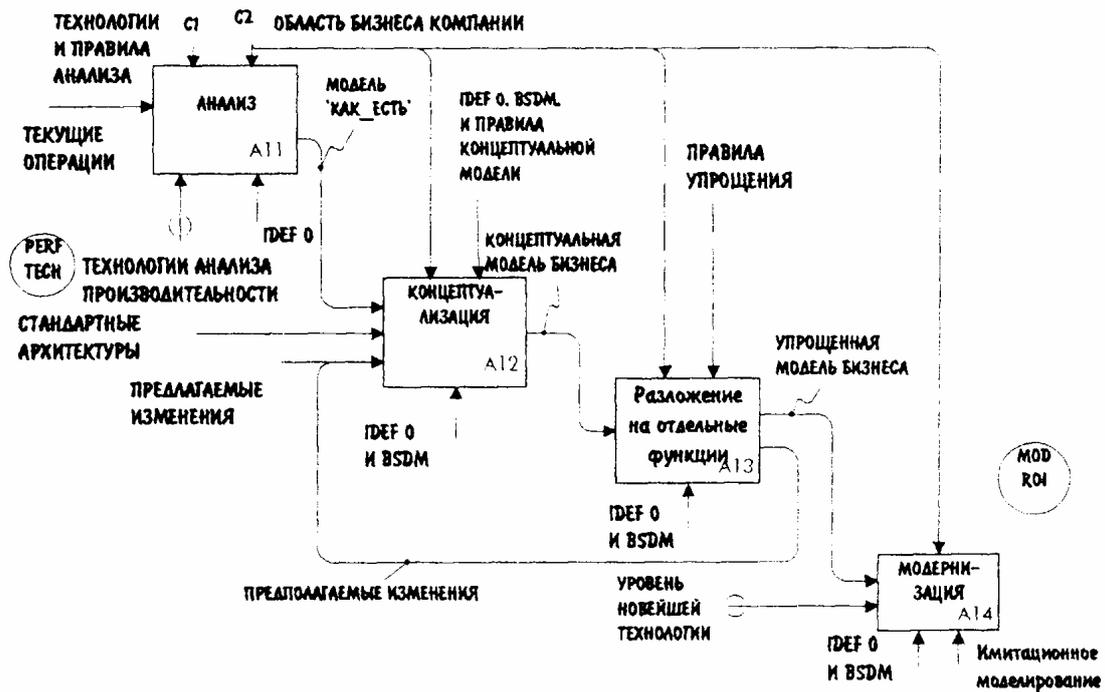


# Методология IDEF0

Стандарт

Русская версия



ARWAL-TR-81-4023  
VOLUME IV

**INTEGRATED COMPUTER-IADED MANUFACTURING (ICAM)  
ARCHITECTURE PART 11  
VOLUME IV - FUNCTION MODELING MANUAL ( IDEF0)**

SofTech, Inc.  
460 Totten Pond Road  
Waltham, -MA 02154

June 1981

Principal contributors to the development of IDEF0 are:

D.T. Ross	MIT and SofTech, Inc.	Developer of original concepts
S. Hori	IITRI	Developer of Cell Modeling Concepts
C.G. Feldmann	MIT and SofTech, Inc.	Primary author of first SADT Reference Manual and manager of first IDEF0 applications projects
D.E. Thornhill	SofTech, Inc.	Primary author of first formal SADT course material and application to government and commercial projects
R.R. Bravoco	SofTech, Inc.	Application of SADT and IDEP to Manufacturing
M.F. Connor and D.A. Marca	SofTech, Inc.	Developer of revised commercial SADT materials and applications to commercial projects
Judy Kornfeld	SofTech, Inc.	Developed criteria for judging quality of IDEF0 models

Русская редакция стандарта по методологии IDEF0 подготовлена фирмой МетаТехнология.

© МетаТехнология 1993

## Содержание

1. Введение.....	5
2. Основные понятия IDEF0.....	6
2.1. Предыстория.....	6
2.2. Концепции IDEF0.....	7
2.3. Обсуждение конкретных понятий.....	8
2.3.1. Графика блочного моделирования.....	8
2.3.2. Постепенно детализируемая информация.....	10
2.3.3. Координация коллективной работы.....	10
3. Понимание IDEF0-диаграмм.....	13
3.1. IDEF0-обозначения.....	13
3.1.1. Диаграммы.....	13
3.1.2. Блоки.....	14
3.1.3. Взаимосвязи между блоками и дугами.....	15
3.1.3.1. Дуги соединяют блоки.....	18
3.1.3.2. Дуги механизмов.....	19
3.1.4. Пример IDEF0-диаграммы.....	21
3.2. Дополнительные обозначения.....	23
3.2.1. Ссылочные выражения.....	23
3.2.1.1. Номера узлов.....	23
3.2.1.2. Имена моделей и номера узлов.....	24
3.2.2. Дуги, выходящие за пределы диаграммы.....	25
3.2.3. Кодирование граничных дуг.....	26
3.2.5. Туннельные дуги.....	27
3.2.6. Пример декомпозиции.....	29
4. Чтение IDEF0-диаграмм.....	34
4.1. Общий обзор модели.....	35
4.2. Основные этапы в процессе чтения диаграммы.....	36
4.3. Семантика блоков и дуг.....	37
4.3.1. Ограничения не показывают "как" и "когда".....	38
4.3.2. Многократные входы, управления и выходы.....	39
Часть 5. Бланки и процедуры IDEF.....	41
5.1. Принцип коллективной работы в IDEF-метопологии.....	41
5.2. Цикл IDEF-папки.....	41
5.2.1. Распределение ролей в цикле IDEF-папки.....	42
5.2.1.1. Авторы.....	43
5.2.1.2. Рецензент.....	43
5.2.2. Руководство для авторов и рецензентов.....	43
5.2.2.1. Руководство для рецензентов.....	43
5.2.2.2. Взаимодействия автора и рецензента.....	43
5.2.2.3. Правила встречи.....	44
5.3. IDEF-папки.....	44
5.3.1. Заполнение титульного листа стандартной папки.....	44
5.3.2. Подготовка стандартной папки.....	46
5.4. Стандартный бланк для диаграмм.....	47
5.4.1. Рабочая информация.....	48
5.4.2. Поле <i>Сообщение</i> .....	49
5.4.3. Поле <i>Название</i> .....	49
5.4.4. Поле <i>Номер</i> .....	50
5.5. Ведение архивов.....	50
5.5.1. Архив стандартных папок.....	50
5.5.2. Архив итоговых папок.....	50
5.5.3. Рабочий архив.....	50
5.6. Процедура сквозного контроля IDEF-модели.....	50
6. Руководство для автора по созданию IDEF0-диаграмм.....	53
6.1. Основные этапы построения диаграммы.....	53

6.1.1. Выбор контекста, точки зрения и цели модели .....	53
6.1.2. Построение контекстной диаграммы .....	54
6.1.3. Построение диаграммы верхнего уровня.....	54
6.1.4. Построение последовательных диаграмм.....	55
6.1.5. Разработка сопроводительного материала.....	55
6.1.6. Выбор блока для декомпозиции .....	55
6.1.7. Основные этапы в работе автора .....	56
6.1.7.1. Стадия сбора данных.....	56
6.1.7.2. Построение.....	56
6.1.7.3. Представление.....	56
6.1.7.4. Взаимодействие .....	57
6.2. Создание IDEF0-диаграммы .....	57
6.2.1. Разработка функциональных блоков.....	57
6.2.2. Создание интерфейсных дуг .....	58
6.2.3. Уровень усилий .....	60
6.3. Перечерчивание IDEF0-диаграмм.....	60
6.3.1. Модифицированные блоки.....	60
6.3.2. Дуги, собранные вместе.....	61
6.3.3. Модификация контекста.....	61
6.3.4. ICOM-синтаксис для соединения диаграмм .....	61
6.4. Графическое представление .....	62
6.4.1. Представление ограничений на диаграмме .....	62
6.4.2. Размещение дуг .....	63
6.4.3. Расположение дуг.....	64
6.5. Текст к диаграмме.....	66
6.5.1. Текст А-0.....	66
6.5.2. Ссылки и замечания .....	67
6.6. Контроль качества модели .....	67
6.6.1. Синтаксис.....	67
6.6.1.1. Синтаксис локальной конструкции .....	68
6.6.1.2. Синтаксис глобальной конструкции.....	69
6.6.1.3. Синтаксис конструкции модели.....	69
6.6.2. Семантика .....	70
6.6.3. Сопряжение и связность .....	76
6.6.4. Показатели, основанные на сопряжении и связности .....	76
6.6.4.1. Связь с другими аспектами системного проектирования.....	77
6.6.4.2. Показатели и типы сопряжения .....	77
6.6.5. Показатели и типы связности.....	80
7. Сбор информации для IDEF-модели .....	86
7.1. Вступление .....	86
7.2. Процесс опроса .....	86
7.3. Папка опроса .....	87
7.4. Правила проведения опроса.....	87
7.4.1. Подготовка к опросу .....	88
7.4.2. Вводная стадия .....	88
7.4.3. Проведение опроса.....	89
7.4.4. Окончание опроса .....	90
7.4.5. Финальная стадия.....	91

## 1. Введение

Программа интегрированной компьютеризации производства (ICAM), предложенная ВВС США, направлена на увеличение производительности в промышленности посредством повсеместного внедрения компьютерных технологий. Подход, лежащий в основе программы ICAM, заключается в разработке структурных методов, способствующих применению компьютерных технологий в промышленном производстве, и использовании этих методов для лучшего понимания путей повышения эффективности производства.

Программа ICAM выявила потребность, в более совершенных способах обмена информацией и методах анализа производственных систем для всех специалистов, занимающихся проблемами повышения производительности. Для удовлетворения этой потребности в рамках программы ICAM разработана методология IDEF (ICAM Definiton), позволяющая проводить исследование определенных характеристик промышленного производства. IDEF состоит из трех методологий моделирования, основанных на графическом представлении производственных систем:

- IDEF0 используется для создания функциональной модели, которая является структурированным изображением функций производственной системы или среды, а также информации и объектов, связывающих эти функции;
- IDEF1 применяется для построения информационной модели, которая представляет структуру информации, необходимой для поддержки функций производственной системы или среды;
- IDEF2 позволяет построить динамическую модель меняющегося во времени поведения функций, информации и ресурсов производственной системы или среды.

Каждая из этих трех моделей или любое их сочетание позволяет сформировать "архитектуру" среды моделируемой системы. Эта среда включает другие системы, организации или технологии, которые должны работать совместно для достижения общей цели производственной среды или системы. Назначение модели, определяемой как "архитектура", заключается в том, что она графически представляет основные взаимоотношения в среде моделируемой системы - функциональные связи, идентификацию информационных потоков (общих, разделяемых, дискретных), а так же динамическое взаимодействие ресурсов. Важно уяснить, что IDEF0-модель становится "архитектурой", когда она используется для более глубокого понимания и анализа не только производственной системы или ее окружения, но и того, как ее компоненты (подсистемы, организации и технологии), взаимодействуют друг с другом.

IDEF является *методологией*, архитектура является *содержанием*, а повышение производительности - *целью*, ради которой в аэрокосмической промышленности была инициирована программа ICAM .

Далее будут рассмотрены основные понятия, методы и процедуры, относящиеся к использованию IDEF0 для построения *функциональной* модели.

## 2. Основные понятия IDEF0

### 2.1. Предыстория

Стремление ВВС США оказать помощь аэрокосмической промышленности в сокращении затрат и времени на модернизацию производства отражено во многих программах модернизации технологии (так называемых "Tech.Mod" - программах). Подобная задача, но в более широком, "индустриальном", плане, чем у других отдельных компаний, поставлена перед программой ICAM.

Целью программы ICAM была разработка "основных подсистем общего назначения", которые могут использоваться большим числом компаний для достижения значительного прогресса в промышленности. Эти "подсистемы" обеспечивают выполнение общих производственных функций, таких, как управление информацией, диспетчерская служба, материальное снабжение.

Такая крупномасштабная цель нуждалась в некотором связующем базисе, вокруг которого осуществлялось бы планирование, разработка и внедрение подсистем в отдельных аэрокосмических компаниях. Этот базис был назван "архитектурой производства", так как это понятие представляло "архитектуру" производства в широком смысле, отражало его современное состояние и являлось основой, опираясь на которую можно было бы спланировать, разработать и внедрить общие подсистемы.

Для разработки "архитектуры производства" нужен был язык, с помощью которого можно описывать функционирование современной аэрокосмической промышленности. В начале работы над программой ICAM ВВС выпустили "Лист предложений", касающихся построения архитектуры. В качестве выразительного языка был выбран "Метод блочного моделирования" (в котором "блок" определен как производственная ячейка или функциональная единица).

Для успешного применения язык должен удовлетворять следующим критериям:

- Так как целью архитектуры является описание производства, язык должен быть в состоянии выразить производственные операции естественным и несложным образом.
- Поскольку моделируемый субъект является весьма обширным и сложным, язык должен быть кратким, и обеспечивать простой и быстрый поиск интересующих деталей.
- Поскольку язык используется широкой аудиторией, он должен обеспечивать взаимодействие как персонала, работающего в аэрокосмической промышленности, так и персонала управления ВВС.
- Так как язык будет служить базой для планирования общих подсистем, их разработки и внедрения, он должен обеспечивать достаточную строгость и точность, чтобы избежать получения неверных результатов.
- Поскольку процесс разработки общих подсистем требует совместных усилий крупных подразделений аэрокосмической промышленности, язык должен включать методологию (правила и процедуры) его использования, что позволит большому числу рабочих групп разрабатывать отдельные компоненты архитектуры системы и сделает возможным широкое обсуждение, рецензирование, критику и утверждение результатов.
- Так как предпочтительным является представление аэрокосмической промышленности в целом, а не какой-либо отдельной компании или промышленного подразделения, методология должна включать средства отделения "организационной"

структуры от "функций". Это означает, что общие соглашения могут быть достигнуты только в том случае, когда различия в организационных структурах отдельных компаний не принимаются в расчет, а рассматриваются только общие функциональные связи.

BBC США в качестве методологии блочного моделирования выбрали методологию SADT (Structured Analysis and Design Technique), разработанную в начале семидесятых годов. Основные элементы этой методологии, использованные в программе ICAM, позже получили название " IDEF0".

## 2.2. Концепции IDEF0

IDEF0-методология основана на следующих концепциях:

1. **Графическое представление блочного моделирования.** Графика "блоков и дуг" IDEF0-диаграммы отображают производственную операцию в виде блока, а интерфейсы входа/выхода в/из операции представляются дугами, соответственно входящими в блок или выходящими из него. Для того чтобы иметь возможность описывать производственные операции, существующие в реальности, было предложено описывать взаимодействие блоков друг с другом посредством интерфейсных дуг, выражающих "ограничения", которые в свою очередь определяют, когда и каким образом операции выполняются и управляются.

2. **Краткость.** Документация архитектуры производственной системы для полноценного охвата материала должна быть точной. Многословные характеристики обычных языковых текстов неудовлетворительны. Двумерная форма графического языка имеет требуемую точность без потери возможности выразить такие взаимоотношения, как интерфейс, обратная связь, ошибочные пути.

3. **Передача информации.** В IDEF существует ряд средств, разработанных для улучшения передачи информации:

- диаграммы, основанные на очень простой графике блоков и дуг;
- метки на естественном языке для описания блоков и дуг, а также глоссарий и сопроводительный текст для определения точного значения элементов диаграммы;
- постепенное представление деталей, при котором на верхнем уровне иерархии показаны основные функции, а на следующих уровнях происходит их более подробное уточнение;
- схема узлов в иерархии диаграмм, обеспечивающая возможность легко составить перечень (индекс) размещенных на них деталей;
- ограничение каждой диаграммы шестью подфункциями для облегчения чтения.

4. **Строгость и точность.** Выполнение правил IDEF0 требует достаточной строгости и точности, чтобы удовлетворить принципам архитектуры ICAM, не накладывая в то же время чрезмерных ограничений на аналитика. Правила IDEF включают:

- Ограничение количества деталей на каждом уровне (правило 3-6 блоков).
- Ограниченный контекст (без пропусков, но и без дополнительных деталей, выходящих за рамки рассмотрения).
- Связность интерфейса диаграмм (номера узлов, номера блоков, C-номера).
- Связность структуры данных (ICOM-коды и использование туннельных дуг).
- Уникальность меток и наименований (отсутствие повторяющихся имен).
- Синтаксические правила для графики (блоков и дуг).

- Ограничения на ветвление дуг данных (метки для ограничений потоков данных на ветвях).
- Разделение входов и управлений (правило определения роли данных).
- Требования к меткам дуг данных (правила минимальных меток).
- Минимальное управление для функций (для каждой функции нужна, по крайней мере, одна управляющая дуга).
- Цель и точка зрения (у каждой модели есть цель и точка зрения).

5. **Методология.** Пошаговые процедуры обеспечивают моделирование, рецензирование и решение задач интеграции. Существуют соответствующие курсы для обучения персонала аэрокосмической промышленности этим методам.

6. **"Организация" из "функций".** Отделение организации от функции включено в цель модели и осуществляется отбором имен функций и связей в процессе разработки модели. Это положение входит в курс по IDEF0, а постоянное рецензирование в ходе создания модели помогает избежать точки зрения, навязанной организацией.

### **2.3. Обсуждение конкретных понятий**

В следующих разделах приведено описание некоторых основных понятий, объяснено и показано их применение в ICAM.

#### **2.3.1. Графика блочного моделирования**

Методология IDEF0 может использоваться для моделирования широкого круга систем, где под системой понимается любая комбинация средств аппаратного и программного обеспечения, а также людей. При создании новых систем IDEF0 может вначале применяться для определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции. Для уже существующих систем IDEF0 может быть использована для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются.

Результатом применения методологии IDEF0 является модель. Модель состоит из диаграмм, фрагментов текста и глоссария, которые имеют ссылки друг на друга. Диаграммы - главные компоненты модели. На диаграммах все функции производственной системы и интерфейсы представлены как блоки (функции) и дуги (интерфейсы). Место соединения дуги с блоком определяет тип интерфейса. Управляющие производством данные входят в блок сверху, в то время как материалы или информация, которые подвергаются производственной операции, показаны с левой стороны блока; результаты выхода показаны с правой стороны. Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу (рис. 2.1).

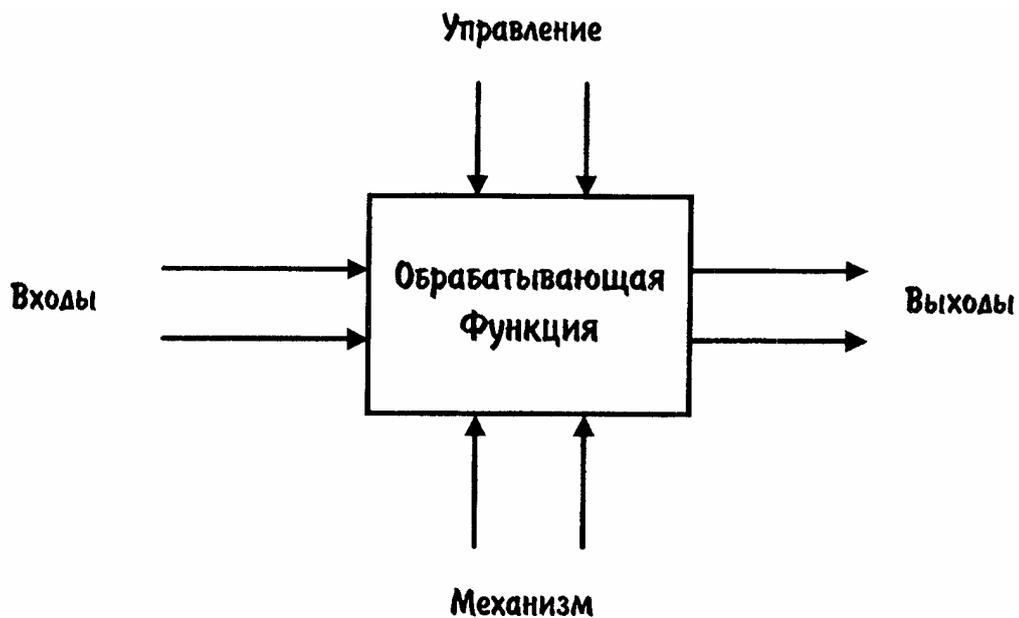


Рис. 2.1. Функциональный блок и интерфейсные дуги

Блоки и дуги в IDEF0-модели используются для представления связей между несколькими подфункциями на диаграмме, описывающей более общую функцию. Эта диаграмма является "подчиненной" диаграммой и показывает конкретные интерфейсы, управляющие каждой подфункцией, а также источники и адресаты этих интерфейсов (см. рис. 2.2).

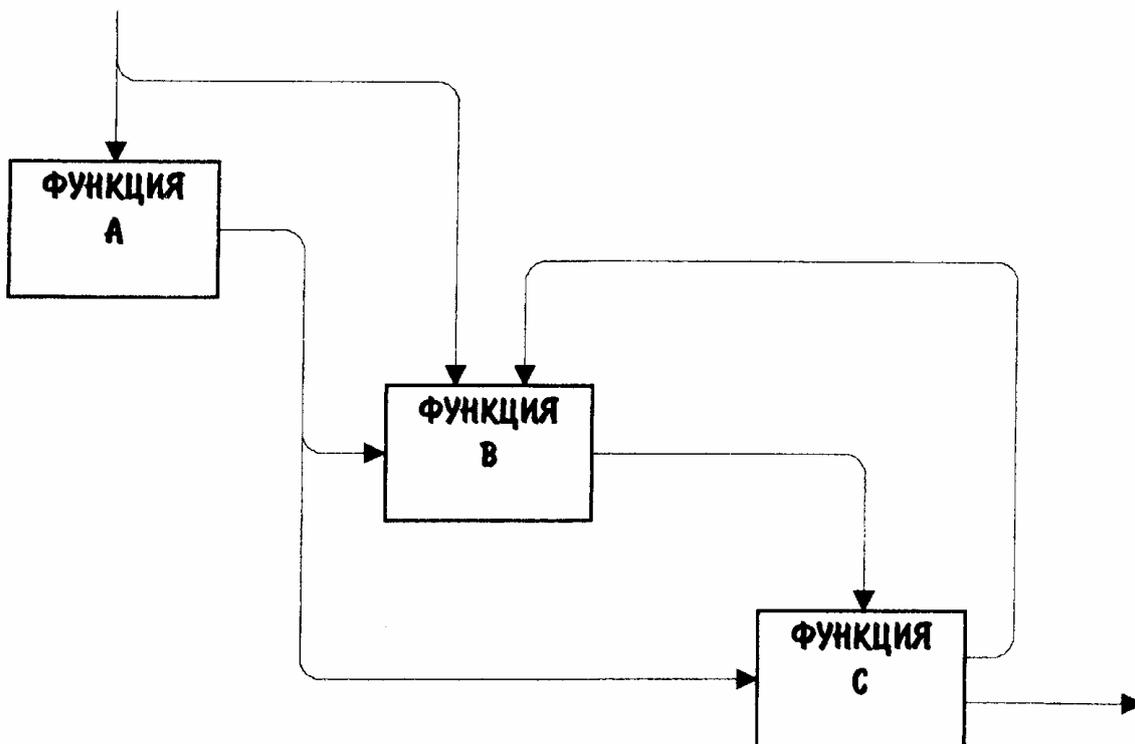


Рис. 2.2. Диаграмма, демонстрирующая ограничения на выполнение функций.

(Функция В зависит от одного входа и двух управлений и производит один выход, от которого зависит функция С)

Здесь термин "ограничение" означает, что функция использует материальные объекты или информацию, изображаемую входящей в блок и, следовательно, ограничена в своем действии зависимостью от интерфейсов; функция не может выполняться, пока не обеспечено содержание интерфейсных дуг, а вариант действия, в соответствии с которым работает функция, зависит от конкретного содержания интерфейсных дуг (объем, количество и т.д.).

### 2.3.2. Постепенно детализируемая информация

Одной из наиболее важных особенностей методологии IDEF0 является постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель. Таким образом обеспечивается представление информации, и читатель располагает хорошо очерченным предметом изучения с приемлемым объемом новой информации на каждой следующей диаграмме.

Структура IDEF0-модели показана на рис. 2.3., где приведены четыре диаграммы и их взаимосвязи.

Построение IDEF0-модели начинается с представления всей системы в виде простейшей компоненты - одного блока и дуг, изображающих интерфейсы с функциями вне системы. Поскольку единственный блок представляет всю систему как единое целое, имя, указанное в блоке, является общим. Это верно и для интерфейсных дуг - они также представляют полный набор внешних интерфейсов системы в целом.

Затем блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки представляют основные подфункции (подмодули) единого исходного модуля. Данная декомпозиция выявляет полный набор подмодулей, каждый из которых представлен как блок, границы которого определены интерфейсными дугами. Каждый из этих подмодулей может быть декомпозирован подобным же образом для более детального представления.

