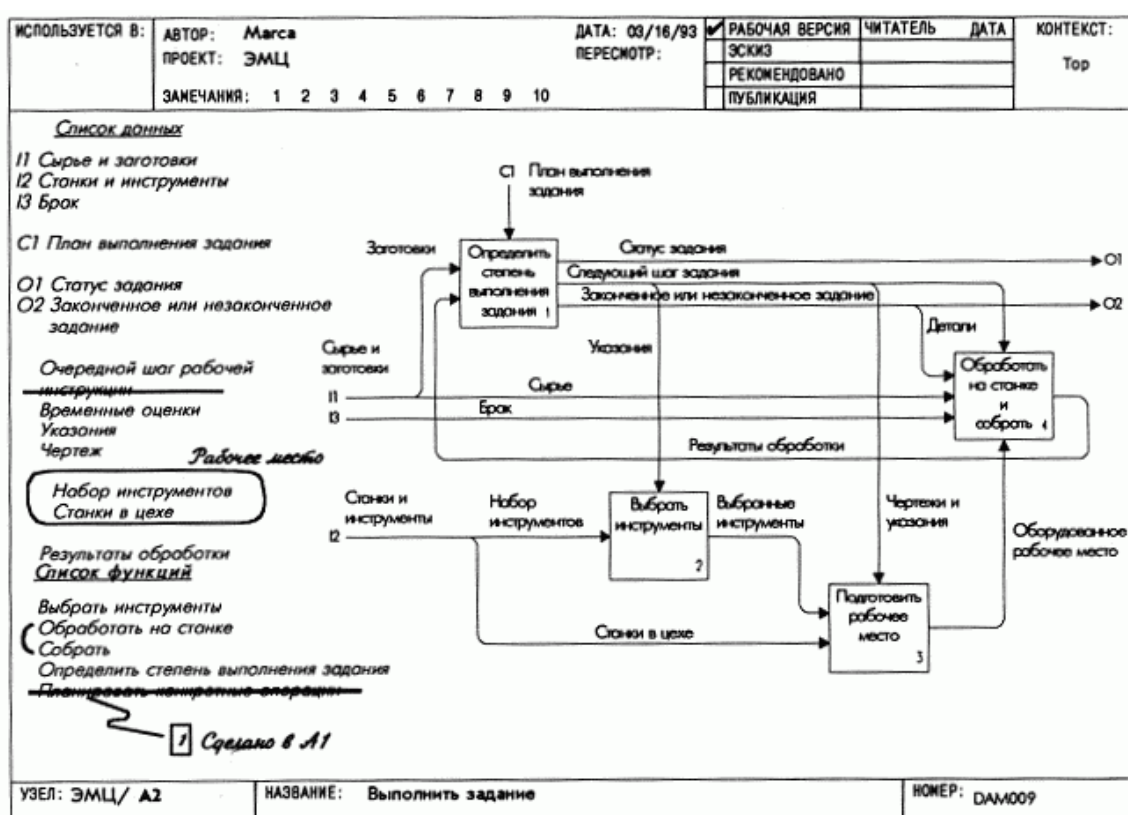


Рис. 1.10. Предварительные наброски для декомпозиции функционального блока

«План выполнения задания» (C1), очевидно, требуется прежде, чем что-нибудь может произойти, потому что содержимое этой внешней дуги определяет последовательность шагов обработки.

Поэтому управляющая дуга очень важна.

Кроме того, аналитик стремился отразить на диаграмме потоки информации, в особенности потоки информации обратной связи.

Необходимо обратить внимание на то, что «результаты обработки» есть обратный вход от блока «*обработать на станке и собрать*» к блоку

«определить степень выполнения задания».

Это означает: *результат обработки есть незаконченное задание, получаемое на каждом шаге выполнения задания, что в полной мере соответствует действительности.*

Выявление интерфейсных ошибок. В IDEF-моделях сбои часто происходят в точках интерфейса. Для IDEF0 интерфейсными являются места соединения диаграмм со своими родителями. Поэтому каждую декомпозицию необходимо аккуратно соединять со своим родителем, используя ICOM-метки.

На примере декомпозиции в этой главе показан один из способов реализации этого подхода: прежде чем составлять список данных, необходимо записать имена и ICOM-коды для всех дуг, образующих границу. Это поможет при декомпозиции уменьшить вероятность пропуска части граничных дуг.

Выполнив декомпозицию, возвращаются назад к исходному блоку родительской диаграммы и соединяют каждую внешнюю дугу новой диаграммы с соответствующей дугой, касающейся этого блока. Это позволит избежать пропуска необходимого соединения.

Принципы и приемы расположения дуг. Дуги выражают связи между блоками. Их вычерчивают не для показа последовательности действий. Они отражают отношения между блоками, независимые от потенциального следования.

Например, диаграмма на рис. 1.10 не указывает возможной последовательности действий. В частности, функции выбрать инструменты может как встретиться, так и не встретиться перед функцией подготовить рабочее место. Это зависит от того, каким является следующий шаг задания. Если же оборудованное рабочее место уже создано, блок обработать на станке и собрать может выполняться неоднократно, пока не возникнет потребность в выполнении следующего шага задания. Такой механизм приводит к реализации различных сценариев, активизируя блоки в различные моменты времени в зависимости от ситуации.

Может случиться, что, начав рисовать дугу, возникнут сомнения в том, что она нужна на диаграмме. Но, не изобразив дугу при первом наброске диаграммы, возникает риск совершить ошибочный пропуск. Поскольку IDEF-диаграммы всегда проверяются другими специалистами и поскольку отсутствующую дугу никто не сможет обнаружить, изъятие сомнительной дуги с диаграммы - прямая дорога к возникновению ошибок и отсечению дополнительных возможностей.

Рекомендуется включать сомнительные дуги в диаграмму с вопросом об их необходимости адресованным читателям. Полученные в ответ комментарии, скорее всего, помогут разрешить возникшие сомнения. IDEF - дуги представляют собой наборы объектов и поэтому они потенциально могут нести много данных.

Например, сырье может быть просто стальным бруском, а может быть

листами, рулонами, кусками дерева, пластика или различных металлов.

Опытные IDEF - аналитики не изображают каждый объект отдельной дугой. В крайнем случае, целесообразно давать определение новой дуги в глоссарии и, возможно, уточнять ее содержимое при декомпозиции тех блоков, которых она касается. Таким образом, предоставляется читателям достаточно информации для понимания сформированной диаграммы и не тратится много времени на избыточно подробное описание модели на слишком ранней стадии, что неизбежно привело бы в дальнейшем к большим переделкам.

2. Разработка моделей с использованием стандарта IDEF1X

IDEF1X (Integration Definition for Information Modeling) - это методология семантического моделирования данных.

Данная методология разработана с учетом следующих требований:

1. Поддерживает разработку концептуальных схем

Синтаксис IDEF1X поддерживает семантические конструкции, необходимые для разработки концептуальной схемы. Окончательная версия IDEF1X-модели обладает желаемыми характеристиками - непротиворечивостью, расширяемостью и адаптируемостью.

2. Обеспечивает ясный язык

IDEF1X имеет простую, ясную, непротиворечивую структуру и четкие семантические понятия. Синтаксис и семантика IDEF1X сравнительно легки для понимания, хотя и являются достаточно мощным средством.

3. Проста для изучения

Семантическое моделирование данных - новое понятие для многих пользователей IDEF1X. Проблема обучаемости этому языку является важным фактором. Язык рассчитан на понимание и использование как профессиональными бизнесменами и системными аналитиками, так и администраторами данных и разработчиками баз данных. Он может служить эффективным средством коммуникации в коллективах, состоящих из различных специалистов.

4. Возможность автоматизации

IDEF1X-диаграммы могут создаваться большим числом графических программных пакетов. Существует также коммерческое программное обеспечение, поддерживающее детализацию, анализ и управление конфигурацией IDEF1X-моделей.

IDEF1X использует подход сущностей-отношений к семантическому моделированию данных.

Исходная разработка IDEF1 заключалась в расширении понятий сущности-отношения по методу П. Ченна, объединенных с понятиями реляционной теории Т. Кодда. Кроме того, для улучшения графического представления и процедур моделирования IDEF1X-методология семантически обогащена введением отношений категоризации (называемых также отношениями обобщения).

Язык IDEF1X включает коммерческие разработки D.Appleton Company и The Database Design Group.

Основными конструкциями IDEF1X-модели являются:

1. Предметы, к которым относятся данные, т.е. люди, места, идеи, события и т.д. Они изображаются блоками.
2. Отношения между этими предметами, изображаемые соединяющими блоки линиями.
3. Характеристики этих предметов, изображаемые именами атрибутов внутри блоков.

Разработка модели в стандарте IDEF1X включает следующие этапы:

1. Планирование проекта.
2. Сбор данных.
3. Определение сущностей.
3. Определение отношений между сущностями.
5. Определение ключевых атрибутов (ключей).
6. Определение неключевых атрибутов (остальных атрибутов сущностей).
7. Проверка правильности модели.
8. Приемка модели

2.1. Планирование и подготовка работы над проектом/моделью

Стадия подготовки работы над проектом/моделью. IDEF1X-модель данных должна быть описана и определена в терминах как ее ограничений, так и целей.

Разработчик модели является одним из главных лиц при определении назначения модели. Вместе с руководителем проекта разработчик модели составляет план для достижения цели на стадии подготовки работы над проектом/моделью, которую иногда называют «стадией 0».

Эта цель включает:

- Определение проекта - общая формулировка того, что должно быть сделано, почему и как это будет достигнуто.
- Исходный материал - план сбора исходной информации, ее регистрация и структуризация.
- Авторские соглашения - выбор основополагающих соглашений (методов), используемых автором при построении модели.

Реализация этого плана наряду с другой описательной и пояснительной информацией является результатом работы на стадии 0.

Определение цели моделирования

Установление цели моделирования охватывает два аспекта:

- Определение направленности - утверждение охватываемых моделью вопросов, т.е. контекстуальных рамок.
- Определение области действия - утверждение функциональных границ модели.

Одним из основных вопросов, на который при установлении цели моделирования должен быть дан ответ, является вопрос о временном интервале для модели: будет ли это модель существующей ситуации (т.е. "как есть" - модель) или модель того, что произойдет в результате задуманных изменений (т.е. "что должно быть" - модель).

Формальное описание проблемной области IDEF1X-проекта может включать обзор, построение, модификацию и выработку одной или нескольких (функциональных) IDEF0-моделей. По этой причине и автор, и руководитель проекта должны обладать некоторым опытом разработки и

использования IDEF0-моделей. Обычно IDEF0-модель уже существует и может служить основой для описания проблемной области.

Хотя целью моделирования данных является установление объективной картины основной инфраструктуры данных всего предприятия, для каждой модели важно определить конкретную *область действия*. Это поможет выявить данные, представляющие особый интерес.

Область действия может быть связана с типом пользователя (например, покупатель или проектировщик), деловой функцией (например, выпуск технического чертежа или планирование организации цеха), типом данных (например, параметры конфигурации продукции или финансовая информация). Утверждение *области действия* и направленности модели определяет цель моделирования.

Приведем пример цели моделирования:

"Целью данной модели является определение текущих данных, используемых начальником производственного подразделения, для производства и проверки, сложных авиационных деталей".

Хотя *область действия* может быть ограничена одним типом пользователя, в процесс моделирования должны быть включены и другие пользователи для обеспечения объективной точки зрения.

2.2. Сбор данных

Одной из первых проблем, возникающих перед разработчиком модели, является определение, какого рода материалы необходимо получить и из каких источников.

Нередко *область действия* и *контекст модели* устанавливаются на основе анализа функциональной IDEF0-модели. Анализ функций и связей между ними выявляет целевые функции в пределах представляемого функциональной моделью предприятия.

Узел целевой функции - это узел, представляющий совокупность информации, репрезентативной для данной проблемной области.

После выявления областей целевых функций и выбора интересующих категорий первичной информации могут быть выбраны участники процесса сбора данных в пределах этих функций.

Такой сбор данных может осуществляться несколькими способами, включая опрос людей, обладающих информацией, наблюдение за деятельностью, анализ документов, линий поведения и процедур, применение специфических, информационных моделей и т.п.

Это требует отображения узлов целевых функций в их эквивалент, соответствующий участникам моделирования. После идентификации групп, участвующих в целевой функции, менеджер проекта может продолжить выявление лиц или заслуживающих внимания областей, которые могут использоваться в качестве источников информации для модели.

Исходный материал может иметь разнообразные формы и довольно широко распространяться по организации.

Он включает:

- результаты опроса;
- результаты наблюдения;
- линии поведения и процедуры;
- выходные данные существующих систем (отчеты и выборки);
- входные данные для существующих систем (бланки входных данных и выборки);
- спецификации баз данных и файлов для существующих систем.

Независимо от применяемых методов целью разработчика в этот момент является разработка план сбора репрезентативной документации, отражающей информацию, соответствующую цели и точке зрения модели. Полученная информация должна быть отмечена таким образом, чтобы можно было проследить ее источник. На различных стадиях разработки модели будут постоянные ссылки на эту документацию и на документацию, получаемую в процессе моделирования. Разработчик будет искать и изучать исходный материал, который может служить обоснованием структурных характеристик модели и смысла представленных данных.

Как отмечалось ранее, целью моделирования данных является определение единого, согласованного в рамках всего предприятия взгляда на ресурсы данных, что в принято называть концептуальной схемой.

Исходные документы, по большей части, представляют либо внешнюю, либо внутреннюю схемы, соответствующие концептуальной схеме, но более ориентированные на практическое использование.

Пользовательские отчеты, например, являются представлением данных из внешней схемы и могут служить исходной документацией.

Описания файлов и проекты баз данных соответствуют представлениям данных из внутренней схемы и также могут использоваться в качестве исходной документации.

Хотя в процессе моделирования структура данных существенно меняется, окончательная модель должна по-прежнему соответствовать внутренней и внешней схеме, на основе которых она построена.

Правильный план сбора данных имеет первостепенное значение для успешного достижения цели. Он должен отражать, какой тип данных является важным, где такие данные можно получить и кто их предоставит.

2.3. Стадия определения сущностей

Целью данной стадии является выявление и определение сущностей, находящихся в пределах моделируемой проблемной области.

Первым шагом в этом процессе является **идентификация сущностей**.

2.3.1. Идентификация сущностей

«Сущность» в контексте IDEF1X-модели представляет множество «предметов», обладающих связанными с ними данными.

Здесь «предмет» может быть отдельной физической субстанцией,

событием, состоянием, действием, идеей, понятием, точкой, местом и т.д.

Элементы представляемого сущностью множества обладают общим набором атрибутов или характеристик.

Например, все элементы множества служащих обладают номером служащего, фамилией и другими общими атрибутами.

Отдельный элемент множества сущности называется *экземпляром сущности*.

Сущности всегда именуются общими существительными в единственном числе.

Сущности должны иметь атрибут (ключ), однозначно идентифицирующий каждый из их экземпляров.

Большинство сущностей могут быть прямо или косвенно определены на основе исходного материала, собранного на стадии «0».

Если при моделировании расширяется или детализируется предшествующая модель данных, то соответствующие сущности должны быть выделены из прежней модели.

Для предварительно не определенных сущностей разработчик должен выявить в списке имен исходного материала предметы, представляющие потенциально возможные сущности.

Для простоты можно выбрать все существительные из этого списка.

Например, такие термины, как деталь, транспортное средство, машина, чертеж и т.д., могут на этой стадии рассматриваться в качестве потенциальных сущностей.

Другой метод состоит в отборе терминов, перед которыми используется слово «код» или «номер» (например, номер детали, номер заказа на покупку, номер маршрута и т.д.).

Предложение, начинающееся словом «код» или «номер», может также рассматриваться на этой стадии в качестве потенциальной сущности.

Что касается оставшихся слов в списке, то разработчик модели должен задать вопрос:

«Представляет ли слово объект или предмет, о котором есть информация, или оно дает информацию о каком-то объекте или предмете?»

Те слова из списка, которые попадают в категорию объектов, о которых известна информация, потенциально являются сущностями.

Сущность образуется в результате объединения основных экземпляров сущности, становящихся элементами этой сущности.

Это означает, что некоторое количество экземпляров сущности, у которых все характеристики однотипны, представляется в качестве сущности.

Каждый экземпляр сущности является элементом сущности, обладающим однотипной определяющей информацией.

Для облегчения отделения сущностей от несущностей разработчик модели должен задать себе следующие вопросы, касающиеся каждой возможной сущности:

- Может ли она быть описана? (Обладает ли она характерными особенностями?)
- Существует ли более одного экземпляра этой сущности?
- Может ли один экземпляр быть отделен от другого (идентифицирован)?
- Называет или описывает это что-либо? (Из положительного ответа следует, что это скорее атрибут, чем сущность).

В конце такого анализа разработчик определяет *начальный пул (накопитель) сущностей*.

Данный пул содержит все известные на данный момент имена сущностей в контексте модели.

При построении пула сущностей разработчик присваивает каждой записи свой идентифицирующий номер и записывает ссылку на ее источник. Таким образом, поддерживается возможность отслеживания информации.

Целостность пула остается ненарушенной, а управление пула – сравнительно легким.

Пример *пула сущностей* показан в таблице 2.1.

Таблица 2.1

Пул Сущностей

Номер	Имя сущности	Номер источника
E-1	Платежное требование	2
E-2	Расходный ордер	2
E-3	Приходный ордер	2
E-4	Реестр платежных поручений	3

По всей вероятности, не все имена в списке останутся сущностями к концу последней стадии.

Кроме того, к этому списку добавится ряд новых сущностей, которые станут частью информационной модели по мере ее развития и улучшения понимания информации.

Обнаруженные на последующих стадиях имена сущностей должны добавляться в пул сущностей и приобретать уникальные идентифицирующие номера. Одним из результатов деятельности на данной стадии является пул сущностей. Он должен обновляться, чтобы оставаться жизнеспособным.

2.3.2. Определение сущностей

Еще одним результатом данной стадии является начало работы над глоссарием сущностей.

На протяжении этой стадии глоссарий представляет собой просто набор *определений сущностей*.

Определение сущности включает следующие компоненты:

1. *Имя сущности*. Имя сущности является уникальным именем, с помощью которого сущность будет распознаваться в IDEF1X-модели. Оно

должно быть описательным. Хотя допускаются аббревиатуры и акронимы, имя сущности должно быть осмысленным.

2. *Определение сущности.* Это то определение сущности, которое обычно используется на предприятии. Оно не задумано как часть словаря. Поскольку смысл отражаемой в модели информации зависит от точки зрения модели и от определенного на начальной стадии контекста модели, то бессмысленно (а может быть, и вредно) включать сюда определения, не входящие в область действия на начальной стадии.

Однако могут быть небольшие различия смысловых оттенков в способе определения сущности, зависящие, главным образом, от контекстуального использования.

В этих случаях или при наличии альтернативных определений (которые могут не быть наиболее общеупотребительными с точки зрения модели) они должны быть также записаны. Рецензенты по своему усмотрению устанавливают, какое определение должно быть связано с термином, употребляемым для определения сущности.

Процесс определения на стадии определения сущности является средством ускорения формирования общепринятого определения.

3. *Синонимы сущности.* Это список других имен, под которыми сущность может быть известна. Единственным относящимся к этому правилом является то, что определение, связанное с именем сущности, должно в точности применяться к каждому из синонимов в списке.

Определения сущностей формулировать легче, если начинать с сущностей, требующих наименьшего количества исследований. Таким образом, объем глоссария увеличится за кратчайшее время.

Затем разработчик сможет проводить исследования для полного определения оставшихся имен в пуле. Четкое планирование времени и усилий на сбор и определение информации обеспечит разумный темп процесса моделирования.

2.3. Стадия определение отношений между сущностями

Целями данной стадии является:

- выявление и определение основных отношений между сущностями, включенными в пул потенциальных сущностей;
- разрешение выявленных неспецифических отношений между сущностями.

Главными результатами стадии являются:

- матрица отношений между сущностями – матрица «сущность – отношения»;
- определения отношений между сущностями;
- диаграммы уровней сущностей;
- диаграммы разрешающие неспецифические отношения между сущностями;
- возможная корректировка пула сущностей.

2.3.1. Установление связанных сущностей. Построение матрицы отношений между сущностями

Отношение между сущностями может быть определено просто как ассоциация или связь между двумя сущностями, точнее, это называется *бинарным отношением*.

IDEF1X ограничивается бинарными отношениями, поскольку исследовать и понимать их легче, чем *n*-арные отношения. Кроме того, они имеют непосредственное графическое представление.

Недостатком является некоторое неудобство при представлении *n*-арных отношений. Но в этом нет ограничения общности, поскольку любое *n*-арное отношение может быть выражено через *n* бинарных отношений.

Экземпляр отношения между сущностями – это имеющая смысл ассоциация или связь между двумя экземплярами этих рассматриваемых сущностей.

IDEF1X-отношение между сущностями представляет множество однотипных образцов отношений между двумя специфическими сущностями.

При этом одна и та же пара сущностей может обладать отношениями нескольких типов.

*Целью IDEF1X является не изображение всех возможных отношений, а определение взаимосвязей между сущностями в терминах отношений зависимости существования отношений **родитель - потомок**.*

Такое отношение – это ассоциация между типом родительской сущности и типом сущности-потомка, при которой каждый экземпляр родительской сущности ассоциирован с произвольным (в том числе нулевым) количеством экземпляров сущности-потомка, а каждый экземпляр сущности-потомка ассоциирован в точности с одним экземпляром родительской сущности.

Это означает, что существование сущности-потомка зависит от существования родительской сущности.

*Если сущность-родитель и сущность-потомок представляют один и тот же объект реального мира, то **родительская сущность является общей сущностью**, а **сущность-потомок является сущностью-категорией**.*

Для каждого экземпляра сущности-категории всегда имеется один экземпляр общей сущности.

Для каждого экземпляра общей сущности может существовать ноль или один экземпляр сущности-категории.

Если несколько сущностей-категорий ассоциируются с общей сущностью в отношении категоризации, то только одна категория может соответствовать данному экземпляру общей сущности.

В начале разработки IDEF1X-модели зачастую невозможно представить все отношения как отношения *родитель-потомок* или отношения *категоризации*.

Поэтому *неспецифические отношения* на стадии в дальнейшем должны быть преобразованы в специфические, т.е. в отношения *родитель-потомок* или отношения *категоризации*.

Неспецифические отношения имеют общую форму – «ноль, один или много – к – ноль, один или много».

Существование любой сущности не зависит от существования другой.

Первым шагом на данной стадии является выявление отношения между элементами различных сущностей. Эта задача может потребовать разработки *матрицы отношений*.

Матрица отношений – это двумерный массив, обладающий горизонтальной и вертикальной осями.

Множество предопределенных факторов (в данном случае – все сущности) записывается вдоль одной из осей, а другое множество факторов (в данном случае – также все сущности) записывается вдоль другой оси.

Для указания на возможное отношение между двумя сущностями в точке пересечения соответствующих осей помещается знак «V». В этот момент суть отношения не важна: достаточно того, что отношение может существовать.

Разработчики-новички обычно устанавливают чрезмерное количество отношений между сущностями. Помните, что целью в конечном итоге является определение модели в терминах отношений родитель-потомок. *Избегайте косвенных отношений.*

Например, если ОТДЕЛ ответствен за один или несколько ПРОЕКТОВ, а каждый ПРОЕКТ инициирует одно или несколько ПРОЕКТНЫХ_ЗАДАНИЙ, то нет необходимости в отношении между ОТДЕЛОМ и ПРОЕКТНЫМ_ЗАДАНИЕМ, поскольку все ПРОЕКТНЫЕ_ЗАДАНИЯ связаны с ПРОЕКТОМ, а все ПРОЕКТЫ связаны с ОТДЕЛОМ.

2.3.2. Определения отношений между сущностями

Определения отношений между сущностями включают:

- установление зависимостей отношения между сущностями;
- определение имени отношения между сущностями;
- комментарии к отношениям между сущностями.

В ходе определения отношений некоторые из них могут отбрасываться, а новые добавляться.

Для *установления зависимости* отношение между двумя сущностями должно быть проверено в обоих направлениях.

Это делается посредством *определения мощности* на каждом конце отношения между сущностями.

Для определения *мощности отношения* предполагают существование экземпляра одной из сущностей. Затем определяют, сколько различных экземпляров второй сущности может быть связано с первой.

После этого повторяют анализ, поменяв сущности ролями.

После установления *зависимости* рассматриваемого отношения назначается *имя* этого отношения.

Имя отношения является кратким выражением, обычно глаголом с союзом, присоединяющим вторую упомянутую сущность.

Это выражение отражает смысл представляемого отношения.

Часто *имя отношения* состоит из одного глагола, хотя наречия и предлоги также появляются в именах отношений.

После выбора *имени отношения* разработчик *читая отношение* должен иметь возможность получать осмысленное предложение, определяющее или описывающее отношение между двумя сущностями.

При *специфической форме отношения* всегда имеются *сущность-родитель* и *сущность-потомок* и *имя отношения* интерпретируется сначала со стороны *сущности-родителя*, а затем от *сущности-потомка* к *сущности-родителю*.

Если между этими сущностями (*сущность-родитель* и *сущность-потомок*) существует отношение категоризации, то отсюда следует, что обе эти сущности относятся к одному и тому же объекту реального мира и мощностью на конце *сущности-потомка* (или *сущности-категории*) всегда является «ноль или один».

Имя отношения в таком случае опускается, поскольку имя «может быть» подразумевается.

При *неспецифической форме отношения* существует два *имени отношения* - по одному для каждой сущности, разделенные знаком «/».

В этом случае *имена сущностей* интерпретируются *сверху вниз* или *слева направо* (в зависимости от относительных положений сущностей на диаграмме), а затем в обратном направлении.

Имена отношений должны быть осмысленными.

Под их именами должна быть реальная основа.

Полное значение, т.е. причина выбора разработчиком данного имени отношения, может быть отражено в *определении отношения*.

Определение отношения – это текст, объясняющий смысл отношения.

Правила для *определения сущностей* применяются и к *определениям отношений*.

Определения отношений (комментарии к отношениям) должны быть:

- специфическими;
- краткими;
- осмысленными.

Например, если *отношение* «один к нулю или к одному» определено между такими двумя сущностями, как ОПЕРАТОР и РАБОЧАЯ_СТАНЦИЯ, то *имя отношения* может читаться «в настоящий момент назначен для обслуживания».

Это отношение может сопровождаться следующим определением: «Каждый оператор на протяжении каждого рабочего дня может быть

назначен для обслуживания нескольких рабочих станций, но это отношение указывает на ту, которую оператор обслуживает в данный момент».

2.3.3. Разрешение неспецифических отношений между сущностями

Неспецифические отношения между сущностями имеют общую форму – «ноль, один или много – к – ноль, один или много» (п.2.3.1).

Данные отношения необходимо привести к виду специфических отношений, т.е. к отношению *сущность-родитель* и *сущность-потомок*, либо *отношению категоризации*.

Чтобы выполнить это требование, разработчик предлагает варианты детализации в виде диаграмм (*диаграммы вариантов детализации*).

Диаграммы вариантов детализации обычно делятся на две части:

- левая часть посвящена субъекту - детализируемому неспецифическому отношению;
- правая часть – вариант детализации.

На рис. 2.3 показан вариант детализации, относящейся к разрешению отношений типа «многое ко многому».

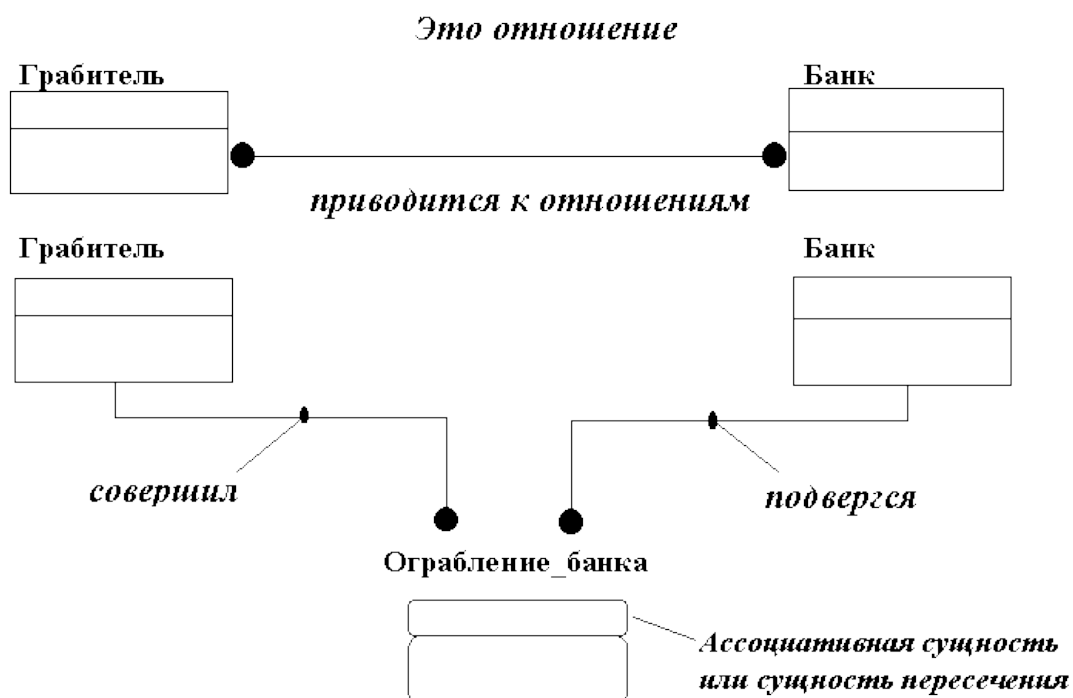


Рис. 2.3. Детализация не специфического отношения

Процесс детализации отношений приводит или конвертирует каждое *неспецифическое отношение* в *два специфических отношения*.

В этом процессе возникают новые сущности.

Неспецифическое отношение на рис. 2.3 указывает, что ГРАБИТЕЛЬ может ограбить много БАНКОВ, а БАНК может быть ограблен многими

ГРАБИТЕЛЯМИ.

Однако мы не можем определить, какой ГРАБИТЕЛЬ грабил какой БАНК, пока не введем для разрешения этого неспецифического отношения третью сущность: ОГРАБЛЕНИЕ_БАНКА.

Каждый экземпляр сущности ОГРАБЛЕНИЕ_БАНКА связан с одним БАНКОМ и с одним ГРАБИТЕЛЕМ.

Ранее мы имели дело с сущностями, которые могли бы неформально назвать *естественными*.

Естественная сущность – это сущность, которую мы, вероятно, будем считать очевидной в списке исходных данных или протоколе исходных материалов при разработке модели.

Естественная сущность будет иметь такие имена, как

- Заказ на покупку.
- Служащий.
- Покупатель.

Сущности пересечения, как видно из приведенного примера, используются для разрешения *неспецифических отношений* и обычно представляют упорядоченные пары предметов с теми же основными характеристиками (уникальный идентификатор, атрибуты и т.д.), что и исходные естественные сущности.

Хотя в предыдущем примере сущность ОГРАБЛЕНИЕ-БАНКА могла бы рассматриваться как естественная сущность, она в действительности представляет объединение сущностей ГРАБИТЕЛИ и БАНКИ.

Одно из небольших различий между *естественной сущностью* и *сущностью пересечения* состоит в именах сущностей.

Именем естественной сущности служит обычно единичное нарицательное существительное; имена сущностей пересечения могут быть составными.

Сущности пересечения являются по своей природе более абстрактными и обычно появляются в результате впервые примененных на стадии определения ключей - *правил определения правильности сущностей*.

Процесс детализации *неспецифических отношений* является очень важным этапом детализации объединенной структуры данных и включает:

- разработку для каждого неспецифического отношения одного или нескольких вариантов детализации;
- выбор разработчиком предпочтительного варианта, который и будет отражен в модели на данной стадии;
- обновление информации стадии определения сущностей с целью включения возникших при детализации новых сущностей в сформированный ранее пул сущностей;
- обновление информации стадии определения отношений с целью определения отношений, связанных с новыми выявленными сущностями, включенными в пул сущностей.