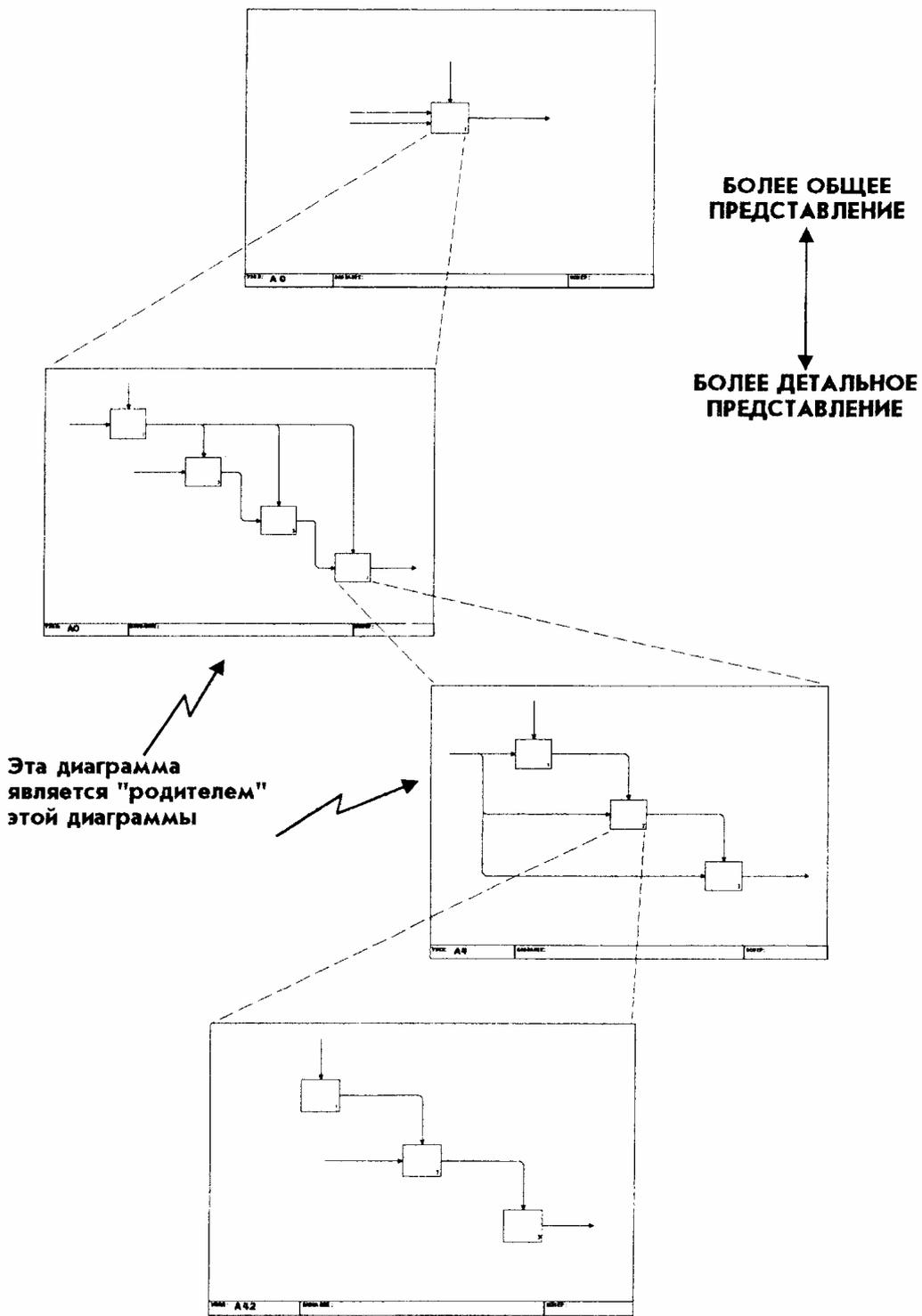
В IDEF0 есть свои правила постепенного добавления деталей в процессе декомпозиции. Модуль всегда делится не менее чем на три, но не более чем на шесть подмодулей. Верхний предел - шесть - позволяет использовать иерархию для описания более сложных объектов. Нижний предел - три - гарантирует введение достаточного количества деталей, чтобы полученная декомпозиция представляла интерес.

Взаимоотношения диаграмм показаны с помощью интерфейсных дуг. Когда модуль декомпозируется на подмодули, интерфейс между ними представляется дугами. Имя каждого блока-подмодуля плюс метки интерфейсных дуг определяют связанный контекст для этого подмодуля.

Во всех случаях каждый подмодуль может содержать только те элементы, которые входят в исходный модуль. Кроме того, модуль не может опустить какие-либо элементы, т.е., как уже отмечалось, родительский блок и его интерфейсы обеспечивают контекст. К нему нельзя ничего добавить, и из него не может быть ничего удалено.

### 2.3.3. Координация коллективной работы

Методология IDEF0 включает методы разработки и критического анализа моделей большим коллективом, а также методы интеграции подсистем в IDEF0-архитектуру. Кроме того, в методологию IDEF0 входят вспомогательные процедуры, например, правила и способы ведения библиотек. Следует отметить, что некоторые из этих правил и способов, такие, как процедура рецензирования "Цикл автор-рецензент", используются и в других IDEF-методологиях.



КАЖДАЯ КОМПОНЕНТА МОЖЕТ БЫТЬ ДЕКОМПОЗИРОВАНА НА ДРУГОЙ  
 ДИАГРАММЕ. КАЖДАЯ ДИАГРАММА ИЛЛЮСТРИРУЕТ "ВНУТРЕННЕЕ СТРОЕНИЕ"  
 БЛОКА НА РОДИТЕЛЬСКОЙ ДИАГРАММЕ

Рис. 2.3. Структура IDEF0-модели

Создание IDEF0-модели является основной компонентой "скоординированной коллективной работы". Создание модели - динамический процесс, требующий обычно участия более чем одного человека. При разработке проекта авторы создают первоначальные диаграммы, которые передаются участникам проекта для рассмотрения и замечаний.

Порядок требует, чтобы каждый эксперт, у которого есть замечания к диаграмме, сделал их письменно и передал автору диаграммы. Этот цикл продолжается до тех пор, пока диаграммы, а затем и вся модель не будут приняты.

В IDEF0 предусмотрено сохранение записей о всех решениях и альтернативных подходах по мере того, как они возникают на протяжении проекта. Копии диаграмм, разработанных автором, критически анализируются компетентными экспертами, которые заносят свои предложения непосредственно на копии. Авторы отвечают на каждое замечание письменно на тех же копиях. Предложения принимаются или отвергаются письменно с указанием причины. После внесения изменений и исправлений старые варианты диаграмм остаются в архиве проекта.

Диаграммы корректируют, чтобы отразить уточнения и замечания. Затем добавляются новые детали с помощью новых диаграмм, которые, в свою очередь, пересматриваются и изменяются. Окончательная модель отражает согласованное представление системы с определенной точки зрения и с определенной целью. Эта модель легко может быть прочитана, использована для представления спецификаций системы во время коротких обсуждений или долгих дискуссий, а также для организации новых проектов, связанных с изменениями системы.

### 3. Понимание IDEF0-диаграмм

Модель - это представление системы. Она может описать, что является сутью системы, что она делает и с какими объектами работает.

Системы состоят из взаимосвязанных или взаимозависимых частей, работающих вместе для выполнения определенных функций. Эти части могут быть любыми, включая машины, информацию, объекты производства, процессы, программное обеспечение или людей. IDEF0 может быть использована для описания функций, выполняемых системами или частями систем, а также информации или предметов, посредством которых функции связаны между собой.

IDEF0 описывает систему с помощью модели, состоящей из диаграмм, текста и глоссария. Диаграммы состоят из блоков и дуг. Блоки в них представляют действия, а дуги - объекты, обрабатываемые системой.

#### 3.1. IDEF0-обозначения

##### 3.1.1. Диаграммы

Модель IDEF0 - представляет собой серию диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части. Первоначальная диаграмма является наиболее общим или абстрактным описанием всей системы. Она показывает каждую основную составляющую системы в виде блока. Детали каждого из основных блоков показаны в виде блоков на других диаграммах. Эти блоки могут быть превращены в еще более подробные диаграммы, и так до тех пор, пока не будет достигнут требуемый уровень детализации.

Каждая детальная диаграмма является декомпозицией блока из более общей диаграммы. На каждом шаге декомпозиции более общая диаграмма называется родительской для более детальной диаграммы. Лучше всего представлять себе, что диаграмма декомпозиции как бы расположена внутри родительского блока (см. рис. 3.1.).

Блоки представляют функции системы (действия, процессы или операции), а дуги - данные (информацию или предметы). Блок на диаграмме рассматриваемого уровня описывается более подробно блоками и дугами диаграммы более низкого уровня. Дуги, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются точно теми же самыми, что и дуги, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграмма представляют одну и ту же часть системы.

В соответствии с одним из фундаментальных правил IDEF0, диаграмма не может иметь менее трех и более шести блоков. Это условие обеспечивает единое, систематическое представление последовательных уровней детализации. Верхнее ограничение - шесть - выбрано потому, что, как показали психологические эксперименты, трудно уловить в одно и то же время более пяти - семи различающихся понятий. Нижнее ограничение - три - выбрано для уверенности в том, что полученная детализация достаточно информативна. Диаграммы верхних уровней охватывают большое количество деталей, поэтому словесные выражения, относящиеся к блокам и дугам, должны быть максимально обобщенными и давать общее понятие. На последующих диаграммах нижних уровней эти детали постепенно уточняются с использованием более конкретных терминов.

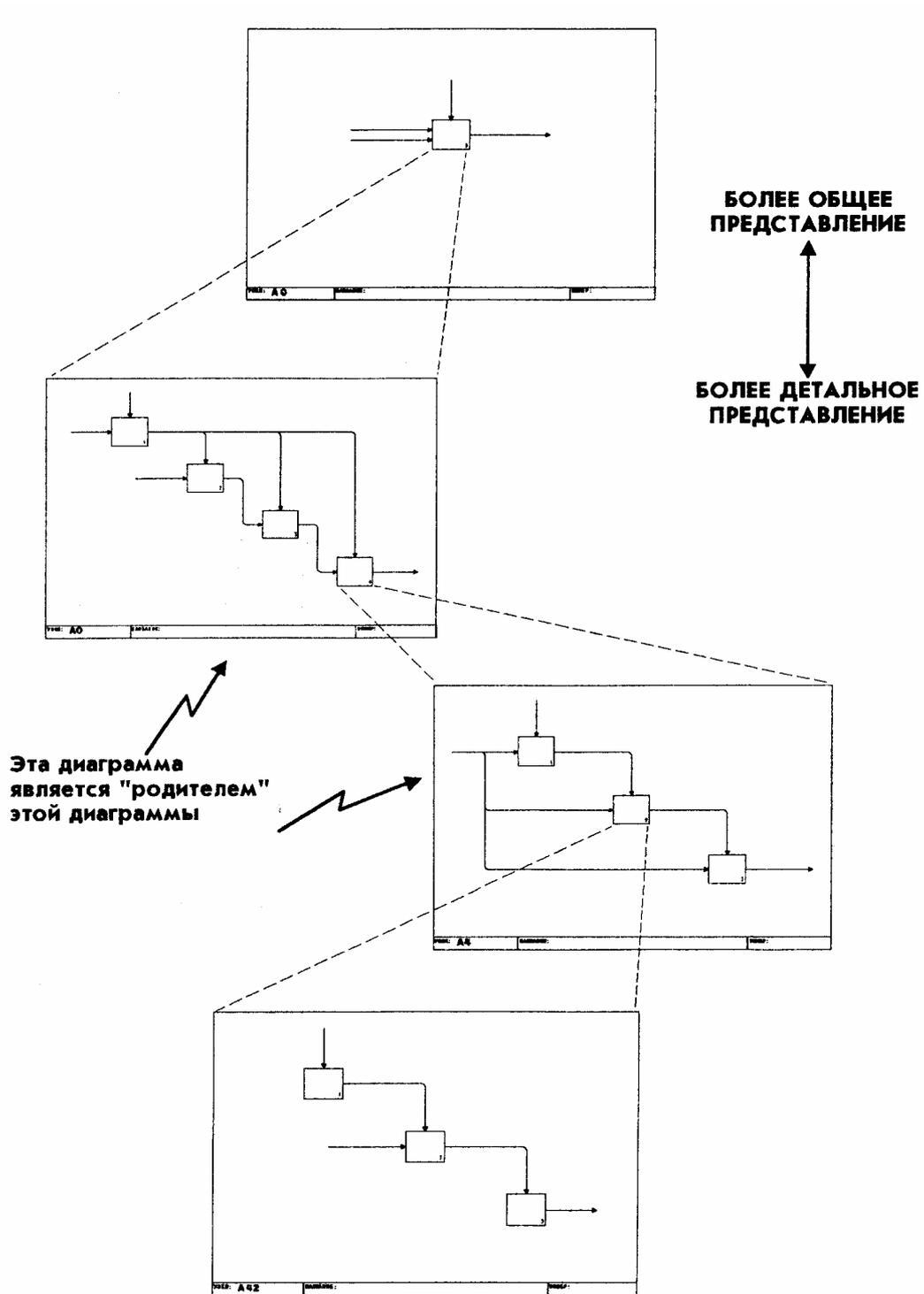


Рис. 3.1. Декомпозиция диаграмм.

### 3.1.2. Блоки

Блоки на диаграмме представляют функции. Функции показывают, что должно выполняться, не идентифицируя при этом какие-либо другие аспекты, такие как необходимость в них или их смысл. Имена функций записываются внутри блоков. Они должны содержать глагольный оборот. Каждый блок на диаграмме имеет номер в диапазоне 1-6, записанный в нижнем правом углу.

**Функция - это все, что может быть выражено активным глагольным оборотом.** Сюда входит все - от конкретного до абстрактного, например:

*сжимать, собирать, классифицировать, прикреплять, переписывать, конструировать, измерять, оценивать, решать, принимать, разрешать, разрабатывать.*

Эти функции выполняются вне временных периодов. Функции не выражаются существительными, такими, как "поддержка".

### 3.1.3. Взаимосвязи между блоками и дугами

Дуги, соединенные с блоками, представляют материальные объекты или информацию, в которой нуждается или которую производит функция. Каждая из дуг имеет метку в виде оборота существительного, записываемую над ней. **"Данными" может служить информация, предметы и все остальное, что может быть описано оборотом существительного.**

Дуги являются для блоков ограничивающим фактором, который определяет существо блоков, но не последовательности или потоки функций (рис. 3.2.).

Сторона блока, в которую дуга входит или из которой выходит, показывает назначение дуги в качестве входа, управления или выхода. Входящие с левой и верхней стороны блока дуги представляют данные, необходимые для выполнения функции. Выходные дуги (с правой стороны блока) изображают данные, полученные в результате выполнения функции. Функции преобразуют данные слева направо (от входа к выходу). Вход превращается функцией в выход. Термины "вход" и "выход" означают, что блок представляет собой переход от состояния "до" к состоянию "после".

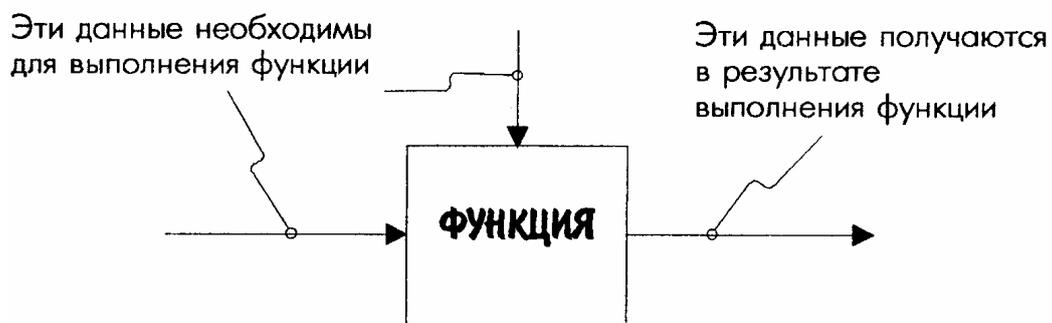


Рис. 3.2. Дуги объясняют и ограничивают смысл каждого блока.

"Управление" описывает условие или обстоятельства, которые управляют функцией. Назначение входа и назначение управления различно. Их различие является важным для понимания работы системы.

Принято, что дуга является управляющей, если не очевидно, что она определяет только вход. Каждый функциональный блок имеет по крайней мере одну управляющую дугу. К нижней части блока может присоединяться дуга "механизма", обозначающая либо человека, либо некоторое средство, выполняющее функцию. Вход и выход показывают, что делается функцией, управление показывает, почему это делается, а механизм показывает, как это делается (рис. 3.3.).



Рис. 3.3. Взаимодействие блок/дуга.

Блоки представляют совокупность взаимосвязанных функций, а не просто отдельные действия. Блок может выполнять при различных обстоятельствах отдельные компоненты своей функции, используя разнообразные комбинации входа и управлений и получая соответствующие выходы. Эти комбинации называются различными активациями блока. С любой из сторон блока могут быть соединены несколько дуг, относящиеся к различным активациям.

Дуги на IDEF0-диаграмме представляют ограничения, накладываемые на данные; это не потоки и не последовательности. Соединение выхода одного блока с входом или управлением другого показывает ограничение (рис. 3.4.). Работа блока, получающего данные, ограничена в том смысле, что функция не может быть выполнена, пока не станут доступными данные, производимые другими блоками. Дуги, входящие в блок, показывают все данные, которые необходимы для выполнения функции.

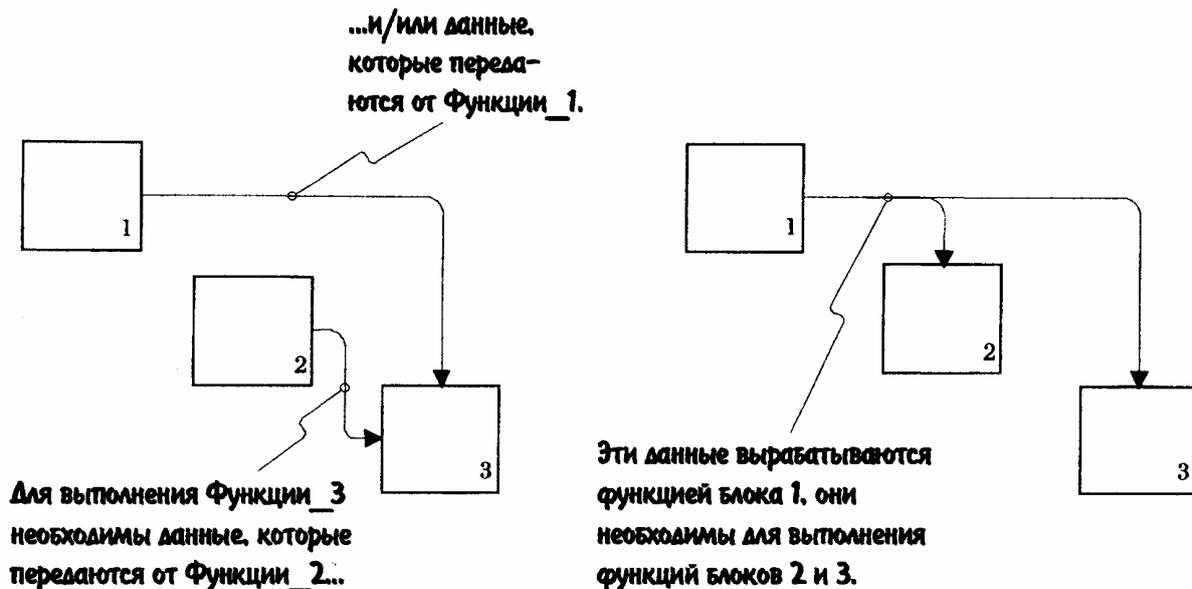


Рис. 3.4. Смысл ограничений.

Несколько функций на одной диаграмме могут выполняться одновременно, если выполнены требуемые условия (ограничения). Дуги соединяют блоки, и выход одного блока может обеспечить некоторые или все данные, необходимые для одного или более блоков (рис. 3.5.).

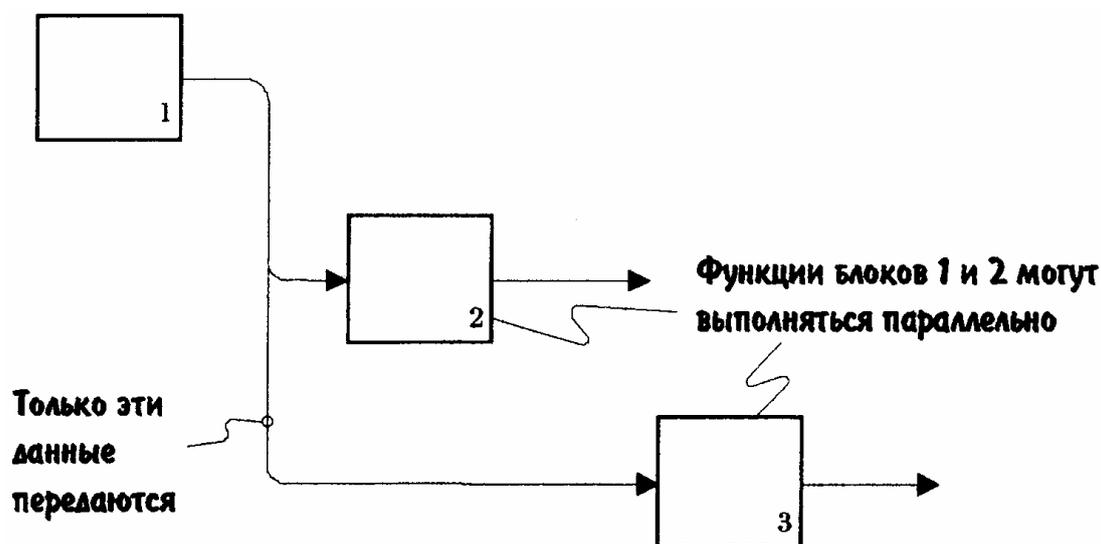


Рис. 3.5. Одновременное выполнение.

Ни последовательность, ни время не указаны явно на IDEF0-диаграммах. Обратные связи, итерации, продолжающиеся процессы и перекрывающиеся (по времени) функции могут быть легко изображены с помощью дуг (рис. 3.6.). Дуги, присоединяющиеся к левой стороне (вход) или к верхней стороне (управление) блока, являются ограничениями. Они представляют собой данные или объекты, необходимые для выполнения некоторых функций. Например, предварительная спецификация системы, представленная на экспертизу, может быть одобрена и признана окончательной, а может быть возвращена с замечаниями и с требованием представить на рассмотрение новый вариант проекта. Это вновь активизирует функцию проектирования. Как проектирование, так и экспертиза выполняются согласно системным требованиям.

Дуги могут разветвляться (в случае, если одни и те же данные требуются более, чем для одной функции) или соединяться (в случае, если однотипные данные могут быть выработаны более чем одной функцией). Каждая из ветвей может представлять один и тот же объект или различные объекты одного и того же типа. Метки указывают назначения дуг. Метки на ветвях и соединениях обеспечивают детализацию содержания более общих дуг, так же, как диаграммы нижнего уровня обеспечивают детализацию блоков.

Дуги с данными, подобно функциональным блокам, имеют разные уровни детализации. Полезно представить верхние уровни дуг данных в виде трубопроводов. Дуги верхнего уровня имеют метки более общего характера. Если они разветвляются, каждая ветвь имеет более конкретную надпись (рис. 3.7). Метки дуг должны передавать замыслы автора читателю. Используя меньшее число дуг, можно уменьшить загроможденность и сделать диаграмму более легкой для восприятия, но при этом требуется более тщательный подбор слов для передачи информации.

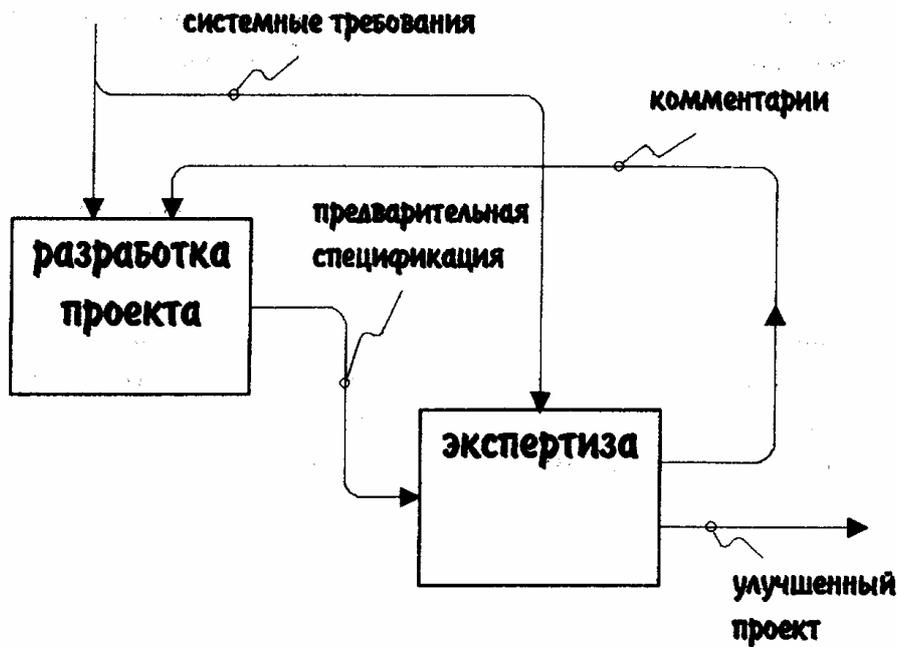


Рис. 3.6. Пример обратной связи.

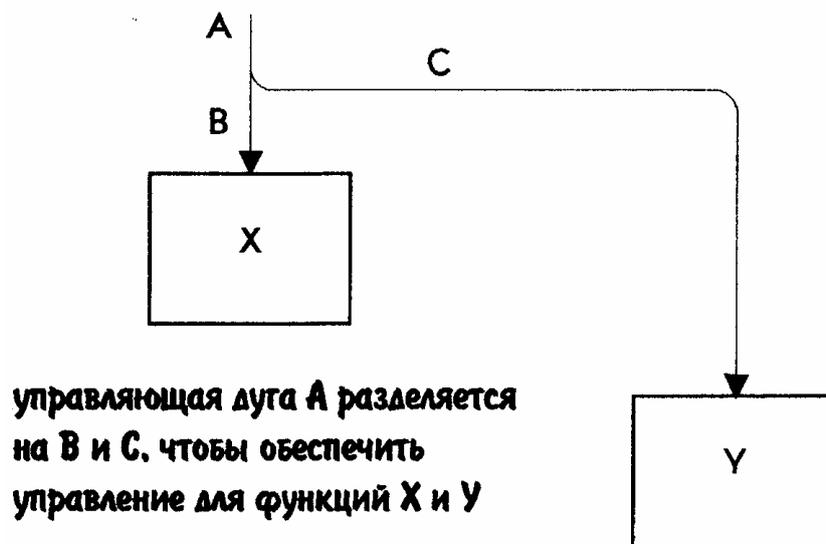


Рис. 3.7. Пример разветвления дуг.

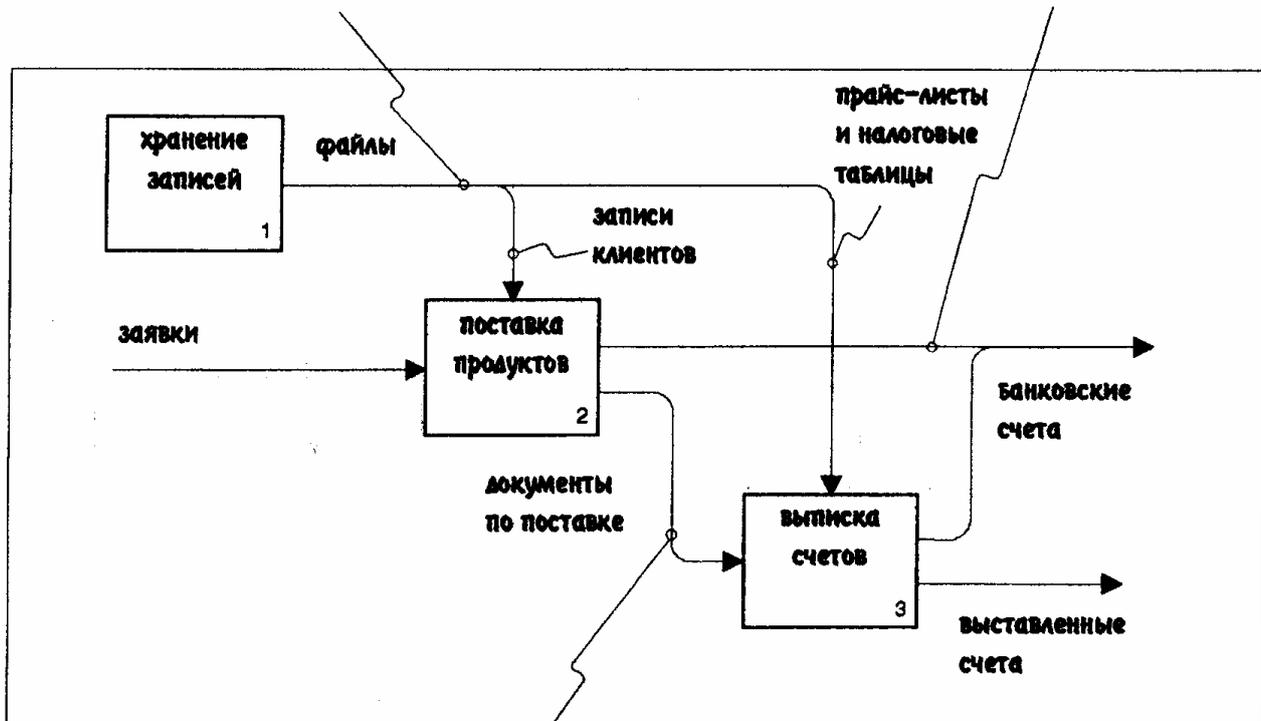
Данные на любой диаграмме могут быть представлены внутренней дугой (оба конца соединены с изображенными на диаграмме блоками) или пограничной дугой (один свободный конец предполагает ее порождение или использование какой-либо функцией за пределами диаграммы).

### 3.1.3.1. Дуги соединяют блоки

Диаграмма состоит из 3-6 блоков, соединенных входными, выходными и управляющими дугами. Каждая выходная дуга может быть единственным или одним из нескольких входов (управлением или механизмом) для какого-либо другого блока. Дуги с одним свободным концом представляют данные, источник которых находится вне диаграммы или которые используются вне ее.

Эта ветвь обозначает, что "файлы" (результат функции 1) являются композицией "записей клиентов" (необходимых для функции 2) и "прайс-листов и налоговых таблиц" (необходимых для функции 3).

Это соединение означает, что "банковские счета" создаются из результатов функций "ПОСТАВКА ПРОДУКТОВ" (функция 2) и/или "ВЫПИСКА СЧЕТОВ" (ФУНКЦИЯ 3).



Эта цепочка входных и выходных дуг означает, что "заявки", под действием поставки (функция 2), записываются как "транзакции", которые под действием функции 3 отражаются в "выставленных счетах".

Рис. 3.8. Соединения дуг.

### 3.1.3.2. Дуги механизмов

Блок представляет функцию. Входные данные (с левой стороны) превращаются в выходные данные (с правой стороны). Управление (сверху) регулирует выполнение функции. Механизмы (с нижней стороны) показывают средства, с помощью которых осуществляется выполнение функции (рис. 3.9). "Механизм" может быть человеком, компьютером, машиной или любым другим устройством, которое помогает выполнять данную функцию. Сам функциональный блок с его входами, управлениями и выходами показывает, что делает система. Механизм определяет, каким образом эта функция выполняется. Диаграммы, изображенные без механизмов, описывают, какие функции должна выполнять система. Механизмы уточняют, как эти функции должны выполняться.

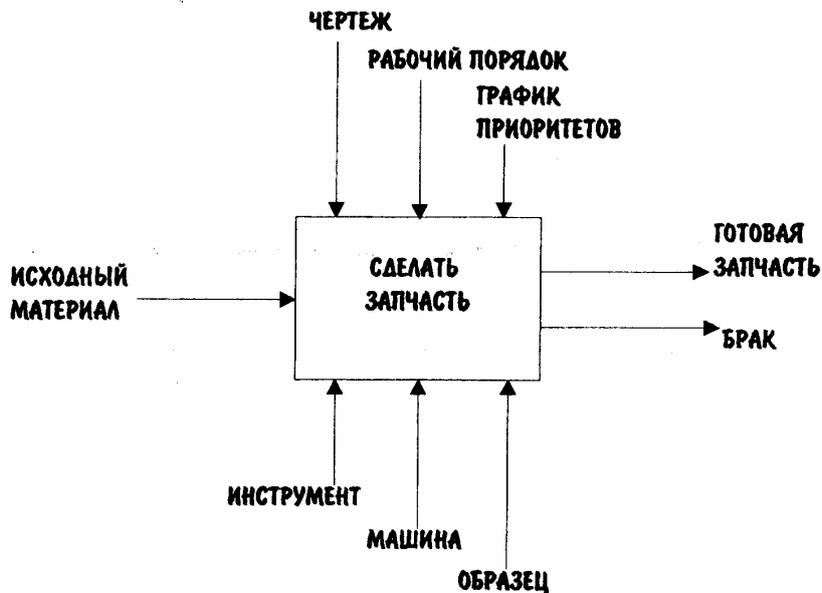


Рис. 3.9. Пример механизма.

Направленная вниз дуга (известная как "вызов") представляет "систему", которая полностью выполняет функцию блока. Если существует необходимость в дальнейшей детализации, она будет определена в отдельной модели самого механизма, (рис. 3.10.).

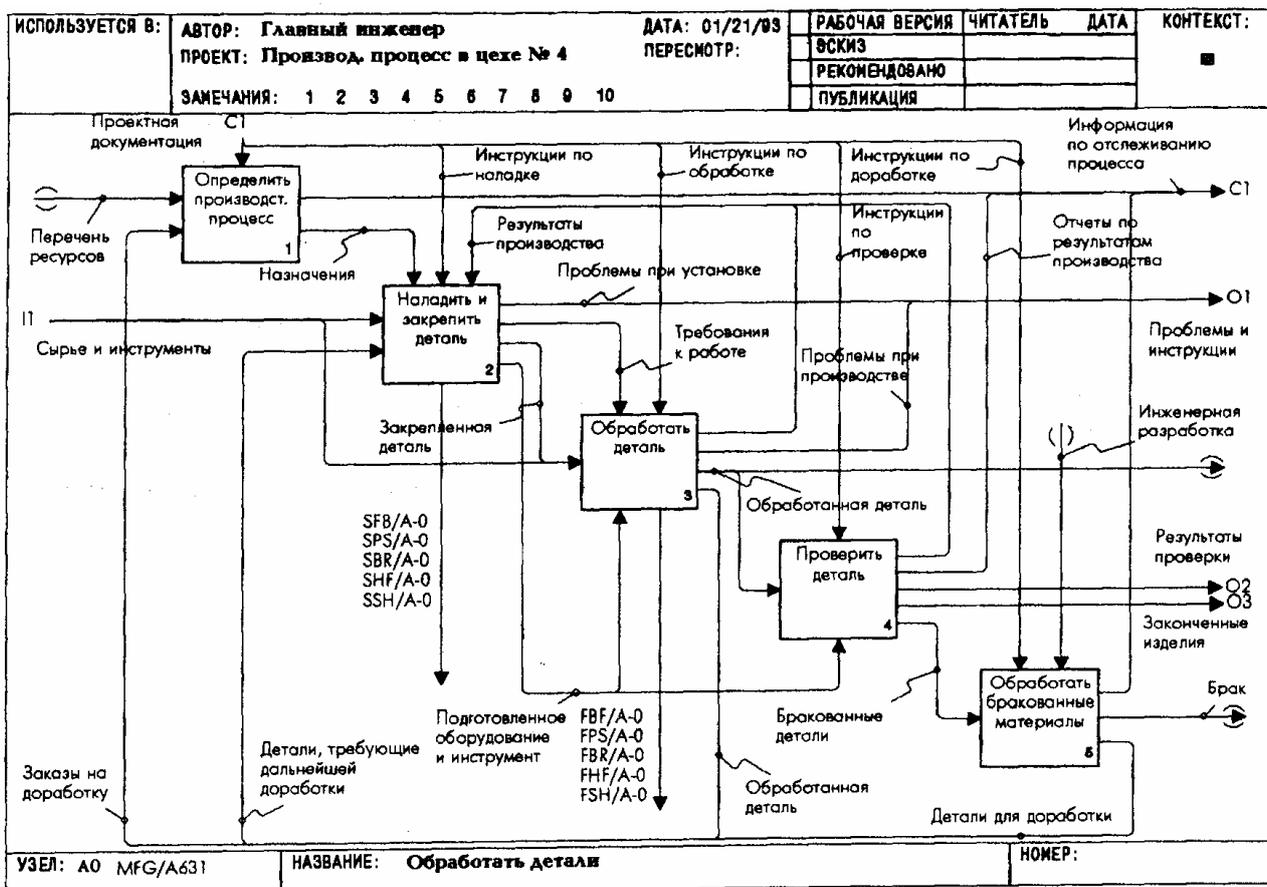


Рис. 3.10. Пример "вызовов" других моделей через дугу механизма.

Дуги механизма могут быть выходами других блоков» если эти функции создают или готовят устройства через свои входные и управляющие дуги (рис. 3.11.).

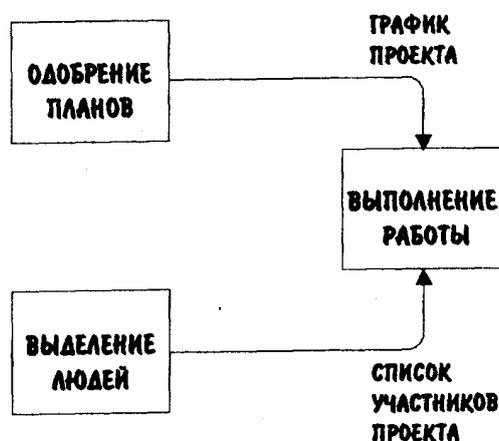


Рис. 3.11. Пример выходной дуги, которая выполняет роль дуги механизма.

#### 3.1.4. Пример IDEF0-диаграммы

На рис. 3.12. изображена полная IDEF0-диаграмма, взятая из модели функции "Произвести изделие (MFGO)".

Три блока из диаграммы A612 (рис. 3.13.) являются декомпозицией блока 2 диаграммы A61 "Управлять последовательностью операций". С учетом требований по изготовлению деталей и на основе существующей загрузки цехов, прогнозируется предполагаемая загрузка цеха (блок 1 в A612). Исходя из (1) ожидаемой загрузки, (2) мощности цеха и (3) конкретного порядка специальных остановок, определяются корректировки к график работ в блоке 2. Затем корректируется сам график (блок 3).

Заметьте, что один и тот же блок может выполнять свои функции при различных условиях. Блок 1 может "сработать", даже если нет "пересмотренного порядка сроков выполнения", выходящего из блока 3. Или он может быть выполнен, когда выработан "пересмотренный порядок сроков выполнения", хотя данные, касающиеся "требований по изготовлению" и "состояния загрузки цехов" не изменились. Эти ситуации известны как различные активации блока. Они могут быть формально определены, но в большинстве случаев будут естественно и интуитивно понятны каждому, кто знаком с нотациями блоков и дуг. Например, блок 2 может "сработать", даже если нет дуги "порядок остановки и сообщения о нехватке материалов". Точно так же блок 2 создает "список проблем" не при каждой своей активации.

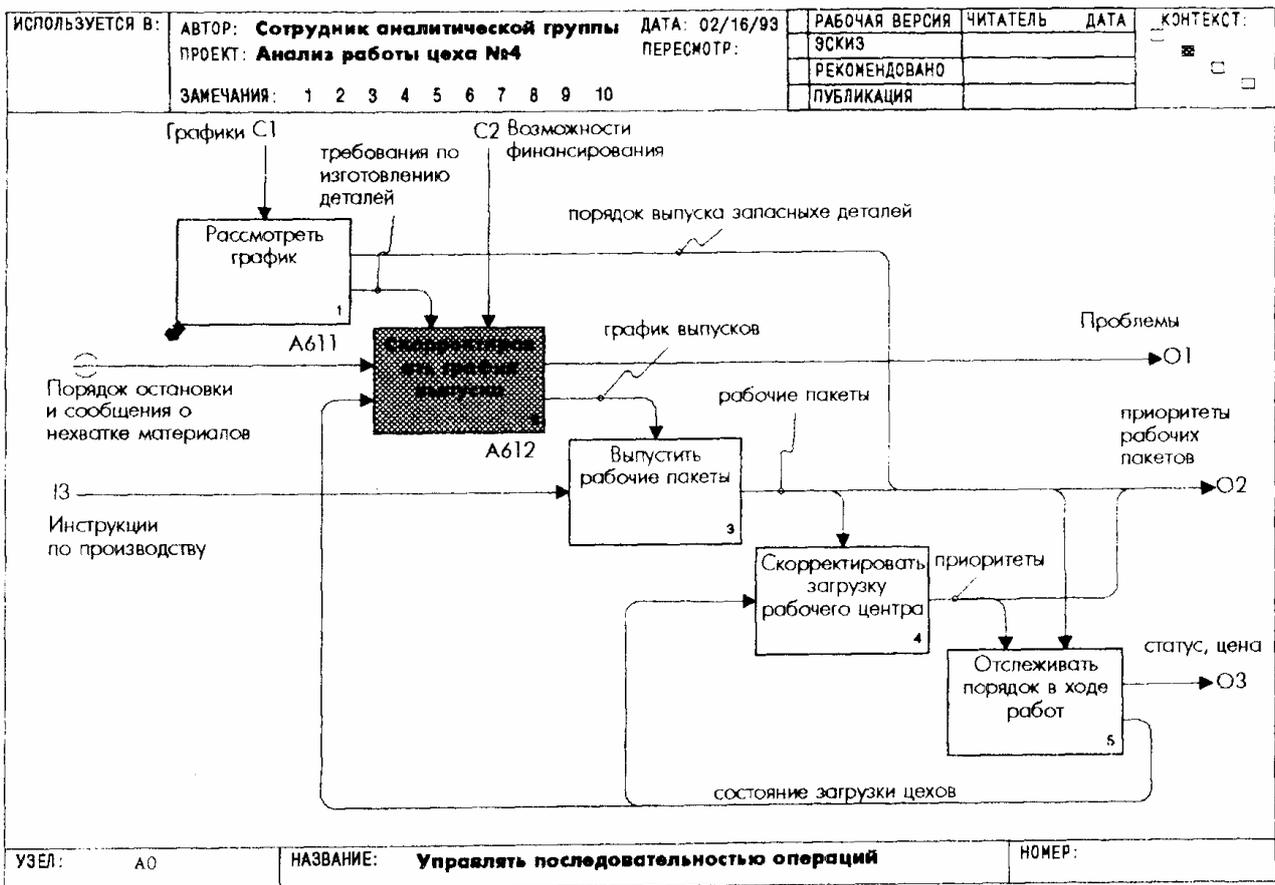


Рис. 3.12. Управление производственным процессом

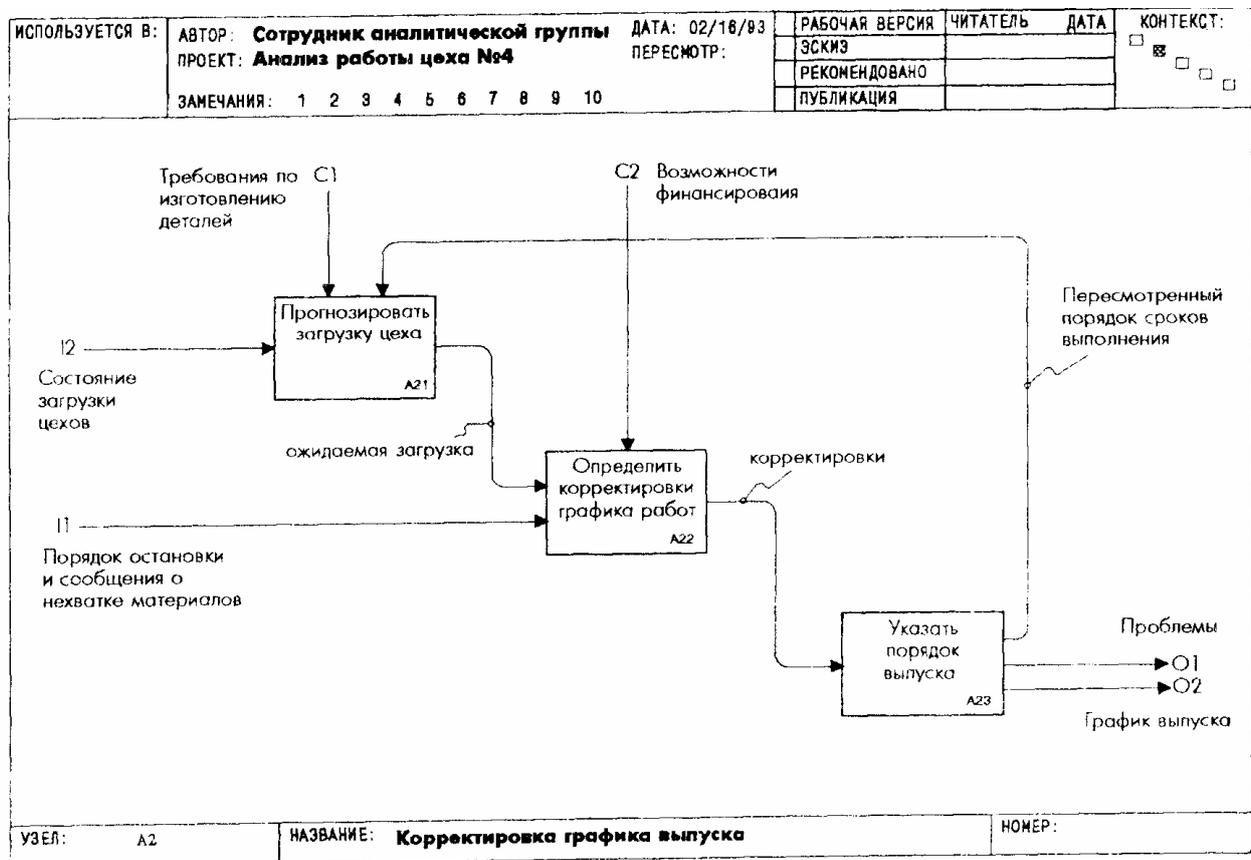


Рис. 3.13. Декомпозиция блока 2, диаграмма A61.

## 3.2. Дополнительные обозначения

Дополнительные обозначения необходимы для структуризации диаграмм, чтобы можно было сформировать взаимосвязанную, непротиворечивую модель.

Для создания модели, состоящей из диаграмм, фрагментов текста, глоссария и FEO-диаграмм, необходимо:

- Показать положение каждой диаграммы в модели и разработать сопроводительную документацию, связанную с каждой диаграммой. Это делается с помощью так называемых ссылочных выражений.
- Определить с помощью ISOM-кодов связь пограничных дуг с дугами исходной диаграммы.
- Предотвратить ненужную детализацию с помощью туннельных дуг.

Эти средства расширяют набор обозначений, позволяющих посредством диаграмм точно отражать функции реальной системы.

### 3.2.1. Ссылочные выражения

#### 3.2.1.1. Номера узлов

Как уже объяснялось в разделе 3.1.1, каждая диаграмма ограничивается тремя-шестью блоками. Каждый блок на диаграмме имеет свой номер. Блок любой диаграммы может быть далее описан диаграммой нижнего уровня, которая, в свою очередь, может быть далее детализирована с помощью необходимого числа диаграмм. Таким образом, формируется иерархия диаграмм.

Для того чтобы указать положение любой диаграммы или блока в иерархии, используются номера узлов. Например, A21 является диаграммой, которая детализирует блок 1 на диаграмме A2. Аналогично, A2 детализирует блок 2 на диаграмме A0, которая является самой верхней диаграммой модели. Эта иерархия может быть показана в таблице имен диаграмм и номеров узлов, образующих дерево узлов. На рис. 3.14. показано типичное дерево узлов.

Номера всех узлов IDEF0-диаграмм начинаются с буквы A, которая идентифицирует их как "диаграммы действия" (Activity), т.е. функциональные диаграммы. В качестве "контекстной" или родительской диаграммы всей модели создается диаграмма, содержащая один блок. Ей принято присваивать узловой номер A-0 (A минус ноль). Если создается полная диаграмма, то для завершения контекста A-0 ей присваивается номер A-1. Этот процесс может, если необходимо, продолжаться вверх. Некоторые сложные объекты начинались с диаграммы A-4, хотя A-0, по-прежнему, являлась вершиной модели.

Иногда используются другие номера узлов. Диаграммой FEO (только для иллюстрации) может считаться любая диаграмма, которая выпадает из строгой иерархии модели. Диаграмма FEO может содержать более шести блоков, неполную структуру дуг, или еще что-либо, необходимое автору для пояснения своей точки зрения. Номера узлов диаграмм FEO содержат букву "F" (например, A2F). Содержание диаграммы поясняется в определениях глоссария. Номера узлов для глоссария включают букву "G" (например, A1G). Номера текстовых узлов содержат букву T и соответствуют номеру узла диаграммы, с которой они связаны (например, A2T). С одной и той же диаграммой может быть связано несколько страниц FEO или глоссария (например, A2F1, A2F2,...), но не более одной страницы текста. (Страницы FEO, глоссария и текста описаны в разделе б).

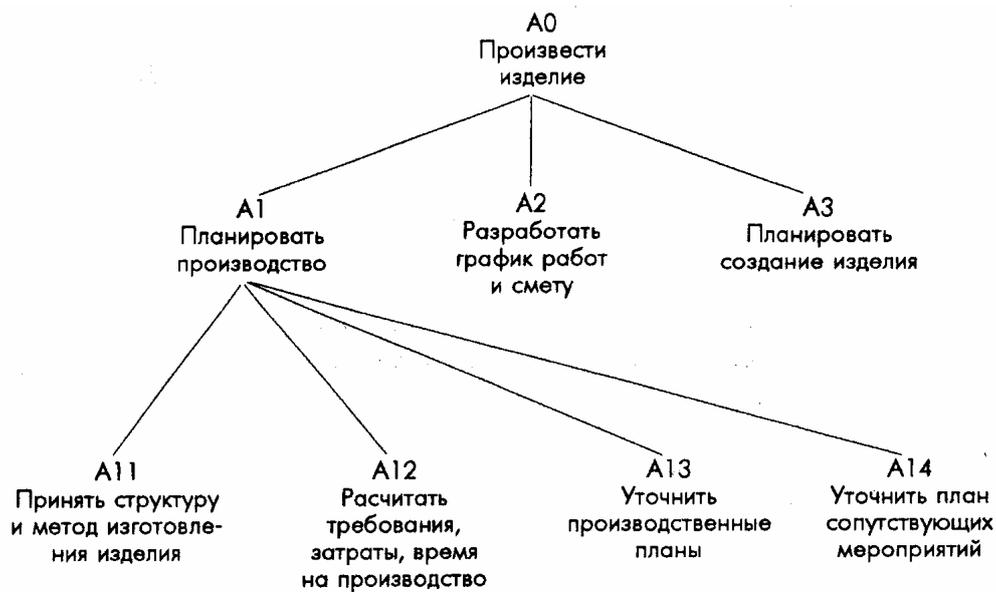


Рис. 3.14. Диаграммы образуют "иерархию", показанную с помощью дерева узлов.

Номер узла используется также для указания на то, что данный блок был декомпозирован. Если блок декомпозирован, то номер узла диаграммы, которая представляет декомпозицию этого блока, записывается под блоком с правой стороны. На рис. 3.15. ссылочные выражения под блоками 1 и 2 указывают на то что эти блоки декомпозированы.

### 3.2.1.2. Имена моделей и номера узлов

Каждая модель имеет имя, которое должно максимально четко отличать ее от других моделей.

Например: ПРОЕКТ

Диаграммы в модели обозначаются добавлением наклонной черты (слэша) и номера узла к имени модели.

Например: ПРОЕКТ/А3

В такой форме обычно пишется полный номер узла каждой диаграммы модели.