2.5. Определение ключевых атрибутов сущностей

Целями данной стадии являются:

- Определение ключевых атрибутов для каждой сущности.
- Перемещение первичных ключей для установления внешних ключей.
- Проверка правильности отношений и ключей.

Результаты стадии изображаются на одной или нескольких диаграммах (диаграммах ключевого уровня). Помимо определения ключевых атрибутов на данной стадии возможно расширение и последующая дополнительная детализация определения сущностей и отношений между ними.

2.5.1. Определение ключевых атрибутов сущностей

На стадии определения *ключевых атрибутов* сущностей - *ключей* методологией IDEF1X идентифицируются и определяются элементы данных об экземплярах сущностей, называемых возможными ключами, первичными ключами, альтернативными ключами и внешними ключами.

Цель определения ключей – установить значения атрибутов, однозначно определяющих каждый экземпляр сущности.

Важно подчеркнуть определение и смысл терминов «*экземпляр атрибута*» и «*атрибут*».

Определение. *Экземпляр атрибута* является свойством или характеристикой экземпляра сущности.

Экземпляры атрибутов состояются из имени и значения.

По своей сути, *экземпляр атрибута* является элементом информации, известной об отдельном экземпляре сущности.

Экземпляры атрибутов являются *описателями*, т.е. они по сути скорее подобны *прилагательным*.

В таблице 2.3 приведены некоторые *экземпляры атрибутов* и их соответствующие *экземпляры сущностей*.

Таблица 2.3

Примеры экземпляров атрибутов

Экземпляры атрибутов					
Имя	Иванов	Имя	Петров	Имя	Сидоров
Номер	1	Номер	2	Номер	3
Должность	Оператор	Должность	Зав. Отд.	Должность	Пилот
Сущность – это множество, к которому принадлежит Иванов, Петров, Сидоров. В этом случае сущность получает имя СЛУЖАЩИЙ					
Атрибут – это признаки, которые описывают в общем виде характеристики сущности, например СЛУЖАЩИЙ. В данном случае каждого служащего описывают атрибуты: имя, номер, должность.					

Первый *экземпляр сущности* или индивидуум идентифицируется *номером служащего 1*, ассоциированное с этим *экземпляром сущности* имя – Иванов, а профессия этого *экземпляра сущности* – оператор.

Эти экземпляры *атрибутов*, взятые вместе, однозначно описывают экземпляр *сущности* и отделяют его от других аналогичных экземпляров *сущностей*.

Каждый экземпляр *атрибута* обладает как *типом*, так и *значением*.

Каждый конкретный экземпляр *сущности* описывает уникальную комбинацию экземпляров *атрибутов*.

Атрибут представляет множество экземпляров *атрибутов* одного типа, относящихся к разным экземплярам одной и той же *сущности*.

Имена *атрибутов* обычно являются описательными существительными в единственном числе.

В приведенном примере (таблица 2.3) с *сущностью* СЛУЖАЩИЙ имеется несколько *атрибутов*:

- Номер служащего.
- Имя служащего.
- Профессия /должность служащего.

В таблице 2.3 также показано, как экземпляры *атрибутов* представляются в качестве *атрибутов*.

Экземпляры *атрибутов* принадлежат экземплярам *сущностей*.

Но и сами *атрибуты* принадлежат *сущности*.

Таким образом, между *сущностью* и некоторым числом *атрибутов* устанавливается *ассоциация собственности*.

У *атрибутов* есть только один *владелец*.

Владелец атрибута – это *сущность*, которой *атрибут* принадлежит.

В приведенном примере *владелец атрибута* НОМЕР-СЛУЖАЩЕГО будет *сущность* СЛУЖАЩИЙ.

Хотя *атрибут* имеет только одного *владельца*, *владелец* может делить его с другими *сущностями*. Ниже будет детально рассмотрено, как это происходит.

Атрибут представляет использование экземпляра *атрибута* для описания *отдельного специфического свойства отдельного экземпляра сущности*.

Определение. Те *атрибуты*, которые представляют использование экземпляра *атрибута* для однозначного установления *специфического экземпляра сущности*, неформально называются **ключевыми атрибутами**.

На стадии *определения ключей* производится *идентификация ключевых атрибутов* в контексте разрабатываемой модели.

На следующей стадии *определения атрибутов* устанавливаются и определяются **неключевые атрибуты**.

Один или несколько *атрибутов* образуют *возможный ключ сущности*.

Определение. *Возможный ключ* определяется как один или несколько *ключевых атрибутов* для однозначной *идентификации* каждого экземпляра *сущности*.

Примером *атрибута*, используемого в качестве *возможного ключа сущности*, является *номер служащего*.

Каждый служащий идентифицируется среди всех других служащих с помощью номера служащего. Поэтому *атрибут* НОМЕР-СЛУЖАЩЕГО является *возможным ключом*, который, *однозначно определяет каждый элемент сущности СЛУЖАЩИЙ*.

Некоторые сущности обладают более чем одной группой атрибутов, которые могут применяться для различения одного экземпляра сущности от других.

Рассмотрим сущность СЛУЖАЩИЙ с атрибутами НОМЕР-СЛУЖАЩЕГО и НОМЕР_СТРАХОВОГО_ПОЛИСА, каждый из которых сам по себе является *возможным ключом*.

Для такой сущности выбирается один возможный ключ для использования в миграции ключей.

Этот ключ называется *первичным ключом*, а остальные возможные ключи – *альтернативными ключами*.

Определение. Если *сущность* обладает только одним *возможным ключом*, то он автоматически является *первичным ключом*.

Таким образом, каждая *сущность* обладает *первичным ключом*, а некоторые *сущности* обладают также *альтернативными ключами*.

Каждый тип возможных ключей может применяться для идентификации экземпляров сущностей, но только *первичный ключ используется в миграции ключей*.

На диаграмме IDEF1X-модели в блоке основной сущности проводится горизонтальная линия, и внутри блока, выше этой линии, указывается *первичный ключ*.

Если в *первичном ключе* имеется более одного атрибута (например, для идентификации проектных заданий требуются и номер проекта, и номер задания), то они все указываются выше горизонтальной линии.

Если сущность обладает *альтернативным ключом*, то ему присваивается *уникальный номер альтернативного ключа*. На диаграмме этот номер указывается в скобках вслед за каждым атрибутом, являющимся частью этого альтернативного ключа.

Если атрибут принадлежит нескольким альтернативным ключам, то каждый из номеров этих ключей указывается в скобках.

Если атрибут принадлежит и альтернативному ключу, и первичному ключу, то он указывается выше горизонтальной линии вместе со следующим после него номером альтернативного ключа.

Если атрибут не принадлежит первичному ключу, то он указывается ниже горизонтальной линии.

На рис. 2.4 приведены различные формы ключей.

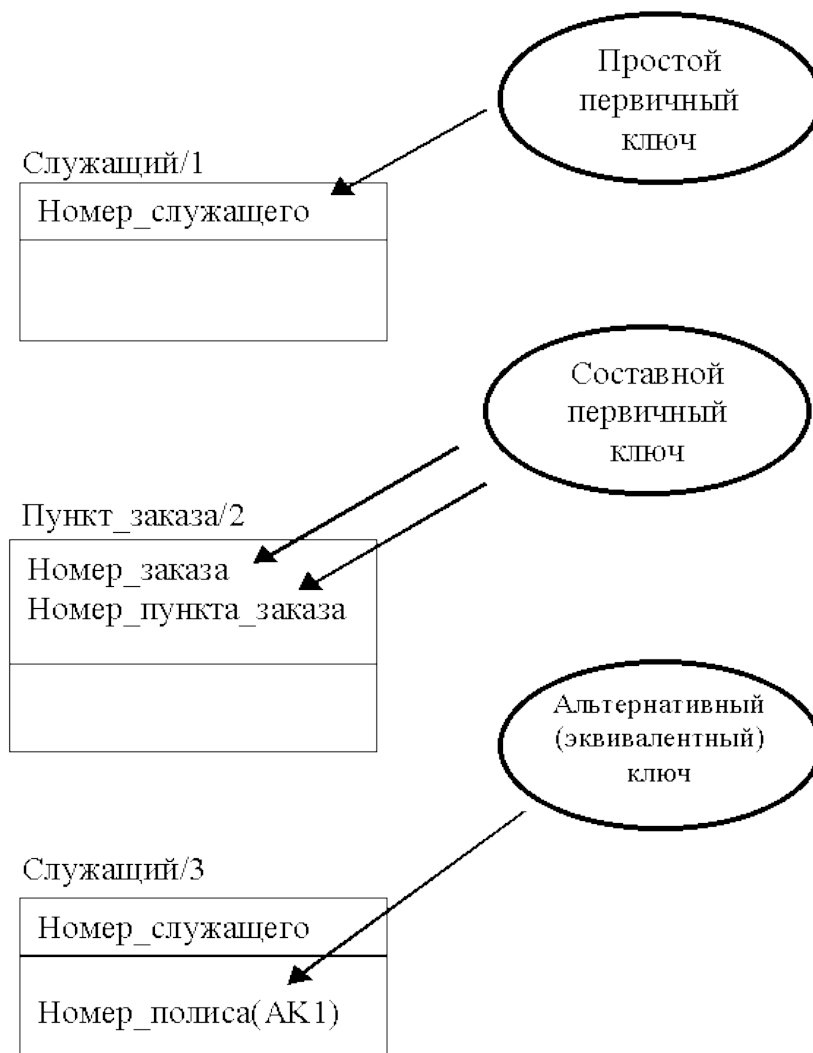


Рис. 2.4. Формы ключей

Процесс идентификации ключей включает следующие процедуры:

- идентификацию возможных ключей сущности;
- выбор одного из них в качестве первичного ключа сущности.

Поскольку некоторые возможные ключи могут возникнуть в результате миграции, то *идентификация ключей* представляет собой *итеративный процесс*.

Идентификацию ключей целесообразно начинать с сущностей, которые не являются ни в каком отношении сущностями-потомками или сущностями-категориями.

Это обычно те сущности, чьи возможные ключи наиболее очевидны.

Они являются также *начальными точками для миграции ключей*, поскольку не содержат *внешних ключей*.

2.5.2. Миграция ключей

Определение. Миграция ключей (перемещение первичных ключей) – это

процесс копирования первичного ключа одной сущности в другую, связанную с ней сущность. *Эта копия называется **внешним ключом**.*

Значение *внешнего ключа* в каждом экземпляре второй сущности совпадает со значением связанного экземпляра первой сущности. Таким образом, атрибут, принадлежащий одной сущности, разделяется с другой сущностью.

Миграция ключей подчиняется следующим трем правилам:

1. Миграция всегда происходит в отношении от родительской или общей сущности к сущности-потомку или сущности-категории.

2. Весь первичный ключ (т.е. все атрибуты, являющиеся элементами первичного ключа) должен мигрировать по одному разу для каждого отношения, разделяемого парой сущностей.

3. Альтернативный ключ и неключевые атрибуты никогда не мигрируют.

Каждый атрибут *внешнего ключа* соответствует атрибуту *первичного ключа* родительской или общей сущности.

Первичный ключ сущности-категории в категориальном отношении должен совпадать с первичным ключом общей сущности.

В других отношениях атрибут *внешнего ключа* может, но не обязан быть частью *первичного ключа сущности-потомка*.

Атрибуты *внешних ключей* не считаются принадлежащими сущностям, в которых они появляются, поскольку они отражают *атрибуты родительских сущностей*.

Таким образом, *каждый атрибут в сущности либо принадлежит этой сущности, либо принадлежит внешнему ключу этой сущности.*

В диаграммах модели *внешние ключи* обозначаются примерно так же, как *альтернативные ключи*, т.е. после каждого атрибута, принадлежащего внешнему ключу, следует (FK).

Если атрибут принадлежит также первичному ключу, то он располагается выше горизонтальной линии, а если нет, то – ниже.

Определение. Если первичный ключ сущности-потомка содержит все атрибуты внешнего ключа, то сущность-потомок называется зависимой от идентификатора относительно родительской сущности, а отношение называется *идентифицирующим отношением*.

Определение. Если какие-либо атрибуты внешнего ключа не принадлежат первичному ключу сущности-потомка, то сущность-потомок не является независимой от идентификатора относительно родительской сущности, а отношение называется *неидентифицирующим отношением*.

На диаграммах IDEF1X *сплошными линиями* изображаются только *идентифицирующие отношения*, а *неидентифицирующие отношения* изображаются *пунктирными линиями*.

Определение. *Сущность, являющаяся сущностью-потомком* в одном или нескольких *идентифицирующих отношениях*, называется *зависимой от идентификатора*.

Определение. Сущность, являющаяся сущностью-потомком только в неидентифицирующих отношениях (или не являющаяся сущностью-потомком ни в одном из отношений), называется *независимой от идентификатора*.

В диаграммах блоками с прямыми углами изображаются только идентификаторно-независимые сущности, а идентификаторно-зависимые сущности изображаются блоками с закругленными углами.

Пример миграции ключей приведен на рис. 2.5.

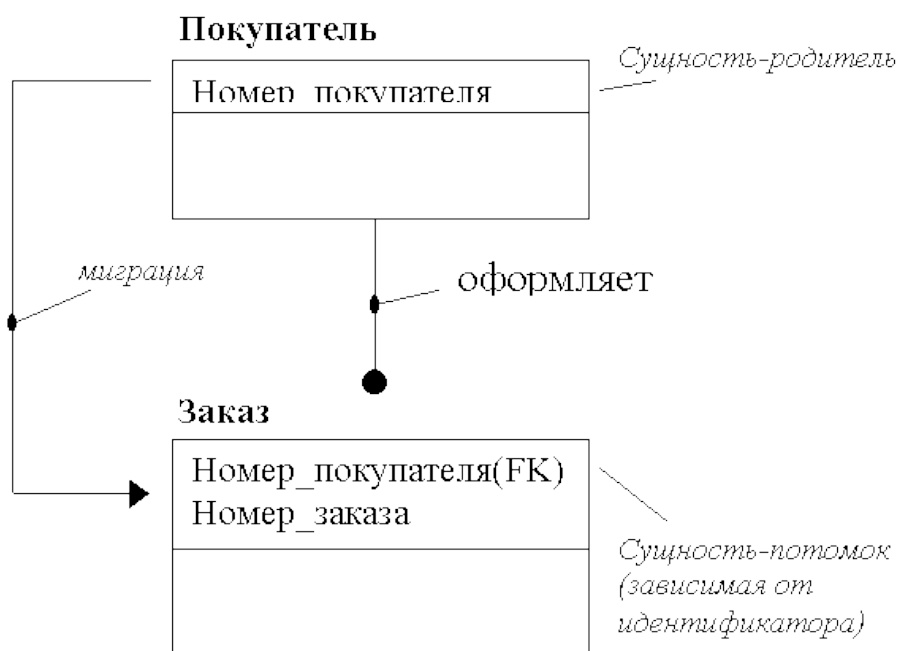


Рис. 2.5. Миграция ключа к зависимой от идентификатора сущности

2.5.3. Проверка правильности ключей и отношений

Идентификация ключей и миграция ключей подчиняется следующим основным правилам:

1. Нельзя использовать синтаксис неспецифических отношений.
2. Миграция ключей от родительских (или общих) сущностей к сущностям-потомкам (или сущностям-категориям) является обязательной.
3. Запрещается использовать атрибуты, которые могут принимать более одного значения для данного экземпляра сущности в одно и то же время (*правило неповторяемости*).
3. Нельзя использовать атрибуты, обращающиеся в ноль (т.е. не принимающие никакого значения) для некоторого экземпляра сущности (*правило необращения в ноль*).
5. Сущности с составными ключами не могут быть разбиты на несколько сущностей с более простыми ключами (*правило наименьшего ключа*).
6. Необходимо объявлять об имеющихся между двумя сущностями двойных путях отношений.

В предыдущих разделах уже рассмотрены первые два правила, поэтому

остановимся на оставшихся правилах.

На рис. 2.6 приведена диаграмма, относящаяся к применению **правила неповторяемости**.

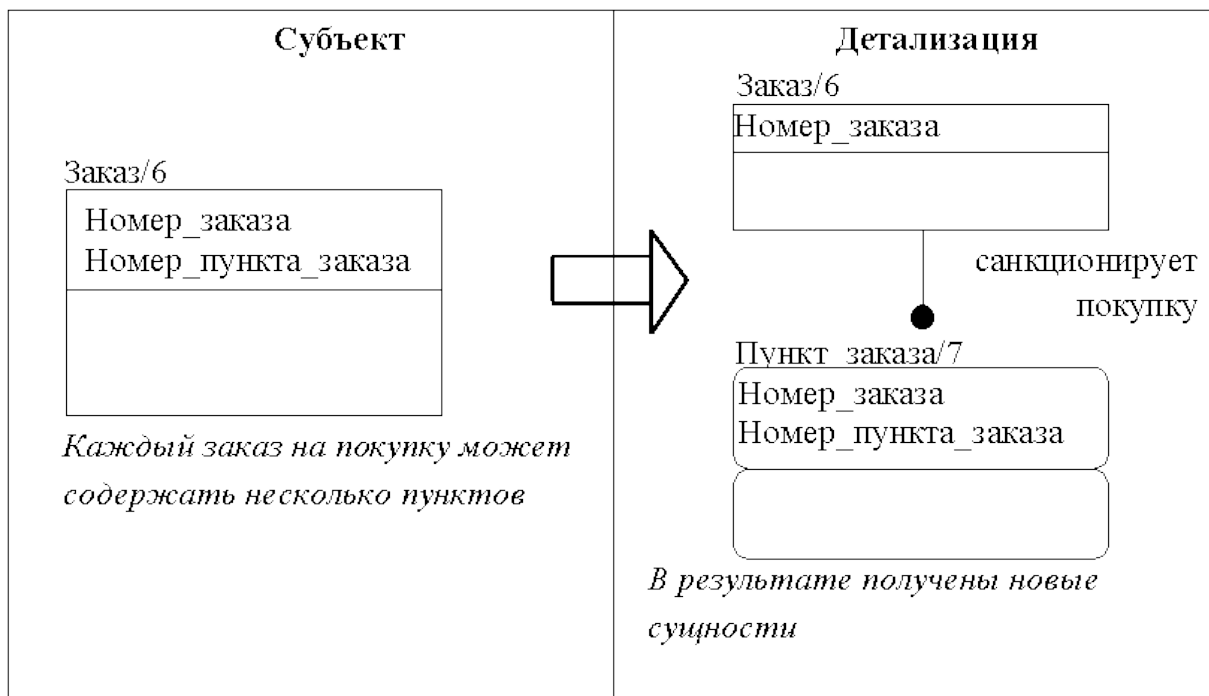


Рис. 2.6. Детализация правила неповторяемости

Из приведенной диаграммы видно, что субъект диаграммы содержит в качестве элементов первичного ключа сущности ЗАКАЗ такие атрибуты, как НОМЕР_ЗАКАЗА и НОМЕР_ПУНКТА_ЗАКАЗА.

Однако, рассмотрев использование НОМЕРА_ПУНКТА_ЗАКАЗА, мы увидим, что один и тот же экземпляр сущности ЗАКАЗ может быть ассоциирован со многими НОМЕРАМИ_ПУНКТА_ЗАКАЗА, по одному на каждый заказываемый пункт.

Для правильного отражения этого факта в модели данных должна быть создана новая сущность, называемая ПУНКТ_ЗАКАЗА с добавлением дуги и метки отношения, синтаксиса и определения.

Таким образом, начинают выясняться истинные характеристики связи между заказами на покупку и пунктами заказов на покупку.

На рис. 2.7 приведена диаграмма вариантов детализации, относящаяся к применению **правила необращения в ноль**.

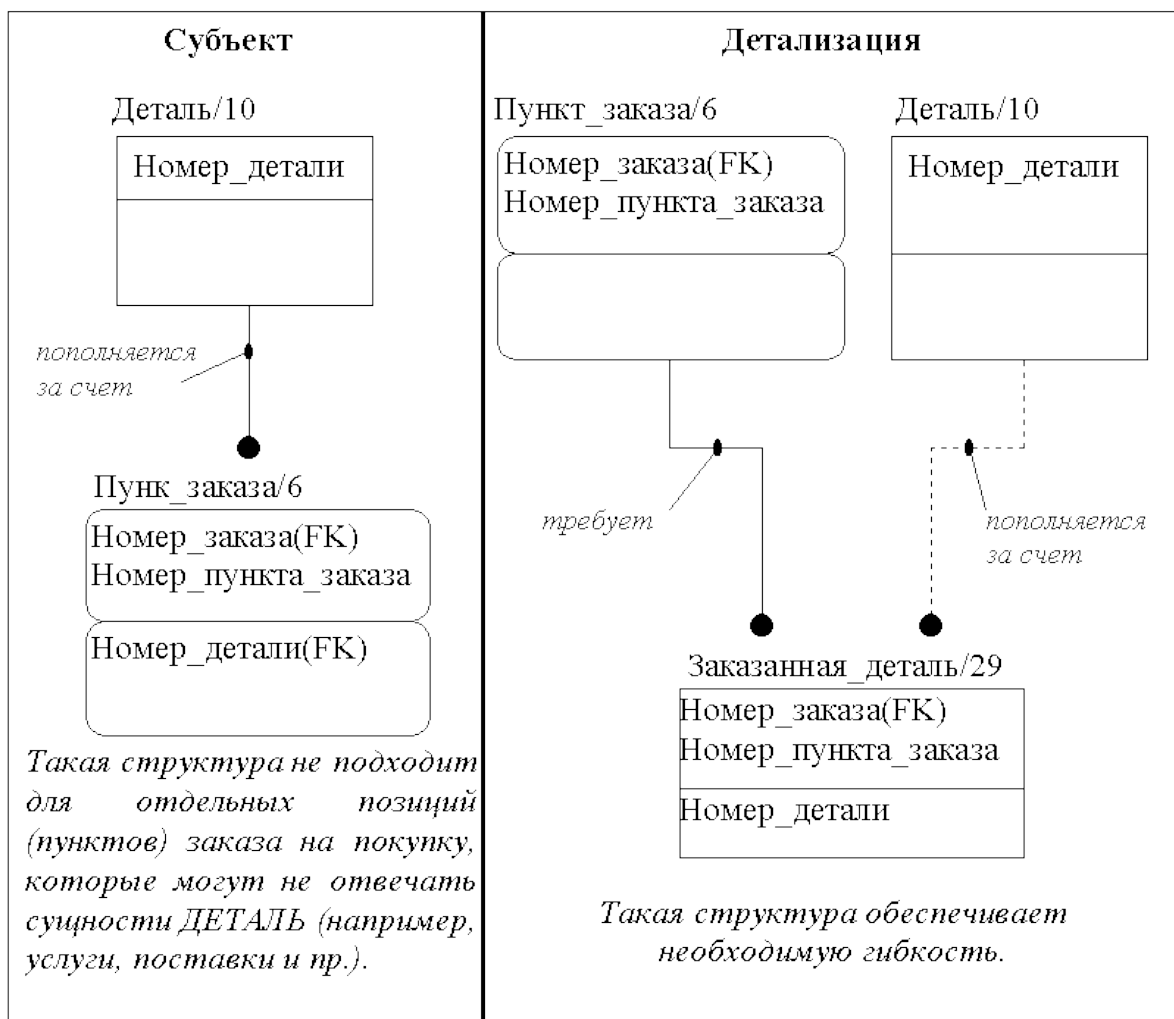


Рис. 2.7. Детализация правила необращения в ноль.

Заметим, что ключ **НОМЕР_ДЕТАЛИ** промигрировал к **ПУНКТУ_ЗАКАЗА**.

Эта ассоциация установилась в связи с тем, что **ПУНКТ_ЗАКАЗА** связан некоторым образом с деталями. Однако показано, что диаграмма утверждает ассоциированность каждого пункта заказа на покупку в точности с одним номером детали.

Исследование (или, возможно, комментарий рецензента) показывает, что не все *пункты заказа на покупку* ассоциируются с *детальями*. Некоторые из них могут быть связаны с услугами или другими товарами, не имеющими номеров деталей. Это запрещает миграцию **НОМЕРА_ДЕТАЛИ** непосредственно в сущность **ПУНКТА_ЗАКАЗА** и требует в нашем примере установления *новой сущности* **ЗАКАЗАННАЯ_ДЕТАЛЬ**.

Как только новая сущность установлена, в соответствии с правилом миграции обязательно должна произойти миграция ключа, и разработчик будет снова проверять правильность соответствия структуры сущность-отношение *правилам необращения в ноль и неповторяемости*.

Каждый составной ключ должен проверяться на соответствие

правилу наименьшего ключа.

Это правило требует, чтобы любая сущность с составным ключом не могла разделяться на несколько сущностей с более простыми ключами (на меньшие компоненты) без потери некоторой информации.

В связи со стадией определения отношений упоминалась тенденция выявления избыточных отношений. Однако на той стадии анализ, в основном, проводится разработчиком по его усмотрению. Установив ключи, разработчик может быть более точным в анализе.

Двойной путь отношений существует тогда, когда существует сущность-потомок с двумя отношениями, ведущими, в конечном итоге, обратно (через одно или несколько отношений) к общей «корневой» сущности-родителю.

В случае существования двойных путей требуется утверждение пути, чтобы определить, являются ли пути равными, неравными или неопределенными.

Определение. Пути равны, если для каждого экземпляра сущности-потомка оба пути отношений всегда ведут к одному и тому же экземпляру корневой родительской сущности.

Определение. Пути являются неравными, если для каждого экземпляра сущности-потомка оба пути отношений всегда ведут к различным экземплярам корневой родительской сущности.

Определение. Пути являются неопределенными, если они равны для некоторых экземпляров сущности-потомка и не равны для остальных экземпляров.

Если один из путей состоит только из одного отношения и пути равны, то путь из одного отношения является излишним и должен быть удален.

Простейшим случаем отношения, имеющего двойственные пути, является отношение, в котором оба пути состоят из единственного отношения.

Например, каждый экземпляр сущности СХЕМА_СБОРКИ может быть связан с двумя различными экземплярами сущности ДЕТАЛЬ, избыточности нет.

В этом случае утверждение пути будет требовать, чтобы пути были неравны, поскольку ДЕТАЛЬ не может быть вмонтирована в себя.

Определение. Если один из путей содержит несколько отношений, а другой путь содержит одно отношение, то такая структура называется **триадой**.

На рис. 2.8 приведен пример **триады**.

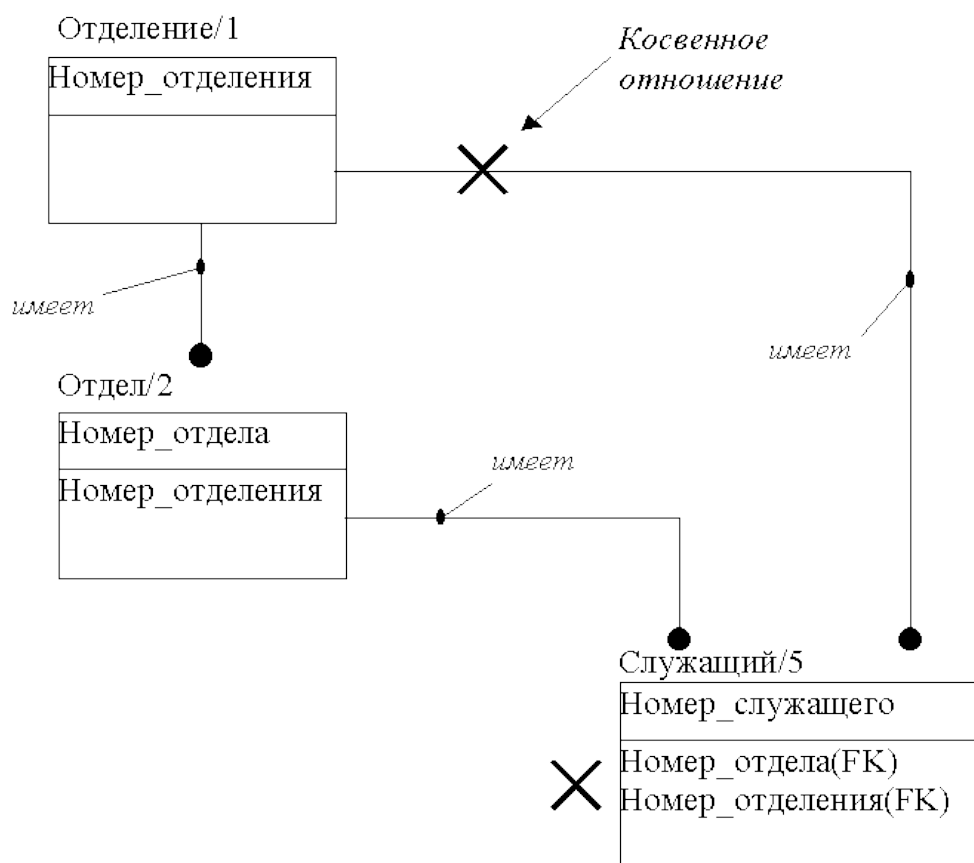


Рис. 2.8. Пример триады

В этом случае СЛУЖАЩИЙ связан с ОТДЕЛЕНИЯМИ как прямо, так и косвенно, через ОТДЕЛ.

Если *утверждение пути* состоит в том, что ОТДЕЛЕНИЕ, которому принадлежит СЛУЖАЩИЙ, содержит ОТДЕЛ, которому принадлежит СЛУЖАЩИЙ, то отношение между сущностями ОТДЕЛЕНИЯ и СЛУЖАЩИЙ является *избыточным* и должно быть удалено.

Заметим, что если некоторые, но не все, СЛУЖАЩИЕ могут в действительности принадлежать двум различным отделениям, то должна быть добавлена еще одна сущность, такая, как ВРЕМЕННЫЙ_СЛУЖАЩИЙ, чтобы НОМЕР_ОТДЕЛЕНИЯ удовлетворял *правилу необращения в ноль* в качестве внешнего ключа сущности СЛУЖАЩИЙ.

Утверждения могут также применяться к отношениям двойственных путей, когда оба пути раскрывают более одного отношения.

В приведенном на рис. 2.9 примере между сущностями ОТДЕЛ и ИСПОЛНИТЕЛЬ_ЗАДАНИЯ существуют *два пути отношений*.

Если СЛУЖАЩИЙ может быть приписан только к тому ПРОЕКТУ, которым руководит его ОТДЕЛ, то *пути равны*.

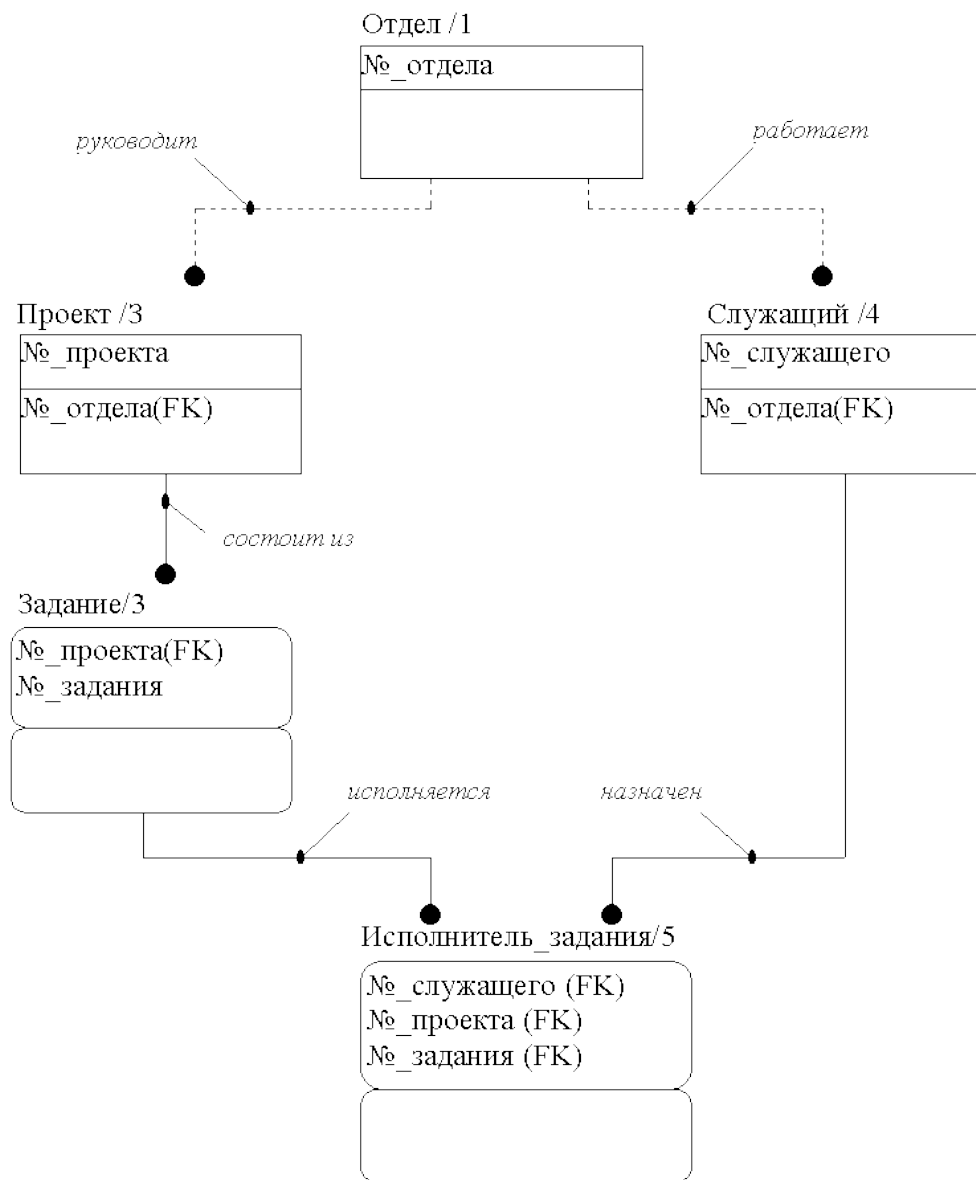


Рис. 2.9. Пример утверждения пути

Если СЛУЖАЩИЙ может быть приписан только к тому ПРОЕКТУ, которым руководит не его отдел, то *пути не равны*.

Если СЛУЖАЩИЙ может быть приписан к ПРОЕКТУ независимо от того, руководит его ОТДЕЛ ПРОЕКТОМ или нет, то *пути являются неопределенными*.

Обычно, пути предполагаются неопределенными, если только утверждения точно не определены.

Утверждения должны быть добавлены в качестве примечаний к диаграммам стадии определения ключей и включены в определение сущности-потомка.

После идентификации элементов первичного ключа производятся записи в пул атрибутов.

Для идентификации распределения и использования атрибутов в модели может применяться матрица сущность/атрибут.

Эта матрица обладает следующими свойствами:

1. Все имена сущностей изображаются с краю.
2. Все имена атрибутов изображаются наверху.
3. Использование атрибутов сущностями изображается в соответствующей строке с помощью следующей кодировки:
 - «О» = владелец,
 - «К» = первичный ключ,
 - «I» = наследуемый.

Пример матрицы сущность/атрибут приведен в таблицах 2.4 и 2.5.

Эта матрица является основным средством поддержки целостности модели.

Таблица 2.3.

Пример матрицы «Сущность/атрибут»

Сущности		Имена соответствующих атрибутов приведены в таблице 3.5																		
		1	2	3	4	5	20	21	22									
Условия покупки	1	О к																		
Покупатель	2		О к																	
Продавец	3			О к																
Заказ	4																			
Оформитель	6																			
Деталь	9																			
Пункт заказа	10	О к																		
Срока заказа	12	О к																		
Администратор	21			О к																
Поставщик	22																			

Таблица 2.5.

Имена атрибутов

№	Атрибуты	№	Атрибуты
1	Номер условий покупки	12	Имя оформителя
2	Код покупателя	13	Код отдела
3	Код продавца	14	Послать через ...
4	Код заказа	15	Имя покупателя
5	Номер измерения	16	Номер заказа
6	Пункт назначения	17	Дата отпуска заказа
7	Имя продавца	18	Код получателя
8	Адрес продавца	19	Код налога
9	Код подтверждения	20	Код дилера
10	Имя подтверждающего лица	21	Номер бланка
11	Код дополнительных копий	22	Условия платежа

2.5.3. Определение ключевых атрибутов

После установления ключей наступает момент для определения атрибутов, которые были использованы в качестве ключей.

Для этих определений будут точными, специфическими, полными и универсально понимаемыми.

Определения атрибутов всегда ассоциированы с сущностями, владеющими этими атрибутами, т.е. они всегда являются элементами набора документов сущностей-владельцев. Поэтому достаточно просто определить атрибуты, которые принадлежат каждой сущности и используются в этом *первичном* или *альтернативном* ключе сущности.

В примере из таблицы 2.4 эти атрибуты кодируются ОК в матрице сущность/атрибут.

Определение атрибута включает:

- имя атрибута;
- определение атрибута;
- синонимы атрибута.

2.5.5. Изображение результатов стадии определения ключей

В результате идентификации и миграции ключей диаграммы функционального представления могут теперь быть обновлены для отражения и детализации отношений.

Диаграммы функционального представления должны изображать:

- атрибуты первичных, альтернативных и внешних ключей;
- независимые от идентификатора (с прямыми углами) и зависимые от идентификатора (с закругленными углами) сущности;
- идентифицирующие (сплошная линия) и неидентифицирующие (штриховая линия) отношения.

Большую часть информации, полученной в результате анализа на этой стадии, содержат сами сущности.

Каждый набор документов сущности содержит:

- определение сущности;
- список атрибутов первичных, альтернативных и внешних ключей;

- определения принадлежащих сущности ключевых атрибутов;
- список отношений, в которых сущность является общей;
- список отношений, в которых сущность является сущностью-категорией;
- список идентифицирующих отношений, в которых сущность является сущностью-родителем;
- список идентифицирующих отношений, в которых сущность является сущностью-потомком;
- утверждения двойных путей (в случае необходимости).

Если нужно, разработчик может построить для сущности отдельную диаграмму, следуя тому же подходу, что и при построении необязательной диаграммы сущности на стадии определения отношений.

Помимо табличных списков определений отношений, полезны перекрестные обратные ссылки на ассоциированные сущности.

В документах, разработанных на стадии определения ключей, необходимо перекрестно ссылаться на те атрибуты, которыми сущности обладают, и на те, которые они разделяют.

2.6. Стадия определения неключевых атрибутов

Данная стадия является завершающей стадией разработки модели.

Она включает:

- разработку *пула атрибутов*;
- установление принадлежности атрибутов;
- определение неключевых атрибутов;

Результаты стадии определения атрибутов изображаются на одной или нескольких диаграммах (диаграммах уровня атрибутов). В конце стадии модель данных полностью детализируется (в соответствии с пятой нормальной формой в реляционной теории).

Модель снабжается полным множеством определений и перекрестных ссылок для всех сущностей, (ключевых и неключевых) атрибутов и отношений.

2.6.1. Разработка пула атрибутов

На стадии определения сущностей в пул сущностей в качестве потенциальных сущностей было введено много имен из списка исходных данных начальной стадии. Некоторые из этих имен, однако, могли не быть признаны на стадии определения ключей в качестве сущностей. По всей вероятности они являются атрибутами.

Кроме того, многие из имен, не выбранные из списка исходных данных сначала, являются, возможно, атрибутами. Этот список, в сочетании со сведениями, полученными на стадиях определения сущностей и отношений, является основой для установления *пула атрибутов*.

Определение. *Пул атрибутов* является списком потенциально жизнеспособных атрибутов, замеченных в контексте модели.

Этот список будет, по всей вероятности, заметно больше пула

сущностей.

Пример пула атрибутов приведен в таблице 2.6.

Таблица 2.6

Номер	Имя атрибута	Номер исх. данных
1	Номер заказа на покупку	1
2	Код покупателя	2
3	Код продавца	3
4	Код заказа	4
5	Номер замены	5
6	Куда отгружен	6
7	Имя продавца	8
8	Адрес продавца	8
9	Код упаковщика	9
10	Имя упаковщика	9
11	Имя заказчика	11,42
12	Код отдела	12
13	Отгружено через	13
14	Имя покупателя	14
15	Номер заказа на покупку	15
16	Дата подписи документа на отгрузку	16
17	Код контроля качества	17
...

Пул атрибутов является источником имен, используемых в модели. Атрибуты, появившиеся на более поздних стадиях моделирования, добавляются в пул атрибутов и им приписываются уникальные идентифицирующие номера. Затем они развиваются для дальнейшего использования в модели.

2.6.2. Определение владельцев атрибутов

На этом этапе каждый *неключевой атрибут* должен быть приписан к одной сущности-владельцу.

Для многих атрибутов сущности-владельцы очевидны.

Например, разработчик без заминки свяжет атрибут ИМЯ-ПРОДАВЦА с сущностью ПРОДАВЕЦ.

Однако некоторые атрибуты могут вызвать у разработчика трудности при определении их сущностей-владельцев.

Если разработчик не уверен в сущности-владельце для атрибута, он может обратиться к исходному материалу, откуда был выбран атрибут. Это поможет в определении владельца.

На начальной стадии был сформирован список исходных данных, ставший основой пула атрибутов. Этот список указывает разработчику места, где значения представленных атрибутов использовались в исходном материале.

Анализ примеров использования атрибута в исходном материале упрощает поиск сущности-владельца в модели. Он должен иметь в виду, что

главенствующим фактором в определении владельцев атрибутов является вхождение экземпляров атрибутов, представленных отражаемыми в исходном материале значениями атрибутов.

После того, как каждому атрибуту будет назначена сущность-владелец, это назначение должно быть зафиксировано.

2.6.3. Определение атрибутов

Для всех выявленных на данной стадии атрибутов должны быть разработаны *определения атрибутов*.

Здесь также применимы основные принципы построения определений, уже используемых в модели данных (в особенности из стадии определения ключей).

Разработанные определения должны быть точными, специфическими, полными и универсально понимаемыми.

Эти определения атрибутов записываются в том же формате, что и определения атрибутов из стадии 3.

Определение атрибута включает:

- имя атрибута;
- определение атрибута;
- синонимы/псевдонимы атрибута.

Каждому атрибуту должно быть присвоено уникальное имя, поскольку в IDEF1X-модели и к сущностям, и к атрибутам применяется правило «одно и то же имя – один и тот же смысл».

Поэтому разработчик может воспользоваться стандартным подходом к именам атрибутов.

Однако для облегчения чтения при проверке правильности лучше использовать привычные для пользователя естественные названия.

Если встречаются *имена атрибутов*, которые должны удовлетворять строгим правилам языка программирования, то они должны всегда идентифицироваться как псевдонимы или не включаться вовсе.

В определении атрибута разработчик может пожелать установить формат атрибута, например, буквенно-цифровой код, текст, денежные единицы, дату и т.д.

В определении атрибута может быть также установлена область допустимых значений в формате списка (например, понедельник, вторник, среда, четверг, пятница) или диапазона (например, больше нуля, но меньше десяти).

В определении атрибута могут быть также указаны утверждения, включающие несколько атрибутов.

Например, атрибут ОКЛАД_СЛУЖАЩЕГО должен быть больше 20000 рублей или СЛУЖЕБНЫЙ_КОД_СЛУЖАЩЕГО равен двенадцати.

2.6.3. Детализация модели

Теперь необходимо начать детализацию отношений на стадии

определения атрибутов.

Здесь используются те же основные правила, что и на стадии определения ключей.

Правила *необращения в ноль* и *неповторяемости* теперь применяются и к *ключевым*, и к *неключевым атрибутам*.

В результате могут возникнуть некоторые новые сущности. После идентификации этих сущностей должно применяться правило миграции ключей точно так же, как на стадии определения ключей.

Единственное отличие в применении на данной стадии правил *необращения в ноль* и *неповторяемости* заключается в том, что эти правила относятся преимущественно к *неключевым атрибутам*.

На рис. 2.10 проиллюстрировано применение к *неключевому атрибуту* правила *необращения в ноль*.

На рис. 2.11 приведен пример применения к *неключевому атрибуту* правила *неповторяемости*.

Вместо того, чтобы немедленно создавать новые сущности для атрибутов, нарушающих правила детализации, можно отмечать такие нарушения по мере их выявления с тем, чтобы позднее создать новые сущности.

Рядом с именами атрибутов-нарушителей в диаграммах атрибутов могут быть сделаны пометки в скобках (буква N для нарушителей *правила обращения в ноль* и буква R для нарушителей *правила повторяемости*).

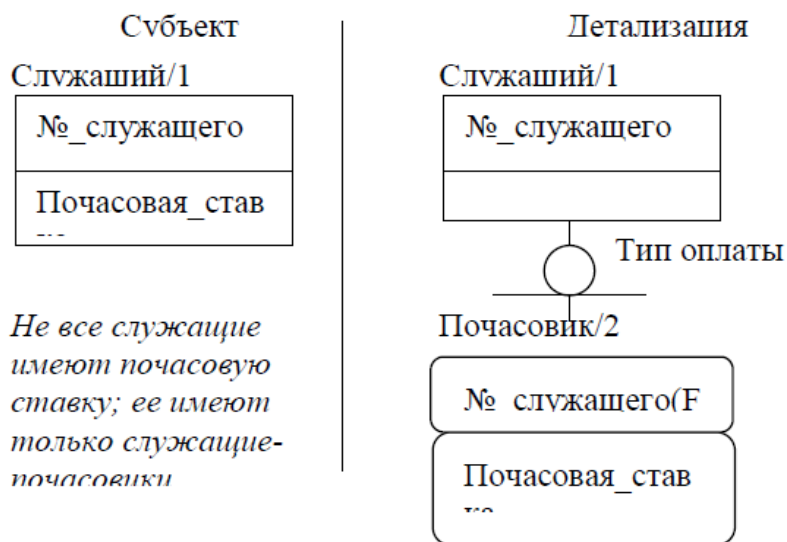


Рис. 2.10. Пример правила обращения в ноль

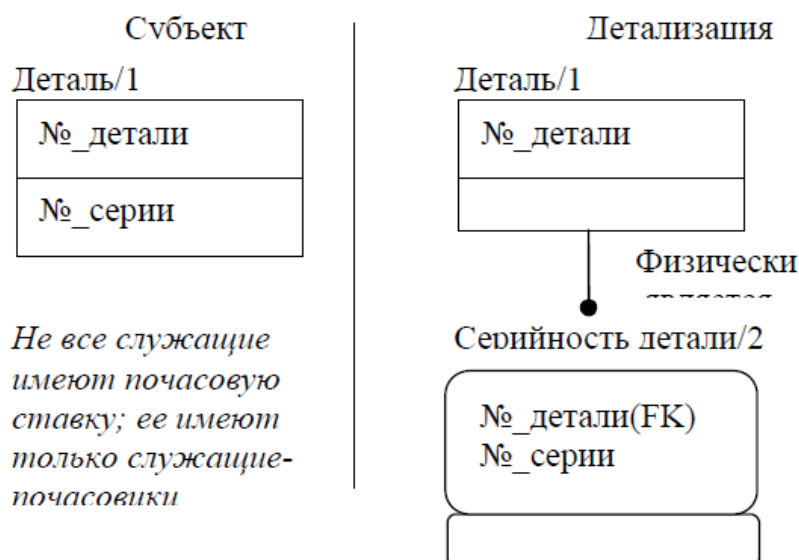


Рис. 2.11. Пример правила неповторяемости

После выявления новых сущностей они должны быть введены в пул сущностей, определены, отражены в матрице отношений и т.д.

Короче говоря, новые сущности должны удовлетворять всем требованиям к документации, созданной на более ранних стадиях, с тем, чтобы их можно было включить в материал данной стадии.

Должна быть также определена принадлежность каждого атрибута в соответствии с *правилом полной функциональной зависимости*.

Это правило утверждает, что *ни одно значение неключевого атрибута, принадлежащего экземпляру сущности, не может быть идентифицировано лишь частью значения ключа данного экземпляра сущности*.

Это правило применимо только к сущностям с составными ключами и эквивалентно второй нормальной форме в реляционной теории.

Все атрибуты модели на данной стадии должны также удовлетворять *правилу отсутствия транзитивной зависимости*.

Это правило требует, чтобы значение принадлежащего экземпляру сущности неключевого атрибута не могло идентифицироваться значением другого принадлежащего экземпляру сущности или наследуемого ею неключевого атрибута.

Простой способ для запоминания правил полной функциональной зависимости и отсутствия транзитивной зависимости можно сформулировать так: *неключевой атрибут должен зависеть от ключа, всего ключа и ни от чего другого, кроме ключа*.

2.6.5. Представление результатов стадии определения атрибутов

После определения атрибутов диаграммы функционального представления могут быть теперь обновлены так, чтобы отразить детали модели, и расширены с тем, чтобы показать неключевые атрибуты.

Неключевые атрибуты перечисляются ниже линии внутри каждого блока сущности.

Для того чтобы внутри блока сущности хватило места, размеры блока могут быть увеличены.

Соответствующие определения и информация для модели должны быть обновлены, чтобы отразить определение неключевых атрибутов и их принадлежности.

Эта дополнительная информация может быть представлена сущностью вместе с ранее определенной информацией.

Теперь каждый набор документов сущности будет содержать:

- определение каждой сущности;
- список первичных, альтернативных и внешних ключевых атрибутов;
- список принадлежащих сущности неключевых атрибутов;
- определение каждого принадлежащего сущности атрибута (как ключевого, так и неключевого);
- список отношений, в которых данная сущность является родительской;
- отношение категоризации;
- идентифицирующие отношения указанного выше типа;
- неидентифицирующие отношения указанного выше типа;
- список отношений, в которых данная сущность является сущностью-потомком;
- отношение категоризации;
- идентифицирующие отношения указанного выше типа;
- неидентифицирующие отношения указанного выше типа;
- утверждения всех двойных путей.

Необязательные диаграммы сущностей также могут быть расширены для указания неключевых атрибутов.

Определения отношений могут быть повторены в комплекте документов для каждой сущности или перечислены отдельно с перекрестными ссылками на эти сущности.

Ключевые и неключевые атрибуты должны быть также перечислены и снабжены перекрестными ссылками на эти сущности

3. Задание курсовой работы по дисциплине «Структурно-функциональное моделирование»

3.1. Цель и задачи курсовой работы

Целью курсовой работы по дисциплине «Структурно-функциональное моделирование» является приобретение студентами практических навыков построения моделей систем с использованием стандартов IDEF0 и IDEF1X.

Задачами выполнения курсовой работы являются:

- углубленное изучение теоретических концепций и разработок, связанных с заданиями курсовой работы;
- приобретение навыков разработки моделей систем с использованием стандартов IDEF0 и IDEF1X;
- развитие ранее приобретенных навыков разработки алгоритмов, использования программных средств, их компонентов, модулей, пакетов расширения, отдельных процедур (функций), автоматизирующих выполнение рутинных операций моделирования систем.

При выполнении курсовой работы по учебной дисциплине «Структурно-функциональное моделирование» необходимо владение материалом следующих учебных дисциплин естественно - научного блока учебного плана специальности/направления 09.03.04 «Программная инженерия», а также дисциплины «Основы системного анализа».

В процессе выполнения курсовой работы студенты закрепляют знания по темам дисциплины «Структурно-функциональное моделирование» и пополняют знания по следующим темам курсов «Основы системного анализа», «Моделирование систем» и «Имитационное моделирование».

Приступая к выполнению курсовой работы, студент в течение первой недели семестра обязан в полном объеме изучить данные методические указания. Итоговый контроль результатов выполнения курсовой работы проводится в форме её защиты.

3.2. Тематика, структура и основные этапы выполнения курсовой работы

Темы курсовой работы (индивидуальные задания) приведены в приложении 1.

3.2.1. Обязанности студента в процессе выполнения курсовой работы

Курсовая работа является учебно-аттестационной работой на заданную тему, выполненной в форме рукописи, обладающей оригинальностью, целенаправленностью и завершающуюся конкретными выводами и предложениями.

В процессе выполнения курсовой работы студент должен:

- сформулировать постановку цели и задач работы;

- выполнить индивидуальное задание курсовой работы, включающее три раздела;
- привести расчеты, иллюстрирующие результаты выполнения заданий курсовой работы;
- сформулировать выводы и заключение, обобщающие полученные результаты и возможность их практического применения;
- разработать и продемонстрировать комплекс моделей системы в среде, используемой для выполнения заданий курсовой работы.

Учебный характер курсовой работы и ограниченность времени самостоятельной работы, выделяемого на её выполнение, допускает условность её результатов. Применимость методических подходов, оценок, выводов и предложений курсовой работы ограничивается условной учебной ситуацией, которая не обязательно должна иметь прямое отношение к реальности. *Вместе с тем наличие у курсовой работы практической значимости является её достоинством, которое может положительно влиять на её оценку.*

Приступая к выполнению курсовой работы, следует отдавать себе отчёт в том, что внеаудиторное учебное время, предусмотренное учебным планом подготовки бакалавров *прикладной информатики*, достаточно (при его строго целевом использовании) для выполнения курсовой работы студентом, успешно выдержавшим вступительные испытания и освоившим предшествующие курсы, как минимум на удовлетворительную оценку.

Пока курсовая работа не зарегистрирована (при условии соблюдения установленных сроков её подготовки и регистрации), студент имеет право получать консультации по вопросу о том, соответствует ли она запланированному баллу и какие недочёты должны быть исправлены для повышения оценки.

Преподаватель осуществляет контроль за использованием по назначению времени самостоятельной работы, выделяемого на выполнение курсовой работы. В его обязанности входит своевременно информировать деканат о нарушениях графика выполнения курсовой работы, обусловленных отклонением студента от самостоятельной работы в объёмах, предусмотренных учебным планом и настоящими методическими указаниями.

Студенты вправе объединять свои усилия для решения отдельных задач курсовой работы — например, для подбора литературы, подготовки и обработки данных или использования программного продукта. В этих случаях в курсовой работе явно должен быть отмечен факт совместной работы студентов над конкретными задачами и чётко указан личный вклад автора курсовой работы в их решение.

Курсовая работа предусматривает обязательную разработку комплекса моделей с использованием рекомендуемого программного продукта.

Электронный носитель данных с комплексом моделей прилагается к рукописи курсовой работы.

По своему правовому статусу курсовая работа является объектом авторского права. Неимущественное авторское право в полном объеме принадлежит выполнившей её студенту, охраняется законами Российской Федерации и международными соглашениями. Университет имеет право использования курсовой работы или любой её части в оригинальном, отредактированном либо видоизменённом виде в учебном процессе, в том числе при подготовке любых учебно-методических материалов и публикаций, без предварительного уведомления автора при условии соблюдения его неимущественных авторских прав, в том числе права на указание авторской принадлежности используемых материалов. Другие имущественные права по поводу курсовых работ регулируются действующим законодательством Российской Федерации и правовыми актами, регламентирующими отношения между студентом и университетом.

3.2.2. Структура курсовой работы

Курсовая работа по учебной дисциплине «Структурно-функциональное моделирование» имеет теоретико-прикладной характер.

Объём курсовой работы, не считая библиографического списка и приложений, составляет не более 45 - 55 страниц компьютерного текста, набранного в соответствии с требованиями, изложенными ниже в разделе «Оформление курсовой работы».

Рекомендуемый объём — 45 - 55 страниц (таблица 3.1).

Таблица 3.1

№ п/п	Элементы структуры курсовой работы	Объем (примерный), страниц
1	Титульный лист	1
2	Содержание	1
3	Введение	1
4	Глава 1	15-25
5	Глава 2	20-25
6	Заключение и выводы	1
7	Список использованной литературы (Литература)	Не менее 5 источников
ИТОГО		Не более 55
Приложения		По необходимости

По своей структуре курсовая работа должна содержать:

- введение, в котором формулируются цели и задачи работы;
- краткое изложение элементов теоретических данных, необходимых для выполнения каждого из разделов задания курсовой работы;
- сопроводительную записку, отражающую разработку моделей в рамках выполнения каждого из разделов задания;

- заключение, в котором содержатся выводы и рекомендации относительно возможностей практического применения материалов работы;
- список используемых источников;
- приложения (если требуются).

Приводимые ниже методические рекомендации по структуризации курсовой работы не являются обязательными. Тем не менее, во избежание недоразумений на этапе защиты, расхождения с ними должны быть мотивированы студентом и согласованы с преподавателем.

Введение

Введение курсовой работы содержит:

- формулировку цели работы;
- перечень задач работы;
- при необходимости — дополнительные сведения, например, информацию о практическом применении полученных результатов, в том числе в учебном процессе, об опубликовании полученных результатов, об участии в студенческих научных конференциях, конкурсах студенческих научных работ и т.п. с данной темой.

Объём введения, как правило, не должен превышать одной – двух страницы.

Во введении не принято использовать таблицы, рисунки и формулы.

Структура и содержание первой главы

Первая глава курсовой работы включает в себе выполнения первого раздела задания работы, посвященному построению в стандарте IDEF0 модели, отражающей предметную область, описание которой приведено ниже в разделе индивидуальных заданий (приложение 1).

Структура данной главы включает в себя:

1. Формулировку задания и задачи, отражающей это задание и выполняемой в рамках данной главы.
2. Исходные данные – описание предметной области (приложение 1).
3. Краткие теоретические данные из методики применения стандарта IDEF0 для моделирования систем, которые будут использоваться непосредственно для выполнения задания.
4. Результаты применения стандарта IDEF0 для моделирования систем, иллюстрирующие выполнение задания, в том числе обоснование:
 - выбора параметров ICOM, исходя из описания предметной области;
 - точки зрения при разработке IDEF0-модели;
 - выбора функций, включаемых в IDEF0-модель;
 - «глубины» декомпозиции диаграммы A-0.

Здесь же приводятся словесное описание функционирования каждой функции диаграммы IDEF0-модели, ее глоссарий, результаты проверки её

синтаксиса, IDEF0-отчет.

Глубина декомпозиции диаграммы А-0 должна быть не менее 2-х уровней и по возможности давать максимально возможное представление о функциональном аспекте предметной области.

5. Выводы и заключение, характеризующие и обобщающие полученные результаты.

Объем первой главы составляет 15 – 25 стр.

Структура и содержание второй главы

Вторая глава курсовой работы включает в себе выполнения первого раздела задания работы, посвященному построению в стандарте IDEF1X модели, связанной с построенной в первом разделе курсовой работы IDEF0-модели и отражающей предметную область, описание которой приведено ниже в разделе индивидуальных заданий (приложение).

Структура данной главы включает в себя:

1. Формулировку задачи, выполняемой в рамках данной главы и отражающей общее задание курсовой работы.
2. Краткие теоретические данные из методики применения стандарта IDEF1X для моделирования систем, которые будут использоваться непосредственно для выполнения задания.
3. Результаты применения стандарта IDEF1X для моделирования систем, иллюстрирующие выполнение задания, в том числе:
 - обоснование выбора сущностей IDEF1X-модели;
 - определение отношений между сущностями и разрешение неспецифических отношений при их наличии;
 - определение ключевых и неключевых атрибутов сущностей;
 - проверка корректности идентифицированных ключевых и неключевых атрибутов сущностей;
 - диаграмму разработанной IDEF1X-модели и отчет о модели.
4. Выводы и заключение, характеризующие и обобщающие полученные результаты.

Объем первой главы составляет 20 – 25 стр.

Содержание библиографического списка

Библиографический список должен отражать необходимую степень изученности классических работ по исследуемой проблеме.

Ссылки на учебную литературу рекомендуется использовать лишь в случаях недоступности соответствующего научного первоисточника.

Рекомендуется использовать в качестве источников диссертации и их авторефераты.

Если при выполнении данной курсовой работы использовались результаты, полученные в других курсовых или дипломных работах или проектах, в библиографическом списке необходимо указывать описания данных источников наряду с остальной литературой.

Библиографический список должен содержать не менее 5 источников (не считая ссылок на курсовые и дипломные работы или проекты), в том числе рекомендуются:

- не менее одной научной статьи, опубликованных в научных журналах и других научных изданиях в течение последних десяти лет;
- не менее одной монографии, изданных за последние десять лет.

Библиографические описания ресурсов из сети Internet не должны составлять более трети от общего числа источников. Сетевой адрес документа (URL) должен точно указывать на используемый документ или на страницу, содержащую именно ту часть документа, которая используется в курсовой работе (это требование не распространяется на электронные словари и энциклопедии, см. ниже). Ссылки на целые сайты не засчитываются в общее количество ссылок и рассматриваются как ошибки при оформлении библиографического списка.

Не разрешается указывать в библиографическом списке:

- лекции (кроме опубликованных, в том числе в сети Internet);
- компьютерные презентации;
- программные средства;
- базы данных;

Каждый используемый словарь либо энциклопедия (в том числе электронные словари и энциклопедии) оформляется одной записью библиографического списка независимо от того, сколько статей из него использовано.

Каждая статья из журналов и сборников научных трудов оформляется отдельной записью независимо от того, опубликованы ли используемые статьи в одном и том же издании или в разных.

В тексте курсовой работы обязательно должны присутствовать ссылки на каждое издание, включённое в библиографический список.

Приложения

В приложения выносятся:

- исходные данные для выполняемых расчётов и их промежуточные результаты;
- таблицы и рисунки, предназначенные для иллюстрации процедур вычислений и не содержащие данных, непосредственно приводящих к значимым выводам;
- любые таблицы и рисунки (включая блок-схемы) в альбомной ориентации либо занимающие более 2/3 страницы;
- подтверждающие документы (копии справок, авторских свидетельств и др.), факсимиле, ксерокопии и сканированные материалы, необходимые для решения задач курсовой работы.

Объём приложений не регламентируется. Тем не менее, чрезмерно объёмные материалы рекомендуется не включать в приложения, а прилагать на электронном носителе (с соответствующей ссылкой в тексте курсовой

работы).

3.3. Требования к оформлению курсовой работы

Курсовая работа печатается на стандартных листах формата А4 (210 297 мм) с соблюдением полей: слева — 30 мм; справа — 10 мм; сверху — 20 мм; снизу — 25 мм.

Распечатанный манускрипт либо переплетается, либо надёжно сшивается (скрепляется) и помещается в папку.

Текст курсовой работы набирается, как правило, шрифтом Times New Roman. Размер шрифта — 14 pt, межстрочный интервал — полуторный. Отступ первой строки абзаца — 12,5 мм. Отступы слева и справа от абзаца, перед абзацем и после него отсутствуют. Висячие строки запрещены. При использовании текстового процессора Microsoft Word для этой цели следует переопределить стиль Обычный в соответствии с вышеуказанными требованиями.

Нумерация страниц должна быть сквозной, включая приложения.

Первым считается титульный лист (приложение 2), но на нём номер не ставится. Нумерация остальных страниц проставляется на расстоянии 15 мм от нижнего края листа по центру относительно полей.

Рукопись курсовой работы имеет следующую структуру:

- 1) титульный лист;
- 2) содержание;
- 3) основной текст (введение, главы 1...3, выводы и предложения);
- 4) библиографический список;
- 5) приложения.

На последней странице курсовой работы ставятся дата окончания работы и подпись автора. Оставляется один чистый лист бумаги для рецензии, замечаний преподавателя.

Титульный лист и содержание курсовой работы

Титульный лист оформляется в соответствии с примером, приведённым в приложении 3. На титульном листе текст «курсовая работа на тему ...» набирается шрифтом размером 18 pt, остальной — размером 14 pt. Выделение полужирным шрифтом и курсивом не используется.