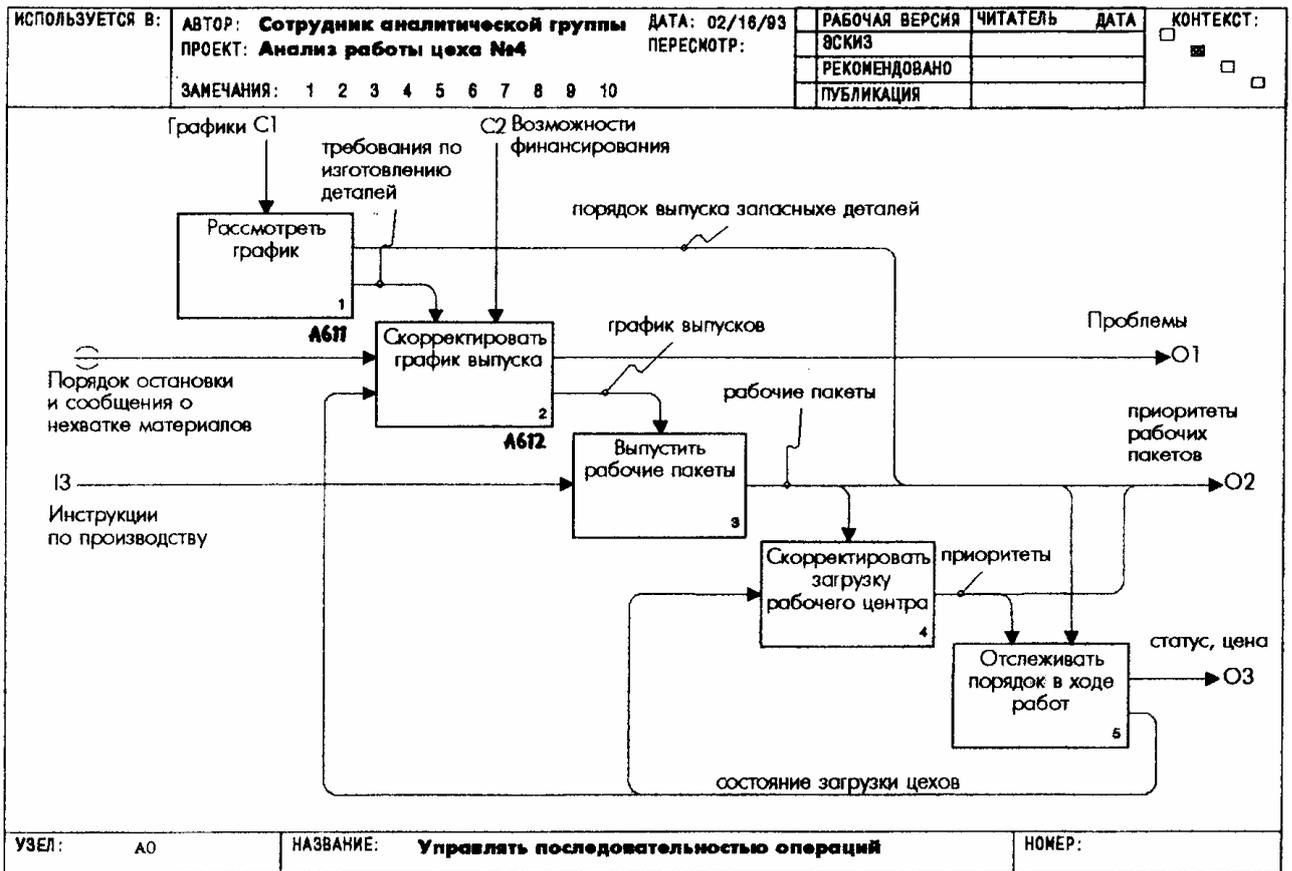


Рис. 3.15. Ссылочные выражения.

### 3.2.2. Дуги, выходящие за пределы диаграммы

Некоторые дуги присоединены к блокам диаграммы обоими концами, у других же один конец остается неприсоединенным (рис. 3.16). Неприсоединенные дуги соответствуют входам, управлениям и выходам родительского блока. Источник или получатель этих пограничных дуг может быть обнаружен только на родительской диаграмме.

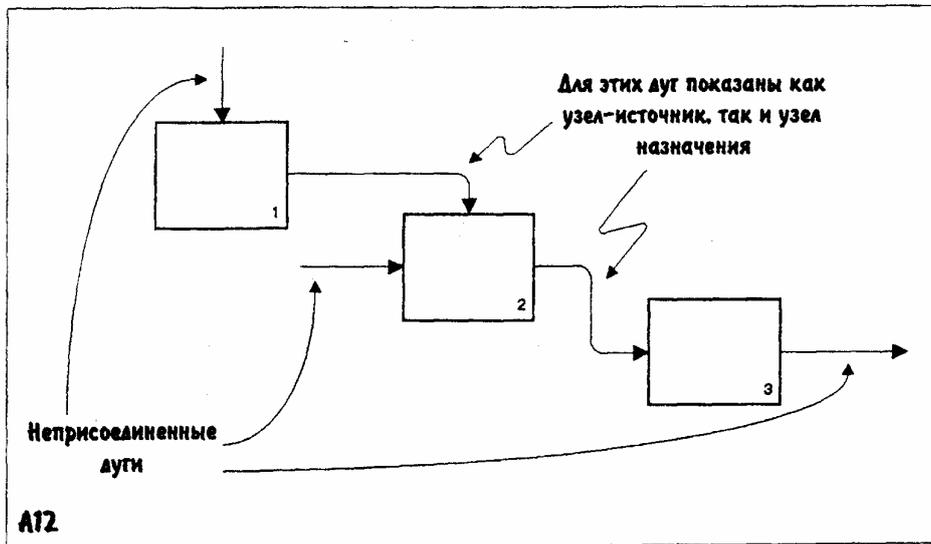


Рис. 3.16. Пограничные стрелки переносятся с родительской диаграммы.

Неприсоединенные концы должны соответствовать дугам на исходной диаграмме. Все граничные дуги должны продолжаться на родительской диаграмме, чтобы модель была полной и непротиворечивой.

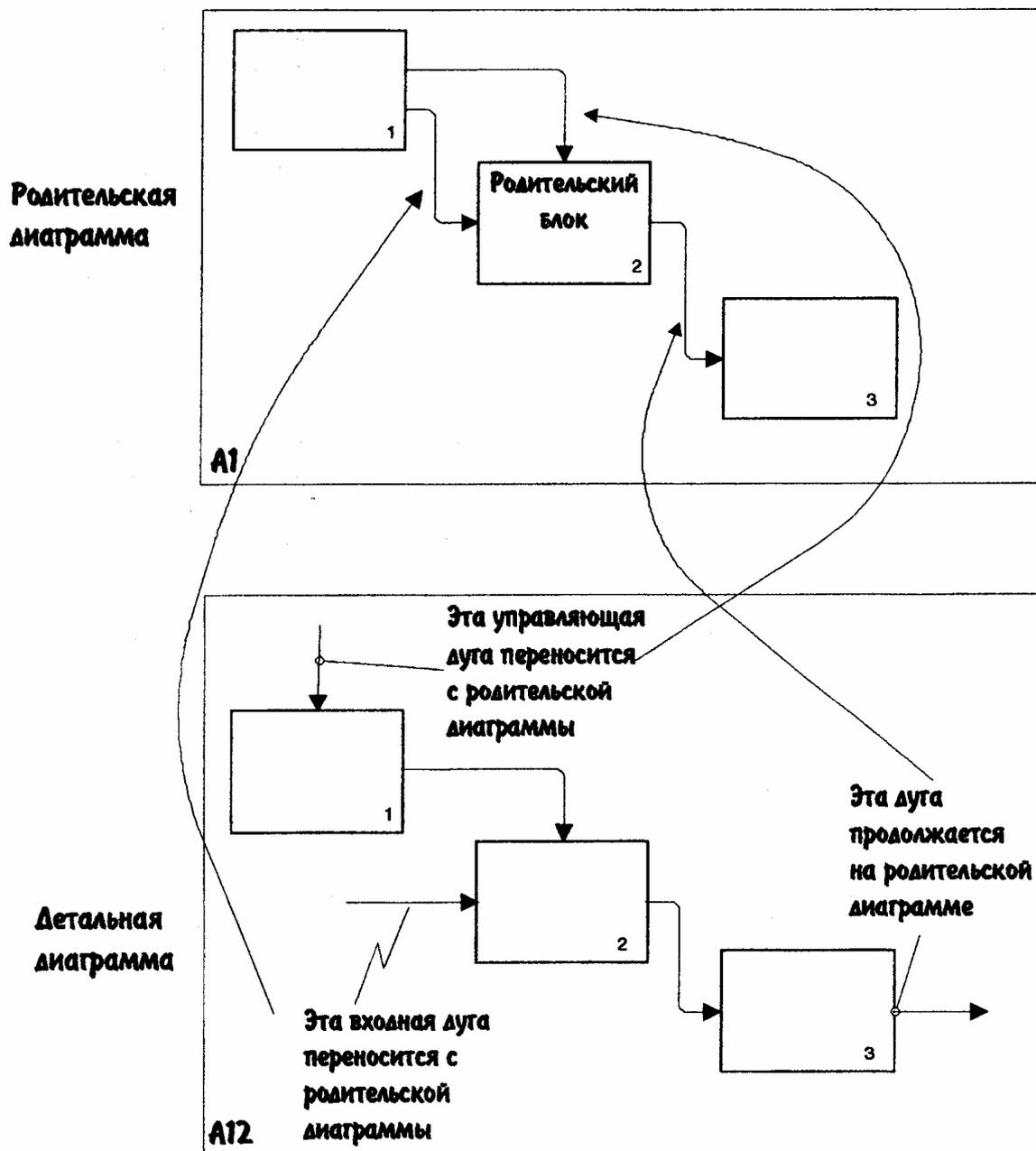


Рис. 3.17. Соответствие должно быть полным и непротиворечивым.

Граничные дуги на самом верхнем уровне А-0 называются внешними дугами, потому что диаграмма А-0 определяет контекст модели, и все дуги связывают ее с тем, что является внешним, по отношению к контексту или сфере действия модели.

### 3.2.3. Кодирование граничных дуг

Соответствующие соединения обозначаются с помощью ICOM-кодов. Буквы I, C, O или M записываются около свободного конца каждой граничной дуги на диаграмме

декомпозиции. Это означает, что дуга изображает, соответственно, вход (Input), управление (Control), выход (Output) или механизм (Mechanism) родительского блока. Эти буквы сопровождаются номером, указывающим позицию, в которой дуга входит в родительский блок или выходит из него. Позиции дуг нумеруются слева направо или сверху вниз. Например, обозначение (код) С3 для дуги на диаграмме-декомпозиции означает, что эта дуга соответствует третьей дуге управления, входящей в родительский блок.

Такое кодирование связывает каждую диаграмму с ее родителем. Когда диаграмма декомпозиции становится родительской диаграммой для дальнейшей декомпозиции, то назначаются новые коды. При использовании такой буквенно-цифровой схемы дуга, являющаяся управляющей или входной на родительской диаграмме, не обязана играть ту же роль на диаграмме декомпозиции.

На рис. 3.18. дуга С2 родительского блока соответствует входу в блок 1 на диаграмме декомпозиции.

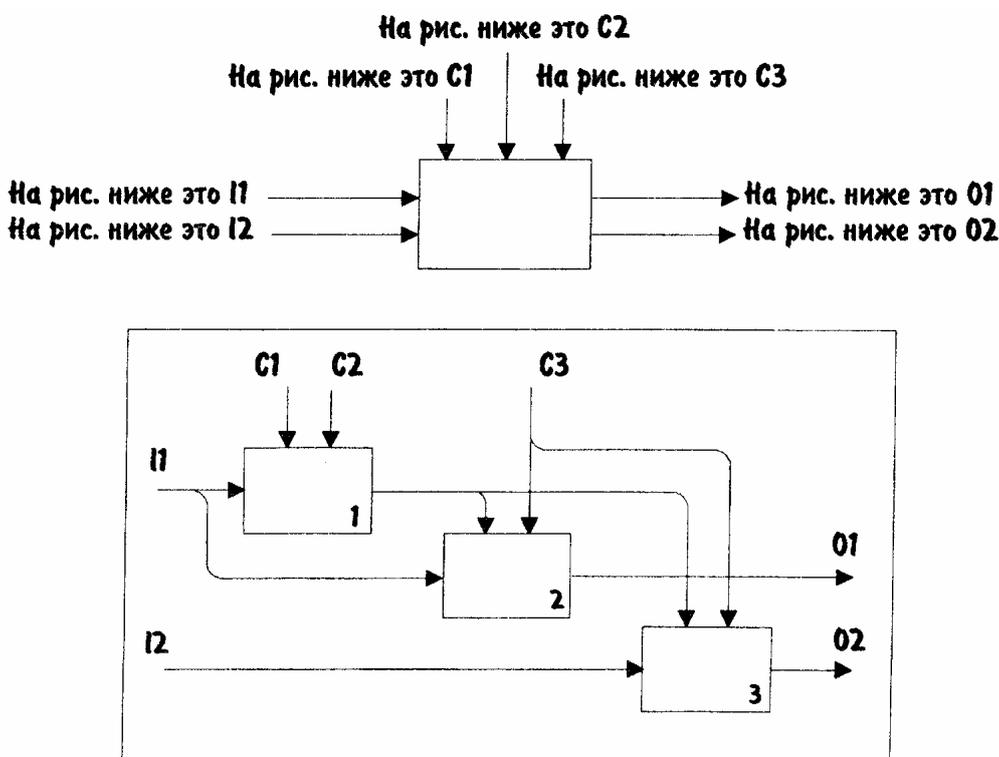


Рис. 3.18. Коды записываются на диаграмме декомпозиции.

ИСОМ-коды должны быть написаны у свободных концов всех граничных дуг, исключая самую верхнюю диаграмму модели, а также дуги, помещаемые в туннель.

### 3.2.5. Туннельные дуги

Туннельные дуги означают, что данные, выраженные этими дугами, не рассматриваются на соответствующем уровне детализации.

Дуга, помещаемая в туннель там, где она присоединяется к блоку (рис. 3.19.), означает, что данные, выраженные ею, не обязательны на следующем уровне декомпозиции.

Дуга, помещаемая в туннель на свободном конце (рис. 3.20.) означает, что данные, выраженные ею, не относятся к исходной диаграмме или на ней не описываются.

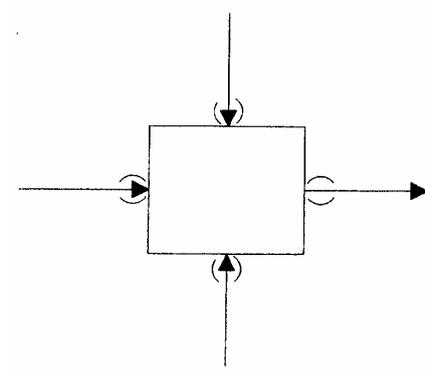


Рис. 3.19. Дуги, имеющие знак туннеля на концах присоединения к блоку.

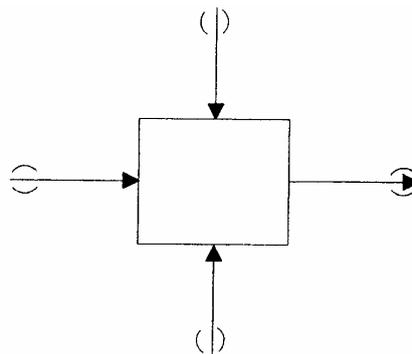


Рис. 3.20. Дуги, имеющие знак туннеля на неприсоединенных концах.

Две круглые скобки на свободном конце означают, что эта дуга отсутствует на родительской диаграмме и она не имеет ISOM-кода. Две круглые скобки на конце, где дуга присоединяется к блоку, означают, что эта дуга не появляется на диаграмме декомпозиции, а ее ISOM-код в дальнейшем не отслеживается и, может быть, никогда не будет использоваться в дальнейшем. Дуга, имеющая взятую в скобки вершину (стрелку), может исчезнуть на одном или нескольких уровнях детализации, а затем появиться на некотором уровне детализации со взятым в скобки началом (основанием). Если первоначальный источник или адресат этой дуги известны, это должно быть отмечено соответствующим ссылочным выражением, помещаемым рядом со скобками.

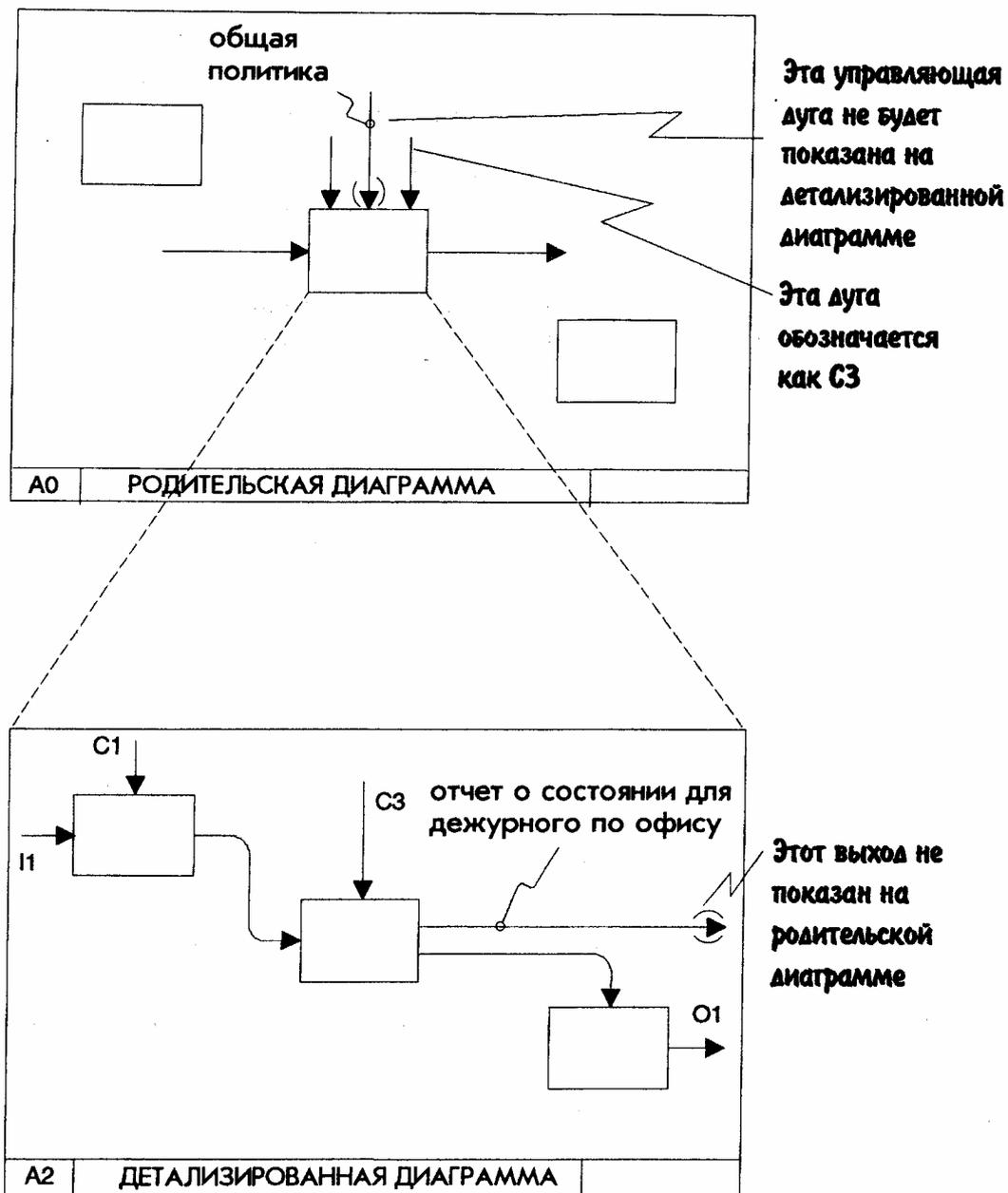


Рис. 3.21. Пример туннельных дуг.

### 3.2.6. Пример декомпозиции

Используя функциональную модель "Произвести изделие" (MFGO), можно рассмотреть самую общую функцию "Произвести изделие", состоящую из следующих подфункций:

- Планировать производство.
- Разработать и вести план-график и сметы.
- Планировать изготовление.
- Обеспечить производственные ресурсы.
- Получить производственные материалы.
- Изготовить изделие.

Любая из этих подфункций может быть в дальнейшем детализирована. Функция "Планировать производство" может содержать подфункции:

- Выбрать структуру и метод производства.
- Оценить требования, стоимость и время производства.
- Разработать планы производства.
- Разработать план отдельной операции.

Каждая из этих подфункций может быть снова декомпозирована с учетом полезности, знаний или имеющегося времени. Эта структура функций и подфункций может быть изображена как дерево узлов (рис. 3.22.).

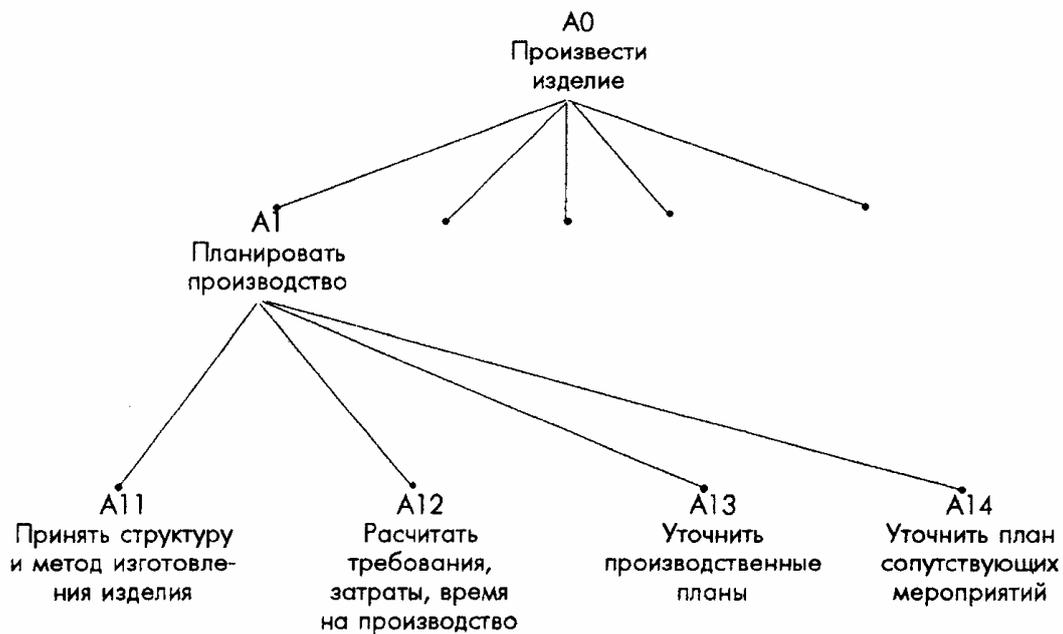


Рис. 3.22. Дерево узлов.

Структура функций и подфункций может быть также изображена как "индекс узлов" (Рис. 3.23.). Формат индекса подобен формату содержания книги или формату списка материалов, используемых в промышленности.

- A0 Произвести изделие
  - A1 Планировать производство
    - A11 Выбрать структуру и метод производства
    - A12 Оценить требования, стоимость и время производства
    - A13 Разработать планы производства
    - A14 Разработать план отдельной операции
  - A2 ...

Рис. 3.23. Индекс узлов.

Приведенные ниже IDEF0-диаграммы начинаются с того же объекта, который мы видим на вершине дерева узлов. На рис. 3.24., диаграмма А-0 представляет информацию и объекты, с которыми оперирует функция "Произвести изделие".

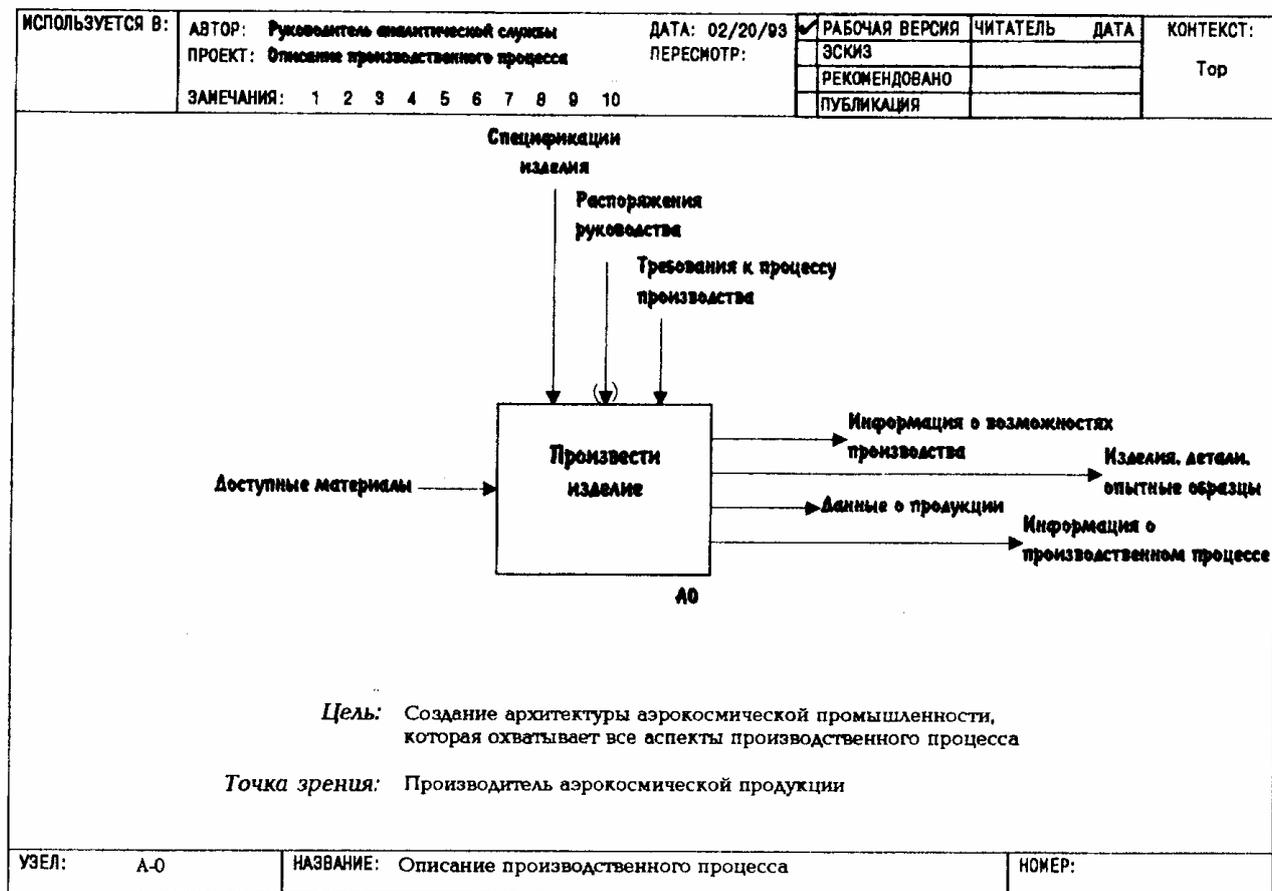


Рис. 3.24. Диаграмма А-0 "Произвести изделие".

Функция "Произвести изделие" включает все, начиная с входов и управлений:

- Доступные материалы
- Спецификации изделия
- Требования к процессу производства

и заканчивая выходами:

- Информация о возможностях производства
- Изделия, детали, опытные образцы
- Данные о продукции
- Производственная информация

То, что происходит в блоке А-0 "Произвести изделие", показано на диаграмме А0, имеющей такое же название.