Шаблон титульного листа курсовой работы может быть изменён в соответствии с указаниями учебно-методического управления университета. В случае изменения требований к оформлению титульного листа преподаватель своевременно сообщает об этом студентам и предоставляет новый образец в печатной либо электронной форме.

Содержание оформляется аналогично содержанию настоящих методических указаний. В него включаются наименования глав и разделов, библиографического списка и приложений. Содержание формируется на основе текста, содержащегося в абзацах, оформленных стилями Заголовок 1 и Заголовок 2.

Для правильного форматирования содержания необходимо переопределить стили его элементов следующим образом.

Оглавление 1: Абзацный отступ слева: 0 см, справа — 1 см, выступ: 1 см. Выравнивание по левому краю. Интервал перед абзацем: 3 пт, после абзаца — 3 пт. Позиция табуляции: 17 см с выравниванием по правому краю и заполнителем «точки». Основан на стиле Обычный.

Оглавление 2: Абзацный отступ слева: 1 см, справа — 1 см, выступ: 1 см. Выравнивание по левому краю. Позиция табуляции: 17 см с выравниванием по правому краю и заполнителем «точки». Основан на стиле Обычный.

В основном тексте курсовой работы заголовки глав, разделов и параграфов следует печатать с прописной буквы без точки в конце, не подчеркивая. Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой.

Заголовком содержания служит слово «Содержание», оформленное стилем Заголовок 1, переопределённым в соответствии с указаниями, изложенными на с. 30.

Основной текст курсовой работы

Основной текст курсовой работы состоит из введения, трёх (как правило) глав, выводов и предложений. Главы состоят из разделов, а разделы, при необходимости, — из параграфов. В секциях «Введение» и «Выводы и предложения» разделы и параграфы не выделяют.

Для оформления названия главы в Microsoft Word используется стиль Заголовок 1, который следует предварительно переопределить, задав следующий формат:

- начинать с новой страницы;
- не разрывать;
- не отрывать от следующего;
- запретить автоматический перенос слов;
- отступы слева и справа — 1 см;
- отступ (выступ) первой строки отсутствует;
- выравнивание по центру;
- интервал после абзаца — 18 пт;
- шрифт полужирный;
- все символы прописные;
- основан на стиле Обычный;
- стиль следующего абзаца — Обычный.

Для оформления названия раздела используется стиль Заголовок 2, для которого необходимо установить следующие параметры:

- не разрывать;
- не отрывать от следующего;
- запретить автоматический перенос слов;
- отступы слева и справа — 1 см;

- отступ (выступ) первой строки отсутствует;
- выравнивание по центру;
- интервал перед абзацем — 12 pt;
- интервал после абзаца — 12 pt;
- основан на стиле Обычный;
- стиль следующего абзаца — Обычный.

Для оформления названия параграфа используется стиль Заголовок 3, для которого необходимо установить следующие параметры:

- не разрывать;
- не отрывать от следующего;
- запретить автоматический перенос слов;
- отступы слева и справа — 1,25 см;
- отступ (выступ) первой строки отсутствует;
- выравнивание по левому краю;
- интервал перед абзацем — 3 pt;
- интервал после абзаца — 3 pt;
- основан на стиле Обычный;
- стиль следующего абзаца — Обычный.

Разрешается оформлять заголовки параграфов курсивом или обычным шрифтом (на усмотрение автора). При этом оформление должно быть единообразным во всей курсовой работе.

Рекомендуемый размер параграфа — 1,5...3 страницы. Разрешается не применять разделение всех или некоторых разделов курсовой работы на параграфы. Никогда не следует выделять в главе только один раздел, а в разделе — только один параграф.

Не допускается наличие в курсовой работе глав, разделов или параграфов, содержащих только таблицы, рисунки или формулы без поясняющего текста.

Точка после заголовков глав, разделов и параграфов не ставится.

В тексте курсовой работы не рекомендуется использовать подчёркивание (за исключением цитирования источников, в которых подчёркивание используется). Полужирным шрифтом выделяют определяемые понятия в определениях. Курсив используется выделения фраз (как правило, не длиннее строки), к которым необходимо привлечь внимание читателя даже при беглом просмотре текста для правильного понимания сути сказанного. Не рекомендуется использовать выделение курсивом чаще трёх раз на странице. Не следует выделять курсивом фрагменты текста длиннее одной-двух строк.

Формулы

При работе в Microsoft Word формулы набираются при помощи встроенного редактора формул либо Mathsoft Mathtype. Формулы длиной более строки либо не уместяющиеся по высоте в полуторный межстрочный интервал необходимо выносить в отдельный абзац и нумеровать

Нумерация формул должна быть сквозной по всей работе. Для её создания можно воспользоваться средствами программы Mathsoft Mathtype.

Ссылка на формулу в тексте курсовой работы представляет собой её номер, заключённый в круглые скобки: например, (1).

В формулах и в тексте:

- числовые значения и имена функций набираются обычным шрифтом;
- имена скалярных переменных (в том числе дискретных и лингвистических) набираются курсивом строчными латинскими либо греческими буквами;
- имена множеств печатаются либо заглавными греческими буквами, набранными обычным шрифтом, либо заглавными латинскими буквами, набранными курсивом (оба способа можно сочетать по необходимости);
- имена векторов печатаются строчными латинскими либо греческими буквами, набранными полужирным шрифтом;

Таблицы

Основное требование к оформлению таблицы состоит в том, что смысл всех представленных в ней данных должен быть полностью ясен из самой таблицы, без обращения к ссылающемуся на неё тексту.

Это достигается умелыми формулировками названия таблицы, названий строк и столбцов, а при необходимости — примечаний к таблице.

Таблицы предназначены для сопоставления данных. Поэтому использовать таблицы, содержащие менее двух строк или менее двух столбцов, не считая содержащих названия и (или) номера строк (столбцов), не следует. В подобных случаях данные приводятся в тексте в форме списка. Иногда для представления подобных данных может оказаться оправданным использование диаграммы.

Создавая таблицу, следует стремиться обойтись минимально необходимым числом строк и столбцов. Лучше всего воспринимаются таблицы, содержащие 3...5 столбцов и 4...7 строк данных.

Шаблон оформления таблицы приведён ниже (см. таблица 3.2).

Таблица 3.2

Пример оформления таблицы

Название столбца	Название группы столбцов		Название группы столбцов	
	Название столбца	Название столбца	Название столбца	Название столбца
Название строки	данные	данные	...	данные
Название строки	—	данные	данные	данные
Название строки	данные	×	данные	×

Примечание: текст примечания.

Источник: [ссылка на источник].

В таблицах, включая их номер, название, примечание (если есть) и ссылку на источник, используется одинарный межстрочный интервал.

Строка, содержащая номер таблицы, выравнивается вправо, название — по центру. Интервалы перед абзацем, содержащим номер таблицы, и после абзаца, содержащего ссылку на источник, должны составлять 18 pt.

После абзаца, содержащего название таблицы, предусматривается интервал 9 pt. Перенос слов в названии таблицы не разрешается. Для всей таблицы — от номера до примечания, исключая лишь ссылку на источник — должен быть установлен формат абзаца «не отрывать от следующего».

Нумерация таблиц в курсовой работе сплошная.

Числовые данные, приводимые в таблицах, должны содержать от 3 до 5 значащих цифр (за исключением случаев, когда большая точность имеет принципиальное значение). Единицы измерения показателей выбираются соответственно.

Точки после названий таблицы, её строк и столбцов не ставятся (если только название не заканчивается общепринятой аббревиатурой, завершающейся точкой). В названиях строк и столбцов не следует использовать сокращения, кроме общепринятых. При крайней необходимости сокращений их следует пояснить в примечании к таблице.

В необходимых случаях разрешается использовать шрифт размером 12 и даже 10 pt, а также поворот текста на 90° в названиях столбцов. Выравнивание текста по ширине в таблице не допускается.

В таблице не должно быть пустых клеток. Клетки таблицы, не подлежащие заполнению (не имеющие смысла), должны содержать знак «X». В клетки, содержащие нулевое значение или результат деления на ноль, помещают прочерк «-». Если клетка содержит малое значение, отличное от нуля, в неё помещают значение 0,000 (число знаков дробной части соответствует другим клеткам столбца и указывает на точность показателя). Если данные для заполнения клетки отсутствуют, в неё помещается эллипс «...».

В таблице не должно быть ни одной клетки, содержащей числовое значение, единица измерения которого не указана. Исключение составляют случаи, когда масштаб показателя или единица его измерения прямо следует из названия строки или столбца (например, «Процент отходов»). Единицы измерения могут указываться в названии таблицы (если они одинаковы во всех её клетках, кроме содержащих проценты), в названиях столбцов либо в названиях строк. Совмещение этих способов не допускается, так как приводит к путанице.

Источники данных таблицы указываются обязательно, следуя правилам, описанным ниже в параграфе «Библиографический список и ссылки на библиографические источники». Если данные, представленные в таблице, — результат труда автора курсовой работы, пишут «Источник: расчёты (наблюдения, опрос и т.п.), выполненные автором» либо «Источник: расчёты автора на основе данных, приведённых в [1]», где 1 — номер

источника по библиографическому списку.

Ссылки на таблицы в тексте курсовой работы оформляются следующим образом: табл. 3. Обычно таблицу размещают ниже первой ссылки на неё либо на той же странице, где находится ссылка, либо, если это невозможно, — на следующей. Если в коротком фрагменте текста имеются ссылки на большое количество таблиц, из-за чего вышеприведённые требования выполнить не удаётся, следует перенести часть таблиц в приложение к курсовой работе.

Расстановка номеров таблиц и ссылок на таблицы вручную не разрешается.

Таблицы размером более 2/3 страницы либо набранные в альбомной ориентации выносятся в приложение к курсовой работе.

Использовать обтекание таблиц текстом в рукописях курсовых работ не разрешается.

Рисунки и диаграммы

Рисунки и диаграммы должны быть выполнены достаточно крупно, чётко и разборчиво. Основное требование к рисунку и диаграмме состоит в том, что он, при достаточной разборчивости всех деталей рисунка занимать меньше места, чем представление той же информации текстом или таблицей.

Выбирая форму диаграммы, следует иметь в виду, что результаты группировки лучше всего представлять в форме столбцов, процесс или непрерывную функциональную зависимость предпочтительно отображать линией, а структуру — либо круговой (кольцевой) диаграммой, либо (в динамике) диаграммой площадей.

Для всех числовых значений на диаграмме должны быть обязательно указаны единицы измерения. Оси координат должны быть подписаны даже в том случае, когда об их смысле можно догадаться по названию рисунка.

Рисунок или диаграмму, как правило, размещают в отдельном абзаце текста, для чего им приписывается атрибут размещения «В тексте».

Абзац, содержащий рисунок или диаграмму, должен быть выровнен по центру, иметь атрибут «не отрывать от следующего» и отступы: 18 pt перед абзацем и 9 pt после.

Точка после названия рисунка не ставится, если только название не заканчивается общепринятой аббревиатурой, требующей точки.

Ссылка на рисунок имеет следующий вид: рис. 1, где 1 — номер рисунка, являющийся результатом вычисления поля.

Нумерация рисунков и расстановка на них ссылок вручную не разрешается.

Для заимствованных рисунков и диаграмм необходимо указание источника. Источник указывается в отдельном абзаце, предшествующем названию рисунка. Для этого абзаца устанавливается атрибут «не отрывать от следующего».

Использовать обтекание рисунков и диаграмм текстом не

рекомендуется.

Библиографический список и ссылки на библиографические источники

Список использованной литературы должен иметь заголовок «Библиографический список», оформленный стилем Заголовок 1 с отключённой нумерацией заголовков. Источники в библиографическом списке располагаются в алфавитном порядке первого слова библиографического описания: вначале русскоязычные, затем иностранные. При большом количестве источников разрешается (но не рекомендуется) их группировать удобным для автора и читателя образом: например, нормативно-правовые акты, статистические сборники, научная литература, учебная литература. Группы в этом случае разделяют соответствующими заголовками, оформленными стилем Заголовок 3 с отключённой нумерацией заголовков. Нумерация библиографических описаний при использовании группировки должна оставаться сплошной.

Как правило, записи библиографического списка оформляют стилем Обычный с автоматической нумерацией списка. При этом следят за тем, чтобы левая граница текста списков отстояла от поля не более чем на 1,25 см. В библиографическом списке рекомендуется использовать выравнивание абзацев по левому краю, а не по ширине.

Структура библиографического описания следующая:

<Автор(ы)> <Название публикации> // <Название издания, номер выпуска> / <Дополнительные сведения об издании>. <Место издания>: <Издательство>, <Год издания>. — <Номера страниц, число страниц либо сетевой адрес (URL) с указанием даты доступа>.

Отдельные элементы (поля) описания могут быть опущены (вместе с предшествующими им разделителями), если они отсутствуют или неизвестны. Название публикации не может быть опущено ни при каких обстоятельствах.

Если издание имеет более трёх авторов, указывают первых трёх, указывая вслед за ними «и др.». Если авторство принадлежит организации или большому авторскому коллективу, оно указывается в поле «Дополнительные сведения об издании». Если издание подготовлено одним или несколькими составителями от имени организации, фамилии и инициалы составителей также указываются в этом поле. Здесь же указываются фамилии научных редакторов изданий (если имеются), а также сведения о переводе издания с иностранных языков.

Поле «Название издания» заполняется при библиографическом описании статей или произведений в журналах, сборниках, газетах, хрестоматиях, на тематических сайтах и т.д. Если описываемое издание представляет собой книгу, это поле не заполняется.

Для отечественных изданий в поле «Место издания» указывается только город, для зарубежных — город и страна (если они указаны на титульном листе издания). Общепринятые сокращения для места издания: М.

— Москва, СПб. — Санкт-Петербург, Л. — Ленинград. Для журналов место издания, как правило, не указывается.

Примеры библиографических описаний можно найти в электронных каталогах научных библиотек.

Автор курсовой работы несёт законодательно установленную ответственность за точность указания сведений об авторах используемых публикаций.

Ссылки на библиографические описания в тексте курсовой работы представляют собой их номера в библиографическом списке, заключённые в квадратные скобки: например, [1], [2-4], [2,5,7]. Допускается указание в квадратных скобках номера конкретной страницы, на которую ссылается автор курсовой работы: [5, с. 27].

Приложения

Заголовком приложений служит слово «Приложения» (или «Приложение», если оно единственное), оформленное стилем Заголовок 1 и выровненное по правому краю. Каждое отдельное приложение имеет заголовок, оформляемый стилем Заголовок 2

Ссылка на приложение оформляется следующим образом: приложение 2. Например: «Блок-схема алгоритма дискретизации значений коэффициента фондовооруженности приведена в приложении 2». В приложениях при необходимости допускается вставка нестандартных листов, размер которых отличается от формата А4, при условии, что они сложены и закреплены надлежащим образом, не нарушая формат рукописи.

Все страницы приложения должны быть пронумерованы в продолжение нумерации страниц предшествующих разделов рукописи.

3.4. Порядок защиты курсовой работы

Если иное расписание защит курсовых работ не установлено деканатом учётно-финансового факультета либо учебно-методическим управлением университета, защита проводится строго в течение 12-й и 13-й недель семестра во внеаудиторное время. Студенты должны быть уведомлены о датах и времени защиты курсовых не позднее чем за три рабочих дня, считая за рабочие те дни, в которые студент данной группы обязан присутствовать в университете. В случае неявки на защиту курсовой в ведомости в графе оценок проставляется «не явился» («не явилась»). Повторная защита назначается только с разрешения деканата факультета и в установленный им срок.

Не позднее чем за три рабочих дня до назначенного срока защиты студенты получают перечень замечаний преподавателя по содержанию курсовой работы. Задача студента в процессе защиты состоит в том, чтобы дать краткие и убедительные ответы или возражения на полученные

замечания. В случае согласия с замечанием студент должен указать, как данное замечание влияет на конечные выводы работы и каким конкретно образом следует его учесть. Студенту рекомендуется представить конкретные результаты (литературные источники, расчёты и выводы), показывающие, каким образом учтено замечание, с которым он соглашается. Если в процессе защиты студент пересматривает некоторые выводы курсовой работы и представляет убедительные подтверждения новым выводам, недостоверность прежних выводов курсовой работы не рассматривается в качестве основания для снижения оценки, за исключением случаев, когда прежние выводы свидетельствуют о серьёзных пробелах в знаниях студента по изучаемому курсу.

Для ответа на замечания преподавателя студенту выделяется до 5 мин. времени. Если объём и содержание замечаний требуют большего времени для защиты курсового проекта, вопрос о регламенте должен быть согласован с преподавателем до начала защиты.

Процедура защиты курсовой работы по дисциплине «Моделирование социально-экономических систем», как правило, не предусматривает устного сообщения о содержании курсовой работы. Исключение может быть сделано для курсовых работ, которые по своему содержанию и полученным результатам представляют существенный интерес для ознакомления с ними всех студентов группы. В этом случае преподаватель заранее (в вышеуказанные сроки) предупреждает студента о целесообразности подготовки сообщения продолжительностью 5...10 мин.

4. Пример разработки моделей в стандартах IDEF0 и IDEF1X

4.1. Описание предметной области «Строительство дома»

Средняя строительная компания занимается малоэтажным домостроительством. Компания использует уже готовые проектные решения для возведения малоэтажных домов под ключ. Компания располагает штатом строительных рабочих, которые в состоянии выполнить все виды работ на строительстве дома от возведения фундамента до отделочных работ. Компания интенсивно развивается на фоне роста заказов на возведение малоэтажных домов и заинтересована во внедрении информационных технологий, позволяющих осуществлять оперативный мониторинг выполненных и выполняемых строительных работ (завершенного и незавершенного строительства) и загруженности своих основных рабочих-строителей. Вы работаете в составе группы IT-специалистов, которых пригласила компания для адаптации уже имеющихся решений в области корпоративных систем к ее деятельности. Вам в рамках работы данной группы поручено разработать первичную информационную модель строительства малоэтажного дома по готовому проекту. В строительстве участвуют строители со следующими специальностями: каменщики, плотники, кровельщики и мастера по отделке – отделочники. При строительстве используются обычные строительные материалы. При разработке первичной информационной модели строительства малоэтажного дома руководством Вашей группы рекомендовано использовать IDEF стандарты.

4.2. Построение модели строительства малоэтажного дома в стандарте IDEF0

Модель строительства малоэтажного дома (далее дом) в стандарте IDEF0 должна отразить следующие ключевые функции процесса строительства:

- возведение фундамента дома;
- возведение стен дома;
- выполнение работ по строительству крыши дома;
- выполнение отделочных работ внутри и снаружи дома.

После выполнения всех перечисленных работ дом считается построенным, регистрируется в соответствующих государственных органах (дому присваивается адрес) и сдается/продается заказчику/покупателю.

Цель разработки модели – определить действия, необходимые для постройки дома.

Точку зрения при разработке данной модели предлагается выбрать, как точку зрения прораба – руководителя строительством нижнего звена.

4.2.1. Построение диаграммы А-0

В качестве основной функции, которая является предметом моделирования, в соответствии с принятой целью разработки модели и выбранной точкой зрения предлагается принять функцию «Построить дом».

В качестве ИСОМ параметров разрабатываемой модели предлагается принять следующие.

Входом (Input) функции «Построить дом» являются материалы, которые используются для строительства дома.

Выходом (output) функции «Построить дом» является непосредственно построенный «дом».

Контролирующим параметром (Control) для функции «Построить дом», т.е. для строительства дома как такового является «проект дома».

В качестве механизма (Mechanism) обеспечивающего реализацию функции «Построить дом» и соответственно строительство дома выступают «строители», т.е. рабочие строительных специальностей фирмы осуществляющие непосредственно строительство дома.

Диаграмма А-0 приведена на рис. 4.1.

4.2.2. Построение диаграммы А0

В п. 3.2 обозначены ключевые функции процесса строительства дома, которые необходимо отразить в модели и на которые соответственно будет декомпозирована функция «Построить дом» на диаграмме А0.

Введем обозначения/названия ключевых функций, которые будут использоваться на диаграмме А0:

- возведение фундамента дома – функция «заложить фундамент» дома;
- возведение стен дома – функция «возвести стены» дома;
- выполнение работ по строительству крыши дома – функция «положить крышу» дома;
- выполнение отделочных работ внутри и снаружи дома – функция «выполнить отделочные работы» дома.

Одним из входов перечисленных выше функций будут материалы, используемые при проведении соответствующих строительных работ. Другие входы перечисленных выше функций и представленные на диаграмме А0 (рис. 4.2) отражают последовательно формируемые в процессе строительства дома его составные части: фундамент, стены, крышу и отделку

его интерьера и наружной части.

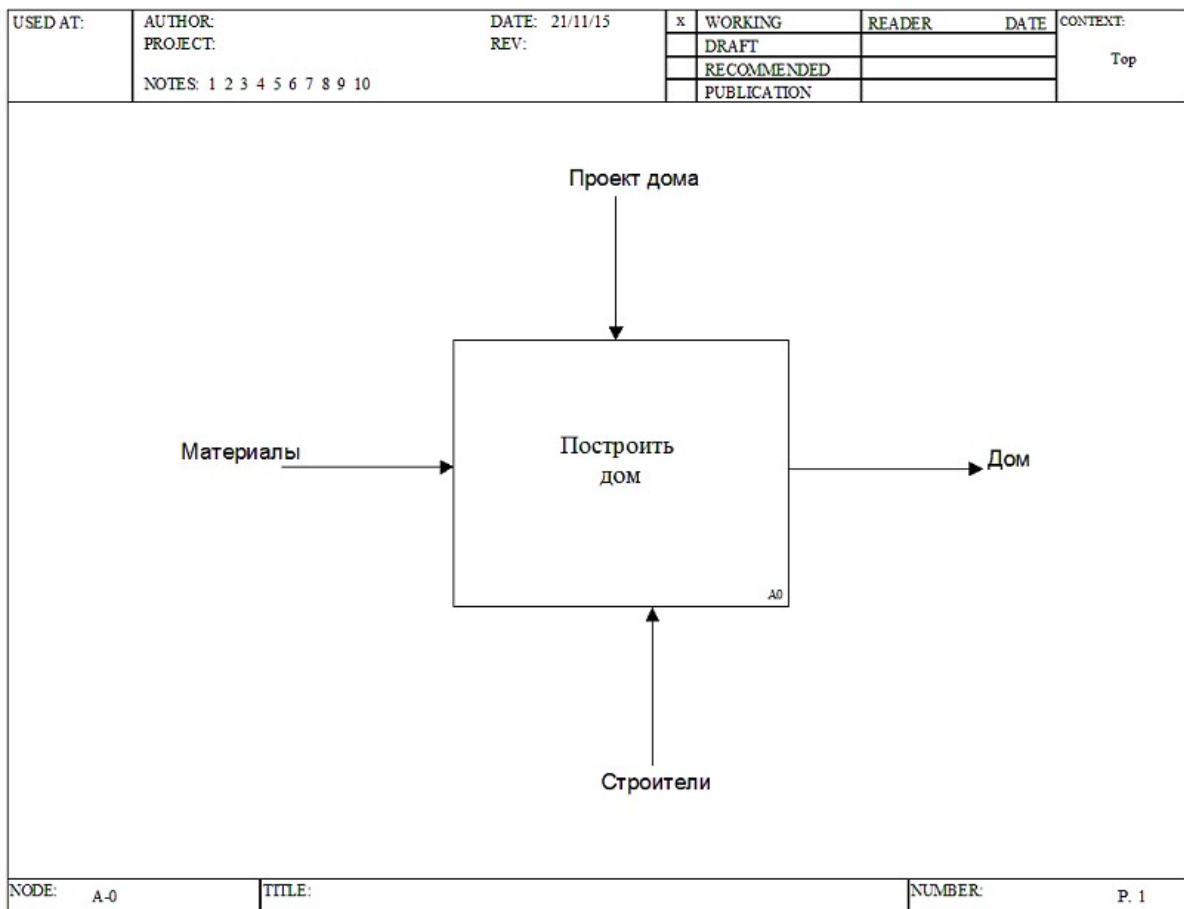


Рис. 4.1

Контролирующим параметром для всех отмеченных функций диаграммы A0 по-прежнему является «Проект дома».

Механизм «Строители», обеспечивающий реализацию рассматриваемых функций, декомпозируется на четыре составляющих, соответствующих специальностям рабочих строителей, осуществляющих возведение дома, а именно на «каменщиков», «плотников», «кровельщиков» и «мастеров по отделке – отделочников».

Проведение более детальной декомпозиции выявленных функций не проводится, т.к. задание по формированию начальной модели строительства малоэтажного дома требует отразить только ключевые функции процесса строительства.

Глоссарий построенной модели приведен ниже.

Glossary Report for:

D:**********\DESIGN~1\idef37\IDEF37\KURS21.IDD

Выполнить отделочные работы (Activity A4)

Возвести стены (Activity A2)
 Дом (Icom)
 Заложить фундамент (Activity A1)
 Каменщики(Icom)
 Крыша (Icom)
 Кровельщики (Icom)
 Материалы (Icom)
 Мастера по отделке (Icom)
 Плотники (Icom)
 Положить крышу(Activity A3)
 Построить дом (Activity A0)
 Проект дома (Icom)
 Стены (Icom)
 Строители (Icom)
 Фундамент (Icom)

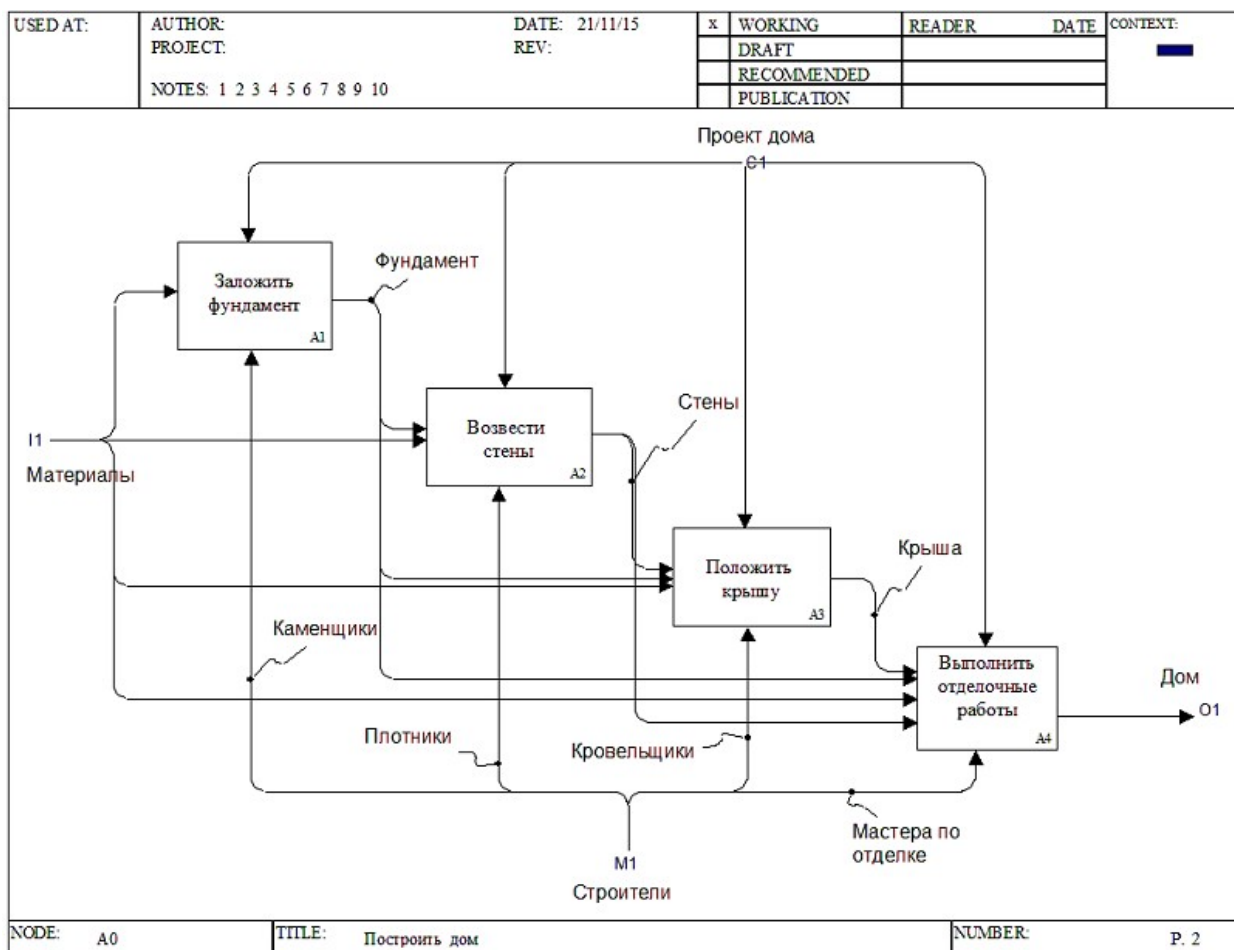


Рис. 4.2

4.3. Построение модели строительства малоэтажного дома в стандарте IDEF1X

В соответствии с обозначенными в разделе 2 стадиями разработки модели в стандарте IDEF1X последовательно выполним данные стадиями применительно к рассматриваемому примеру.

4.3.1. Определение сущностей

Целью данной стадии является выявление и определение сущностей, находящихся в пределах моделируемой проблемной области.

Результатом реализации разработчиком этой стадии является формирование пула сущностей и глоссария определений сущностей.

В качестве кандидатов в сущности (члены пула потенциальных сущностей) для формируемой модели предлагается использовать ICOM разработанной модели в стандарте IDEF0:

1. *Проект дома*
2. *Дом*
3. *Фундамент*
3. *Стены*
5. *Крыша*
6. *Материалы*
7. *Строители*
8. *Каменщики*
9. *Плотники*
10. *Кровельщики*
11. *Мастера по отделке – Отделочники.*

4.3.2. Определение отношений между сущностями

Целью данной стадии является выявление и определение основных отношений между сущностями.

На этой стадии моделирования некоторые отношения могут быть неспецифическими и потребуют дополнительной детализации на последующих стадиях.

Обращаясь к IDEF0-модели, отметим, что функции, связывающие ICOM, трансформируются в отношения между этими сущностями.

Главными результатами этой стадии являются:

- матрица отношений между сущностями;
- определение выявленных отношений между сущностями;
- диаграммы уровней сущностей.

4.3.2.1. Построение матрицы отношений между сущностями.

Как отмечалось ранее, целью IDEF1X является не изображение всех возможных отношений, а определение взаимосвязей между сущностями в терминах, отношений зависимости существования (отношений родитель - потомок).

Такое отношение «родитель – потомок» – это ассоциация между типом *родительской сущности* и типом *сущности-потомка*, при которой каждый экземпляр *родительской сущности* ассоциирован с произвольным (в том числе нулевым) количеством экземпляров *сущности-потомка*, а каждый экземпляр *сущности-потомка* ассоциирован в точности с одним экземпляром *родительской сущности*.

Если *сущность-родитель* и *сущность-потомок* представляют один и тот же объект реального мира, то ***родительская сущность является общей сущностью, а сущность-потомок является сущностью-категорией***.

Для каждого экземпляра *сущности-категории* всегда имеется один экземпляр общей сущности.

Для каждого экземпляра *общей сущности* может существовать ноль или один экземпляр *сущности-категории*

В начале разработки модели часто невозможно представить все отношения как отношения *родитель-потомок* или *отношения категоризации*, т.е. как специфические отношения.

Поэтому неспецифические отношения на этой стадии должны быть преобразованы в специфические.

Неспецифические отношения имеют общую форму – «ноль, один или много – к – ноль, один или много».

Существование любой сущности не зависит от существования другой.

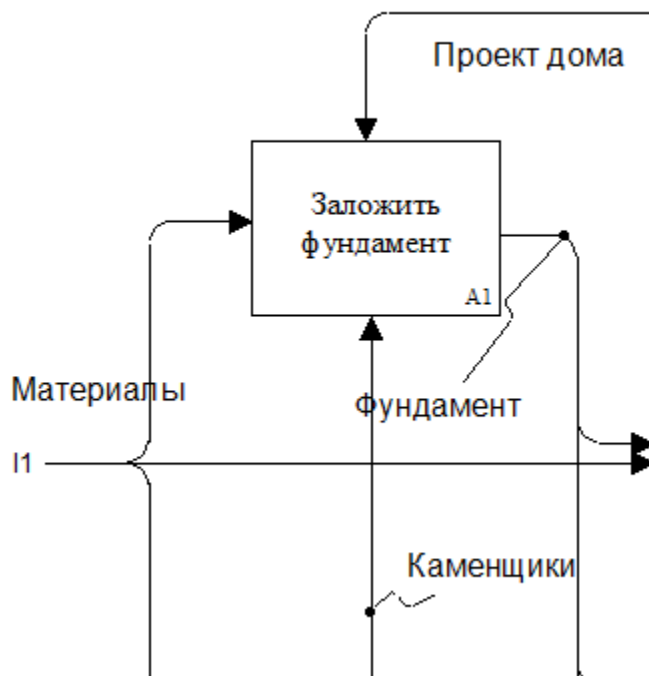
Первым шагом на данной стадии является выявление отношения между элементами различных сущностей.

Для решения этой задачи построим *матрицу отношений* между сущностями (таблица 4.1), включенными в пул потенциальных сущностей (п.4.3.1).

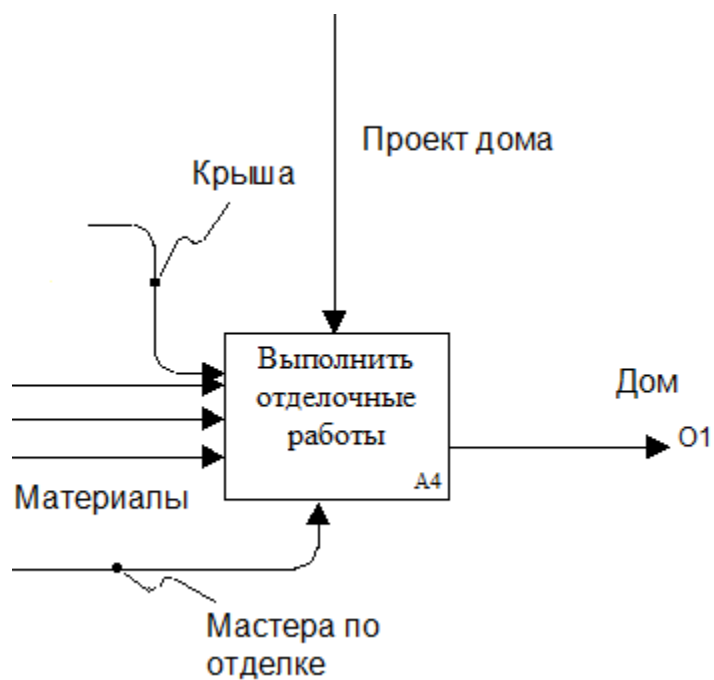
Для удобства и минимизации возможных ошибок при формировании таблицы 3.1 можно «разбить» диаграммы IDEF0 - модели части, каждая из которых будет включать только одну функцию. Для этой одной функции, которая будет характеризовать отношения между потенциальными сущностями, соответствующими ICOM этой функции, необходимо провести заполнение части таблицы 4.1. Подобную процедуру целесообразно

повторить для всех функций IDEF0 – модели.

На рис. 4.3.а, б приведены две функции «Заложить фундамент» и «Выполнить отделочные работы» с соответствующими этим функциям ICOM.



а.



б.

Рис. 4.3.

Таблица 4.1

Матрица «сущность – отношение»

№	Сущность	1	2	3	4	5	6	7	8	9	10	11
1	<i>Проект дома</i>		•	•	•	•	•	•	•	•	•	•
2	<i>Дом</i>	•		•	•	•	•		•	•	•	•
3	<i>Фундамент</i>	•	•		•		•		•		•	•
4	<i>Стены</i>	•	•	•		•	•	•	•		•	•
5	<i>Крыша</i>	•	•		•		•	•		•	•	•
6	<i>Материалы</i>	•	•	•	•	•		•	•	•	•	•
7	<i>Строители</i>	•	•	•	•	•	•		•	•	•	•
6	<i>Каменщики</i>	•	•	•	•		•	•				
9	<i>Кровельщики</i>	•	•		•	•	•	•				
10	<i>Плотники</i>	•	•	•	•	•	•	•				
11	<i>Отделочники</i>	•	•	•	•	•	•	•				

• – обозначение наличия отношения между сущностями, включенными в пул потенциальных сущностей модели.

4.3.2.2. Определение выявленных отношений между сущностями

При определении выявленных отношений между сущностями выполняют следующие процедуры:

- установление зависимостей между сущностями;
- назначение имени отношений между сущностями;
- комментарии к отношениям между сущностями.

При установлении зависимостей между сущностями необходимо проверить рассматриваемое отношение в обоих направлениях связи между сущностями.

Это делается посредством *определения мощности* на каждом конце отношения, связывающего сущности. Для определения *мощности* на одном конце отношения предполагаем существование экземпляра одной из сущностей. Затем определяем, сколько различных экземпляров второй сущности может быть связано с первой. Проводим этот анализ, поменяв сущности ролями.

Результаты установления зависимостей между сущностями, т.е. определения *мощности* отношения между ними, приведены в таблице 4.2, являющейся фрагментом таблицы П.3.1.

Например, в строке 1 таблицы 4.2 приведено отношение между сущностями «Проект дома» и «Дом». Предположим, что существует одна сущность «Проект дома». По одному проекту дома можно построить «много» (N) домов, соответствующих именно этому проекту. Теперь

предположим, что существует только одна сущность «Дом». Очевидно, что одному определенному дому будет соответствовать только одна сущность «Проект дома», в соответствии с которым, т.е. проектом дома, он и был или будет построен.

В столбце 3 таблицы 4.2 совместно приведены «Зависимость отношения» (располагается над чертой) и «Имя отношения» (располагается под чертой). «Имя отношения» сформулировано в соответствии с рекомендациями п. 2.3 Курсивом выделено пояснение, которое может использоваться или не использоваться непосредственно при построении IDEF1X – модели.

Определение рассматриваемого отношения «один к многим» может сопровождаться следующим комментарием или определением: *Каждый проект дома может быть использован для строительства нескольких домов, в том числе и рассматриваемого дома.*

По аналогии со строкой 1 таблицы 4.2 осуществляем заполнение таблицы П.3.1.

Таблица 4.2

Зависимости (мощности) выявленных отношений между потенциальными сущностями

№	Сущность	<u>Зависимость отношения</u> Имя отношения	Сущность
1	2	3	4
1	Проект дома	<u>1 : N</u> Используется для <i>строительства</i>	Дом
2	Проект дома	<u>1 : 1</u> Определяет	Фундамент
3	Проект дома	<u>1 : 1</u> Определяет	Стены
4	Проект дома	<u>1 : 1</u> Определяет	Крыша
5	Проект дома	<u>1 : N</u> Определяет	Материалы
6	Проект дома	<u>1 : N</u> Используют для <i>строительства</i>	Строители
7	Проект дома	<u>1 : N</u> Используют для <i>строительства</i>	Каменщики
8	Проект дома	<u>1 : N</u> Используют для <i>строительства</i>	Кровельщики
10	Проект дома	<u>1 : N</u>	Плотники

		Используют для <i>строительства</i>	
11	Проект дома	1 : N ----- Используют для <i>строительства</i>	Отделочники

В таблицу П.3.1 включен дополнительный столбец, в котором напротив каждого отношения проставлено обозначение, характеризующее вид отношения, а именно отношение «сущность родитель – сущность потомок» (Р-П), отношение категоризации (О К) и неспецифическое отношение (Н О).

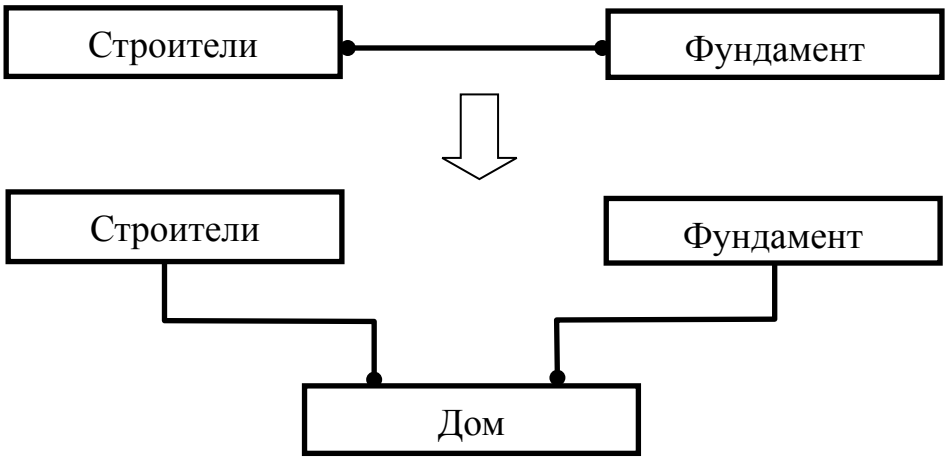
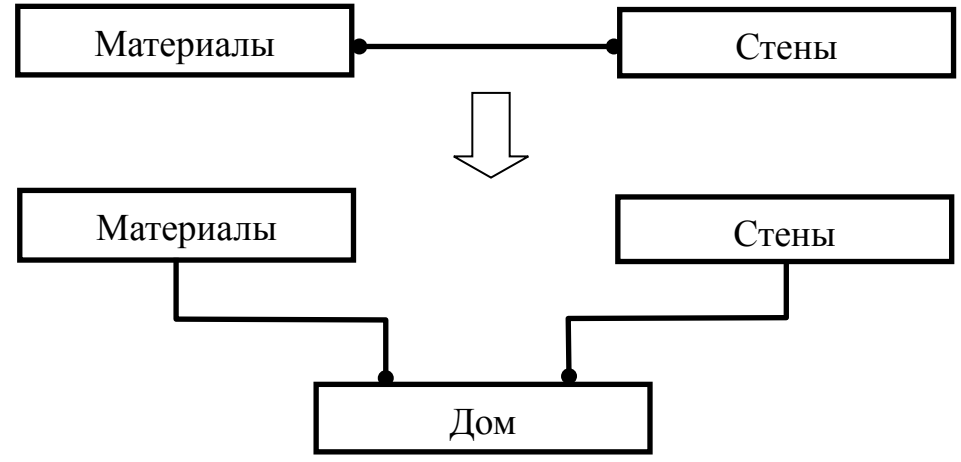
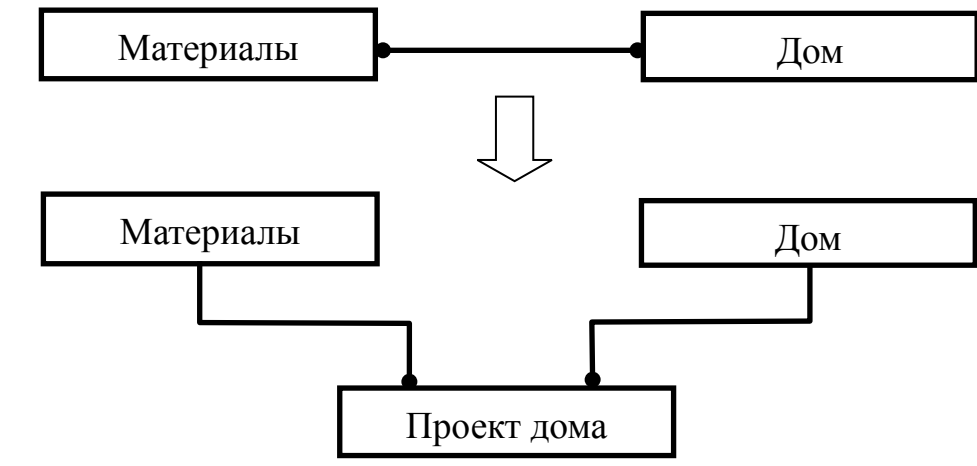
Между сущностями «Строители», а также «Каменщики», «Кровельщики», «Плотники» и «Отделочники» имеет место быть *отношение категоризации*, т.к. эти сущности относятся к одному и тому же объекту реального мира. Мощность на конце *сущности-потомка* (или *сущности-категории*) всегда является «*ноль или один*», а в рассматриваемом примере она равна «*один*».

Неспецифическое отношение имеет место между сущностью «Материалы» и сущностями «Строители», «Каменщики», «Кровельщики», «Плотники» и «Отделочники», что выявляется на основании использования ранее использованного подхода к оценке мощности отношения при следующих рассуждениях. Так, при строительстве используется много материалов, которые в свою очередь используются многими строителями, т.е. строителями различных специальностей, как то каменщики, кровельщики и т.п. Верно и обратное отношение – много строителей (строителей различных специальностей) используют при строительстве много материалов.

Выявленные неспецифические отношения необходимо разрешить, приведя эти отношения к отношению вида «сущность родитель – сущность потомок» (п.2.3.3) с мощностью «*один к одному или ко многим*».

На рис. 4.4 в качестве примера представлены диаграммы, позволяющие разрешить следующие неспецифические отношения «Материалы – Дом», «Материалы – Стены», «Строители – Фундамент», «Фундамент – Крыша». Разрешение этих неспецифических отношений осуществляется за счет использования сущностей «Проект дома» и «Дом».

Таким образом, можно предположить, что сущности «Фундамент», «Стены» и «Крыша» предположительно могут быть перенесены в категорию атрибутов сущности «Дом».



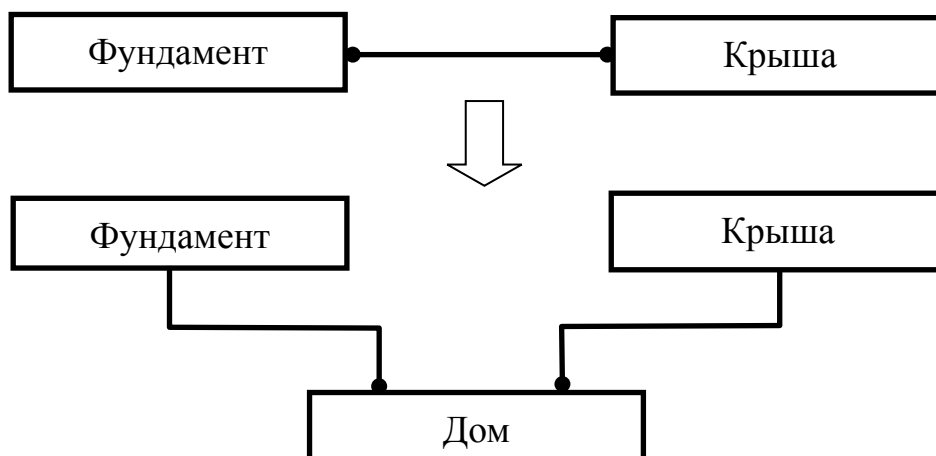


Рис. 4.4

4.4. Определение атрибутов сущностей модели

Поскольку в качестве кандидатов в сущности (пул потенциальных сущностей) для формируемой IDEF1X-модели предлагается использовать ICOM разработанной IDEF0-модели, то декомпозиция последней от уровня A-0 формально приводит к генерации кандидатов-сущностей. Эти кандидаты-сущности возникают между «входом» и «выходом» диаграммы уровня A-0 и представляют собой «входы» и «выходы» функций, которые появляются в процессе декомпозиции. Целесообразно эти промежуточные «входы» и «выходы» рассматривать, как атрибуты сущности, соответствующей «выходу» диаграммы уровня A-0. В рассматриваемом примере такими промежуточными «входами» и «выходами» являются сущности «Фундамент», «Стены» и «Крыша». Следовательно их следует перевести в категорию атрибутов сущности «Дом». Это не противоречит сделанному выше аналогичному предположению, когда рассматривались разрешения неспецифических отношений между указанными сущностями.

В соответствии с п. 2.5, где рассматривались такие понятия как атрибут, экземпляр атрибута сущности, ее ключевые и неключевые атрибуты сформируем следующую таблицу (таблица 4.3).

Таблица 4.3

№	Сущность	Атрибут	Ключ	Домен		Примечания
				Тип	Размер	
1	2	3	4	5	6	7
1	Проект_дома	№_проекта	П	Целое полож. число	До 100000	
		ФИО_архитектора	С	Текст	До 30 символов	
		Дата_создания	П	Формат	10	ДД.ММ.ГГГГ

				даты	СИМВОЛОВ	
		Стоимость_проекта	П	Целое полож. число	До 10000000	В руб.
2	Материалы	Код_материала	С	Текст+ Численное обознач.	До 30 СИМВОЛОВ	
		Название	П	Текст	До 30 СИМВОЛОВ	
		Вид_материала	П	Текст	До 30 СИМВОЛОВ	
		Характеристики	С	Текст	До 200 знаков	
3	Строитель	Табельный_номер	П	Целое полож. число	До 10000	
		ФИО	С	Текст	До 30 СИМВОЛОВ	
		Профессия	П	Текст	До 30 СИМВОЛОВ	
		Стаж_работы	П	Целое полож. число	До 70	
		Адрес	С	Текст	До 200 СИМВОЛОВ	
		Адрес_дома	С	Текст	До 200 СИМВОЛОВ	
		ФИО_хозяина	С	Текст	До 60 СИМВОЛОВ	
		Фундамент	П	Текст	До 30 СИМВОЛОВ	
		Стены	П	Текст	До 30 СИМВОЛОВ	
4	Дом	Крыша	П	Текст	До 30 СИМВОЛОВ	
		Вид	П	Текст	До 30 СИМВОЛОВ	Одноэтажный ; Двухэтажный; Трехэтажный
		Дата_сдачи	П	Формат даты	10 СИМВОЛОВ	ДД.ММ.ГГГГ

Обозначения: (П) – простой атрибут; (С) – составной/сложный атрибут; (О) – однозначный атрибут; (М) – многозначный атрибут; (В) – вычисляемый/производный атрибут.

В колонке 3 таблицы 4.3 приведены все атрибуты сущностей, приведенных в колонке 2 этой таблицы.

Рассмотрим атрибуты сущности «Проект дома». В качестве атрибутов данной сущности предлагается принять следующие

параметры/характеристики:

- №_проекта;
- ФИО_архитектора;
- Дата_создания;
- Стоимость_проекта.

Данные атрибуты в целом полностью идентифицируют рассматриваемую сущность и представляют собой список ее возможных ключей, из которого необходимо выделить первичный ключ, т.е. атрибут, *однозначно* определяющий или идентифицирующий данную сущность. Очевидно, что в качестве первичного ключа целесообразно выбрать атрибут «№_проекта», т.к. остальные перечисленные выше атрибуты при их использовании в качестве первичного ключа в общем случае будут несколько вариантов сущности «Проект дома». Оставшиеся атрибуты этой сущности представляют собой неключевые атрибуты.

Выделенный первичный ключ по своей структуре относится к категории простых первичных ключей и удобен для частого использования, т.е. отвечает следующим требованиям:

- принимает не очень большие числовые или длинные текстовые значения;
- вероятность изменения значения минимальна;
- вероятность потери уникальности минимальна.

Каждый атрибут сущности, в том числе и ее ключи, имеет область допустимых значений, которая указана в колонке 5 таблице 3.3.

В колонке 6 данной таблицы указывается обязательность атрибута для соответствующей сущности, т.е. без этого атрибута сущность не может быть корректно и однозначно идентифицирована.

В колонке 7 таблицы 4.3 приводятся комментарии к атрибутам, актуальные с точки зрения разработчика модели, а также все значения атрибутов, которые относятся к так называемому «перечисляемому типу» атрибутов. Например, это может быть атрибут, характеризующий дни передачи некоторой информации/отчета. И вот именно эти дни должны быть вписаны в колонке 7 для данного атрибута.

Все имена атрибутов всех сущностей целесообразно занести в специальную таблицу имен атрибутов (таблица 4.4).

Таблица 4.4.

Имена атрибутов сущностей

№	Атрибуты	№	Атрибуты	№	Атрибуты
1	№_проекта	8	Характеристики	15	ФИО_хозяина
2	ФИО_архитектора	9	Табельный_номер	16	Фундамент

3	Дата_создания	10	ФИО	17	Стены
4	Стоимость_проекта	11	Профессия	18	Крыша
5	Код_материала	12	Стаж_работы	19	Вид
6	Название	13	Адрес	20	Дата_сдачи
7	Вид_материала	14	Адрес_дома		

Рассмотрим миграцию ключей между сущностями модели. Как отмечалось в п. 2.5.2 миграция ключей имеет место между сущностями, связанными отношением «родитель-потомок», и сущностями, связанными отношением категоризации.

Между сущностями «Проект дома» и «Дом» имеет место отношение вида «родитель-потомок», поэтому первичный ключ «№_проекта» сущности «Проект дома» мигрирует и включается в список атрибутов сущности «Дом» в группу ее неключевых атрибутов. Это обусловлено тем, что данный атрибут не в состоянии однозначно идентифицировать сущность «Дом». Аналогичная ситуация имеет место и для первичных ключей сущностей «Материалы» - «Код_материала» и «Строители» - «Табельный_номер». Все перечисленные ключи - «№_проекта», «Код_материала», «Табельный_номер» являются внешними ключами сущности «Дом».

Сущность «Строители» связана отношением категоризации с сущностями «Каменщики», «Кровельщики», «Плотники» и «Отделочники». Поэтому первичный ключ сущности «Строители» мигрирует в перечисленные сущности и становится для них внешним первичным ключом. Неключевые атрибуты сущности «Строители» эквивалентны атрибутам этого типа сущностей «Каменщики», «Кровельщики», «Плотники» и «Отделочники».

Завершающим действием стадии определения атрибутов сущностей модели является формирование матрицы «Сущность – атрибут» (таблица 4.5).

4.5. Проверка правильности назначения атрибутов

Проверка правильности назначения атрибутов сущностей модели проводится с помощью правил приведенных в п.п. 2.5.3, 2.6.4 в два этапа – сначала проверяются ключевые атрибуты (ключи), а затем проверяются неключевые атрибуты.

Проведем проверку первичных ключевых атрибутов всех сущностей разрабатываемой модели:

- сущность «Проект дома» - первичный ключ «№_проекта»;
- сущность «Дом» - первичный ключ «Адрес_дома»;

- сущность «Материалы» - первичный ключ «Код_материала»;
- сущность «Строители» - первичный ключ «Табельный номер».

Сущность «Проект дома» - первичный ключ «№_проекта».

1. Нельзя использовать синтаксис неспецифических отношений.

Поскольку все неспецифические отношения были разрешены в процессе разработки модели, то данное требование выполнено для рассматриваемого первичного ключа.

2. Миграция ключей от родительских (или общих) сущностей к сущностям-потомкам (или сущностям-категориям) является обязательной.

Данный первичный ключ мигрирует к сущности – потомку – сущность «Дом» и включен в группу неключевых атрибутов этой сущности. Таким образом, данное требование выполнено.

3. Запрещается использовать атрибуты, которые могут принимать более одного значения для данного экземпляра сущности в одно и то же время (*правило неповторяемости*).

Значение атрибута «№_проекта» в одно и то же время не может принимать более одного значения для данного экземпляра сущности, т.к. любой проект, в том числе и дома, имеет уникальный номер. Правило выполнено.

3. Нельзя использовать атрибуты, обращающиеся в ноль (т.е. не принимающие никакого значения) для некоторого экземпляра сущности (*правило необращения в ноль*).

Атрибут «№_проекта» не обращается в ноль, т.е. для любого экземпляра сущности «Дом» данный атрибут всегда принимает фиксированное и уникальное значение. Правило выполняется.

5. Сущности с составными ключами не могут быть разбиты на несколько сущностей с более простыми ключами (*правило наименьшего ключа*).

Поскольку ключевой атрибут «№_проекта» является простым, то данное требование автоматически выполняется.

6. Необходимо объявлять об имеющихся между двумя сущностями двойных путях отношений.

Для всех сущностей модели двойные пути отсутствуют.

По аналогичной схеме выполняется проверка всех ключевых и неключевых атрибутов сущностей модели.

Результаты проверки сводятся в таблицу, фрагмент которой приведен ниже.

В рассматриваемом примере отсутствуют ситуации, соответствующие требованиям 4, 5, 6, как для ключевых, так и для неключевых атрибутов.

Примеры разрешения возможных ситуаций с указанными правилами приведены в п.п. 2.5.3, 2.6.3.

Таблица 4.5.

Сущности		Атрибуты (имена атрибутов приведены в таблице 3.4)																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Проект_дома	О К	О	О	О																
2	Материалы					О К	О	О	О												
3	Строители									О К	О	О	О	О							
4	Дом	I				I				I					О К	О	О	О	О	О	О

Обозначения: «О» = владелец, «К» = первичный ключ, «I» = наследуемый атрибут (внешний ключ).

Таблица 4.6.

№	Атрибут	Номер правила проверки корректности атрибутов					
		1	2	3	4	5	6
1	№_проекта	+	+	+	+	+	+
2	ФИО_архитектора	+	+	+	+	+	+
3	Дата_создания	+	+	+	+	+	+
4	Стоимость_проекта	+	+	+	+	+	+

Диаграмма разработанной в среде Design/IDEF 3.X модели в стандарте IDEF1X приведена на рис. 4.4.

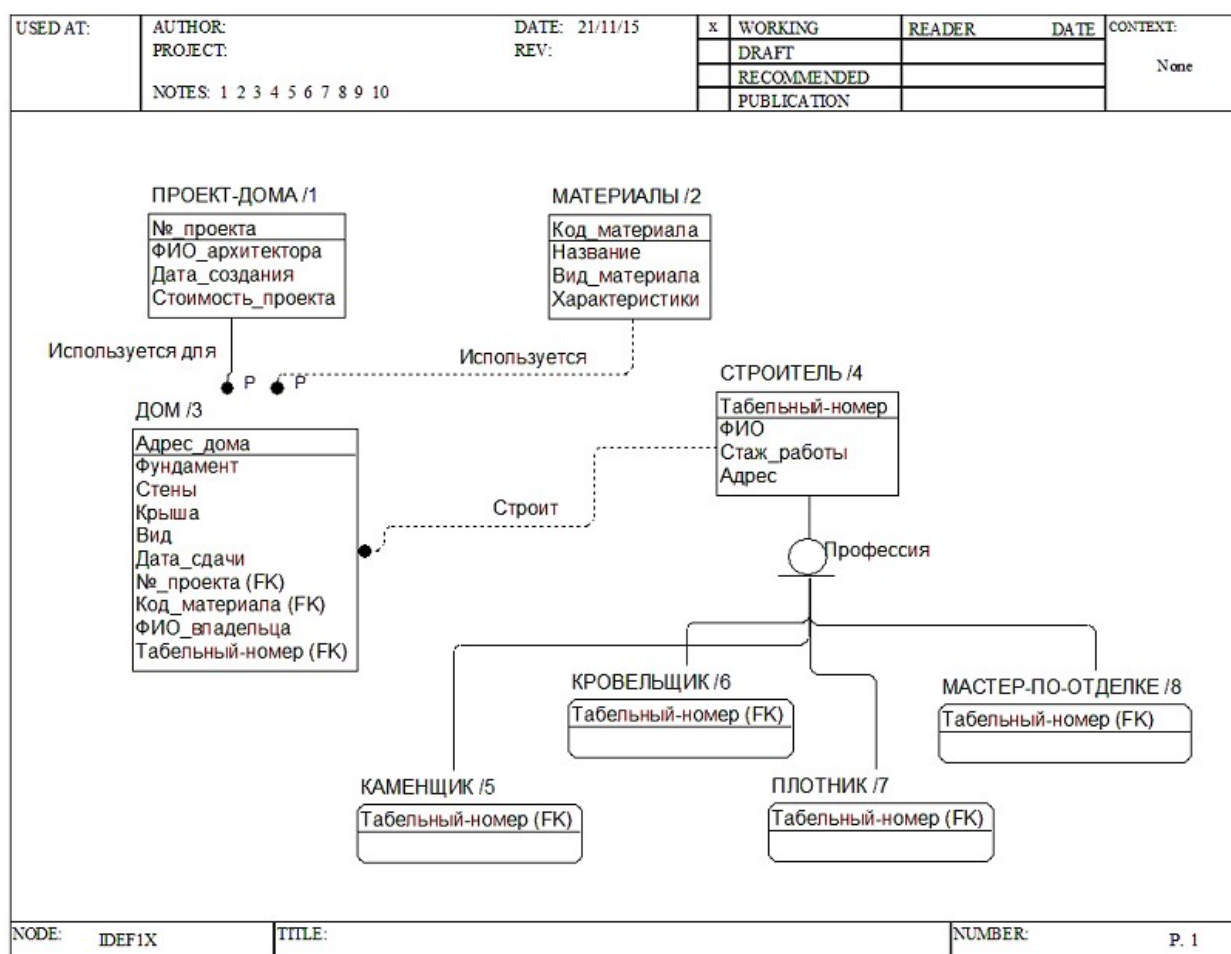


Рис. 4.4

При выполнении курсовой работы необходимо в приложении привести все рассуждения, на основании которых осуществлялось заполнение таблицы 4.6. Непосредственно в тексте пояснительной записки курсовой работы приводится только таблица 4.6.

Ниже приведен отчет по модели в стандарте IDEF1X, построенной в среде DESIGN/IDEF.

IDEF1X Report for: D:****_24C8~1\DESIGN~1\idef37\IDEF37\KURS1X.IDD

```

{ENTITY, 1, ПРОЕКТ-ДОМА, INDEPENDENT, {№_проекта},
{ФИО_архитектора,Дата_создания,Стоимость_проекта}}
{ENTITY, 2, МАТЕРИАЛЫ, INDEPENDENT, {Код_материала},
{Название,Вид_материала,Характеристики}}
{ENTITY, 3, ДОМ, INDEPENDENT, {Адрес_дома},
{Фундамент,Стены,Крыша,Вид,Дата_сдачи,№_проекта (FK),Код_материала
(FK),ФИО_владельца,Табельный-номер (FK)}}
{ENTITY, 4, СТРОИТЕЛЬ, INDEPENDENT, {Табельный-номер},
{ФИО,Стаж_работы,Адрес}}
{ENTITY, 5, КАМЕНЩИК, DEPENDENT, {Табельный-номер (FK)}, {}}
{ENTITY, 6, КРОВЕЛЬЩИК, DEPENDENT, {Табельный-номер (FK)}, {}}
{ENTITY, 7, ПЛОТНИК, DEPENDENT, {Табельный-номер (FK)}, {}}
{ENTITY, 8, МАСТЕР-ПО-ОТДЕЛКЕ, DEPENDENT, {Табельный-номер
(FK)}, {}}
{CATEGORIZATION, 1, Профессия, PARTIAL, {4}, {8,7,6,5}}
{RELATION, 1, Используется для, IDENTIFYING, 1, 3, ZOM}
{RELATION, 2, Используется, NON-IDENTIFYING, 2, 3, ZOM}
{RELATION, 3, Строит, NON-IDENTIFYING, 4, 3, ZOM}

```

5. Учебно-методическое и информационное обеспечение курсовой работы

5.1. Нормативные документы

1. Межгосударственный стандарт ГОСТ 2.105-05: Единая система конструкторской документации. Общие требования к текстовым документам. М., ИПК Изд-во стандартов, 2006.
2. Межгосударственный стандарт ГОСТ 7.1-2003: Система стандартов по информации, библиотечному и издательскому делу: Библиографическая запись. Библиографическое описание: Общие требования и правила составления. М., ИПК Изд-во стандартов, 2004.
3. Рабочая программа учебной дисциплины «Структурно-функциональное моделирование» / Сост. А.В. Кутышкин.