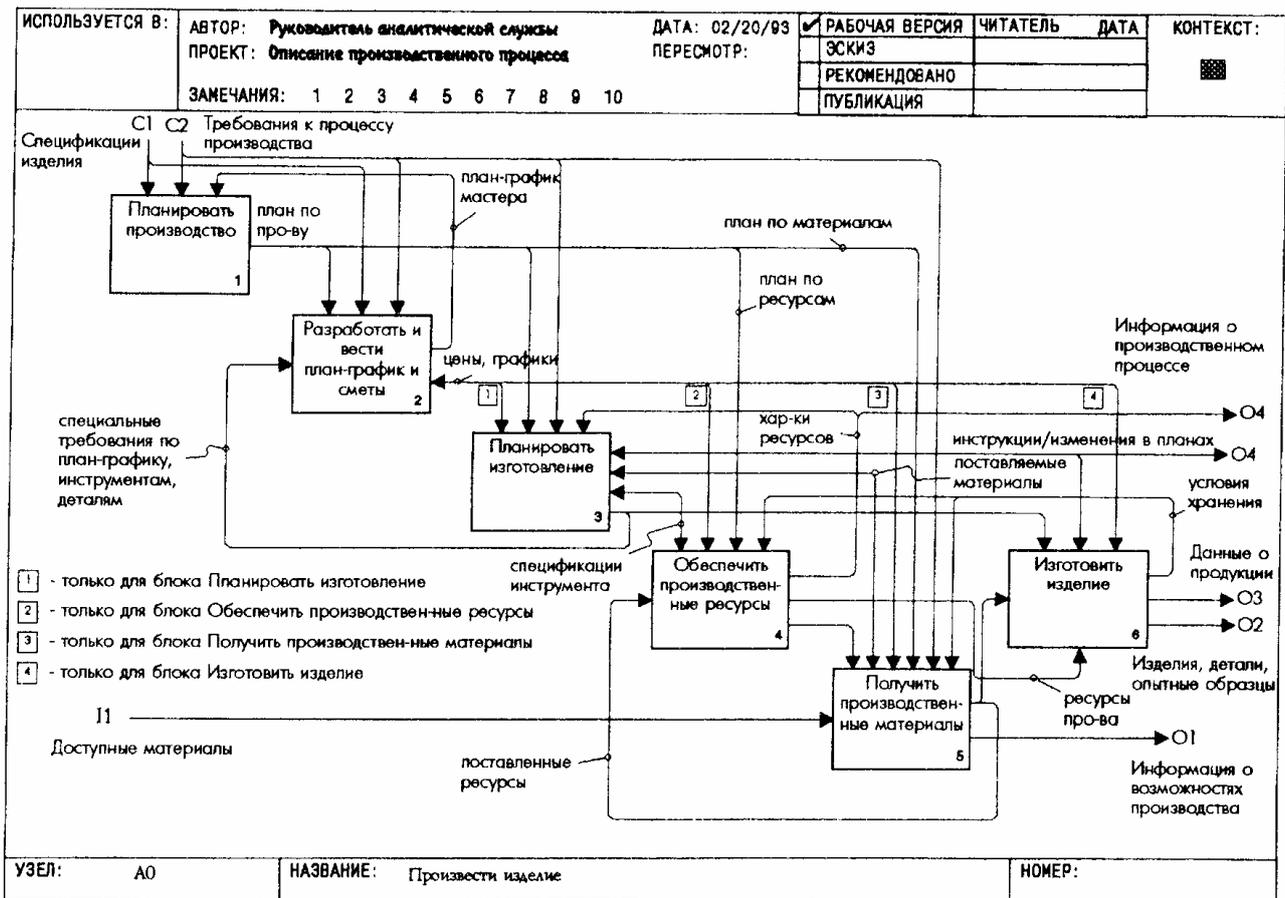


Рис. 3.25. А-0, Произвести изделие.

Каждая функция, составляющая часть функции "Произвести изделие", является некоторым объединением последовательности решений и действий, составляющих сложный процесс производства. Объединение функций осуществляется на основе взаимоотношений данных, т. е. той информации и объектов, которые передаются между функциями. Эти взаимоотношения определяют границы функции и терминологию, используемую для обозначения каждой из них.

Каждый блок диаграммы А0 может быть далее описан с помощью другой диаграммы. Первый блок диаграммы А0 детализируется в диаграмме А1 "Планировать производство". А1 показывает только те функции и данные, которые касаются функции "Планировать производство", как определено дугами на А0.

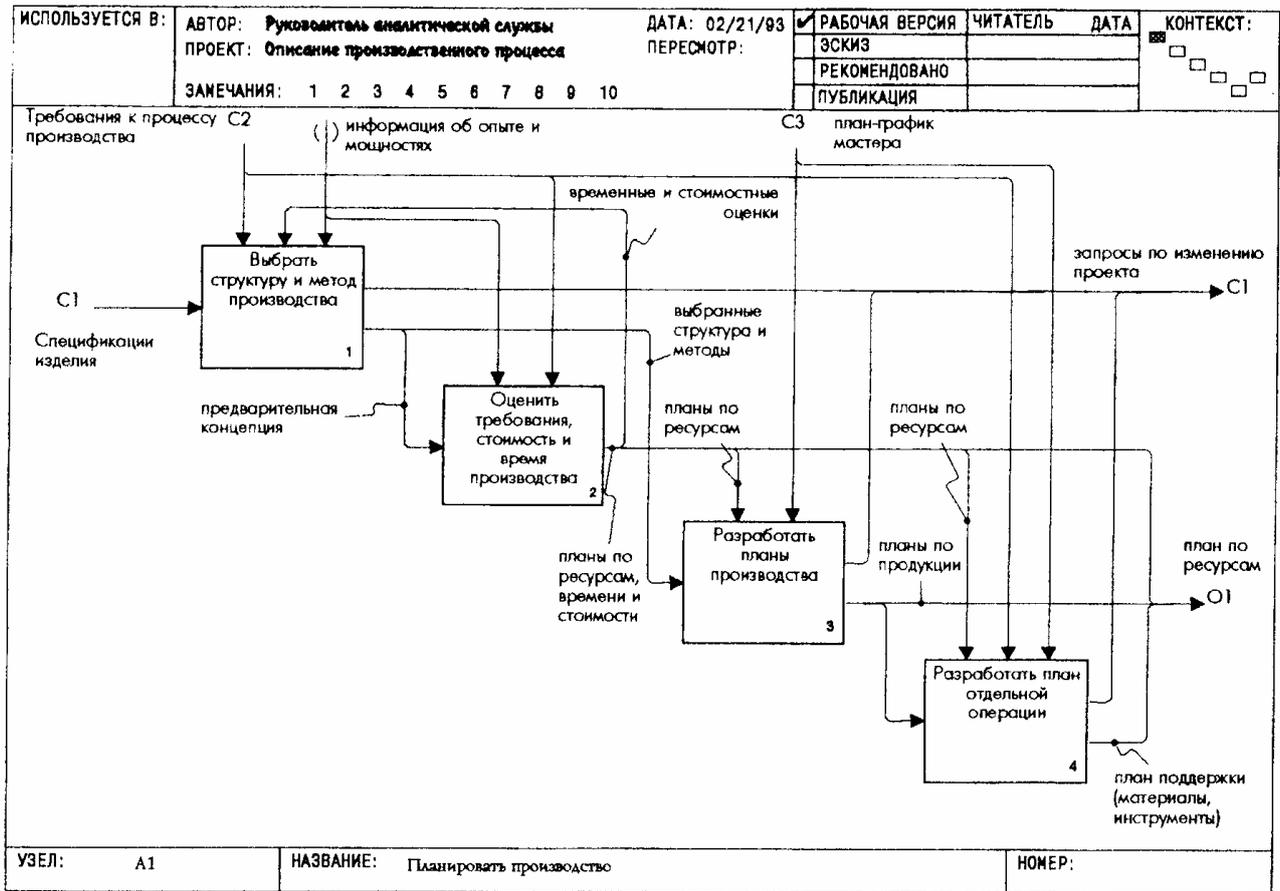


Рис. 3.26. Блок А1, Планировать производство.

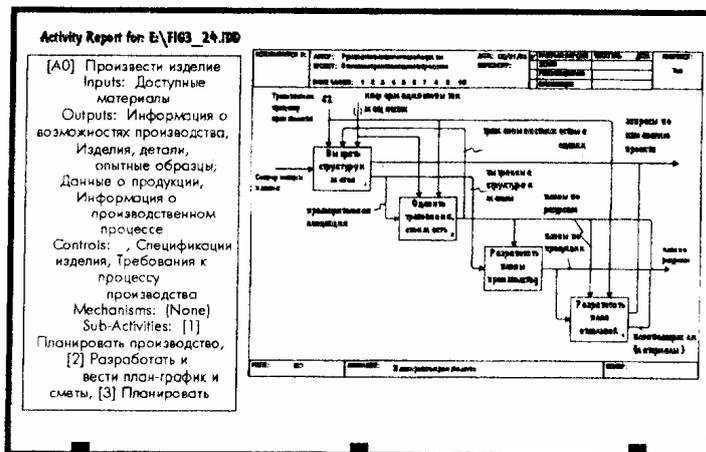
#### 4. Чтение IDEF0-диаграмм

Модель состоит из набора диаграмм, организованных иерархически, и связанных с ними материалов. Представляется также индекс узлов или оглавление. Размещение диаграмм в иерархическом порядке дает общее представление о модели и облегчает доступ к любой ее части.

Чтение диаграммы осуществляется сверху вниз. После прочтения верхних уровней прочитывается диаграмма первого уровня, затем второго и т. д. Если необходимы конкретные подробности модели, то используется указатель узлов для движения вниз по уровням к требуемой детальной диаграмме.

При публикации модель помещается в формат "пара страниц" в порядке, соответствующем индексу узлов. Формат "пара страниц" означает, что каждая диаграмма и весь текст, связанный с ней, размещаются на развороте двух страниц (рис. 4-1).

Текст,  
поясняющий  
текущую  
диаграмму



Диаграмма,  
являющаяся  
родительской  
по отношению  
к текущей  
диаграмме.



Изображена в  
уменьшенном  
размере

Это  
текущая  
диаграмма

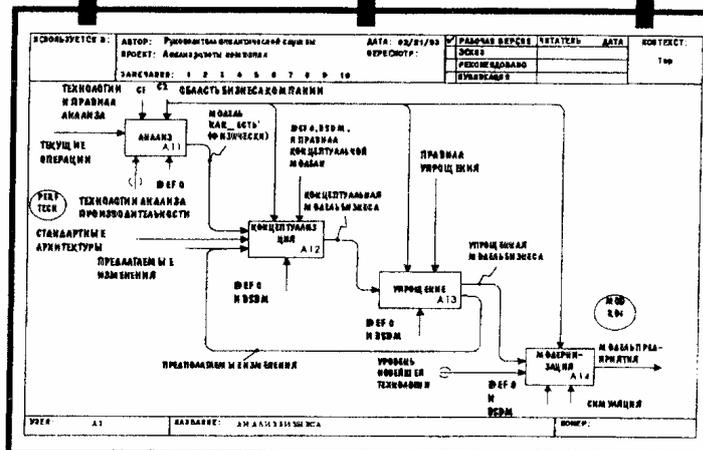


Рис. 4.1. Формат "пара страниц".

Порядок, соответствующий индексу узлов, означает, что все диаграммы декомпозиции, относящиеся к одному блоку на диаграмме, представляются прежде, чем декомпозиция следующего блока. При таком порядке диаграммы располагаются так же, как и в обычном оглавлении.

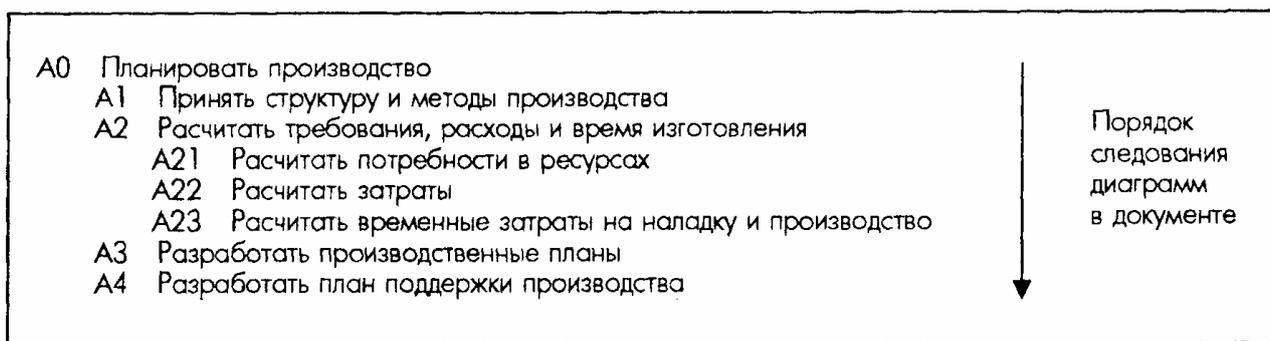


Рис. 4.2. Индекс узлов, иллюстрирующий последовательность диаграмм.

#### 4.1. Общий обзор модели

Модель обеспечивает обзор как всей системы, так и отдельных деталей. Для рассмотрения модели в самом общем виде отберите с помощью индекса все диаграммы верхнего уровня.

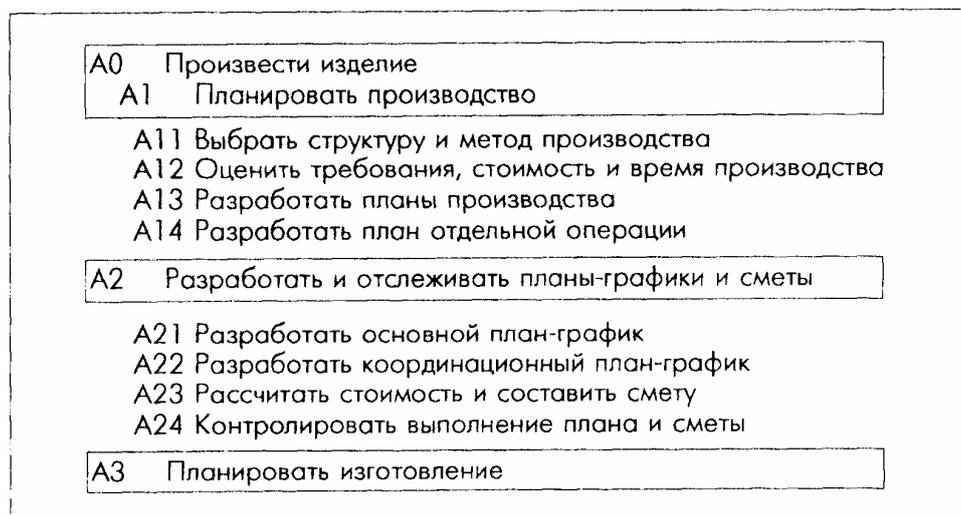


Рис. 4.3. Индексы узлов, указывающие на более общие диаграммы.

Чтобы прочесть модель детально, используйте индекс для нахождения всех диаграмм, детализирующих интересующий вас объект (рис. 4.4.).

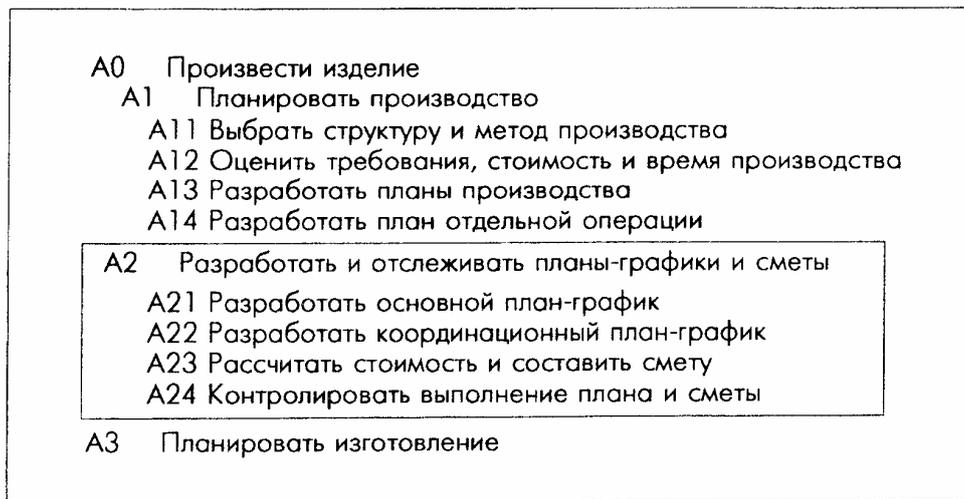
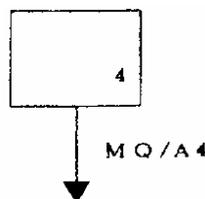


Рис. 4.4. Индексы узлов, показывающие на конкретные детализирующие диаграммы.

Дальнейшая детализация модели может быть прослежена с помощью ссылочного выражения, расположенного сразу под номером блока. Оно означает либо номер узла, либо номер страницы с диаграммой декомпозиции для этого блока. Если ссылочное выражение отсутствует, то блок еще не был детализирован. Например, на диаграмме A24 изображение блока, означает, что детали блока 3 находятся на диаграмме с номером узла A243.

Одна детализирующая диаграмма может использоваться в нескольких местах модели, или находиться в другой модели. В последнем случае нижняя стрелка, направленная вниз, показывает, где находится детализирующая диаграмма (См. рис. 3.10.) :



Блок 4 детализирован в модели MQ диаграммой A4. (Такая ситуация известна как "вызов").

#### 4.2. Основные этапы в процессе чтения диаграммы

Точная информация о системе находится в самих диаграммах. Рекомендуется следующая последовательность чтения:

1. Окиньте взглядом блоки диаграммы, чтобы получить представление о ее содержании.
2. Вернитесь к родительской диаграмме и рассмотрите дуги, присоединенные к диаграмме. Попробуйте определить наиболее важные вход, управление и выход.
3. Рассмотрите дуги текущей диаграммы. Попробуйте определить, имеется ли главный путь, соединяющий "наиболее важный" вход или управление и наиболее важный выход.
4. Мысленно двигайтесь по диаграмме от левого верхнего угла в правый нижний, используя главный путь в качестве направляющего. Обратите внимание, как другие дуги взаимодействуют с каждым блоком. Определите, имеются ли побочные пути. Вы можете проверить версию, полученную из анализа диаграммы, рассмотрев, как трактуются другие похожие ситуации.
5. Проверьте, существует ли "FEO" диаграмма, связанная с моделью.

6. В заключение прочитайте текст и глоссарий, если они имеются.

Такая последовательность гарантирует, что основным особенностям каждой диаграммы будет уделено должное внимание. Текст будет направлять внимание на все то, что автор решил выделить. Глоссарий будет определять авторскую интерпретацию используемой терминологии.

Каждая диаграмма имеет центральную идею, проходящую от самой важной входящей граничной дуги до самой важной выходящей граничной дуги. Этот главный путь через блоки и дуги определяет основную функцию диаграммы. Другие части диаграммы дают уточняющие или альтернативные условия, которые являются побочными для главного пути.

Функционирование системы Вы можете мысленно представить, следуя главному пути. Особые виды входных данных, трактовка ошибок и возможные альтернативные выходы добавляют специфические детали к материалу. Такой просмотр улучшает понимание диаграммы.

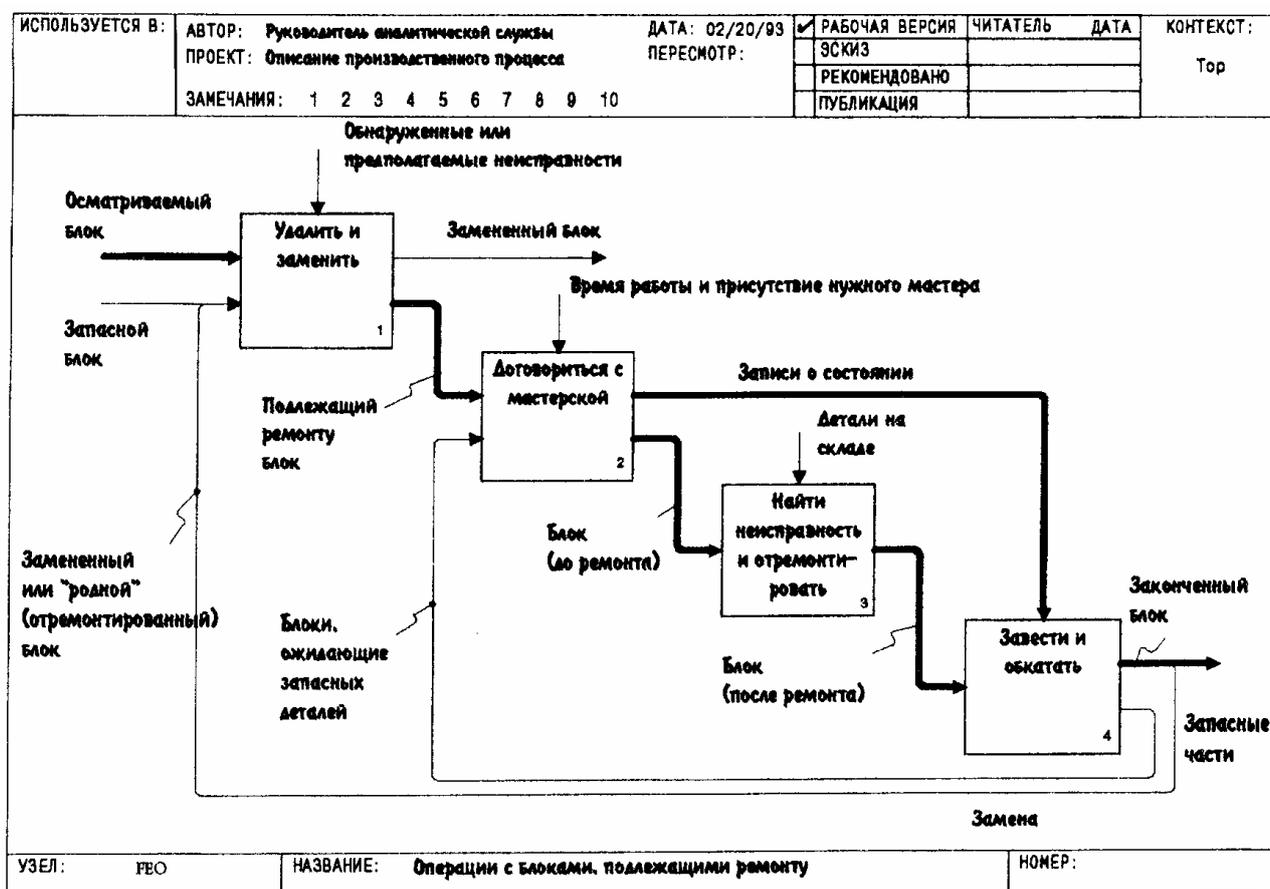


Рис. 4.5. Пример основного пути.

### 4.3. Семантика блоков и дуг

Основным критерием, которым необходимо руководствоваться при интерпретации любой диаграммы или набора диаграмм, является следующее:

**Только то, что изложено ясно и точно, должно быть выполнено обязательно.**

Это следует из самой природы ограничений диаграмм. Не нужно предполагать существование неуказанных ограничений. Обязательные ограничения должны быть четко сформулированы.

Отсюда:

**Любая дальнейшая декомпозиция, явно не запрещенная, неявно разрешена.**

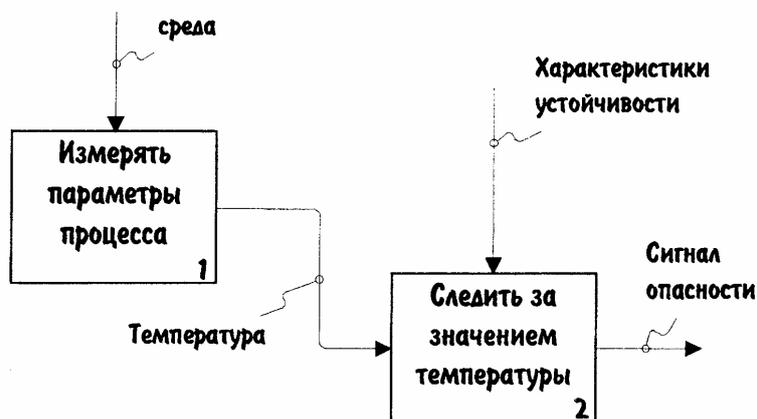


Рис. 4.6. Пример ограничения.

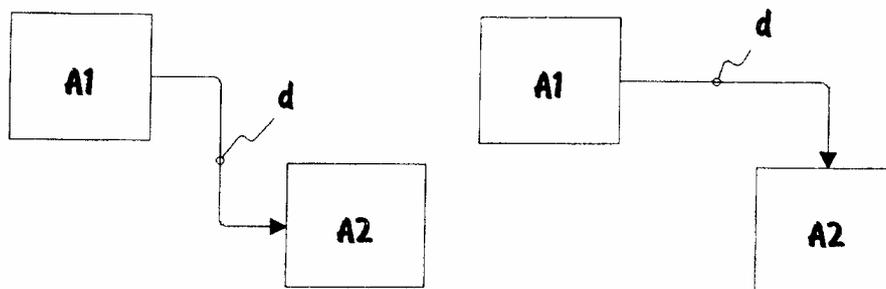
При рассмотрении рис. 4.6. можно сделать предположение, что температура измеряется "достаточно часто", допустимое отклонение изменяется "когда необходимо", а температура регулируется по отношению к допустимой погрешности "достаточно часто" таким образом, что сигнал об опасности будет выдаваться "достаточно быстро". Никакое из этих интуитивных предположений не противоречит последующей детализации, показавшей, что:

- a) температура измеряется путем периодической выборки, или
- b) текущие допустимые отклонения запрашиваются только тогда, когда температура увеличивается на некоторое фиксированное значение, или
- c) ряд температурных величин, произведенных блоком накапливается в блоке 2, который проверяет значение изменения для определения, находится ли оно в допустимых пределах,
- d) и т.д.; и т.д.

Графические обозначения на диаграмме сами по себе являются абстрактными. Однако они должны выявлять важные фундаментальные особенности. Их абстрактная сущность не должна умиляться из-за широкого спектра возможных интерпретаций, которые она допускает.

#### 4.3.1. Ограничения не показывают "как" и "когда"

Любое из двух представлений:



говорит, что:

**действие A2 зависит от d, которое создается или модифицируется действием A1.**

Каждое из этих представлений определяет взаимосвязь между двумя блоками. То, что явно выражено связующей дугой, в каждом случае можно сформулировать следующим образом: некая активация блока 2 требует нечто такого, именуемого d, что производится при некоторой активации блока 1.