5.2. Основная учебная литература

1. Черемных С. В. Моделирование и анализ систем. IDEF-технологии: практикум/ С.В.Черемных, И.О. Семенов, В.С. Ручкин. – М.: Финансы и статистика, 2006.
2. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. – 2-е изд., испр. и дополн. – М.: Диалог-МИФИ, 2007.
3. Дубейковский В.И. Эффективное моделирование с AllFusion Process Modeler 4.1.4 и AllFusion PM. – М.: Диалог-МИФИ, 2007.

5.3. Дополнительная учебная литература

1. Марка Д.А., МакГоуэн К. Методология структурного анализа и проектирования: Пер.с англ. – М.: МетаТехнология, 1993.
2. Маклаков С.В. Vpwin и Erwin. CASE-средства разработки информационных систем. – М.: Диалог-МИФИ, 1999.
3. Методология функционального моделирования IDEF0. РД IDEF0 – 2000. – М.: Издательство стандартов, 2000.
4. Разработка и стандартизация программных средств и информационных технологий: Методические указания к выполнению лабораторных работ для студентов специальности 080801 «Прикладная информатика (в экономике)» очной и заочной форм обучения / И.А. Рычка. – Петропавловск-Камчатский: КамчатГТУ, 2008. – 65 с.
5. Рыбанов А.А. Инструментальные средства автоматизированного проектирования баз данных: Учебное пособие и варианты заданий к лабораторным работам по дисциплине «Базы данных». - Волгоград, ВолгГТУ, 2007. – 96 с.
6. Орел А.А., Ромакина О.М. Проектирование и моделирование бизнес-

процессов: Учебное пособие по курсу «Проектирование бизнес – процессов» для студентов механико-математического факультета. – Саратов, СГУ, 2012. – 56 с.

7. <http://dit.isuct.ru/ivt/books/CASE/case10/>

Приложения

Индивидуальные задания для выполнения курсовой работы

1. Страховая компания

Описание предметной области

Вы работаете в страховой компании. Вашей задачей является отслеживание финансовой деятельности компании.

Компания имеет различные филиалы по всей стране. Каждый филиал характеризуется названием, адресом и телефоном. Деятельность компании организована следующим образом: к Вам обращаются различные лица с целью заключения договора о страховании. В зависимости от принимаемых на страхование объектов и страхуемых рисков, договор заключается по определенному виду страхования (например, страхование автотранспорта от угона, страхование домашнего имущества, добровольное медицинское страхование). При заключении договора Вы фиксируете дату заключения, страховую сумму, вид страхования, тарифную ставку и филиал, в котором заключался договор.

2. Гостиница

Описание предметной области

Вы работаете в гостинице. Вашей задачей является отслеживание финансовой стороны работы гостиницы.

Ваша деятельность организована следующим образом: гостиница предоставляет номера клиентам на определенный срок. Каждый номер характеризуется вместимостью, комфортностью (люкс, полулюкс, обычный) и ценой. Вашими клиентами являются различные лица, о которых Вы собираете определенную информацию (фамилия, имя, отчество и некоторый комментарий). Сдача номера клиенту производится при наличии свободных мест в номерах, подходящих клиенту по указанным выше параметрам. При поселении фиксируется дата поселения. При выезде из гостиницы для каждого места запоминается дата освобождения.

3. Ломбард

Описание предметной области

Вы работаете в ломбарде. Вашей задачей является отслеживание финансовой стороны работы ломбарда.

Деятельность Вашей компании организована следующим образом: к Вам обращаются различные лица с целью получения денежных средств под залог определенных товаров. У каждого из проходящих к Вам клиентов Вы запрашиваете фамилию, имя, отчество и другие паспортные данные. После оценивания стоимости принесенного в качестве залога товара Вы определяете сумму, которую готовы выдать на руки клиенту, а также свои

комиссионные. Кроме того, определяете срок возврата денег. Если клиент согласен, то Ваши договоренности фиксируются в виде документа, деньги выдаются клиенту, а товар остается у Вас. В случае если в указанный срок не происходит возврата денег, товар переходит в Вашу собственность.

4. Реализация готовой продукции

Описание предметной области

Вы работаете в компании, занимающейся оптово-розничной продажей различных товаров. Вашей задачей является отслеживание финансовой стороны работы компании.

Деятельность Вашей компании организована следующим образом: Ваша компания торгует товарами из определенного спектра. Каждый из этих товаров характеризуется наименованием, оптовой ценой, розничной ценой и справочной информацией. В Вашу компанию обращаются покупатели. Для каждого из них Вы запоминаете в базе данных стандартные данные (наименование, адрес, телефон, контактное лицо) и составляете по каждой сделке документ, запоминая наряду с покупателем количество купленного им товара и дату покупки.

5. Ведение заказов

Описание предметной области

Вы работаете в компании, занимающейся оптовой продажей различных товаров. Вашей задачей является отслеживание финансовой стороны работы компании.

Деятельность Вашей компании организована следующим образом: Ваша компания торгует товарами из определенного спектра. Каждый из этих товаров характеризуется ценой, справочной информацией и признаком наличия или отсутствия доставки. В Вашу компанию обращаются заказчики. Для каждого из них Вы запоминаете в базе данных стандартные данные (наименование, адрес, телефон, контактное лицо) и составляете по каждой сделке документ, запоминая наряду с заказчиком количество купленного им товара и дату покупки.

6. Бюро по трудоустройству

Описание предметной области

Вы работаете в бюро по трудоустройству. Вашей задачей является отслеживание финансовой стороны работы компании.

Деятельность Вашего бюро организована следующим образом: Ваше бюро готово искать работников для различных работодателей и вакансии для ищущих работу специалистов различного профиля. При обращении к Вам клиента- работодателя, его стандартные данные (название, вид деятельности, адрес, телефон) фиксируются в базе данных. При обращении к Вам клиента-соискателя, его стандартные данные (фамилия, имя, отчество, квалификация, профессия, иные данные) также фиксируются в базе данных. По каждому

факту удовлетворения интересов обеих сторон составляется документ. В документе указываются соискатель, работодатель, должность и комиссионные (доход бюро).

7. Нотариальная контора

Описание предметной области

Вы работаете в нотариальной конторе. Вашей задачей является отслеживание финансовой стороны работы компании.

Деятельность Вашей нотариальной конторы организована следующим образом: Ваша фирма готова предоставить клиенту определенный комплекс услуг. Для наведения порядка Вы формализовали эти услуги, составив их список с описанием каждой услуги. При обращении к Вам клиента, его стандартные данные (название, вид деятельности, адрес, телефон) фиксируются в базе данных. По каждому факту оказания услуги клиенту составляется документ. В документе указываются услуга, сумма сделки, комиссионные (доход конторы), описание сделки.

8. Фирма по продаже запчастей

Описание предметной области

Вы работаете в фирме, занимающейся продажей запасных частей для автомобилей. Вашей задачей является отслеживание финансовой стороны работы компании.

Основная часть деятельности, находящейся в Вашем ведении, связана с работой с поставщиками. Фирма имеет определенный набор поставщиков, по каждому из которых известны название, адрес и телефон. У этих поставщиков Вы приобретаете детали. Каждая деталь наряду с названием характеризуется артикулом и ценой (считаем цену постоянной). Некоторые из поставщиков могут поставлять одинаковые детали (один и тот же артикул). Каждый факт покупки запчастей у поставщика фиксируется в базе данных, причем обязательными для запоминания являются дата покупки и количество приобретенных деталей.

9. Курсы по повышению квалификации

Описание предметной области

Вы работаете в учебном заведении и занимаетесь организацией курсов повышения квалификации.

В Вашем распоряжении имеются сведения о сформированных группах студентов. Группы формируются в зависимости от специальности и отделения. В каждой из них включено определенное количество студентов. Проведение занятий обеспечивает штат преподавателей. Для каждого из них у Вас в базе данных зарегистрированы стандартные анкетные данные (фамилия, имя, отчество, телефон) и стаж работы. В результате распределения нагрузки Вы получаете информацию о том, сколько часов занятий проводит каждый преподаватель с соответствующими группами.

Кроме того, хранятся также сведения о виде проводимых занятий (лекции, практика), предмете и оплате за 1 час.

10. Определение факультативов для студентов

Описание предметной области

Вы работаете в высшем учебном заведении и занимаетесь организацией факультативов.

В Вашем распоряжении имеются сведения о студентах, включающие стандартные анкетные данные (фамилия, имя, отчество, адрес, телефон). Преподаватели Вашей кафедры должны обеспечить проведение факультативных занятий по некоторым предметам. По каждому факультативу существует определенное количество часов и вид проводимых занятий (лекции, практика, лабораторные работы). В результате работы со студентами у Вас появляется информация о том, кто из них записался на какие факультативы. Существует некоторый минимальный объем факультативных предметов, которые должен прослушать каждый студент. По окончании семестра Вы заносите информацию об оценках, полученных студентами на экзаменах.

11. Распределение учебной нагрузки

Описание предметной области

Вы работаете в высшем учебном заведении и занимаетесь распределением НА-нагрузки между преподавателями кафедры.

В Вашем распоряжении имеются сведения о преподавателях кафедры, включающие наряду с анкетными данными сведения об их ученой степени, занимаемой административной должности и стаже работы. Преподаватели Вашей кафедры должны обеспечить проведение занятий по некоторым предметам. По каждому из них существует определенное количество часов. В результате распределения нагрузки у Вас должна получиться информация следующего рода: «Такой-то преподаватель проводит занятия по такому-то предмету с такой-то группой».

12. Распределение дополнительных обязанностей

Описание предметной области

Вы работаете в коммерческой компании и занимаетесь распределением дополнительных разовых работ. Вашей задачей является отслеживание хода выполнения дополнительных работ.

Компания имеет определенный штат сотрудников, каждый из которых получает определенный оклад. Время от времени, возникает потребность в выполнении некоторой дополнительной работы, не входящей в круг основных должностных обязанностей сотрудников. Для наведения порядка в этой сфере деятельности Вы проклассифицировали все виды дополнительных работ, определившись с суммой оплаты по факту их выполнения. При возникновении дополнительной работы определенного

вида Вы назначаете ответственного, фиксируя дату начала. По факту окончания Вы фиксируете дату и выплачиваете дополнительную сумму к зарплате с учетом Вашей классификации.

13. Техническое обслуживание станков

Описание предметной области

Ваше предприятие занимается ремонтом станков и другого промышленного оборудования. Вашей задачей является отслеживание финансовой стороны деятельности предприятия.

Клиентами Вашей компании являются промышленные предприятия, оснащенные различным сложным оборудованием. В случае поломок оборудования они обращаются к Вам.

Ремонтные работы в Вашей компании организованы следующим образом: все станки проклассифицированы по странам-производителям, годам выпуска и маркам. Все виды ремонта отличаются названием, продолжительностью в днях, стоимостью. Исходя из этих данных, по каждому факту ремонта Вы фиксируете вид станка и дату начала ремонта.

14. Туристическая фирма

Описание предметной области

Вы работаете в туристической компании. Ваша компания работает с клиентами, продавая им путевки. Вашей задачей является отслеживание финансовой стороны деятельности фирмы.

Работа с клиентами в Вашей компании организована следующим образом: у каждого клиента, пришедшего к Вам, собираются некоторые стандартные данные – фамилия, имя, отчество, адрес, телефон. После этого Ваши сотрудники выясняют у клиента, куда он хотел бы поехать отдыхать. При этом ему демонстрируются различные варианты, включающие страну проживания, особенности местного климата, имеющиеся отели разного класса. Наряду с этим, обсуждается возможная длительность пребывания и стоимость путевки. В случае если удалось договориться, и найти для клиента приемлемый вариант, Вы регистрируете факт продажи путевки (или путевок, если клиент покупает сразу несколько путевок), фиксируя дату отправления. Иногда Вы решаете предоставить клиенту некоторую скидку.

15. Грузовые перевозки

Описание предметной области

Вы работаете в компании, занимающейся перевозками грузов. Вашей задачей является отслеживание стоимости перевозок с учетом заработной платы водителей.

Ваша компания осуществляет перевозки по различным маршрутам. Для каждого маршрута Вы определили некоторое название, вычислили примерное расстояние и установили некоторую оплату для водителя. Информация о водителях включает фамилию, имя, отчество и стаж. Для

проведения расчетов Вы храните полную информацию о перевозках (маршрут, водитель, даты отправки и прибытия). По факту некоторых перевозок водителям выплачивается премия.

16. Учет телефонных переговоров

Описание предметной области

Вы работаете в коммерческой службе телефонной компании. Компания предоставляет абонентам телефонные линии для междугородних переговоров. Вашей задачей является отслеживание стоимости междугородних телефонных переговоров.

Абонентами компании являются юридические лица, имеющие телефонную точку, ИНН, расчетный счет в банке. Стоимость переговоров зависит от города, в который осуществляется звонок, и времени суток (день, ночь). Каждый звонок абонента автоматически фиксируется в базе данных. При этом запоминаются город, дата, длительность разговора и время суток.

17. Учет внутриофисных расходов

Описание предметной области

Вы работаете в бухгалтерии частной фирмы. Сотрудники фирмы имеют возможность осуществлять мелкие покупки для нужд фирмы, предоставляя в бухгалтерию товарный чек. Вашей задачей является отслеживание внутриофисных расходов.

Ваша фирма состоит из отделов. Каждый отдел имеет название. В каждом отделе работает определенное количество сотрудников. Сотрудники могут осуществлять покупки в соответствии с видами расходов. Каждый вид расходов имеет название, некоторое описание и предельную сумму средств, которые могут быть потрачены по данному виду расходов в месяц. При каждой покупке сотрудник оформляет документ, где указывает вид расхода, дату, сумму и отдел.

18. Библиотека

Описание предметной области

Вы являетесь руководителем библиотеки. Ваша библиотека решила зарабатывать деньги, выдавая напрокат некоторые книги, имеющиеся в небольшом количестве экземпляров. Вашей задачей является отслеживание финансовых показателей работы библиотеки.

У каждой книги, выдаваемой в прокат, есть название, автор, жанр. В зависимости от ценности книги Вы определили для каждой из них залоговую стоимость (сумма, вносимая клиентом при взятии книги напрокат) и стоимость проката (сумма, которую клиент платит при возврате книги, получая назад залог). В библиотеку обращаются читатели. Все читатели регистрируются в картотеке, которая содержит стандартные анкетные данные (фамилия, имя, отчество, адрес, телефон). Каждый читатель может обращаться в библиотеку несколько раз. Все обращения читателей

фиксируются, при этом по каждому факту выдачи книги запоминаются дата выдачи и ожидаемая дата возврата.

19. Прокат автомобилей

Описание предметной области

Вы являетесь руководителем коммерческой службы в фирме, занимающейся прокатом автомобилей. Вашей задачей является отслеживание финансовых показателей работы пункта проката.

В Ваш автопарк входит некоторое количество автомобилей различных марок, стоимостей и типов. Каждый автомобиль имеет свою стоимость проката. В пункт проката обращаются клиенты. Все клиенты проходят обязательную регистрацию, при которой о них собирается стандартная информация (фамилия, имя, отчество, адрес, телефон). Каждый клиент может обращаться в пункт проката несколько раз. Все обращения клиентов фиксируются, при этом по каждой сделке запоминаются дата выдачи и ожидаемая дата возврата.

20. Выдача банком кредитов

Описание предметной области

Вы являетесь руководителем информационно-аналитического центра коммерческого банка. Одним из существенных видов деятельности Вашего банка является выдача кредитов юридическим лицам. Вашей задачей является отслеживание динамики работы кредитного отдела.

В зависимости от условий получения кредита, процентной ставки и срока возврата все кредитные операции делятся на несколько основных видов. Каждый из этих видов имеет свое название. Кредит может получить юридическое лицо (клиент), при регистрации предоставивший следующие сведения: название, вид собственности, адрес, телефон, контактное лицо. Каждый факт выдачи кредита регистрируется банком, при этом фиксируются сумма кредита, клиент и дата выдачи.

21. Инвестирование свободных средств

Описание предметной области

Вы являетесь руководителем аналитического центра инвестиционной компании. Ваша компания занимается вложением денежных средств в ценные бумаги.

Ваши клиенты – предприятия, которые доверяют Вам управлять их свободными денежными средствами на определенный период. Вам необходимо выбрать вид ценных бумаг, которые позволят получить прибыль и Вам и Вашему клиенту. При работе с клиентом для Вас весьма существенной является информация о предприятии – название, вид собственности, адрес и телефон.

22. Занятость актеров театра

Описание предметной области

Вы являетесь коммерческим директором театра, и в Ваши обязанности входит вся организационно-финансовая работа, связанная с привлечением актеров и заключением контрактов.

Вы поставили дело следующим образом: каждый год театр осуществляет постановку различных спектаклей. Каждый спектакль имеет определенный бюджет. Для участия в конкретных постановках в определенных ролях Вы привлекаете актеров. С каждым из актеров Вы заключаете персональный контракт на определенную сумму. Каждый из актеров имеет некоторый стаж работы, некоторые из них удостоены различных наград и званий.

23. Платная поликлиника

Описание предметной области

Вы являетесь руководителем службы планирования платной поликлиники. Вашей задачей является отслеживание финансовых показателей работы поликлиники.

В поликлинике работают врачи различных специальностей, имеющие разную квалификацию. Каждый день в поликлинику обращаются больные. Все больные проходят обязательную регистрацию, при которой в базу данных заносятся стандартные анкетные данные (фамилия, имя, отчество, год рождения). Каждый больной может обращаться в поликлинику несколько раз, нуждаясь в различной медицинской помощи. Все обращения больных фиксируются, при этом устанавливается диагноз, определяется стоимость лечения, запоминается дата обращения.

24. Анализ динамики показателей финансовой отчетности различных предприятий

Описание предметной области

Вы являетесь руководителем информационно-аналитического центра крупного холдинга. Вашей задачей является отслеживание динамики показателей для предприятий Вашего холдинга.

В структуру холдинга входят несколько предприятий. Каждое предприятие имеет стандартные характеристики (название, реквизиты, телефон, контактное лицо). Работа предприятия может быть оценена следующим образом: в начале каждого отчетного периода на основе финансовой отчетности вычисляется по неким формулам определенный набор показателей. Принять, что важность показателей характеризуется некоторыми числовыми константами. Значение каждого показателя измеряется в некоторой системе единиц.

25. Учет телекомпанией стоимости прошедшей в эфире рекламы

Описание предметной области

Вы являетесь руководителем коммерческой службы телевизионной компании. Вашей задачей является отслеживание расчетов, связанных с

прохождением рекламы в телеэфире.

Работа построена следующим образом: заказчики просят поместить свою рекламу в определенной передаче в определенный день. Каждый рекламный ролик имеет определенную продолжительность. Для каждой организации-заказчика известны банковские реквизиты, телефон и контактное лицо для проведения переговоров. Передачи имеют определенный рейтинг. Стоимость минуты рекламы в каждой конкретной передаче известна (определяется коммерческой службой, исходя из рейтинга передачи и прочих соображений).

26. Интернет-магазин

Описание предметной области

Вы являетесь сотрудником коммерческого отдела компании, продающей различные товары через Интернет. Вашей задачей является отслеживание финансовой составляющей работы компании.

Работа Вашей компании организована следующим образом: на Интернет-сайте компании представлены (выставлены на продажу) некоторые товары. Каждый из них имеет некоторое название, цену и единицу измерения (штуки, килограммы, литры). Для проведения исследований и оптимизации работы магазина Вы пытаетесь собирать данные с Ваших клиентов. При этом для Вас определяющее значение имеют стандартные анкетные данные, а также телефон и адрес электронной почты для связи. В случае приобретения товаров на сумму свыше 5000 руб. клиент переходит в категорию «постоянных клиентов» и получает скидку на каждую покупку в размере 2%. По каждому факту продажи Вы автоматически фиксируете клиента, товары, количество, дату продажи, дату доставки.

27. Ювелирная мастерская

Описание предметной области

Вы работаете в ювелирной мастерской.

Ваша мастерская осуществляет изготовление ювелирных изделий для частных лиц на заказ. Вы работаете с определенными материалами (платина, золото, серебро, различные драгоценные камни и т.д.). При обращении к Вам потенциального клиента Вы определяетесь с тем, какое именно изделие ему необходимо. Все изготавливаемые Вами изделия принадлежат к некоторому типу (серьги, кольца, броши, браслеты), бывают выполнены из определенного материала, имеют некоторый вес и цену (включающую стоимость материалов и работы).

28. Парикмахерская

Описание предметной области

Вы работаете в парикмахерской.

Ваша парикмахерская стрижет клиентов в соответствии с их пожеланиями и некоторым каталогом различных видов стрижки. Так, для каждой стрижки

определены название, принадлежность полу (мужская, женская), стоимость работы. Для наведения порядка Вы, по мере возможности, составляете базу данных клиентов, запоминая их анкетные данные (фамилия, имя, отчество). Начиная с 5-ой стрижки, клиент переходит в категорию постоянных и получает скидку в 3% при каждой последующей стрижке. После того, как закончена очередная работа, в кассе фиксируются стрижка, клиент и дата производства работ.

29. Химчистка

Описание предметной области

Вы работаете в химчистке.

Ваша химчистка осуществляет прием у населения вещей для выведения пятен. Для наведения порядка Вы, по мере возможности, составляете базу данных клиентов, запоминая их анкетные данные (фамилия, имя, отчество). Начиная с 3-го обращения, клиент переходит в категорию постоянных клиентов и получает скидку в 3% при чистке каждой последующей вещи. Все оказываемые Вами услуги подразделяются на виды, имеющие название, тип и стоимость, зависящую от сложности работ. Работа с клиентом первоначально состоит в определении объема работ, вида услуги и, соответственно, ее стоимости. Если клиент согласен, он оставляет вещь (при этом фиксируется услуга, клиент и дата приема) и забирает ее после обработки (при этом фиксируется дата возврата).

30. Сдача в аренду торговых площадей

Описание предметной области

Вы работаете в крупном торговом центре, сдающим в аренду коммерсантам свои торговые площади.

Вашей задачей является наведение порядка в финансовой стороне работы торгового центра.

Работа Вашего торгового центра построена следующим образом: в результате планирования Вы определили некоторое количество торговых точек в пределах Вашего здания, которые могут сдаваться в аренду. Для каждой из торговых точек важными данными являются этаж, площадь, наличие кондиционера и стоимость аренды в день. Со всех потенциальных клиентов Вы собираете стандартные данные (название, адрес, телефон, реквизиты, контактное лицо). При появлении потенциального клиента Вы показываете ему имеющиеся свободные площади. При достижении соглашения Вы оформляете договор, фиксируя в базе данных торговую точку, клиента, период (срок) аренды.

Синтаксис и семантика моделей IDEF0 □

1. Синтаксис графического языка IDEF0

Синтаксис языка – это набор структурных компонентов языка, их характеристики и правила, определяющие связи между его компонентами.

Компоненты синтаксиса IDEF0 — блоки, стрелки, диаграммы и правила. Блоки представляют функции, определяемые как деятельность, процесс, операция, действие или преобразование (см. ниже).

Стрелки представляют данные или материальные 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.

Блок. Блок описывает функцию. Типичный блок показан на рис. П.2.1.

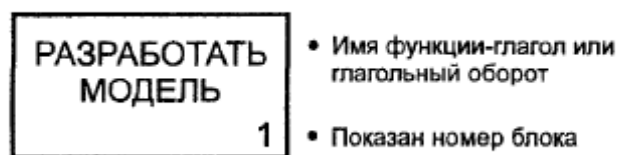


Рис. П.2.1

Внутри каждого блока помещаются его *имя* и *номер*.

Имя должно быть активным глаголом или глагольным оборотом, описывающим функцию.

Номер блока размещается в правом нижнем углу. Номера блоков используются для их идентификации на диаграмме и в соответствующем тексте.

Стрелка. Стрелка формируется из одного или нескольких отрезков прямых и наконечника на одном конце.

Сегменты стрелок могут быть прямыми или ломаными; в последнем случае горизонтальные и вертикальные отрезки стрелки сопрягаются дугами, имеющими угол 90°.

Стрелки не представляют поток или последовательность событий, как в традиционных блок-схемах потоков или процессов (потоковых диаграммах). Они лишь показывают, какие данные или материальные объекты должны поступить на вход функции для того, чтобы эта функция могла выполняться.

2. Синтаксические правила

Блоки

Для блоков установлены следующие синтаксические правила:

- размеры блоков должны быть достаточными для того, чтобы включить имя и номер блока.
- блоки должны быть прямоугольными, с прямыми углами;
- блоки должны быть нарисованы сплошными линиями.

Стрелки

Для стрелок установлены следующие синтаксические правила:

- ломаные стрелки изменяют направление только под углом 90°;
- стрелки должны быть нарисованы сплошными линиями. Можно использовать линии различной толщины;
- стрелки могут состоять только из вертикальных или горизонтальных отрезков; отрезки, направленные по диагонали, не допускаются;
- концы стрелок должны касаться внешней границы функционального блока, но не должны пересекать ее;
- стрелки должны присоединяться к блоку на его сторонах. Присоединение в углах не допускается.

3. Границы и связи

Описание любого блока должно, как минимум, включать в себя описание объектов, которые блок создает в результате своей работы («выхода»), и объектов, которые блок потребляет или преобразует («вход»).

В IDEF0 также моделируются *управление* и *механизмы исполнения*.

Под *управлением* понимаются объекты, воздействующие на способ, которым блок преобразует вход в выход.

Механизм исполнения — объекты, которые непосредственно выполняют преобразование входа в выход, но не потребляются при этом сами по себе.

Для отображения категорий информации, присутствующих на диаграммах IDEF0, существует *аббревиатура ICOM*, отображающая четыре возможных типа стрелок:

I (Input) — вход — нечто, что потребляется в ходе выполнения процесса;

C (Control) — управление — ограничения и инструкции, влияющие на ход выполнения процесса;

O (Output) — выход — нечто, являющееся результатом выполнения процесса;

M (Mechanism) — исполняющий механизм — нечто, что используется для выполнения процесса, но не потребляется само по себе.

Рис. П.2.2 показывает возможные типы стрелок в IDEF0, каждый из типов соединяется со своей стороной функционального блока.



Рис. П.2.2

Для названия стрелок, как правило, употребляются имена существительные.

Стрелки могут представлять собой людей, места, вещи, идеи или события.

Стрелки входа. Вход представляет собой сырье, или информацию, потребляемую или преобразуемую функциональным блоком для производства выхода. Стрелки входа всегда направлены в левую сторону прямоугольника, обозначающего в IDEF0 функциональный блок. Наличие входных стрелок на диаграмме не является обязательным, так как возможно, что некоторые блоки ничего не преобразуют и не изменяют.

Примером блока, не имеющего входа, может служить «принятие решения руководством», где для принятия решения анализируется несколько факторов, но ни один из них непосредственно не преобразуется и не потребляется в результате принятия какого-либо решения.

Стрелки управления. Стрелки управления отвечают за регулирование того, как и когда выполняется функциональный блок, и, если он выполняется, какой выход получается в результате его выполнения.

Так как управление контролирует поведение функционального блока для обеспечения создания желаемого выхода, каждый функциональный блок

должен иметь, как минимум, одну стрелку управления.

Стрелки управления всегда входят в функциональный блок сверху.

Управление часто существует в виде правил, инструкций, законов, политики, набора необходимых процедур или стандартов. Влияя на работу блока, оно непосредственно не потребляется и не трансформируется в результате. Может оказаться, что целью функционального блока является как раз изменение того или иного правила, инструкции, стандарта и т.п. В этом случае стрелка, содержащая соответствующую информацию, должна рассматриваться не как управление, а как вход функционального блока.

Управление можно рассматривать как специфический вид входа.

В случаях, когда неясно, относить ли стрелку к входу или к управлению, предпочтительно относить ее к управлению до момента, пока неясность не будет разрешена.

Стрелки выхода. Выход — это продукция или информация, получаемая в результате работы функционального блока.

Каждый блок должен иметь, как минимум, один выход. Действие, которое не производит никакого четко определяемого выхода, не должно моделироваться вообще (по меньшей мере, должно рассматриваться в качестве одного из первых кандидатов на исключение из модели).

При моделировании непроизводственных предметных областей выходами, как правило, являются данные, в каком-либо виде обрабатываемые функциональным блоком. В этом случае важно, чтобы названия стрелок входа и выхода были достаточно различимы по своему смыслу.

Например, блок «Прием пациентов» может иметь стрелку «Данные о пациенте» как на входе, так и на выходе. В такой ситуации входящую стрелку можно назвать «Предварительные данные о пациенте», а исходящую — «Подтвержденные данные о пациенте».

Стрелки механизма исполнения. Механизмы являются ресурсом, который непосредственно исполняет моделируемое действие.

С помощью механизмов исполнения могут моделироваться: ключевой персонал, техника и (или) оборудование.

Стрелки механизма исполнения могут отсутствовать в случае, если оказывается, что они не являются необходимыми для достижения поставленной цели моделирования.

4. Структура IDEF0-модели

IDEF0-модели состоят из документов трех типов:

- графических диаграмм;
- текста;
- глоссария.

Эти документы имеют перекрестные ссылки друг на друга.

4.1. Графические диаграммы IDEF0-модели

Графическая диаграмма — главный компонент IDEF0-модели, содержащий блоки, стрелки, соединения блоков и стрелок и ассоциированные с ними отношения.

Блоки представляют основные функции моделируемого объекта. Эти функции могут быть разбиты (декомпозированы) на составные части и представлены в виде более подробных диаграмм.

Процесс декомпозиции продолжается до тех пор, пока объект не будет описан на уровне детализации, необходимом для достижения целей конкретного проекта.

Диаграмма верхнего уровня обеспечивает наиболее общее описание объекта моделирования.

За этой диаграммой следует серия дочерних диаграмм, дающих более детальное представление об объекте.

Контекстная диаграмма верхнего уровня. Каждая модель должна иметь контекстную диаграмму верхнего уровня, на которой объект моделирования представлен единственным блоком с граничными стрелками.

Эта диаграмма называется А—0 (А минус ноль). Стрелки на этой диаграмме отображают связи объекта моделирования с окружающей средой.

Поскольку единственный блок представляет весь объект, его имя — общее для всего проекта.

Это же справедливо и для всех стрелок диаграммы, поскольку они представляют полный комплект внешних интерфейсов объекта.

Диаграмма А—0 устанавливает область моделирования и ее границу.

Пример диаграммы А—0 показан на рис. П.2.3.

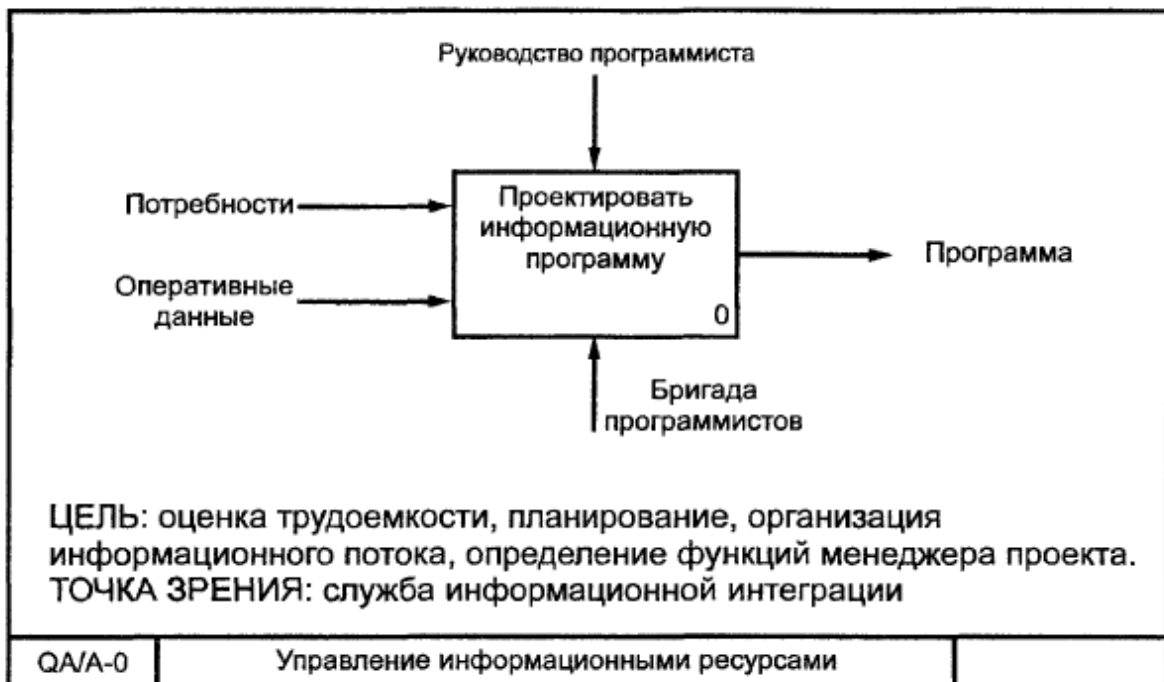


Рис. П.2.3.

Контекстная диаграмма $A—0$ также должна содержать краткие утверждения, определяющие точку зрения должностного лица или подразделения, с позиций которого создается модель, и цель, для достижения которой ее разрабатывают. Эти утверждения помогают руководить разработкой модели и ввести этот процесс в определенные рамки.

Точка зрения определяет, что и в каком разрезе можно увидеть в пределах контекста модели. 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.

Формулировка цели выражает причину создания модели, то есть содержит перечень вопросов, на которые должна отвечать модель, что в значительной мере определяет ее структуру.

Наиболее важные свойства объекта обычно выявляются на верхних уровнях иерархии; по мере декомпозиции функции верхнего уровня и разбиения ее на подфункции, эти свойства уточняются.

Каждая подфункция, в свою очередь, декомпозируется на элементы следующего уровня, и так происходит до тех пор, пока не будет получена релевантная структура, позволяющая ответить на вопросы, сформулированные в цели моделирования.

Каждая подфункция моделируется отдельным блоком.

Каждый родительский блок подробно описывается дочерней диаграммой на более низком уровне.

Все дочерние диаграммы должны быть в пределах области контекстной диаграммы верхнего уровня

Дочерняя диаграмма. Единственная функция, представленная на контекстной диаграмме верхнего уровня, может быть разложена на основные подфункции посредством создания *дочерней диаграммы*.

В свою очередь, каждая из этих подфункций может быть разложена на составные части посредством создания *дочерней диаграммы* следующего, более низкого уровня, на которой некоторые или все функции также могут быть разложены на составные части.

Каждая *дочерняя диаграмма* содержит дочерние блоки и стрелки, обеспечивающие дополнительную детализацию родительского блока.

Дочерняя диаграмма, создаваемая при декомпозиции, охватывает ту же область, что и родительский блок, но описывает ее более подробно.

Таким образом, *дочерняя диаграмма* как бы вложена в свой родительский блок. Эта структура иллюстрируется рис. П.2.4.

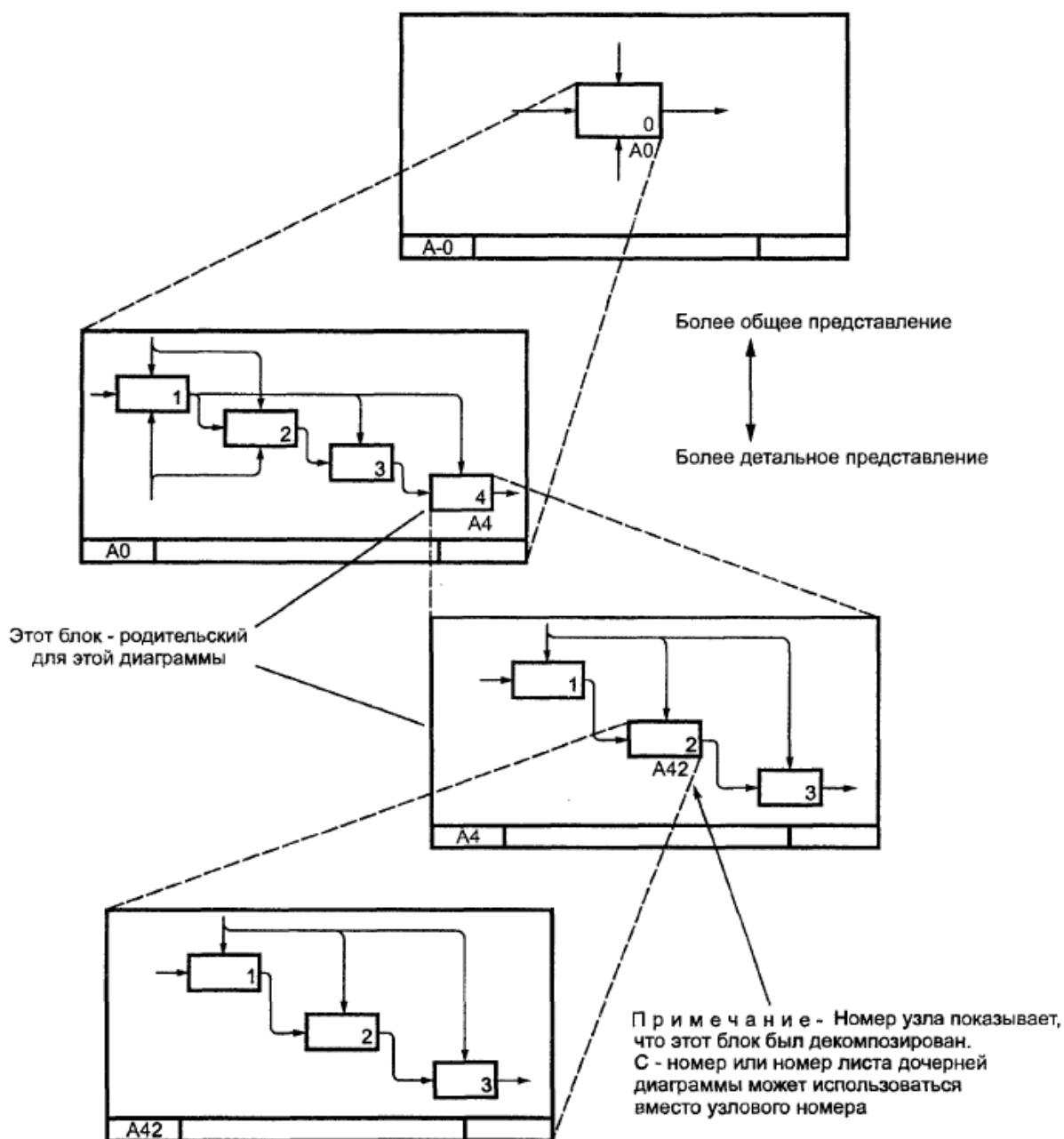


Рис. П.2.4.

Родительская диаграмма — та, которая содержит один или более родительских блоков.

Иными словами, это диаграмма верхнего уровня, на которой располагается декомпозируемая функция со всеми ее связями.

Каждая обычная (неконтекстная) диаграмма является также дочерней диаграммой, поскольку, по определению, она подробно описывает некоторый родительский блок.

Таким образом, *любая диаграмма* может быть как *родительской диаграммой* (содержать родительские блоки), так и *дочерней диаграммой* (подробно описывать собственный родительский блок).

Аналогично, блок может быть как *родительским* (подробно описываться дочерней диаграммой), так и *дочерним* (появляющимся на дочерней диаграмме).

Основное иерархическое отношение существует между родительским блоком и дочерней диаграммой, которая его подробно описывает (рис. П.2.4).

То, что блок является дочерним и раскрывает содержание родительского блока на диаграмме предшествующего уровня, указывается специальным ссылочным кодом, написанным ниже правого нижнего угла блока.

Этот ссылочный код может формироваться несколькими способами, из которых самый простой заключается в том, что код, начинающийся с буквы А (по имени диаграммы А—0), содержит цифры, определяемые номерами родительских блоков.

Например показанные на рис. П.2.5 коды означают, что диаграмма является декомпозицией 1-го блока диаграммы, которая, в свою очередь, является декомпозицией 6-го блока диаграммы А0, а сами коды образуются присоединением номера блока.

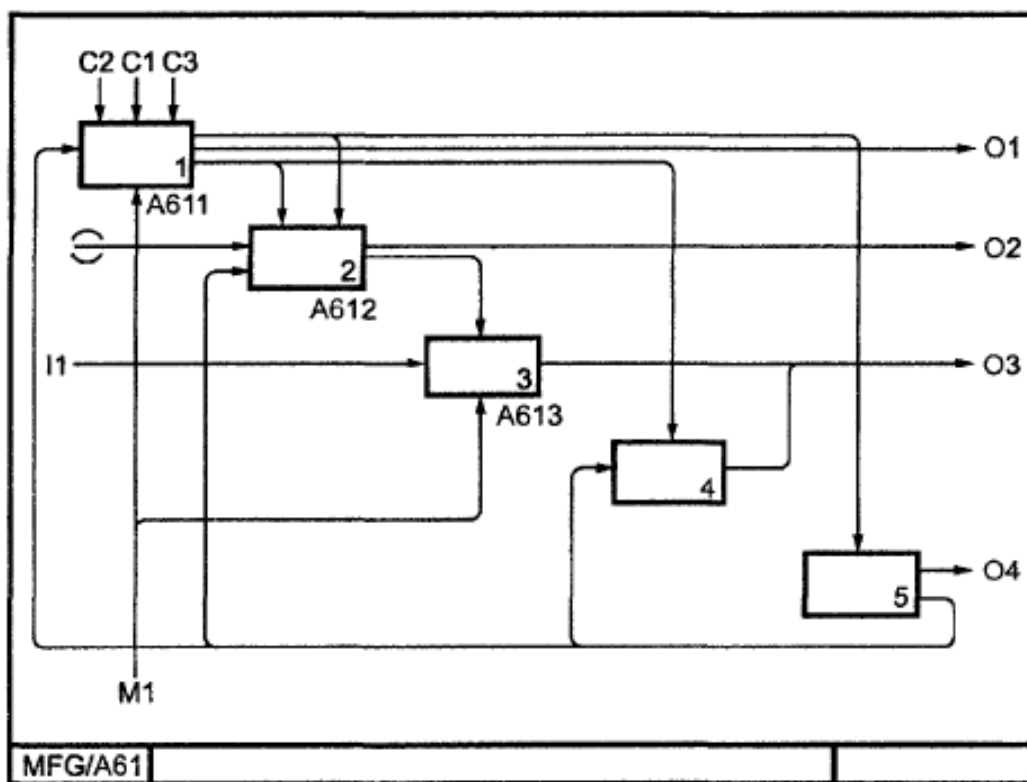
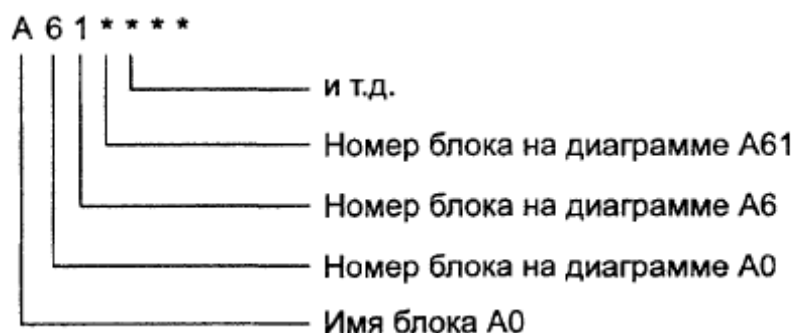


Рис. П.2.5

Следовательно, код формируется так:



Диаграммы-иллюстрации (FEO diagrams - For Exposition Only diagrams).

Эти диаграммы используются в качестве дополнений, поясняющих специфику содержания основных диаграмм в случаях, когда это необходимо.

Диаграмма FEO не должна подчиняться синтаксическим правилам IDEF0.

Презентационные диаграммы (FEO diagrams) часто включают в модели, чтобы проиллюстрировать другие точки зрения или детали, выходящие за рамки традиционного синтаксиса IDEF0.

Диаграммы FEO допускают нарушение любых правил построения диаграмм IDEF0 в целях выделения важных с точки зрения аналитика частей модели. Естественно, если диаграмма FEO включена в модель исключительно для отображения другой точки зрения на систему, она, скорее всего, будет выглядеть как обыкновенная диаграмма IDEF0, удовлетворяя всем ограничениям IDEF0.

Один из способов использования FEO-диаграмм состоит в отделении функционального блока от его окружения посредством создания диаграммы с единственным блоком и всеми относящимися к нему стрелками наподобие контекстной диаграммы (рис. П.2.6).

Это может оказаться полезным в ситуациях, когда необходимо быстро получить информацию об интерфейсе (стрелках) функционального блока, а соответствующая диаграмма декомпозиции содержит слишком много объектов.

Кроме того, встречаются следующие виды презентационных диаграмм:

- копия диаграммы IDEF0, которая содержит все функциональные блоки, и стрелки, относящиеся только к одному из функциональных блоков, — это позволяет отразить взаимодействие между этим блоком и другими объектами диаграммы;
- копия диаграммы IDEF0, которая содержит все функциональные блоки, и стрелки, непосредственно относящиеся только к входу и (или) к выходу родительского блока;
- различные точки зрения, как правило, на глубину одного уровня декомпозиции.

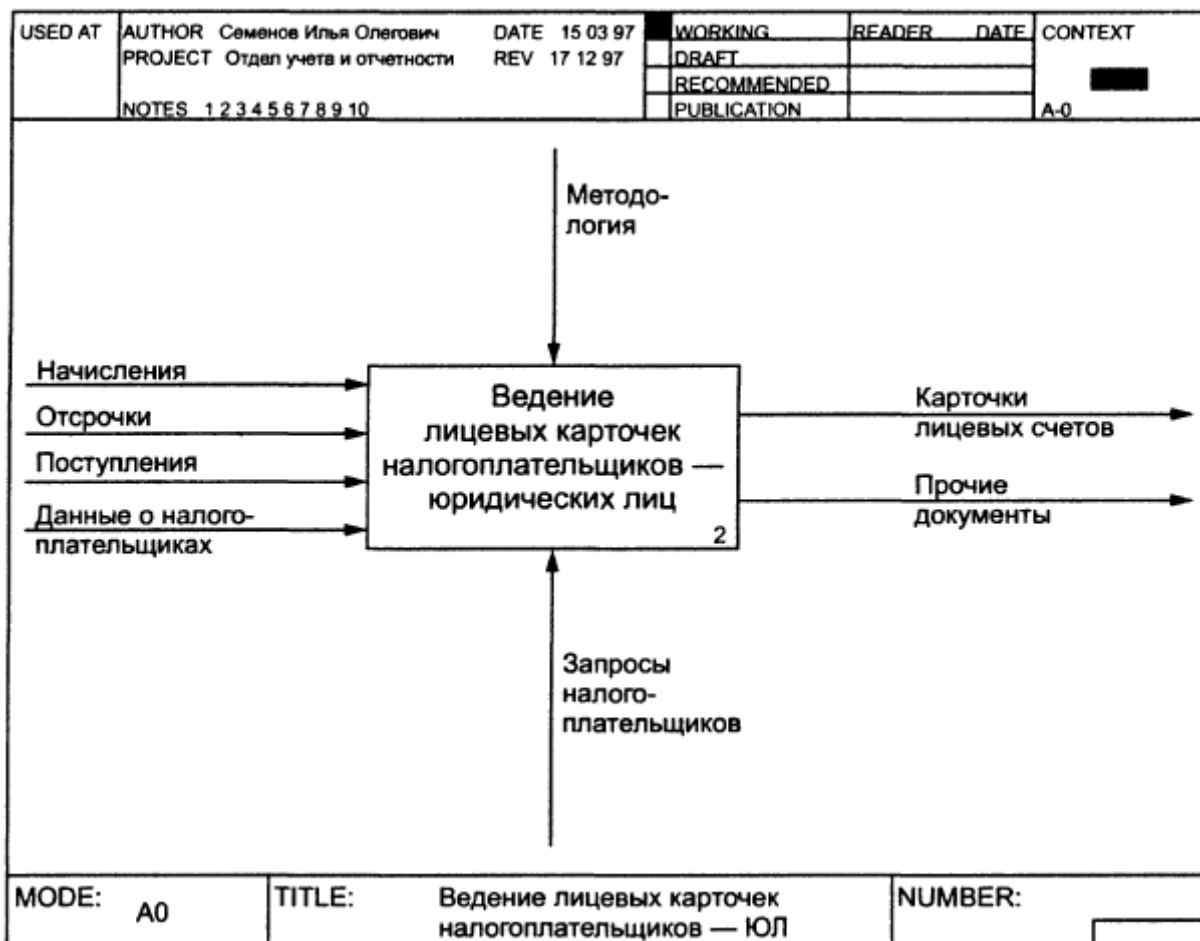


Рис. П.2.6. Диаграмма FEO для выделения функционального блока и его стрелок

4.2. Текст и глоссарий IDEF0-модели

Диаграмме может быть поставлен в соответствие структурированный 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 стрелок на диаграммах.

При большом объеме текст располагается на отдельном листе модели (текстовой странице).

Глоссарий 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 понимаемы всеми участниками разработки и пользователями модели, чтобы правильно интерпретировать ее содержание.

Глоссарий составляется с любой необходимой степенью подробности.

5. Свойства диаграмм IDEF0-модели

Стрелки как ограничения. Стрелки на диаграмме IDEF0, представляя данные или материальные объекты, одновременно задают своего рода ограничения (условия).

Входные и управляющие стрелки блока, соединяющие его с другими блоками или внешней средой, описывают условия, которые должны быть выполнены для того, чтобы реализовалась функция, записанная в качестве имени блока.

Рис. П. 2.7 иллюстрирует случай, при котором «функция 3» может быть выполнена только после получения данных, выработанных «функцией 1» и «функцией 2».



Рис. П. 2.7.

Параллельное функционирование. Различные функции в модели могут быть выполнены параллельно, если удовлетворяются необходимые ограничения (условия).

Как показано на рис. П. 2.8, один блок может создать данные (а) или материальные объекты (б), необходимые для параллельной работы нескольких блоков.

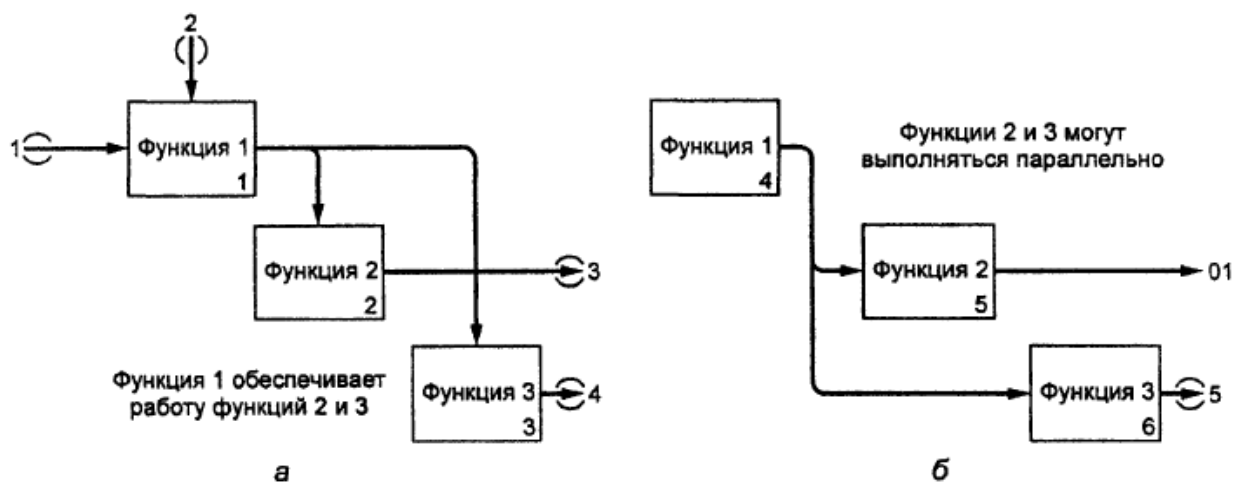


Рис. П. 2.8.

Ветвление и слияние сегментов стрелок. Ветвление и слияние стрелок призвано уменьшить загроуженность диаграмм графическими элементами (линиями).

Чтобы стрелки и их сегменты правильно описывали связи между блоками-источниками и блоками-потребителями, используется аппарат меток. Метки связываются с сегментами посредством тильд.

При этом между сегментами возникают определенные отношения, описанные ниже:

- непомеченные сегменты (рис. П. 2.9) содержат все объекты, указанные в метке стрелки перед ветвлением (то есть все объекты принадлежат каждому из сегментов);

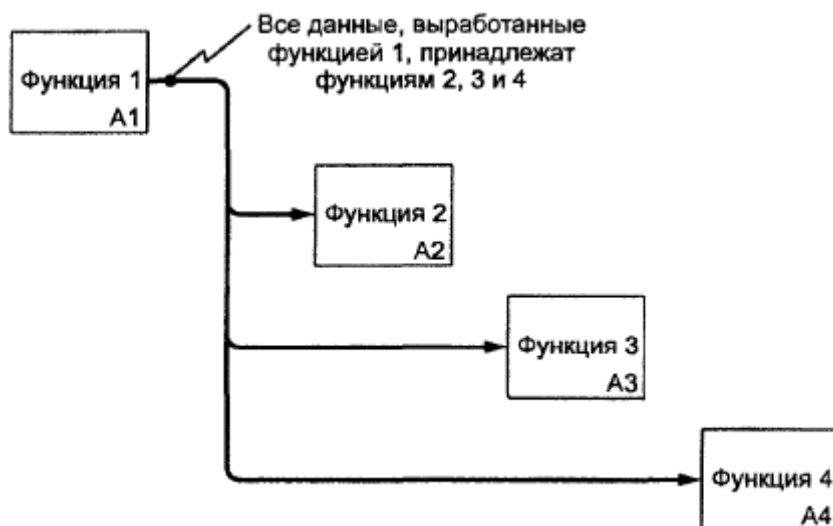


Рис. П. 2.9.

- сегменты, помеченные после точки ветвления (рис. П. 2.10), содержат все объекты, указанные в метке стрелки перед ветвлением, или их часть, описываемую меткой каждого конкретного сегмента;

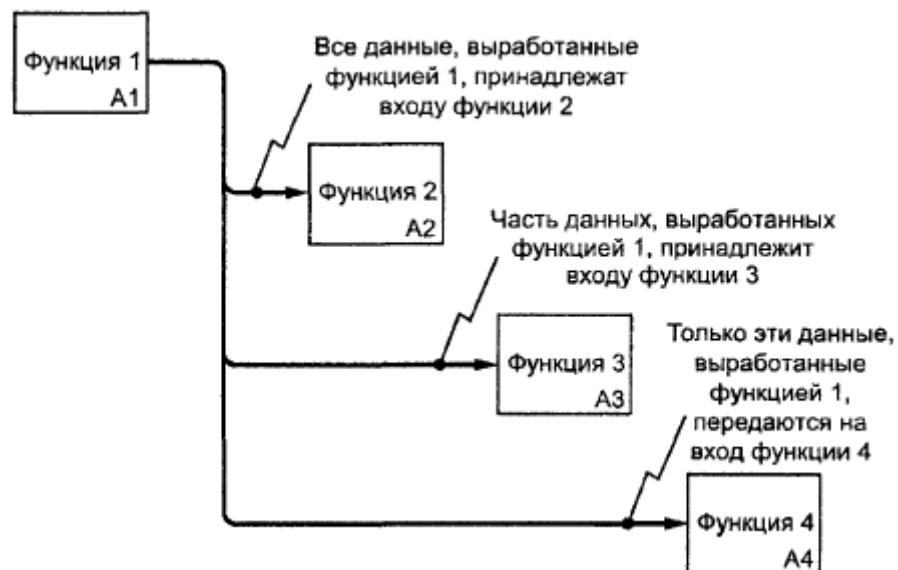


Рис. П. 2.10.

- при слиянии непомеченных сегментов объединенный сегмент стрелки содержит все объекты, принадлежащие сливаемым сегментам и указанные в общей метке стрелки после слияния (рис. П. 2.11);



Рис. П. 2.11.

- при слиянии помеченных сегментов (рис. П. 2.12) объединенный сегмент содержит все или некоторые объекты, принадлежащие сливаемым сегментам и перечисленные в общей метке после слияния; если общая метка после слияния отсутствует, это означает, что общий сегмент передает все объекты, принадлежащие сливаемым сегментам;



Рис. П. 2.12.

Отношения блоков на диаграммах. В методологии IDEF0 существует шесть типов отношений между блоками в пределах одной диаграммы:

- доминирование;
- управление;
- выход — вход;
- обратная связь по управлению;
- обратная связь по входу;
- выход — механизм.

Первое из перечисленных отношений определяется взаимным расположением блоков на диаграмме. Предполагается, что блоки, расположенные на диаграмме выше и левее, «доминируют» над блоками, расположенными ниже и правее.

«Доминирование» понимается как влияние, которое один блок оказывает на другие блоки диаграммы.

Остальные пять отношений описывают связи между блоками и изображаются соответствующими стрелками.

Отношения управления и выход — вход являются простейшими, поскольку отражают прямые взаимодействия, которые понятны и очевидны.

Отношение управления (рис. П. 2.13) возникает, когда выход одного блока служит управляющим воздействием на блок с меньшим доминированием.



Рис. П. 2.13.

Отношение выход — вход (рис. П. 2.14) возникает при соединении выхода одного блока с входом другого блока с меньшим доминированием.



Рис. П. 2.14.

Обратная связь по управлению и **обратная связь по входу** являются более сложными типами отношений, поскольку они представляют итерацию (выход функции влияет на будущее выполнение других функций с большим доминированием, что впоследствии влияет на исходную функцию).

Обратная связь по управлению (рис. П. 2.15) возникает, когда выход некоторого блока создает управляющее воздействие на блок с большим доминированием



Рис. П. 2.15.

Отношение обратной связи по входу (рис. П. 2.16) имеет место, когда выход блока становится входом другого блока с большим доминированием.



Рис. П. 2.16.

Связи «выход — механизм» (рис. П. 2.17) 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.).



Рис. П. 2.17.

6. Отношения между блоками диаграммы и другими диаграммами (окружающей средой)

Все описанные выше отношения отображаются внутренними стрелками, то есть такими, у которых оба конца связаны с блоками диаграммы. Отношения между блоками диаграммы и другими диаграммами, являющимися по отношению к рассматриваемой диаграмме окружающей средой (окружением), описываются *границными стрелками*. Обе ситуации отражены на рис. П. 2.18.

Граничные стрелки. На обычной (неконтекстной) диаграмме граничные стрелки представляют входы, управления, выходы или механизмы родительского блока диаграммы. Источник или потребитель граничных стрелок можно обнаружить, только изучая родительскую диаграмму. Все граничные стрелки на дочерней диаграмме (за исключением стрелок, помещенных в туннель) должны соответствовать стрелкам родительского блока, как показано на рис. П. 2.19.

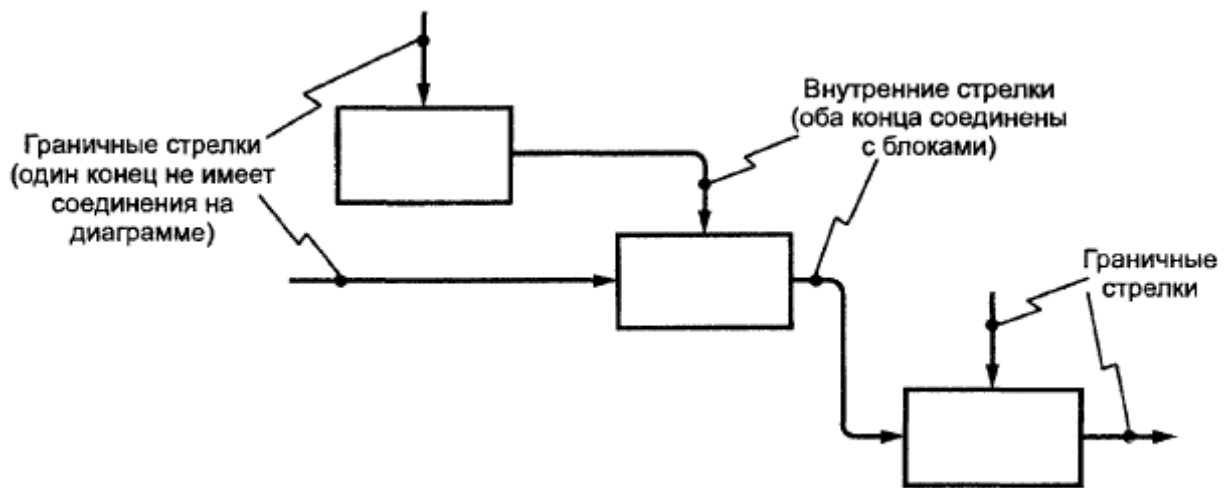


Рис. П. 2.18.

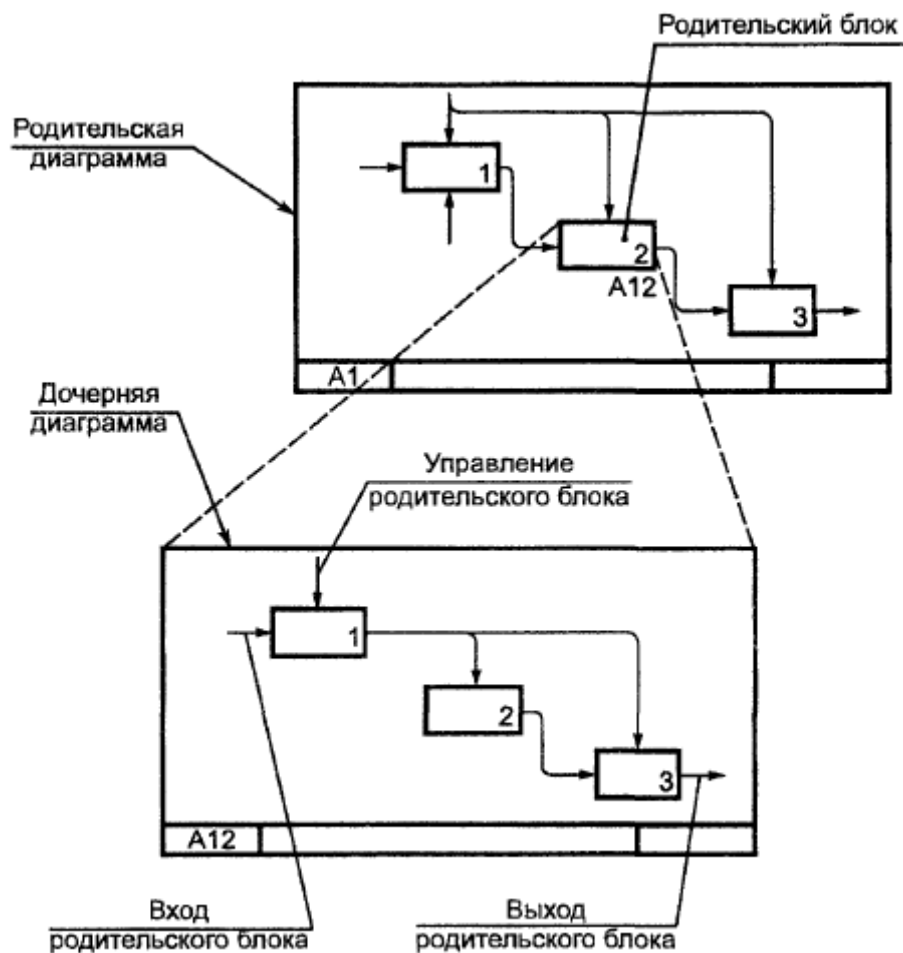


Рис. П. 2.19.