Часто в диаграмме строго определено, что два или более блоков должны "конкурировать" в получении информации, представляемой некоторой дугой. Значение блоков и дуг, показанных на рис. 4.7., состоит в том, что нечто, произведенное блоком 1, является необходимым для блоков 2 и 3.

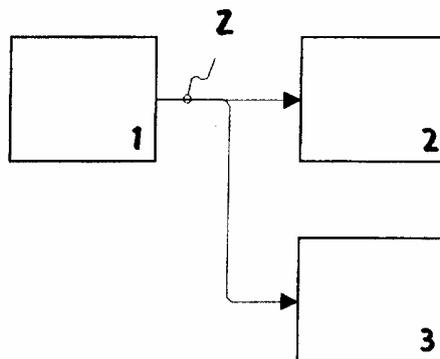


Рис. 4.7. Блоки, конкурирующие за содержание дуги.

Это может значить, что активация "источника" дуги (блока 1) должна предшествовать каждой активации его "адресата" (блока 2 или блока 3). А может быть, что одной активации источника достаточно для всей активации любого адресата. Без дополнительной информации блоки и дуги сами по себе допускают обе интерпретации.

#### 4.3.2. Многократные входы, управления и выходы

Основная интерпретация блока, показанного на рис. 4.8., заключается в следующем: для того чтобы получить любое подмножество выходов [O1, O2, O3], может потребоваться любое подмножество входов [I1, I2, I3, C1, C2, C3, C4]. При отсутствии дальнейшей декомпозиции блока нельзя считать, что:

- а) любой из выходов может быть произведен без присутствия всех входов, или
- б) какой-либо из выходов требует все входы для своего создания.

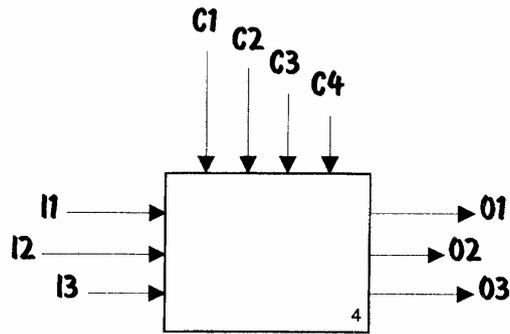


Рис. 4.8. Иллюстрация множественных ICOM-меток.

Частичная детализация предыдущего блока (которая может быть изображена на диаграмме FEO) показывает, что управления C2, C3, C4 не требуются для создания выхода O1. Это означает, что:

- a) Дальнейшая декомпозиция может определить точное взаимоотношение входов и управлений с выходами.
- b) До тех пор пока такая декомпозиция не сделана, не следует принимать каких-либо предположений о взаимоотношениях "внутри" блока.
- c) При чтении диаграммы следует уделять больше внимания дугам, которые точно определены, чем на содержание блоков, которое только подразумевается.

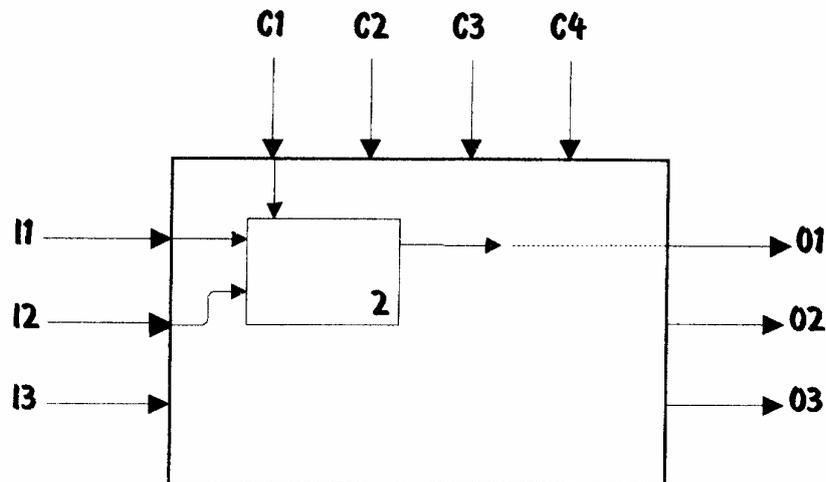


Рис. 4.9. Представление FEO, детализирующее множественные ICOM—метки.

## Часть 5. Бланки и процедуры IDEF

### 5.1. Принцип коллективной работы в IDEF-методологии

Разработка любой IDEF-модели ( IDEF0, IDEF1, IDEF2) является сложным динамическим процессом, который требует участия более чем одного человека. Во время разработки проекта авторами создаются эскизы отдельных частей модели, которые рассылаются другим участникам проекта для рассмотрения. Эти эскизы модели называются папками и могут содержать диаграммы, фрагменты текста, глоссарий или любую другую информацию, необходимую, по мнению автора, для разработки модели.

При коллективной разработке IDEF-модели все, кто участвует в ее обсуждении, называются обозревателями. Обозреватели, представляющие письменные замечания по содержанию папки, называются рецензентами. Обозреватели, которые получают папки только для ознакомления, называются читателями.

Методология требует, чтобы каждый, кто делает замечания по папке, представлял бы их автору в письменном виде. Автор отвечает каждому рецензенту письменно на том же экземпляре документов. Этот процесс продолжается до тех пор, пока модель не завершена и не рекомендована к опубликованию.

Каждые три месяца папки с новыми уточнениями, внесенными в модель, рассылаются читателям, чтобы держать их в курсе текущей информации о развитии модели.

Такая организация совместной работы обеспечивает гарантию, что окончательная версия IDEF-модели будет верной. В папки вносятся изменения с учетом корректив и принятых замечаний. Более подробная детализация достигается построением большего числа диаграмм, текста и глоссария. Делаются новые замечания, вносятся новые изменения. Окончательная модель соответствует представлениям автора и обозревателей о системе, смоделированной с данной точки зрения и для данной цели.

### 5.2. Цикл IDEF-папки

Материалы, записанные или собранные автором, рассылаются рецензентам в форме стандартной папки. Рецензенты изучают материал и делают письменные замечания. Затем они возвращают папки автору, который отвечает на их замечания или использует их для пересмотра материала. Папка с ответами автора снова передается рецензентам. Этот процесс известен как Цикл папки. Он включает следующие этапы:

- Автор собирает материалы, предназначенные для рассмотрения, в стандартную папку (типы IDEF-папок рассмотрены в разделе 5.3). Заполняется титульный лист. Копии папки рассылаются каждому рецензенту и автору. Оригинал регистрируется и хранится для ссылок.
- В течение времени, отведенного для ответа, рецензент читает папку и вносит замечания в копии диаграмм. Папка возвращается автору.
- Автор отвечает письменно на копиях каждому рецензенту. Он может принять замечания, отметив это в своем рабочем экземпляре, и учесть их в следующей версии модели. Если у автора есть возражения, он делает соответствующую запись в папке и возвращает ее рецензенту.
- Рецензент знакомится с ответами автора и, если они его удовлетворяют, оставляет папку в своем архиве. (Папки с замечаниями всегда возвращаются рецензенту). Если рецензент не согласен с ответами автора, назначается встреча с ним для обсуждения разногласий. Если это невозможно, составляется список спорных вопросов, который передается авторитетному лицу для разрешения.

Этот цикл продолжается до тех пор, пока не будет получен документ, отражающий мнение всех участников проекта. К тому же фиксируется документально вся история процесса.

Конечной целью цикла папки является документ, в который автор и рецензенты внесли свой вклад, и, если в этом есть необходимость, список вопросов, которые требуют решения руководства.

В течение всего цикла библиотека проекта осуществляет размножение, рассылку, регистрацию и обмен ими между авторами и рецензентами.

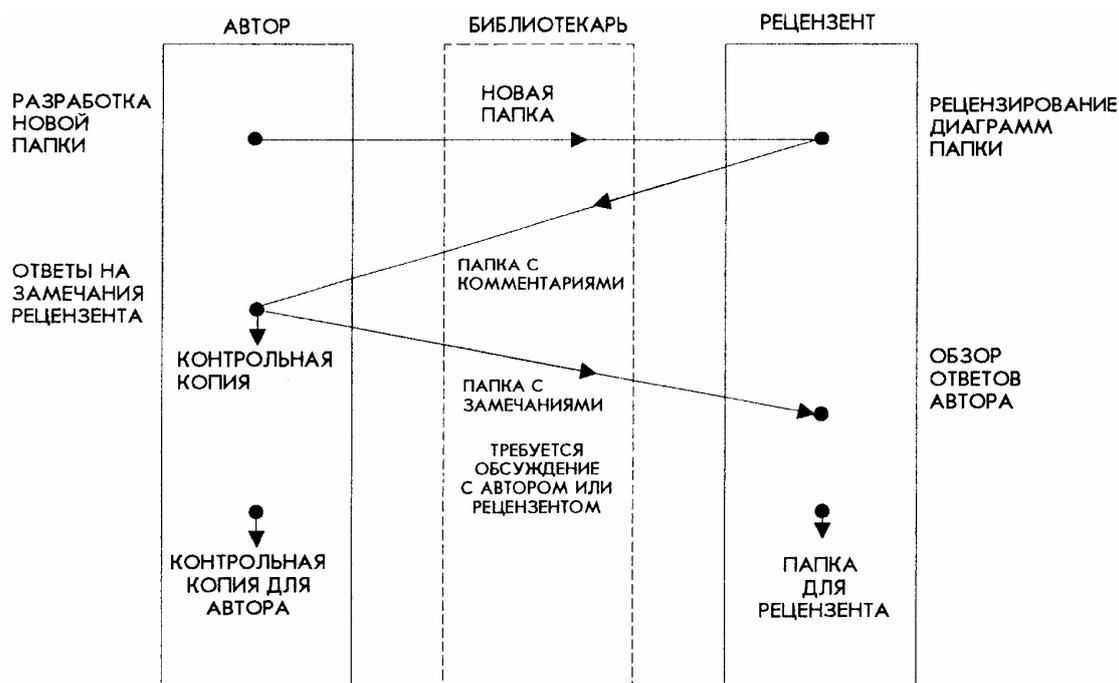


Рис. 5.1. Цикл папки.

### 5.2.1. Распределение ролей в цикле IDEF-папки

Роли и функции людей, участвующих в цикле IDEF-папки, следующие:

- Авторы - лица, создающие IDEF-модели.
- Рецензенты (эксперты) - лица, хорошо знающие предмет моделирования. От них автор может получать информацию с помощью опроса. Рецензенты владеют методологией IDEF и могут делать письменные структурированные замечания.
- Читатели (эксперты) - лица, достаточно хорошо знающие предмет моделирования. От них автор также может получать информацию с помощью опроса. Читатели знакомятся с документами, не делая письменных комментариев.
- Библиотекарь - лицо, ответственное за хранение документации, изготовление копий, рассылку папок и регистрацию.

"Роль" не зависит от должности. Один и тот же человек может выполнять несколько функций. Однако роль каждого участника индивидуальна и зависит от обсуждаемой папки.

### 5.2.1.1. Авторы

Автор опрашивает экспертов и составляет документы. Однако источником технического содержания документа автор может и не быть. Иногда автор дает только техническое описание или регистрирует материал, полученный из других источников. Автор часто выступает в роли редактора: проводит сортировку, идентификацию, организацию и представление информации, полученной от экспертов.

### 5.2.1.2. Рецензент

Рецензенты рассматривают предоставленный автором материал с точки зрения его технической правильности. Они должны выявить ошибки и внести исправления.

Роль рецензента является ключевой в получении высококачественных результатов. Рецензент определяет, насколько автор следовал методологии IDEF, придерживался ли он принятого подхода в достижении цели, есть ли ошибки или упущения, на которые нужно обратить внимание автора.

## 5.2.2. Руководство для авторов и рецензентов

### 5.2.2.1. Руководство для рецензентов

Не существует стандартного набора вопросов или правил, которые могли бы всегда применяться при рецензировании, так как объекты, стиль и методы слишком разнообразны. Однако существуют общие рекомендации. Основными критериями оценки качества документа являются: понятен ли документ лицам, для которых он предназначен, отвечает ли он поставленной цели, точен ли он в рамках заданного контекста. Общие принципы рецензирования:

- Делайте записи кратко, тщательно и конкретно. Автор должен опускать мелкие подробности для увеличения наглядности и уменьшения путаницы.
- Используйте обозначение, чтобы пометить замечание. Чтобы сделать - пометку, уточните порядковый номер того замечания, к которому относится пометка. Запишите этот номер, обведите его кружком и соедините пометку с соответствующей частью диаграммы с помощью знака " " (см. раздел 5.4.).
- Замечания должны быть конструктивны. Старайтесь не только отметить негативные моменты, но и предложить решение.
- Выделите время для обобщения замечаний на титульном листе или на отдельном листе бумаги. (Включайте сюда частные моменты, относящиеся к конкретным страницам). Здесь же можно сделать пометки для беседы с автором.

Продолжительность работы, связанной с замечаниями, зависит от разных факторов: степени знакомства с тем, что описывается, количества рассмотрения, опыта рецензента и автора и т.д. Папка, возвращенная автору без замечаний, означает, что рецензент в целом согласен с автором. Рецензент должен понимать, что он разделяет с автором ответственность за качество работы.

### 5.2.2.2. Взаимодействия автора и рецензента

Когда рецензент возвращает папку, автор отмечает каждое замечание знаком " " или "X". " " означает, что он согласен с рецензентом и учтет замечание. "X" означает, что автор не согласен. Рядом с таким замечанием автор должен письменно объяснить причину несогласия. После того, как автор ответит на все замечания, папка возвращается рецензенту.

После изучения ответов автора рецензент должен по оставшимся разногласиям сделать запрос на встречу с автором. Список конкретных разногласий составляет повестку встречи.

### 5.2.2.3. Правила встречи

Пока замечания и ответы на них не записаны, рецензентам и авторам беседовать не рекомендуется.

Процедура встречи такова:

1. Продолжительность каждой встречи должна быть заранее ограничена.
2. Каждое заседание должно начинаться с уточнения списка вопросов, которые необходимо обсудить.
3. Каждое заседание должно закончиться, когда участники придут к соглашению, что продуктивность встречи падает и индивидуальная работа была бы более результативна.
4. Каждая встреча должна заканчиваться согласованным перечнем действий, в который может быть включен график следующих встреч со своими повестками дня.
5. На каждом заседании следует назначать секретаря для ведения протокола.
6. Серьезные разногласия должны рассматриваться специалистами с документированием точек зрения обеих сторон.

По окончании встречи принимают резолюцию или составляют список разногласий, которые должны быть разрешены руководством. В резолюции может указываться необходимость дальнейшего изучения вопроса кем-либо из участников проекта.

### 5.3. IDEF-папки

IDEF-папка представляет собой технический документ. Она может содержать диаграммы, текст, глоссарий, итоговые решения, исходную информацию или какие-либо другие материалы, необходимые для рассмотрения.

Существует два типа IDEF-папок:

- Стандартная папка.

Все папки, рассылаемые для рецензирования и комментирования. Они рассматриваются как "рабочий материал", помогающий автору в "улучшении" модели, содержат не более двадцати страниц.

- Итоговая папка.

Папка с последней версией модели. Она рассылается только для ознакомления. Цель ее - предоставление текущей информации о модели в целом, по мере того, как части модели проходят цикл папки.

Стандартная папка содержит описание отдельных частей модели. Она составляется регулярно по мере выполнения работы и представляется на рассмотрение в рамках цикла IDEF-папки. В данном руководстве обсуждается этот тип IDEF-папок.

Итоговая папка представляется на рассмотрение каждые три месяца. Она содержит последний вариант модели. От получателей итоговых папок замечаний не ожидается, хотя, по желанию, они могут их делать. Итоговые папки помещаются получателями в свои архивы.

#### 5.3.1. Заполнение титульного листа стандартной папки

Оформите титульный лист для каждой папки, заполнив на нем следующие поля (рис. 5.2.).

1. Описание модели/документа

*Название* - должно быть указано наименование папки.

*Этап цикла* - "Как есть (AS IS)" или "Как должно быть (TO BE)"

*IDEF* - методология - 0, 1 или 2.

*Система* - акроним для системы или подсистемы.

*Тип рассылки* - указать, если он отличен от рассылки стандартной папки.

2. Информация о проекте.

*Автор* - фамилия человека, представляющего папку на рассмотрение. Если стандартная папка предназначается на рассмотрение как коллективная работа (проблемной группы, комитета, соавторов) представитель авторского коллектива берет на себя ответственность в качестве "автора".

*Дата* - дата отсылки в библиотеку.

*Компания* - название компании, представляющей папку на рассмотрение.

*Номер Проекта*

*Номер Задачи*

3. Информация о папке.

Отметьте, что папка является стандартной, укажите номер документа, присвоенный библиотекой, если папка является исправленным вариантом предыдущей стандартной папки.

4. Цикл рассмотрения.

После рассмотрения должны быть поставлены подписи рецензента и автора, а также указана дата.

5. Индекс узлов/содержание.

Номер узла, название, порядковый номер (С-номер) каждой страницы документа (включая титульный лист) вносятся в "содержание", которое всегда размещается на странице 2 (рис. 5.3.).

6. Замечания / специальные инструкции.

Замечания и любая другая информация для обозревателей. Они могут быть использованы для специальных инструкций библиотеке относительно способов обработки документов. Также библиотека может использовать это поле для инструкций получателю папки.

ИСПОЛЬЗУЕТСЯ В:	АВТОР:	ДАТА: <date>	РАБОЧАЯ ВЕРСИЯ	ЧИТАТЕЛЬ	ДАТА	КОНТЕКСТ: Тор	
	ПРОЕКТ:	ПЕРЕСМОТР:	ЭСКИЗ				
	ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10			РЕКОМЕНДОВАНО			
				ПУБЛИКАЦИЯ			
<input type="checkbox"/> Реестр <input type="checkbox"/> Файл <input type="checkbox"/> Автор	Номер документа	Получена _____ Закончена _____	ИНСТРУКЦИЯ ПО КОПИРОВАНИЮ: копия _____ страниц = _____ всего <input type="checkbox"/> как можно быстрее <input type="checkbox"/> к _____				
Читатели:	Новая папка к читателю	Срок возврата	Комментарий к автору	Срок возврата	Ответ Читателю		
Требуемый ответ: <input type="checkbox"/> Быстрый <input type="checkbox"/> Нормальный <input type="checkbox"/> Медленный <input type="checkbox"/> Не нужен							
Содержание:		Комментарии:		Файл модели:		СПЕЦИАЛЬНЫЕ ИНСТРУКЦИИ: <input type="checkbox"/> Авторский экземпляр не нужен _____ дополнительных авторских экземпляров	
Стр.	Узел	Название	С-номер	Статус			
А		Титульный лист			<input type="checkbox"/> Обновить <input type="checkbox"/> Заменить		
Б					} _____ этой папки		
В							
Г							
Д							
Е							
Ж							
З							
И							
К							
Л							
М							
Н							
О							
П							
Р							
УЗЕЛ:	РБО	НАЗВАНИЕ:	Титульный лист		НОМЕР:		

Рис. 5.2. Титульный лист.

### 5.3.2. Подготовка стандартной папки

Чтобы избежать неточностей, рассматривайте папку как единственный имеющийся в наличии источник информации. Исправьте все типографские ошибки. Дайте необходимые пояснения в виде кратких записей. Термины, появившиеся в результате такой работы, должны быть включены в глоссарий.

Собранный вами полезный материал предоставьте в помощь рецензенту. Никогда не используйте этот дополнительный материал для передачи информации, которая должна быть представлена соответствующим образом самими диаграммами. Если это возможно, используйте диаграммы как наиболее естественное средство взаимодействия, чтобы показать детали, важные для понимания концепции. Объедините все материалы с заполненным титульным листом и индексом узлов/содержанием и отправьте в библиотеку.



ИСПОЛЬЗУЕТСЯ В:	АВТОР:	ДАТА: <date>	РАБОЧАЯ ВЕРСИЯ	ЧИТАТЕЛЬ	ДАТА	КОНТЕКСТ: <context>	
	ПРОЕКТ:	ПЕРЕСМОТР:	ЭСКИЗ				
	ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10			РЕКОМЕНДОВАНО			
				ПУБЛИКАЦИЯ			
УЗЕЛ:	<node>	НАЗВАНИЕ:	<title>	НОМЕР:			

Рис. 5.4. Стандартный бланк для диаграмм.

Бланк построен так, чтобы рабочая информация в верхней его части могла быть отрезана после принятия окончательной версии. Бланк для диаграмм используется также для всей письменной документации.

#### 5.4.1. Рабочая информация

##### Поле *Автор /Дата /Проект*

В этом поле указывают имя автора диаграммы, дату, когда она была представлена впервые, название проекта, в рамках которого она разрабатывалась. Поле "Дата" может содержать и дополнительные даты, записываемые ниже первоначальной. Эти даты! относятся к изменениям, внесенным в первоначальный лист. Если лист принят без изменений, дополнительные даты не проставляются.

##### Поле *Замечания*

Это поле позволяет отслеживать замечания, записанные на диаграмме. По мере появления замечаний на странице их номера последовательно вычеркиваются. Вычеркивание обеспечивает быструю проверку номеров замечаний; номер, обведенный кружком, представляет собой однозначную ссылку на конкретное замечание.

##### Поле *Статус*

Классификация по статусу обеспечивает распределение диаграмм по следующим уровням:

- *Рабочая версия* - это диаграмма с большим числом изменений. Новые диаграммы являются, конечно же, рабочими экземплярами.

- *Диаграмма* - эскиз имеет меньше изменений по сравнению с предыдущей. Она свидетельствует о достижении некоторого согласия ряда читателей. Проектная диаграмма предлагается руководителем работ, но пока еще она не одобрена совещанием рецензентов из технического комитета или комиссии.
- *Рекомендованная диаграмма* и сопутствующие ей тексты прорецензированы и утверждены совещанием технического комитета или комиссии. В них не предполагается вносить изменения.
- *Публикация*. Этот материал может рассылаться для окончательной печати и опубликования.

#### Поле *Читатель / Дата*

В этом поле рецензент должен указать свою фамилию и дату рецензирования.

#### Поле *Контекст*

В данном поле определен контекст, в котором следует рассматривать диаграмму на данной странице. Схема контекста всегда относится к предыдущему (верхнему) уровню по отношению к текущей диаграмме. Текущая диаграмма показана на схеме контекста в виде заштрихованного прямоугольника, а все другие части диаграммы верхнего уровня изображаются в виде прямоугольников (или эллипсов). Дуги не представлены. Номер узла диаграммы верхнего уровня записывается в нижнем левом углу поля "Контекст" (рис. 5.5.).

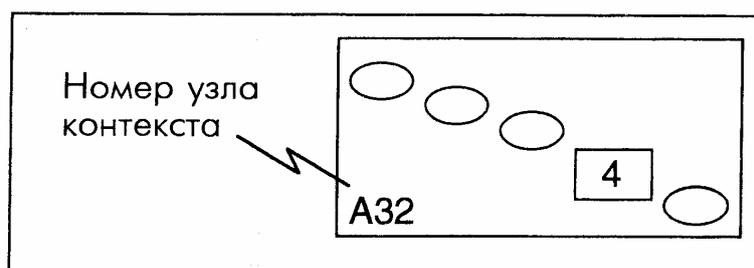


Рис. 5.5. Поле "контекст".

Поле *Контекст* контекстной диаграммы (A-0) пустое (NONE). Поле *Контекст* диаграммы верхнего уровня (A0) является "вершиной" (TOP).

#### Поле *Использована в...*

Это список диаграмм, отличных от контекста, которые каким-либо образом используют диаграмму на данной странице.

#### 5.4.2. Поле *Сообщение*

Поле *Сообщение* содержит основную информацию, которая обычно используется для построения диаграммы. Однако оно может служить и для иных целей: для размещения глоссария, замечаний, набросков и т.д. Автор не должен использовать никаких других бумаг кроме бланков для диаграмм.

#### 5.4.3. Поле *Название*

Поле *Название* содержит наименование материала, представленного на бланке данной страницы. Если поле *Сообщение* содержит диаграмму, то поле *Название* должно в точности повторять текст, записанный в исходном блоке.

#### 5.4.4. Поле *Номер*

Поле *Номер* содержит все номера, с помощью которых можно ссылаться на данную страницу.

*С-номер* состоит из двух или трех букв авторских инициалов и порядкового номера, присвоенного автором. Он размещается в нижнем левом углу поля *Номер* и служит основным средством ссылки на лист. Каждая диаграмма, используемая автором, получает свой *С-номер*. Когда модель опубликована, *С-номер* может быть заменен обычным номером страницы.

*Номер страницы* папки записывается библиотекарем с правой стороны поля *Номер*. Он формируется как номер документа, за которым следует номер, обозначающий лист внутри документа.

### 5.5. Ведение архивов

Каждый участник проекта должен вести архивы (подшивки) полученных документов. Библиотекарь ведет основной и справочный архивы для каждой папки, составленной в процессе разработки проекта.

Ведение архивов может быть разным в зависимости от личных вкусов. Однако, мы рекомендуем ведение следующих архивов:

- Архивы стандартных папок, формируемые авторами и рецензентами.
- Архивы итоговых папок, формируемые авторами, рецензентами и читателями.
- Рабочие архивы, формируемые авторами.

#### 5.5.1. Архив стандартных папок

Этот архив содержит стандартные папки, созданные владельцем архива или полученные извне. Папки, помещаемые в архив, должны регистрироваться. При регистрации следует включать информацию, позволяющую осуществлять удобный доступ к содержанию папки.

#### 5.5.2. Архив итоговых папок

Этот архив содержит итоговые папки, созданные или полученные извне. Они также должны регистрироваться.

#### 5.5.3. Рабочий архив

Этот архив содержит всю документацию, которая не была помещена в папку. В нем должен быть отражены процесс работы и текущие записи.

### 5.6. Процедура сквозного контроля *IDEF-модели*

В дополнение к циклу папки существует процедура сквозного контроля. Эта процедура может быть использована, когда участники разработки модели собираются для обсуждения.

1. Представьте модель, предназначенную для рассмотрения, с помощью индекса узлов. Индекс узлов позволяет обозревателю получить общее представление о содержании приведенного далее материала.
2. Представьте глоссарий терминов. Он позволит каждому обозревателю заменить свои термины на те, которые выбрала группа, представляющая модель. В этот момент термины не подвергаются сомнению. Изменение их потребует многих изменений в диаграммах.
3. Представьте на рассмотрение все диаграммы.

4. Обсуждение диаграмм - многошаговый последовательный процесс, в ходе которого задаются вопросы, выявляющие потенциальные недостатки диаграммы. Ниже приведено 6 шагов этого процесса.

Изменения диаграмм могут быть предложены на любом шаге. Их вносят немедленно или отмечают для последующего внесения.

### ***ШАГ 1. ПРОСМОТРИТЕ ДИАГРАММУ***

Этот шаг позволяет читателю получить общее представление о содержании диаграммы. Обычно читатель вначале изучает родительскую диаграмму, которая описывает текущую диаграмму как один из блоков. Затем он проверяет, как автор произвел декомпозицию этого блока,

Критерии принятия:

1. Декомпозиция удовлетворяет своему назначению и выполняется в контексте родительского блока. Все функции или данные нижнего уровня могут быть четко определены по каждому из блоков.
2. По мнению обозревателя, диаграмма отражает точку зрения и соответствует цели модели.
3. По мнению обозревателя, представлен достаточный объем новой информации, расширяющей знания о родительском блоке. В диаграмме нет чрезмерного количества деталей, усложняющих ее и вызывающих трудности при прочтении.

Кроме совершенно очевидных случаев, критику лучше отложить до шага 3. Однако первое впечатление не должно быть потеряно. Его можно зафиксировать на диске или каком-либо другом носителе информации.

### ***ШАГ 2. ПОСМОТРИТЕ НА РОДИТЕЛЬСКУЮ ДИАГРАММУ.***

После того, как читатель разобрался в декомпозиции, представленной на текущей диаграмме, следует обратиться к родительской диаграмме для проверки согласованности.

Критерии для принятия:

1. Декомпозиция охватывает все аспекты, которые обозреватель ожидает увидеть при чтении родительской диаграммы.
2. После детализации соответствующей части родительской диаграммы результат детализации мнение читателя этого блока покажется верным. Если это не так, отметьте пропущенные детали.

На этом шаге может оказаться весьма важным вернуться к родительской диаграмме, добавьте новые -пометки или дополнить существующие в соответствии с новым пониманием, которое появилось при рассмотрении декомпозиции.

### ***ШАГ 3. СОЕДИНИТЕ РОДИТЕЛЬСКИЙ БЛОК И ДИАГРАММУ ДЕКОМПОЗИЦИИ***

На этом шаге проверяются интерфейсные дуги, соединяющие родительскую диаграмму и диаграмму декомпозиции.

Критерии для принятия:

1. Нет пропущенных или лишних интерфейсных дуг.
2. Интерфейсные дуги представляются соответствующими ICOM-кодами.
3. Метки дуг диаграммы декомпозиции совпадают с метками дуг родительской диаграммы или уточняют их. Метки выражают правильное и полное содержание дуг.

4. Проверка соединяющих дуг не выявила проблем в родительской диаграмме. (Дополнительный интерфейс может внести неясность в содержание родительской диаграммы.)

Просмотр четырех сторон родительского блока по часовой стрелке с проверкой каждой дуги - хороший способ проверки соответствия ICOM-кодов дугам родительской диаграммы.

#### ***ШАГ 4. ПРОВЕРЬТЕ ВНУТРЕННЕЕ СОДЕРЖАНИЕ ДУГ***

Содержание блоков и дуг составляет суть создаваемой модели. Каждый блок должен быть проверен в соответствии с его порядковым номером и каждая дуга каждого блока должна быть просмотрена в соответствии с ее номером и ICOM-кодом. После завершения этого процесса читатель должен "пройти" через всю последовательность знакомых ему ситуаций, представленных на диаграмме, и проверить, действительно ли диаграмма моделирует известные взаимосвязи.

Критерии для принятия:

1. Диаграмма не выглядит хаотичной. Количество пересечений и изгибов дуг минимизировано.
2. Блоки должны быть сбалансированы в отношении детализации. В каждом блоке должно быть одинаковое количество деталей. Однако от этого критерия возможны отступления ради ясности понимания.
3. Диаграмма не должна быть в противоречии с опытом обозревателя и его знанием существа проблемы. Обратная связь и ошибки должны быть показаны так, как их представляет обозреватель.

#### ***ШАГ 5. ПРОЧТИТЕ СОПРОВОДИТЕЛЬНУЮ ДОКУМЕНТАЦИЮ***

На этом шаге рассматриваются моменты, которые автор проясняет в тексте, глоссарии и диаграммах FEO.

Критерии для принятия:

1. Текст подтверждает информацию, полученную из проверяемой диаграммы.
2. Некоторые направления, обратные связи, обработка ошибок и другие особенности, освещаемые в тексте, имеются в самой диаграмме или в диаграмме FEO.
3. Важные особенности диаграммы, не раскрытые на этапах 1-4, находятся в тексте, глоссарии или FEO-диаграммах.
4. Ссылки к диаграмме достаточно подробны, чтобы связать текст, глоссарий и диаграммы FEO с конкретными деталями диаграммы.

#### ***ШАГ 6. УСТАНОВИТЕ СТАТУС ДИАГРАММЫ***

Возможные варианты:

1. Рекомендуются в существующем виде.
2. Рекомендуются после модификации.
3. Эскиз: внесено слишком много изменений, необходимо перерисовать.
4. Не принимается: требуется полный повторный анализ.

## 6. Руководство для автора по созданию IDEF0-диаграмм

При создании любой IDEF0-диаграммы должны быть удовлетворены следующие требования:

- a) цель и точка зрения диаграммы должны соответствовать установленной цели и точки зрения всей модели;
- b) граничные дуги должны соответствовать дугам на родительской диаграмме;
- c) содержание диаграммы должно точно соответствовать содержанию родительского блока.

### 6.1. Основные этапы построения диаграммы

Поэтапный порядок построения диаграмм позволяет создавать диаграммы, образующие и взаимосвязанные модели. Необходимо соблюдать следующую последовательность:

1. Определите суть проблемы более точно, чем это предложено в названии функционального блока. Это можно сделать с помощью списка данных (объектов или информации), которые воздействуют на функцию или обрабатываются ею.
2. Изучите определенное таким образом содержание и сформируйте возможные подфункции общей функции.
3. Постарайтесь соединить эти подфункции естественными связями.
4. Разъедините и скомбинируйте подфункции для образования других блоков.
5. Нарисуйте окончательный вариант диаграммы, уделяя особое внимание размещению деталей и ясности изображения.

#### 6.1.1. Выбор контекста, точки зрения и цели модели

Перед началом разработки любой модели важно определить ее направленность: контекст, точку зрения и цель.

Контекст определяет объект модели как часть целого. Он очерчивает границы модели с ее внешним окружением посредством описания внешних интерфейсов.

Точка зрения специфицирует, что можно "увидеть" в контексте и под каким "углом". Она определяет позицию автора как наблюдателя системы или ее элемента и выбирается таким образом, чтобы получить максимально полезную информацию из разрабатываемой модели. В зависимости от интересов аудитории (управляющие, персонал, клиенты и т.д.) могут быть приняты различные точки зрения, выявляющие различные аспекты объекта.

**Каждая модель представляет только одну точку зрения.**

**Одна модель преследует только одну цель с единственной точки зрения.**

Цель определяет назначение модели или обеспечиваемых ею взаимодействий. Она воплощает причину, по которой модель создана (спецификация функций, проектная разработка, обслуживание клиентов и т.д.).

Описанные концепции определяют рамки создаваемой модели. Их содержание может уточняться в процессе работы, но при этом оно должно сохранять соответствие назначению модели, если оно верно и не меняется.

### **Сформулируйте четко цель.**

**Без ясно очерченного направления автор в процессе детализации может отклониться от первоначальной цели.**

**В самом начале любого анализа необходимо сформулировать ограничения контекста. Решите, что является сердцевинной содержанием перед тем, как будет создан самый верхний блок. Избегайте расширения этой тщательно выбранной области.**

**Каждый шаг должен сверяться с конечной целью. То, что не соответствует ей, может быть намечено для дальнейшего моделирования с соответствующими точками зрения.**

**Ясность вытекает из строгости детализации. Определение того, как далеко идти, когда остановиться, когда изменить "снаряжение", каким должно быть сопряжение отдельных частей, всегда зависит от цели создаваемой модели.**

#### 6.1.2. Построение контекстной диаграммы

Моделирование следует начинать с построения диаграммы A-0. Нарисуйте один блок, содержащий имя функции, которая охватывает всю сферу деятельности описываемой системы. Используйте дуги, входящие в блок и выходящие из него, чтобы представить обмен данными системы и ее окружения. Эта диаграмма с единственным блоком определяет контекст всей модели и образует основу для дальнейшей декомпозиции.

Некоторые авторы считают, что легче набросать A0, а затем начертить единственный блок и интерфейсные дуги, показанные на уровне A-0. Для получения хорошей основы для декомпозиции может оказаться полезным несколько раз переключиться с разработки диаграммы уровня A-0 на уровень A0 и наоборот.

Если диаграмма A-0 начата на слишком низком уровне детализации, сделайте блок A-0 основой нового уровня диаграммы A0. Перейдите на более высокий уровень и создайте новую диаграмму A-0. Повторяйте этот процесс до тех пор, пока A-0 не достигнет достаточного охвата всех аспектов системы. (Иногда такое повышение уровня скорее расширяет, чем уточняет выбранную точку зрения. Если это так, то постройте многоблочную контекстную диаграмму A-1, а диаграмму A0 оставьте для первоначальной цели.)

#### 6.1.3. Построение диаграммы верхнего уровня

Все функции системы содержатся в единственном блоке, показанном на диаграмме A-0. Диаграмма очерчивает границы контекста системы. Диаграмма A0 декомпозирует диаграмму A-0 на подфункции (от 3 до 6).

Настоящая "вершина" модели - диаграмма A0. Она является первым и наиболее важным выражением точки зрения модели. Ее структура ясно показывает, что пытается "сказать" диаграмма A-0. Термины и структура диаграммы A0 ограничивают и каждый последующий уровень, поскольку она является полным описанием выбранного объекта.

Нижние уровни описывают каждую из функций (блоков) A0. Чтобы достичь цели модели, эта цепочка детализации должна тщательно проследиваться на каждом шаге. Решающую роль играет правильное построение диаграммы верхнего уровня. Первый шаг

для автора связан с серьезными проблемами. Они заставляют автора поддерживать определенный уровень абстракции, следить за степенью углубленности модели, направлять детализацию к более низкому уровню.

#### 6.1.4. Построение последовательных диаграмм

Для формирования иерархии диаграмм декомпозируйте каждый блок диаграммы A0 на его основные части. Постройте новую диаграмму, на которой представлено то же, что на родительском блоке, но более подробно.

Для декомпозиции каждого блока на 3-6 блоков соберите дополнительную информацию. Сделайте первую диаграмму-набросок, перечислив все виды данных, содержащихся в декомпозируемом блоке. Стремитесь к тому, чтобы эти данные охватывали бы всю тематику родительского блока, без потери каких-либо частей при декомпозиции. Начертите блоки, основанные на этих перечнях, и начертите интерфейсные дуги между блоками.

Модифицируйте или нарисуйте диаграмму несколько раз, добиваясь максимальной наглядности. Для этого при необходимости разбивайте блоки на части или, наоборот, объединяйте несколько блоков в один.

Постройте части диаграмм более детализированных уровней, чтобы исследовать моменты, нуждающиеся в уточнении. Желательно создавать сразу 3-4 диаграммы, а не делать их по одной.

#### 6.1.5. Разработка сопроводительного материала

На конечном этапе разработки модели к каждой диаграмме должны быть приложены страница с сопроводительным текстом, глоссарий и, возможно, диаграммы FEO. Текст, связанный с диаграммой A-0, должен подтверждать назначение модели. Его пишут после построения диаграммы A-0. Текст дополняет содержание диаграммы A-0, связывая его с точкой зрения и целью модели.

Каждая диаграмма, включая A0, сопровождается текстом, относящимся только к ней. В нем коротко и четко излагает процесс. Он не дублирует то, что очевидно из диаграммы, но соединяет ее с исходными материалами. На каждом уровне он освещает направление, которое ведет к цели.

Глоссарий объясняет определения, предложенные автором для функций и данных диаграммы. Эти определения очень важны, потому что терминология, используемая в модели, может иметь совершенно другое значение в других ситуациях.

FEO-диаграммы (FEO - For Exposition Only) являются диаграммами, которые проясняют особенно интересные или тонкие аспекты диаграммы. Они не ограничены синтаксисом IDEF и могут содержать только часть дуг, снабжаться пометками и т.п. для усиления смысла.

#### 6.1.6. Выбор блока для декомпозиции

Исходя из законченной родительской диаграммы, хорошо проработайте верхние уровни, прежде чем приступить к детализации. Получив диаграмму A0, сосредоточьтесь на построении диаграмм A1, A2, A3. Декомпозиция A1 в A11, A111 должна быть выполнена позже. Это исключит потенциальную повторную работу при возможных изменениях в диаграммах верхних уровней.

Не следует стремиться всегда достигать одинаковой глубины декомпозиции. Глубина декомпозиции конкретного блока зависит от того, будет ли иерархия диаграмм передавать смысл лучше, чем одна диаграмма. Не откладывайте построение диаграммы более низкого уровня, например, A111. Сделайте набросок по свежим следам. Важно относиться к этим наброскам, как к эскизам, пока горизонтальный уровень не согласован полностью. Будьте

готовы вернуться к разработке нижнего уровня, если он противоречит верхнему, например A1, A2, A3 и т.д.

При выборе блока для декомпозиции полезно руководствоваться следующими рекомендациями:

1. Начинайте с "трудной части", т.е. с того, что менее знакомо или менее ясно.
2. Выбирайте блок, декомпозиция которого даст наибольший объем информации о других блоках.

Более простые объекты легче декомпонировать позже с меньшим риском ошибок или потерь. Они могут быть легко проработаны для приведения к соответствию декомпозиции более сложных объектов.

### 6.1.7. Основные этапы в работе автора

#### 6.1.7.1. Стадия сбора данных

##### ***Чтение исходной информации***

Автор собирает данные, читая исходную информацию.

##### ***Опрос***

Автор лично опрашивает эксперта - специалиста по данному вопросу. Такое интервью не проводится при обзорных встречах и при одновременном опросе нескольких человек.

##### ***Осмысление***

Перед построением диаграмм систематизируйте информацию, полученную в процессе чтения исходных документов и опросов.

##### ***Выбор блока***

Решите, какой блок наиболее подходит для детализации, основанной на полученной информации.

#### 6.1.7.2. Построение

##### ***Построение***

Построение включает процесс реальной разработки диаграммы. Он не ограничен только вычерчиванием блоков и дуг. В него входят также составление перечня элементов различных данных, создание набросков и т.д., которые предшествуют вычерчиванию блоков и дуг.

##### ***Переделка***

На этом этапе происходит осмысление построенной диаграммы, аналогичное редактированию и переработке текста. Задача сводится не к созданию диаграммы заново, а к выбору наиболее наглядного графического представления.

##### ***Создание мастер-диаграммы***

На этом этапе все улучшения переносятся на мастер-диаграмму. Это в основном механическая операция - мастер-диаграммы сохраняются, а изменения вносятся в их копии.

#### 6.1.7.3. Представление

##### ***Создание и редактирование текста***

Текст, сопровождающий любую диаграмму, должен быть точным. При его редактировании часто отсеиваются ненужные детали.

## ***Объединение***

Соберите все диаграммы, деревья узлов, глоссарии, текст и другие материалы, относящиеся к существу вопроса. Включите заполненный титульный лист.

### **6.1.7.4. Взаимодействие**

#### ***Реагирование***

Это относится к реакции автора в процессе чтения замечаний и ответам на них.

#### ***Беседа***

Этап сводится к встрече автора и рецензента для обсуждения ответов на замечания.

#### ***Собрания, группы***

Проводите собрания группы для обсуждения того, как продвигается работа, или "мозгового штурма" следующих этапов. Протоколы таких собраний должны отражать обсуждаемый предмет.

## **6.2. Создание IDEF0-диаграммы**

Создание диаграммы - индивидуальная и наиболее творческая работа. Различные авторы делают это по-разному. Никакая последовательность действий не будет в равной степени приемлема для всех авторов. Ниже в помощь начинающим авторам приведен примерный план действий при создании IDEF0-диаграмм.

1. Составьте относящийся к делу, но еще не структуризованный перечень данных, а также список первых появившихся соображений (в контексте родительского блока). Близкие по смыслу вещи по возможности объединяйте
2. Дайте имена функциям, которые воздействуют на перечисленные данные, и нарисуйте блоки вокруг имен.
3. Набросайте соответствующие дуги. В процессе создания блока рисуйте только "зачатки" дуг, чтобы лучше выделить блок. Закончите соединения, когда смысл диаграммы прояснится.
4. Выберите такое расположение блоков и дуг, которое максимально проявляет их взаимосвязи. Объединяйте дуги вместе, если структура слишком детализирована. Оставьте только существенные элементы и преобразуйте диаграмму.
5. Создайте текст, глоссарий и диаграммы FEO, если нужно выделить важные аспекты. При необходимости внесите изменения в родительскую диаграмму.

### **6.2.1. Разработка функциональных блоков**

Функциональные блоки создаются на базе основных подфункций родительской функции. Записав имена подфункций, начертите вокруг них прямоугольники, чтобы начать работу по построению диаграммы. На этой стадии количество блоков несущественно. Оно может быть изменяться за счет объединения или расщепления.

Посредством объединения несколько блоков группируются в единый блок для того, чтобы объединить родственные функции в единую, более общую функцию. Это исключит преждевременную детализацию, которая "затемняет" информацию, предназначенную для отображения на данном уровне.

Посредством расщепления единый блок делят на несколько частей. Процесс расщепления противоположен объединению. Его цель -обеспечить большую детализацию для лучшего понимания декомпозируемого объекта.

Проанализируйте получившийся набор функциональных блоков. Найдите оптимальный баланс между выбранными факторами. Посмотрите, нельзя ли сделать имена более конкретными и понятными для аудитории. Специальные термины и аббревиатуры используйте только при необходимости улучшить контакт с определенной аудиторией и только на уровне детальных диаграмм. Не используйте их на верхних (A-0 и A0) уровнях. Тщательно определите специальные термины в глоссарии.

Во всех случаях в качестве имен функциональных блоков выбирайте глаголы. Если слово может быть интерпретировано как глагол и как существительное, используйте для глагола обозначение (V).

### **Блоки**

1. В большинстве случаев блоки располагаются по диагонали от левого верхнего угла к правому нижнему. В принципе приемлемо любое расположение, ясно выражающее авторскую мысль, однако вертикальное или горизонтальное расположение часто приводит к скученности дуг, затрудняющей структурный анализ.
2. Блоки, расположенные вверху слева, "доминируют" над блоками, расположенными внизу справа, что выражается посредством управляющих дуг, которые связывают их. Такой стандартный способ изображения облегчает читателю понимание того, что вы хотите показать в диаграмме.
3. Номер каждого функционального блока указывается в его нижнем правом углу. Нумеруйте блоки на диаграмме слева направо и сверху вниз. Таким образом определяется номер узла для каждого блока. Начальные цифры полного номера узла блока те же, что и номер узла диаграммы. Последней цифрой номера узла является номер этого блока. Если блок на рис. 6-1 находится на диаграмме A4, то полный номер узла для этого блока будет A42.
4. На рабочей или эскизной диаграмме указывается также авторский C-номер, размещаемый ниже правого угла декомпозируемого блока.
5. Диаграмма не может содержать более шести блоков.



Рис. 6.1. Номер блока.

### 6.2.2. Создание интерфейсных дуг

Сделайте набросок интерфейсных дуг для каждого блока. Соедините концы дуг, чтобы показать, какие выходы обеспечивают те или иные входы и управления.

**Помните, что входные данные преобразуются функцией для получения выходных.** Если дуга содержит как входные, так и управляющие данные, то представьте ее как дугу управления.

Если не определено, является ли дуга управлением или входом, сделайте ее управляющей. Если есть сомнения в необходимости какой-либо части данных, отбросьте их.

Выходные дуги показывают результаты возможных реализации функции. Синтаксис для выходных дуг не идентифицирует, какие выходные дуги могут появляться при определенных обстоятельствах. Если их последовательность представляет особый интерес,

начертите диаграммы FEO, иллюстрирующие структуру. Не беспокойтесь о последовательности. Проверьте только, учитывает ли диаграмма все важные случаи.

Объединяйте группы взаимосвязанных дуг, где только возможно. Основная ошибка при создании дуг заключается в слишком большой детализации структуры дуг или их меток. Уровень детализации дуг должен соответствовать уровню детализации блоков. На верхних уровнях как имена блоков, так и метки дуг должны быть наиболее общими.

Для окончательной проверки сравните все дуги с перечнем данных, чтобы убедиться в том, что ни один нужный элемент реальных данных не упущен. Элементы, которые не удалось обнаружить при этом, либо не нужны на данном уровне детализации, либо были упущены из виду при создании дуг.

### ***Думайте об управлении и ограничениях, но не о потоке.***

Основным правилом для размещения структуры дуг является "взаимозависимость, а не последовательность".

Даже если описываемая функция изменяется шаг за шагом, достигая при этом ожидаемого конечного результата, старайтесь описать взаимосвязи, или ограничения, которым необходимо удовлетворить, или свойства инвариантности, которые всегда должны быть истинны, а не какую-то специальную последовательность шагов, которые приводят к этому результату.

Дело в том, что все блоки могут действовать одновременно, поэтому последовательность не имеет значения.

Всегда более полезно и информативно отразить "ограничения", а не последовательность. Насколько это возможно, диаграммы должны создаваться таким образом, чтобы правильно показать взаимоотношения независимо от того, какие шаги делаются первыми. Очевидно, что это нужнее, чем ограничиваться только одной из возможных последовательностей.

Часто гораздо проще сначала подумать о функциях подмодели в какой-то отдельной последовательности и сделать наброски на бумаге. Такой подход полезен, но необходимо всегда перерабатывать первоначальный вариант, чтобы он вписывался в структуру ограничений.

### ***Будьте внимательны к меткам***

Подчиненное положение несущественных деталей подчеркивает более важные. Не загружайте диаграммы большим, объемом информации и большим количеством дуг.

Не тратьте слишком много времени на один уровень. Не следует стремиться выразить все сразу, иначе неизбежны упущения и искажения.

Основная идея заключается в том, чтобы в конце концов сказано было все.

Чрезмерная насыщенность диаграмм делает их негибкими. Если это произошло, ценность диаграммы несколько снижается. Сила диаграммы в ее структурированности. Нужную структуру можно получить перенесением излишних деталей в подфункции.

Для выявления всех важных деталей полезно периодически возвращаться от диаграмм декомпозиции к диаграмме верхнего уровня.

### ***Отбросьте сомнительные дуги***

Часто трудно определить, нужно ли показывать ту или иную дугу. Наиболее легким решением будет следующее: "если есть сомнение, отбросьте дугу". Если есть сомнение в необходимости дуги, то, вероятно, было бы неправильным создавать ее. Неверное исключение сомнительной дуги на данном этапе не приведет к непоправимым последствиям.

Необходимость в дуге станет ясной при рассмотрении подфункций. Выполнение правил ICOM заставит вернуть дугу на этот уровень. Тогда отпадут все вопросы.

### 6.2.3. Уровень усилий

Первоначальной целью при разработке диаграммы является получение ясной диаграммы, представляющей вполне определенную информацию без нарушения правил синтаксиса. Когда диаграмма построена, критический подход при чтении ее и рассмотрении другими специалистами может улучшить первоначальный вариант. Большинство диаграмм может быть модифицировано для получения второго варианта, который в некотором смысле лучше, чем первый. Первый вариант редко бывает лучшим. После получения некоторых навыков первые диаграммы станут точнее и авторы почувствуют себя увереннее в применении методологии IDEF0.

Тем не менее, переработка диаграмм будет всегда необходимой частью процесса. Основная идея заключается в использовании цикла рецензирования для совершенствования диаграммы. При последовательном улучшении будут учтены все наиболее важные моменты.

IDEF0 - это методология, формирующая стиль мышления, а не просто упражнения в построении диаграмм. Излагая свои мысли на бумаге и заставляя работать систему обозначений и другие элементы методологии, вы придете к необходимому уровню детализации. Полагайтесь скорее на умение задавать хорошие вопросы, чем на способность давать безупречные ответы.

## 6.3. Перечерчивание IDEF0-диаграмм

### 6.3.1. Модифицированные блоки

При создании первой диаграммы получают 3-6 функциональных блоков приблизительно одного и того же уровня детализации. Путем их объединения и расщепления можно достичь такого разбиения, которое легче воспринимается или обеспечивает наиболее простое взаимодействие между функциональными блоками.

Чаще всего используют и объединение, и расщепление. Блоки расщепляют и полученные части объединяют в новые, которые более точно передают соответствующую идею. Охват остается тем же, но части группируются более понятным образом.

#### ***Расщепляйте и перефразируйте***

Важно, чтобы все блоки диаграммы имели согласующиеся формулировки. Никакие изменения не должны приводить к тому, чтобы какая-либо подфункция казалась не на месте. Разъединяйте и перефразируйте, чтобы восстановить равновесие.

Иногда один блок не соответствует другим блокам текущей диаграммы.

Часто это происходит потому, что другие аспекты подвергались изменениям и уточнениям. То, что раньше было хорошо, теперь стало противоречивым.

Разделите неудачный блок на две или более частей, одна из которых будет содержать суть первоначальной идеи.

Будьте готовы изменить формулировку имени блока (или блоков). При разделении новые идеи становятся более отчетливыми и больше увязываются с соответствующими блоками.

#### ***Объединяйте и заменяйте***

Часто более абстрактное понятие яснее и строже, чем преждевременная детализация. Объедините соответствующие блоки и замените их одним блоком.

Часто хороший уровень абстракции можно улучшить путем объединения нескольких блоков в более общий и переносом деталей в следующий нижний уровень. Нарисуйте линию

вокруг объединенных блоков и замените их все на один с соответствующим наименованием. Дополнительный уровень не приводит к усложнению. Он дает лучшее представление, потому что более ясно показывается структура.

Данный способ часто применяется совместно с расщеплением и является одним из наиболее мощных методов представления функций.