ICOM - кодирование граничных стрелок. ICOM-коды связывают граничные стрелки на дочерней диаграмме со стрелками родительского блока.

Нотация, названная ICOM - кодом, определяет значения соединений.

Буквы I, C, O или M, приведенные около несвязанного конца граничной стрелки на дочерней диаграмме, идентифицируют стрелку как Вход (Input.), Управление (Control), Выход (Output) или Механизм (Mechanism) в родительском блоке.

Буква следует за числом, определяющим относительное положение точки подключения стрелки к родительскому блоку; это положение определяется слева направо или сверху вниз. Например, код «3С» возле граничной стрелки на дочерней диаграмме указывает, что эта стрелка соответствует третьей (считая слева) управляющей стрелке родительского блока.

Это кодирование связывает каждую 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 стрелки этих диаграмм со стрелками их родительских блоков.

Иногда буквенные ICOM-коды, определяющие роли граничных стрелок (вход, управление, механизм), могут меняться при переходе от родительского блока к дочерней диаграмме.

Например, управляющая стрелка в родительском блоке может быть входом на дочерней диаграмме. Аналогично, вход родительского блока может быть управлением для одного или нескольких дочерних блоков.

Примеры изменения ролей стрелок можно видеть на рис. П. 2.20.

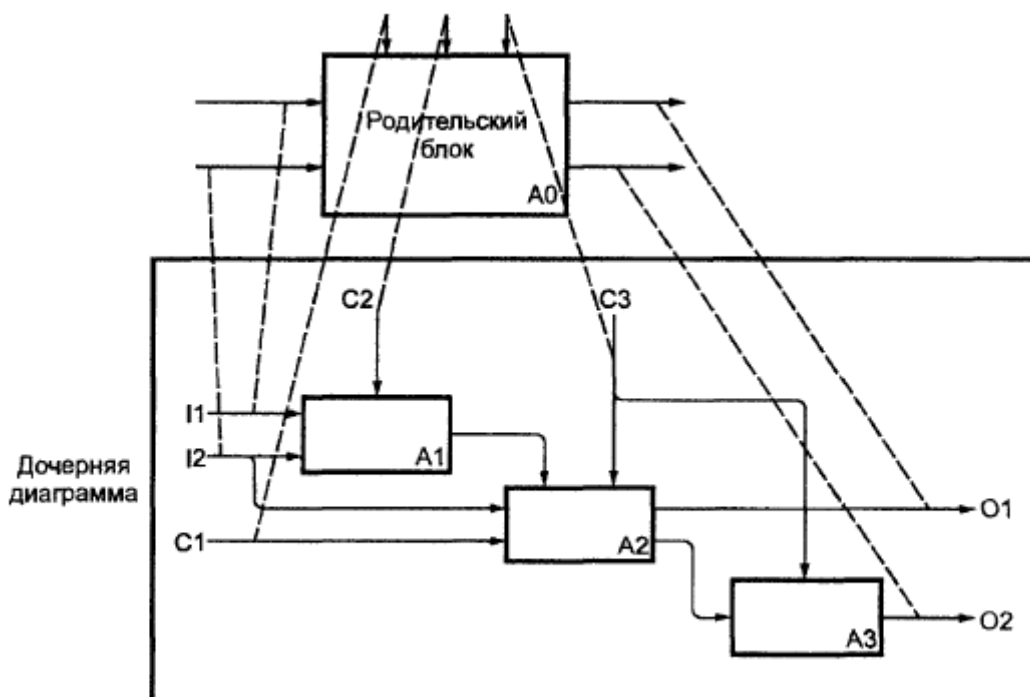


Рис. П. 2.20.

Примечание — Штриховые линии показывают отношения между

граничными стрелками и стрелками родительского блока.

Стрелки, помещенные в «туннель». *Туннель* — это круглые скобки в начале и/или в конце стрелки.

Туннельные стрелки означают, что данные, выраженные этими стрелками, не рассматриваются на родительской диаграмме и/или на дочерней диаграмме.

Стрелка, помещенная в туннель там, где она присоединяется к блоку (рис. П. 2.21), означает, что данные, выраженные этой стрелкой, не обязательны на следующем уровне декомпозиции.

Стрелка, помещаемая в туннель на свободном конце (рис. П. 2.22), означает, что представляемые ею данные отсутствуют на родительской диаграмме.

Более детально эта ситуация поясняется на рис. П. 2.23.

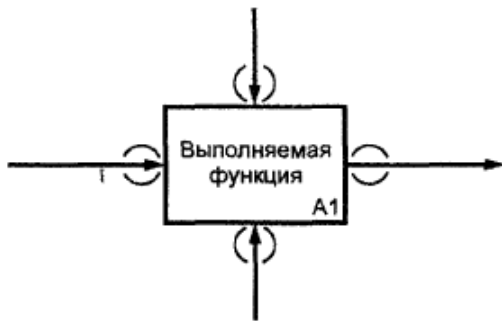


Рис. П. 2.22

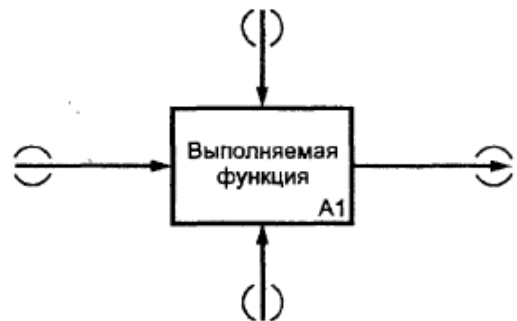


Рис. П. 2.22

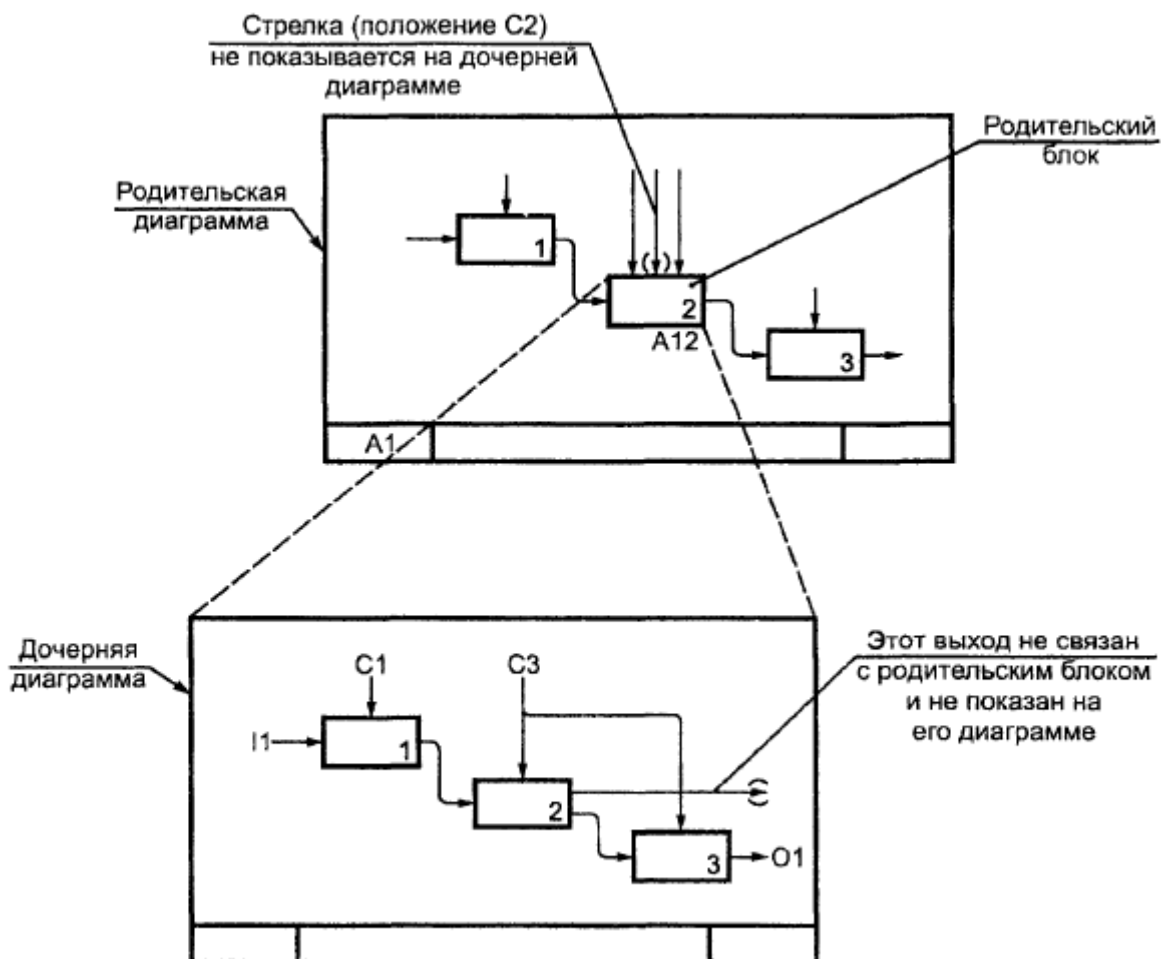


Рис. П. 2.23.

7. Правила построения диаграмм

При построении диаграмм необходимо выполнять следующие правила.

1. В составе модели должна присутствовать контекстная диаграмма А—0, которая содержит только один блок. Номер единственного блока на контекстной диаграмме А—0 должен быть 0.

2. Блоки на диаграмме должны располагаться по диагонали — от левого верхнего до правого нижнего угла диаграммы в порядке присвоенных номеров. Блоки на диаграмме, расположенные вверху слева, «доминируют» над блоками, расположенными внизу справа.

«Доминирование» понимается как влияние, которое блок оказывает на другие блоки диаграммы. Расположение блоков на листе диаграммы отражает авторское понимание доминирования.

Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные. При параллельном функционировании блоков отношение доминирования часто носит

формальный характер, и диагональное расположение блоков лишь способствует читаемости диаграммы.

3. Диаграммы (кроме диаграммы А—0) 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.

4. Каждый блок неконтекстной диаграммы получает номер, помещаемый в правом нижнем углу; порядок нумерации — от верхнего левого к нижнему правому блоку (от 1 до 6).

5. Каждый блок, подвергнутый декомпозиции, должен иметь ссылку на дочернюю диаграмму; ссылка (например, узловой номер, С-номер или номер страницы) помещается под правым нижним углом блока.

6. Имена блоков (выполняемых функций) и метки стрелок должны быть уникальными. Если метки стрелок совпадают, это значит, что стрелки отображают тождественные данные.

7. При наличии стрелок со сложной топологией целесообразно повторить метку для удобства ее идентификации.

8. Следует обеспечить максимальное расстояние между блоками и поворотами стрелок, а также между блоками и пересечениями стрелок для облегчения чтения диаграммы. Одновременно уменьшается вероятность того, что две разные стрелки будут перепутаны.

9. Блоки всегда должны иметь хотя бы одну управляющую и одну входную стрелку, но могут не иметь выходных стрелок.

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

11. Максимально увеличенное расстояние между параллельными стрелками облегчает размещения меток, их чтение и позволяет проследить пути стрелок (рис. П. 2.24).

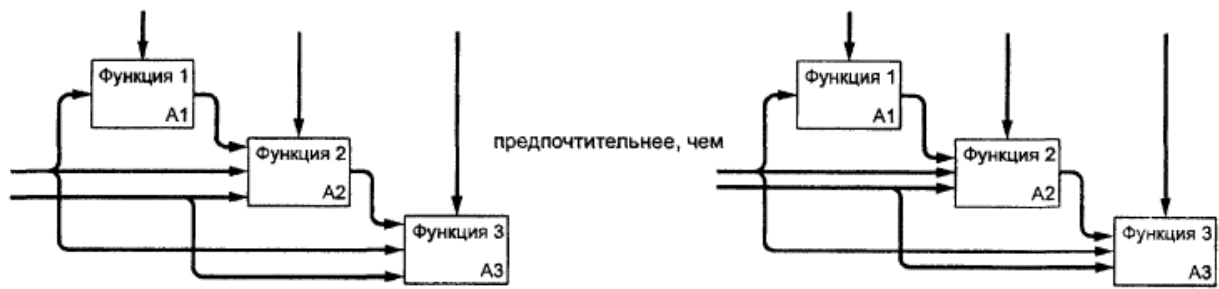


Рис. П. 2.24.

12 Стрелки связываются (сливаются), если они представляют сходные данные и их источник не указан на диаграмме (рис. П. 2.25).

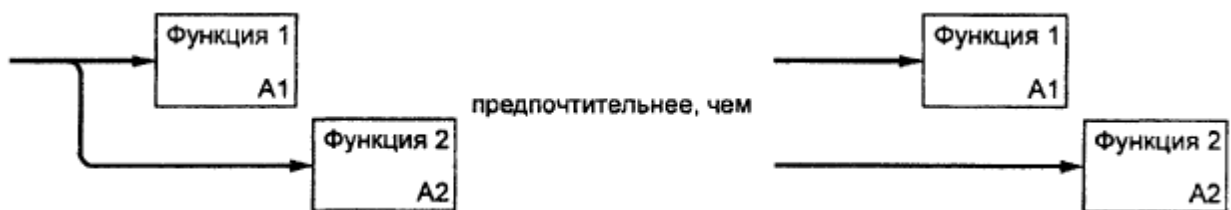


Рис. П. 2.25.

13 Обратные связи по управлению должны быть показаны как «вверх и над» (рис. П. 2.26а).

Обратные связи по входу должны быть показаны как «вниз и под» (рис. П. 2.26б). Так же показываются обратные связи посредством механизма (рис. П. 2.26в). Таким образом, обеспечивается показ обратной связи при минимальном числе линий и пересечений.

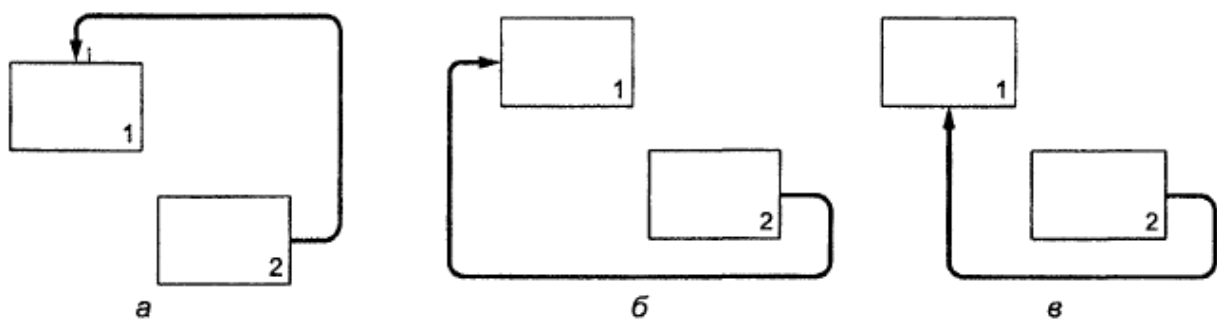


Рис. П. 2.26

14 Циклические обратные связи для одного и того же блока изображаются только для того, чтобы их выделить. Обычно обратную связь изображают на диаграмме, декомпозирующей блок. Однако иногда требуется выделить повторно используемые объекты (рис. П. 2.27).

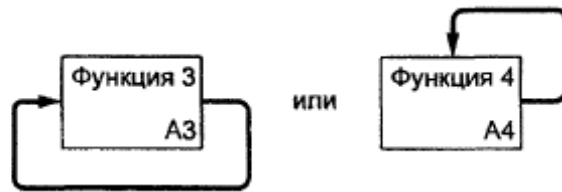


Рис. П. 2.27

15 Стрелки объединяются, если они имеют общий источник или приемник, или они представляют связанные данные. Общее название лучше описывает суть данных. Следует минимизировать число стрелок, касающихся каждой стороны блока, если, конечно, природа данных не слишком разнородна (рис. П.2.28).



Рис. П.2.28.

16. Если возможно, стрелки присоединяются к блокам в одной и той же позиции. Тогда соединение стрелок конкретного типа с блоками будет согласованным, и чтение диаграммы упростится (рис. П.2.29).



Рис. П.2.29.

17. При соединении большого числа блоков необходимо избегать необязательных пересечений стрелок (рис. П.2.30). Следует минимизировать число петель и поворотов каждой стрелки (рис. П.2.31)

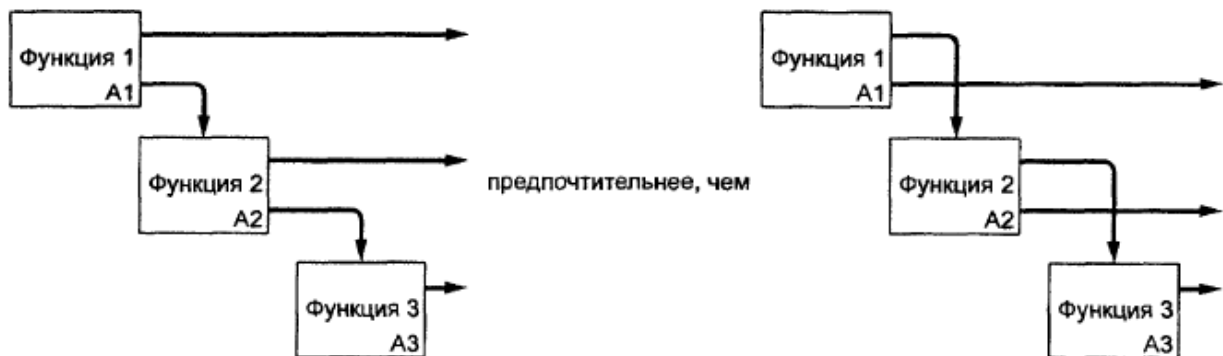


Рис. П.2.30

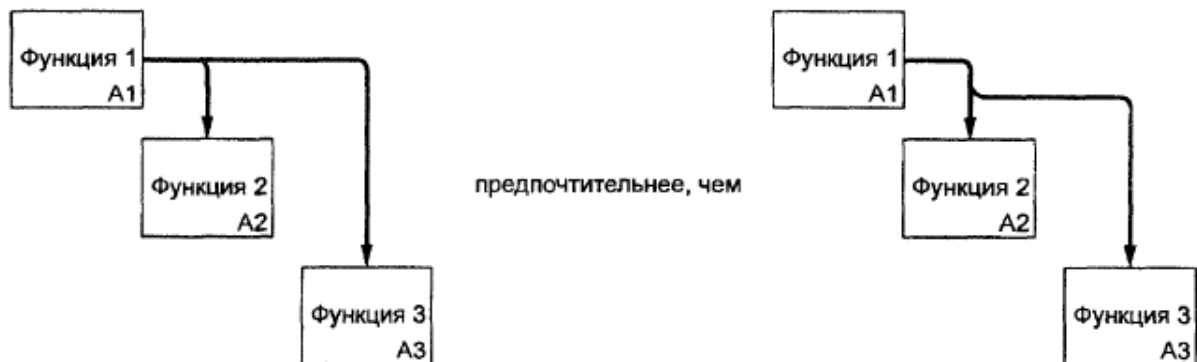


Рис. П.2.31

18 Блоки (функции) дочерней диаграммы являются сопряженными через среду (диаграмму или родительский блок), если они имеют связи с источником, генерирующим данные, без конкретного определения того, как отдельные части данных относятся к какому-либо блоку (рис. П.2.32).

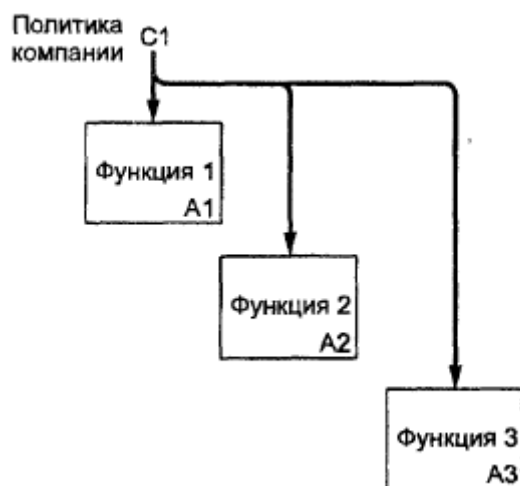


Рис. П.2.32.

19 Тип интерфейса, показанный на рис. П.2.33, предпочтителен, поскольку в этом случае определяются конкретные данные, относящиеся к каждому блоку.

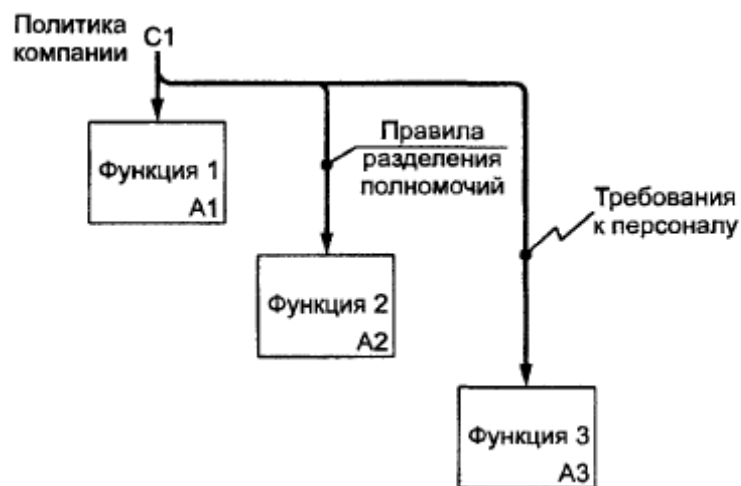


Рис. П.2.33

20 Необходимо использовать (там, где это целесообразно) выразительные возможности ветвящихся стрелок (рис. П.2.34).

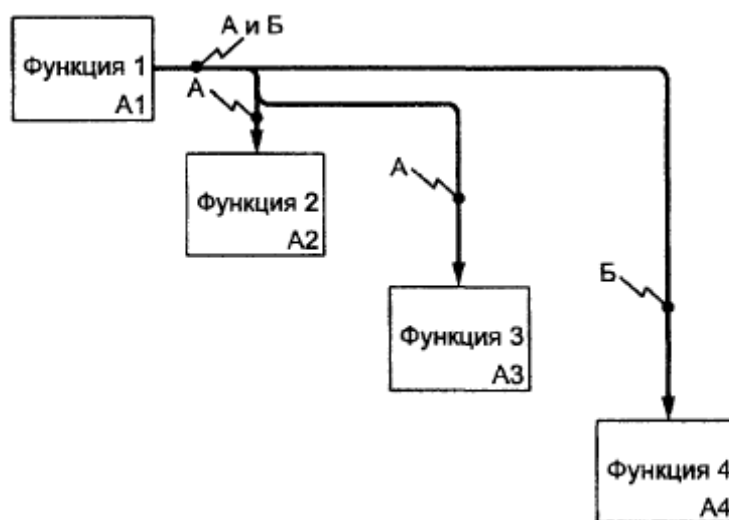


Рис. П.2.34

21. Каждая диаграмма IDEF0 изображается на стандартном бланке, именуемом мастер-страницей. Бланк снабжен верхним и нижним штампами, содержащими информацию, как о конкретной диаграмме, так и в целом о проекте, в состав которого входит диаграмма.

Ссылочные выражения (коды). *Ссылочные выражения (коды)* присваиваются всем элементам модели: диаграммам, блокам, стрелкам и примечаниям.

Ссылочные выражения затем могут использоваться в различных контекстах для точного указания на нужный элемент модели.

Основное ссылочное выражение — узловой номер, который появляется там, где выполняется декомпозиция функционального блока и создается его подробное описание на дочерней диаграмме. Все остальные ссылочные коды базируются на узловых номерах.

Номера блоков. Каждому блоку на диаграмме присваивается номер, помещаемый в нижнем правом внутреннем углу блока.

Эта система нумерации необходима для однозначной идентификации блоков в пределах диаграммы и для генерации узловых номеров. Эти номера используются также для ссылок на блоки в тексте и глоссарии.

На контекстной диаграмме А—0 единственному блоку присваивается номер 0 (ноль). На всех других диаграммах блоки нумеруются от 1 до 6, начиная с верхнего левого блока (при их диагональном размещении) и кончая нижним правым блоком. 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 против часовой стрелки.

Узловые номера

Узловой номер базируется на положении блока в иерархии модели. Обычно узловой номер формируется добавлением номера блока к номеру диаграммы, на которой он появляется.

Например, узловой номер блока 2 на диаграмме А25—А252. Все узловые номера IDEF0 начинаются с заглавной буквы, например «А». Когда родительский блок подробно описывается дочерней диаграммой, узловые номера родительского блока и дочерней диаграммы совпадают.

Контекстные диаграммы и дочерняя диаграмма верхнего уровня — исключения в вышеуказанной схеме узловой нумерации.

Каждая модель IDEF0 имеет контекстную диаграмму верхнего уровня — диаграмму А—0. Эта диаграмма содержит единственный «высший блок», который является уникальным родителем всей модели и несет уникальный номер 0 (ноль) и узловой номер А0.

Каждая модель IDEF0 должна также иметь по крайней мере одну дочернюю диаграмму, содержащую декомпозицию блока А0 на 3 ... 6 дочерних блоков. Этим блокам присваиваются уникальные узловые номера А1, А2, А3, . . . , А6. Таким образом, последовательность [А0, А1, . . . , А2, . . . , А3, . . .] начинает нумерацию узлов для любой модели.

Например, модель может иметь следующие узловые номера:

...	
А—1	Дополнительная контекстная диаграмма
А—0	Обязательная контекстная диаграмма верхнего уровня (содержащая высший блок А0)
А0	Верхняя дочерняя диаграмма
А1, А2, . . . , А6	Дочерние диаграммы
А11, А12, . . . , А16, . . . , А61, . . . , А66	Дочерние диаграммы
А111, А112, . . . , А161, . . . , А611, . . . , А666	Дочерние диаграммы
...	Дочерние диаграммы нижнего уровня

Узловой номер используется также для обозначения того, что блок подвергнут декомпозиции. В этом случае узловой номер, совпадающий с номером дочерней диаграммы, помещается под правым нижним углом блока на родительской диаграмме (рис. П.2.35).

Перечень 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. П.2.35.

- 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. П.2.35.

Дерево узлов. Разработанная модель IDEF0 со всеми уровнями структурной декомпозицией может быть представлена на единственной диаграмме в виде дерева узлов, дополняющего перечень узлов. Для изображения этого дерева нет стандартного формата. Единственное требование состоит в том, что вся иерархия узлов модели должна быть представлена наглядно и понятно.

Пример дерева узлов показан на рис. П.2.36.



Рис. П.2.36

Таблица П 3.1

№	Сущность	<u>Зависимость отношения</u> Имя отношения	Сущность	Вид отноше- ния
1	2	3	4	5
1	Проект дома	$\frac{1 : N}{\text{Используется длястроительства}}$	Дом	(Р-П)
2	Проект дома	$\frac{1 : 1}{\text{Определяет}}$	Фундамент	(Р-П)
3	Проект дома	$\frac{1 : 1}{\text{Определяет}}$	Стены	(Р-П)
4	Проект дома	$\frac{1 : 1}{\text{Определяет}}$	Крыша	(Р-П)
5	Проект дома	$\frac{1 : N}{\text{Определяет}}$	Материалы	(Р-П)
6	Проект дома	$\frac{1 : N}{\text{Используют длястроительства}}$	Строители	(Р-П)
7	Проект дома	$\frac{1 : N}{\text{Используют длястроительства}}$	Каменщики	(Р-П)
8	Проект дома	$\frac{1 : N}{\text{Используют длястроительства}}$	Кровельщики	(Р-П)
10	Проект дома	$\frac{1 : N}{\text{Используют длястроительства}}$	Плотники	(Р-П)
11	Проект дома	$\frac{1 : N}{\text{Используют длястроительства}}$	Отделочники	(Р-П)
12	Дом	$\frac{1 : 1}{\text{Включает в себя}}$	Фундамент	(Р-П)
13	Дом	$\frac{1 : 1}{\text{Включает в себя}}$	Стены	(Р-П)
14	Дом	$\frac{1 : 1}{\text{Включает в себя}}$	Крышу	(Р-П)
15	Дом	$\frac{1 : N}{\text{Используется длястроительства}}$	Материалы	(Р-П)

		Требует для <i>строительства</i>		
16	Дом	$\frac{1 : N}{\text{Строят}}$	Строители	(Р-П)
17	Дом	$\frac{1 : N}{\text{Строят}}$	Каменщики	(Р-П)
18	Дом	$\frac{1 : N}{\text{Строят}}$	Кровельщики	(Р-П)
19	Дом	$\frac{1 : N}{\text{Строят}}$	Плотники	(Р-П)
20	Дом	$\frac{1 : N}{\text{Строят}}$	Отделочники	(Р-П)
21	Фундамент	$\frac{1 : 1}{\text{Определяет}}$	Стены	(Р-П)
22	Фундамент	$\frac{1 : 1}{\text{Требует для строительства}}$	Материалы	(Р-П)
23	Фундамент	$\frac{1 : N}{\text{Строят}}$	Строители	(Р-П)
24	Фундамент	$\frac{1 : N}{\text{Строят}}$	Каменщики	(Р-П)
25	Фундамент	$\frac{1 : N}{\text{Строят}}$	Плотники	(Р-П)
26	Фундамент	$\frac{1 : N}{\text{Строят}}$	Отделочники	(Р-П)
27	Стены	$\frac{1 : 1}{\text{Определяют}}$	Крышу	(Р-П)
28	Стены	$\frac{1 : 1}{\text{Требуют для строительства}}$	Материалы	(Р-П)
29	Стены	$\frac{1 : N}{\text{Строят}}$	Строители	(Р-П)
30	Стены	$\frac{1 : N}{\text{Строят}}$	Каменщики	(Р-П)
31	Стены	$\frac{1 : N}{\text{Строят}}$	Плотники	(Р-П)
32	Стены	$\frac{1 : N}{\text{Строят}}$	Отделочники	(Р-П)
33	Крыша	$\frac{1 : 1}{\text{Требует для строительства}}$	Материалы	(Р-П)
34	Крыша	$\frac{1 : N}{\text{Строят}}$	Строители	(Р-П)
35	Крыша	$\frac{1 : N}{\text{Строят}}$	Кровельщики	(Р-П)
36	Крыша	$\frac{1 : N}{\text{Строят}}$	Плотники	(Р-П)

37	Крыша	$\frac{1 : N}{\text{Строят}}$	Отделочники	(Р-П)
38	Материалы	$\frac{M : N}{\text{Используют при строительстве}}$	Строители	(Н О)
39	Материалы	$\frac{M : N}{\text{Используют при строительстве}}$	Каменщики	(Н О)
40	Материалы	$\frac{M : N}{\text{Используют при строительстве}}$	Кровельщики	(Н О)
41	Материалы	$\frac{M : N}{\text{Используют при строительстве}}$	Плотники	(Н О)
42	Материалы	$\frac{M : N}{\text{Используют при строительстве}}$	Отделочники	(Н О)
43	Строители	$\frac{N : 1}{\text{Включают в себя}}$	Каменщики	(О К)
44	Строители	$\frac{N : 1}{\text{Включают в себя}}$	Кровельщики	(О К)
45	Строители	$\frac{N : 1}{\text{Включают в себя}}$	Плотники	(О К)
46	Строители	$\frac{N : 1}{\text{Включают в себя}}$	Отделочники	(О К)

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Югорский государственный университет»
Институт систем управления и информационных технологий
Кафедра компьютерного моделирования
и информационных технологий

КУРСОВАЯ РАБОТА
по дисциплине «Структурно-функциональное моделирование»

09.03.04 «Программная инженерия»

Группа:

Студент:

Преподаватель:

Ханты – Мансийск

20__ г.