### 6.3.2. Дуги, собранные вместе

Как дуги, так и блоки на диаграмме должны иметь соответствующий уровень абстракции. Для достижения этого существует два способа:

1. Соберите вместе дуги, имеющие один и тот же источник и одно и то же назначение под единым, более общим названием и сделайте одну дугу.
2. Переименуйте некоторые блоки (используя расщепление и объединение), чтобы лучше распределить подфункции, и переименуйте результирующие дуги.

Избыток дуг только в редких случаях свидетельствует об ошибке. Вполне возможно, что все они точны. Но избыточность всегда является недостатком, если картина перегружена. Число используемых дуг определяется способностью читателя разобраться в сказанном с их помощью.

### 6.3.3. Модификация контекста

Понимание деталей в результате построения новой диаграммы помогает выявить ошибки в родительской диаграмме. Модификация родительской диаграммы является естественным процессом. При создании структуры дуг следует выполнять правило: "если есть сомнения в необходимости той или иной дуги данных для функционального блока, ее отбрасывают; последующая детализация покажет, нужна ли эта дуга". На данном этапе такие вопросы решают окончательно, исходя из конкретной ситуации.

Изменения родительской диаграммы различны по степени сложности. Если изменение может быть внесено только при пересмотре ближайшей родительской диаграммы, то это проще, чем пересмотр более ранних диаграмм. Предлагая изменение, тщательно обдумайте его и оцените степень его сложности. Предпочитая простые изменения более сложным, можно снизить качество детализации. Когда коррекция закончена, проверьте все граничные связи, чтобы убедиться, что ISOM-коды указаны правильно. Об изменениях сообщайте другим авторам, работающим над диаграммой.

Всегда имейте в виду родительскую диаграмму декомпозируемого блока. Это поможет в процессе создания диаграммы.

Если детализированная диаграмма не соответствует контексту, то либо она, либо контекст неверны. Исправьте контекст или диаграмму. Они должны соответствовать друг другу.

### 6.3.4. ISOM-синтаксис для соединения диаграмм

Важным аспектом понимания диаграмм является способность находить необходимые факты и интерпретировать их. Номера узлов показывают структуру декомпозиции блока. Соединения и местоположения дуг представляет интерфейсные связи.

ISOM-коды пишутся на всех дугах, имеющих на диаграмме один неприсоединенный конец. Эти граничные дуги соединяют дуги "сквозь" все диаграмм. Каждая граничная дуга обозначается ISOM-кодом, чтобы определить связь дуги с родительской диаграммой.

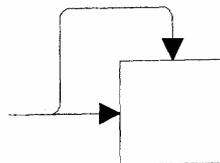
## 6.4. Графическое представление

Расположите блоки по диагонали (в соответствии с требованиями к структуре) от левого верхнего угла к правому нижнему так, чтобы дуги обратной связи шли вверх и влево. Пронумеруйте блоки слева направо.

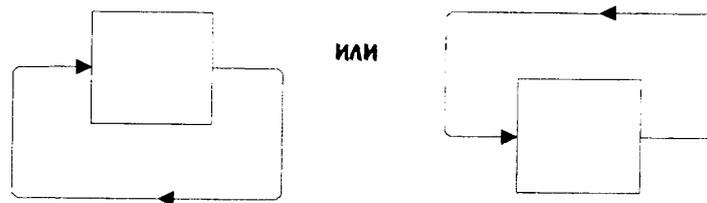
Это размещение лучше начинать с наиболее часто используемых управляющих дуг, добавляя затем реже используемые дуги. Подмножество дуг позволит определить расположение блоков. Вычертите все граничные дуги, показанные на родительской диаграмме, и добавьте оставшиеся дуги.

### 6.4.1. Представление ограничений на диаграмме

1. Когда входная дуга служит как управление и вход, покажите ее как управление. Если есть сомнения, сделайте ее управляющей. Аута» являющаяся на родительской диаграмме управляющей, может появиться на следующем уровне как управление или вход, либо как то и другое, в зависимости от ее взаимоотношений с подфункцией на этом уровне.
2. Функциональные блоки должны всегда иметь управляющие дуги, хотя могут не иметь входных.
3. Вообще говоря, не следует расщеплять дугу на управление и вход для одного и того же блока. Эта деталь лучше будет представлена на диаграмме нижнего уровня, где очевидны назначение каждой ветви и причина такого расщепления. Когда это необходимо, выберите такие метки для обеих ветвей, которые отобразят обоснованность вашего решения:

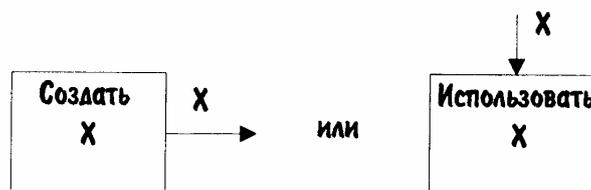


4. Циклический процесс или накопление данных на IDEF-диаграмме могут быть показаны следующим образом



Этот способ применяется только тогда, когда накопление происходит "на этом уровне". В противном случае покажите петлю обратной связи на следующем уровне детализации, т.е. "внутри" блока.

5. Пытайтесь избегать избыточности, например:



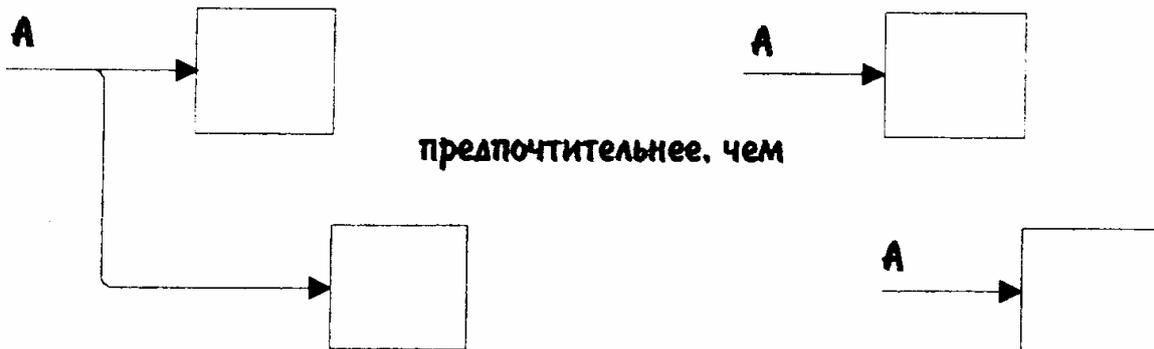
В представленных примерах имена блоков повторяют сообщение, изображенное дугами. Дополнительное обдумывание обычно приводит к более информативным именам блоков.

## 6.4.2. Размещение дуг

1. Размещайте дуги вдоль горизонтальных и вертикальных линий, а не по диагонали или в виде кривых (за исключением углов).
2. Помещайте углы дуг, их пересечения и метки (надписи) на приемлемом расстоянии от блоков.
3. Не используйте ключевые слова, такие как "данные", "функции", "вход", "выход", "управление" или "механизм" в именах или надписях, если это не вызвано особой необходимостью.
4. Если дуга длинная, надписывайте ее дважды:



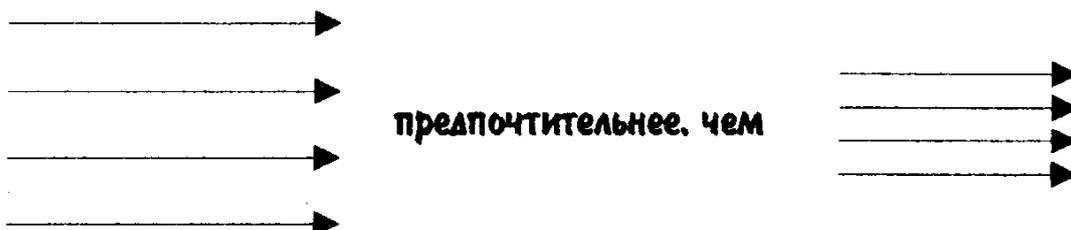
5. Помещайте ICOM-коды на свободных концах дуг.
6. Связывайте граничные дуги, имеющие свободные концы, чтобы показать все взаимосвязанные места. Иначе читатели могут не заметить связи:



7. Не доводите дуги до самого края диаграммы:



8. Оставляйте достаточные промежутки между параллельными дугами. Их трудно отслеживать визуально, если они очень длинные и находятся близко друг от друга:



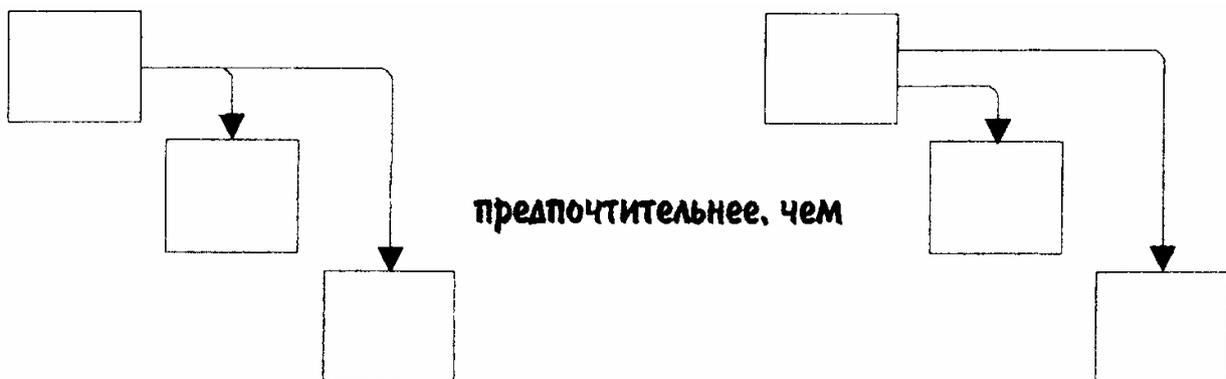
9. Помещайте дополнительные стрелки на дугах, когда это необходимо для ясности.

### 6.4.3. Расположение дуг

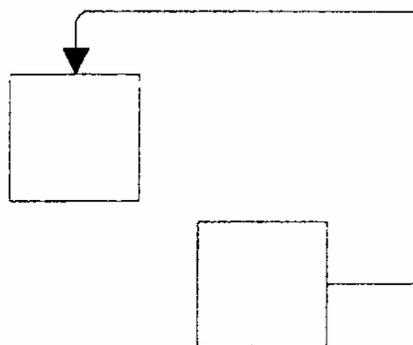
1. Объединяйте в связку дуги с одними и теми же источниками и пунктами назначения, если дуга не является настолько важной, что включение ее в пучок уменьшит ясность схемы:



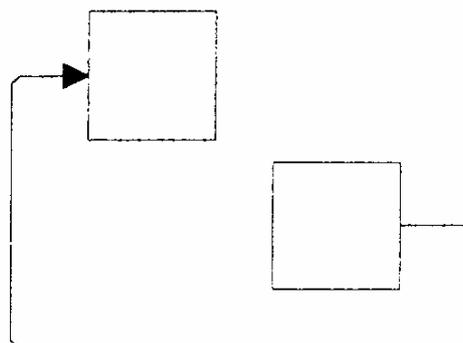
2. На любой из сторон блока не должно быть более четырех дуг. Если их больше, объедините дуги, присвойте общую метку и сделайте ответвления к их конечным пунктам:



3. Управляющая обратная связь становится более ясной, если соответствующая дуга нарисована направленной вверх и проходящей над блоком:



Входная обратная связь становится более ясной, когда соответствующая дуга нарисована направленной вниз и проходящей под блоком;



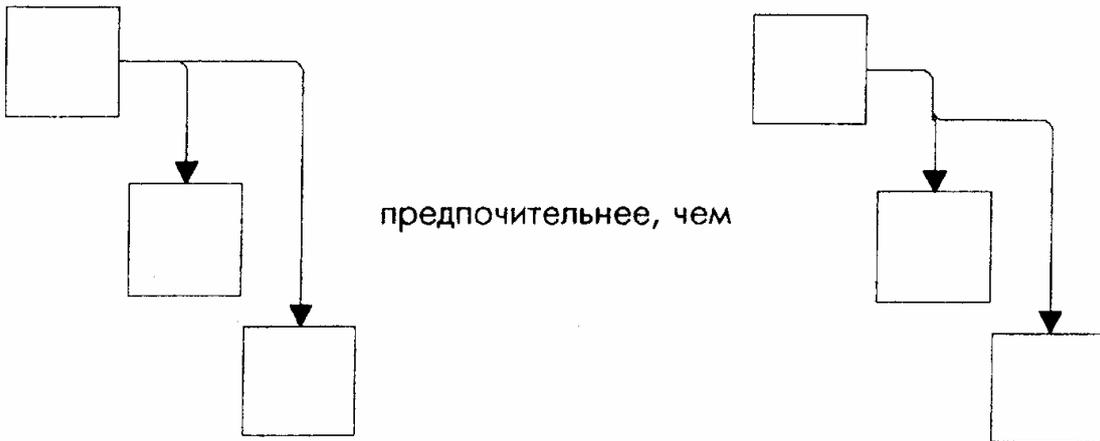
4. Если дуга разветвляется и входит в несколько блоков, изобразите ее на одних и тех же местах в соответствии с ICOM-кодами на каждом блоке, если это возможно:



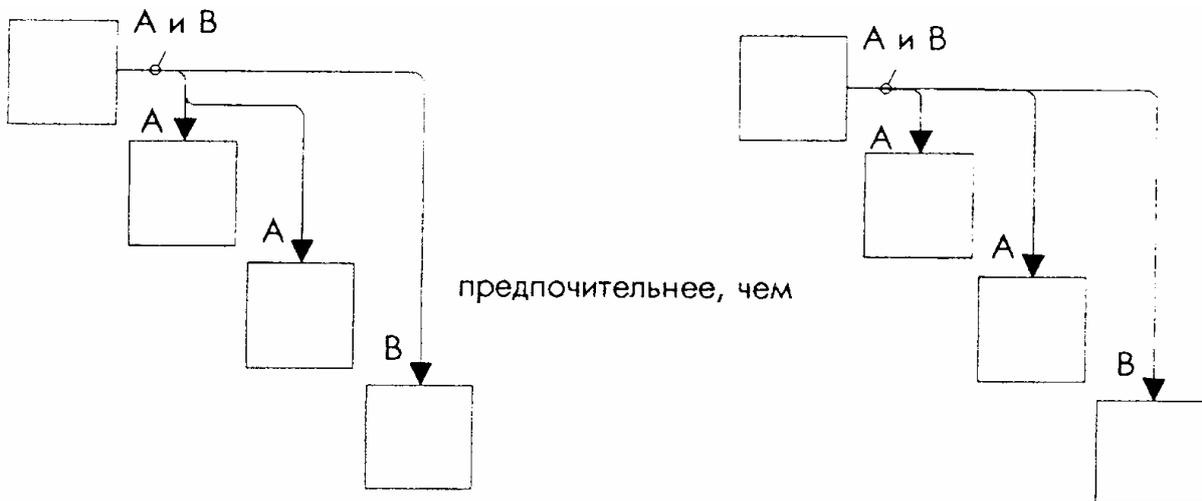
5. Располагайте дуги так, чтобы минимизировать пересечения несвязанных дуг:



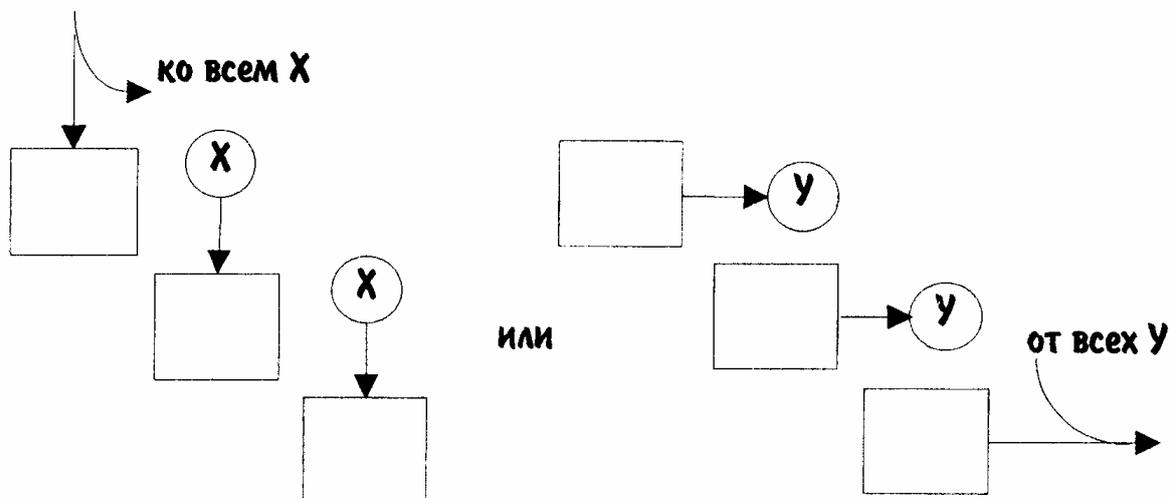
6. Сводите к минимуму число кривых линий и углов, где это возможно:



7. Используйте, где это возможно, выразительные возможности ветвящихся дуг:



8. Чтобы избежать нагромождений при изображении внешних дуг, которые одинаково подходят к каждому или одинаково приходят от каждого блока, используйте следующие обозначения;



## 6.5. Текст к диаграмме

Текст, который сопровождает каждую диаграмму, представляет собой краткий ее обзор. Объем текста всегда должен быть меньше одной страницы. В тексте освещаются особенности, которые, по мнению автора, представляют особый интерес для читателя или являются существенными для понимания идеи диаграммы. Он не повторяет всех деталей диаграммы. Если диаграмма позволяет получить всю необходимую информацию, текст можно опустить.

Пишите текст только после того, как диаграмма получила одобрение. Откладывая момент написания текста, вы усиливаете изобразительные свойства диаграммы и способствуете передаче необходимой информации. Текст, основанный на тщательно вычерченной диаграмме, будет так же структуризован и организован, как сама диаграмма.

Включите определения в глоссарий, чтобы пояснить термины, применяемые в диаграмме. Слово может иметь различные дополнительные толкования в различных фирмах и даже внутри одной фирмы. Например, "директивы" могут исходить из проектных или финансовых подразделений, от правительства и т.д.

Попытайтесь получить хороший текст без добавления FEO-диаграмм. FEO использоваться для иллюстрации малозаметных моментов, которые проясняют суть диаграммы, но которые могли бы загромоздить ее, если бы были в нее включены. Если FEO-диаграмма необходима, текст, который сопровождает ее, должен ссылаться на соответствующую диаграмму.

### 6.5.1. Текст А-0

Пишите текст уровня А-0, когда диаграмма А-0 начерчена, и представляйте его, когда уровень А-0 принимается на рассмотрение. Так как вся декомпозиция начинается с диаграммы А-0, помещайте любые значимые факты, относящиеся к полной модели, в текст, связанный с этой диаграммой. Текст А-0 должен содержать обсуждение точки зрения и цели модели.

## 6.5.2. Ссылки и замечания

В тексте и замечаниях, чтобы сделать ссылку на диаграмму или какую-либо ее часть, можно использовать краткий язык ссылок. Например:

02	означает	"Граничная дуга с ICOM-кодом 02",
211	означает	"Блок 2, вход I",
202 к 3C1	означает	"дуга от 202 до 3C1"
Ⓜ	означает	"пометка R"

Эти элементы языка могут использоваться отдельно, если они отсылают к текущей диаграмме (например, в замечании или тексте). В других случаях им будет предшествовать номер узла и, если необходимо, имя модели. Точка означает "смотри" определенный элемент на указанной диаграмме:

MODEL /A21.3C2	означает	"в "MODEL" на диаграмме A21 смотри блок 3, управление 2",
A 42.3	означает	"на диаграмме A42 в этой модели смотри замечание 3."

При каждой ссылке давайте ровно столько информации, сколько необходимо для полной однозначности. Самой полной формой является:

АОСТ / A21 (T56). 102 к 4C3, что означает: "в модели "АОСТ", диаграмме A21, варианте T56, смотри дугу от блока 1, выход 2 до блока 4, управление 3."

Давайте точные и полные ссылки на текст, глоссарий и FEO. Нетрудно быть точным и не оставлять места сомнениям читателя.

При написании текста описывайте диаграмму целиком, создавая содержательный рассказ.

Не заботьтесь о дополняющих ссылках. Они будут тормозить работу и сделают предложения трудночитаемыми .

Закончив текст, пройдите по нему и добавьте ссылки номеров блоков для точной привязки текста к диаграмме.

Чаще всего достаточно номера блока (блок 3) или ссылки к паре дуг (блок 3, 01 и C2). Когда читателю необходимо "посмотреть" другую диаграмму, в языке ссылок используйте точку.

Ссылки должны служить для пояснения. Они не будут навязчивыми, если их добавить после завершения работы с основным текстом.

## 6.6. Контроль качества модели

В материале, представленном в следующих разделах, рассматриваются методы контроля качества разрабатываемых IDEF0-моделей.

### 6.6.1. Синтаксис

Синтаксические правила являются ограничениями (условиями) в том смысле, что они описывают логически то, что представлено графически. Компоненты диаграммы в IDEF0 могут рассматриваться как "примитивы"; синтаксические ограничения являются

утверждениями, которые определяют допустимые или имеющие место отношения и операции, воздействующие на эти примитивы. Синтаксические ограничения могут рассматриваться как критерии: если ограничение не удовлетворяется или нарушается, то графическое изображение не соответствует критерию, и результирующая диаграмма или модель несовершенна с точки зрения этого ограничения.

Выделяются по крайней мере три различных типа синтаксических ограничений в зависимости от уровня анализа или уровня графического представления, к которому они относятся:

1) Локальная конструкция

Эти правила относятся к простым, первого порядка сложности или связанности примитивам. Они определены для диаграмм и FEO-диаграмм. Им нужно следовать для обеспечения правильности конструкции. Диаграммы, имеющие недостающие компоненты или показывающие отношения между примитивами, отличные от тех, что установлены ниже, являются "синтаксически дефектными".

2) Глобальная конструкция

Эти правила относятся ко всей диаграмме, но не к тексту или FEO. Таким образом, они установлены для нескольких конструкций и их выполнение может быть проверено только после окончания разработки диаграммы.

3) Конструкция модели

Ограничение этой категории определяет расположение узлов и буквенно-цифровые обозначения, которые создают упорядоченную иерархию диаграмм.

Материал, изложенный ниже, может служить руководством при определении минимального уровня приемлемости IDEF0-диаграмм и моделей. Недостатки могут быть выявлены на локальном и глобальном уровнях или на уровне всей модели. "Завершенность" и "корректность" следует понимать как всеобщие определяющие стандарты. Рабочие определения (для IDEF0-синтаксиса) приводятся ниже.

### **Завершенность**

**Определение.** Степень, с которой все элементы - входы, надписи, блоки, дуги и идентифицирующие записи - присутствуют на диаграмме или в модели.

### **Корректность**

**Определение.** Степень точности, с которой диаграмма или модель сконструирована, пояснена надписями и идентифицирована, т.е. степень точности синтаксических отношений, представленных в графическом изображении.

**Показатели.** Проверка точности выполнения ограничений (см. разделы 6.6.1.1.-6.6.1.3).

#### **6.6.1.1. Синтаксис локальной конструкции**

- а) Однонаправленные сегменты дуг состоят из упорядоченных пар концевых точек (источник, назначение), где:

Исходная точка принадлежит к одному из следующих типов:

- граничная концевая точка около I-, C- или M-границы диаграммы;
- концевая точка блока на стороне O блока;
- конечная точка ветвления;
- конечная точка соединения;

Точка назначения принадлежит к одному из следующих типов:

- граничная концевая точка около О-границы диаграммы;
- концевая точка на сторонах I, С, М блока;
- источник (начало) соединения;
- источник (начало) ветвления.

Заметьте, что все комбинации удовлетворяют правилам, исключая дугу, соединяющую границу диаграммы с границей диаграммы.

- b) Метки дуг связаны с сегментами дуг. Чтобы сделать эту связь ясной, можно использовать зигзагообразные линии.
- c) Каждый ICOM-код родительского блока должен быть связан с граничным концом некоторой дуги.
- d) Блоки могут иметь целочисленный номер. Номера блоков начинаются с первого и увеличиваются на единицу.
- e) Имена связаны с блоками и не могут существовать без них.
- f) Концевые точки или дуги на границах диаграмм имеют только либо знак "туннеля" "()", либо ICOM-код.
- g) Концевые точки дуг на концевых точках блоков могут иметь знак туннеля "()".

#### 6.6.1.2. Синтаксис глобальной конструкции

- Законченная. IDEF0-диаграмма должна включать не менее трех и не более шести блоков, если это не диаграмма А-0, состоящая только из одного большого блока.
- Каждый блок должен быть назван и пронумерован. На каждой диаграмме номера начинаются с первого и последовательно увеличиваются на единицу. Блок на диаграмме А-0 нумеруется 0. Два блока на диаграмме не могут иметь один и тот же номер.
- Каждый блок должен иметь по крайней мере одну выходную дугу на стороне О (справа) и одну входную дугу на стороне С (сверху).
- Все концы сегментов дуг на границах диаграмм должны иметь либо ICOM-код, либо знак туннеля "()", за исключением диаграммы А-0, у которой нет ни ICOM-кодов, ни знаков туннеля "()".

#### 6.6.1.3. Синтаксис конструкции модели

##### **Правила и соглашения в методологии IDEF0 для структуры принятой модели**

Модель является иерархически структурированным набором узлов, представляющим собой последовательность "уточнений", каждое из которых может быть детализировано на одной или нескольких диаграммах. Верхний узел IDEF0-диаграммы в модели обозначается А0. Блок имеет номер узла, получаемый добавлением номера блока к номеру узла диаграммы, которой блок принадлежит. Диаграмма имеет тот же номер узла, что и блок, который она детализирует. Однако при комбинации с номером блока "О" (из узла А0) этот номер опускается. Например, декомпозиция блока 2 на диаграмме, детализирующей узел А0, обозначается А2. Декомпозиция блока 1 на диаграмме, детализирующей узел А2315, обозначается А23151.

ФЕО-диаграмма, которая поясняет IDEF-диаграмму, обозначается путем соединения буквы F с номером диаграммы, расположенным внизу справа. Текст или глоссарий, которые

поясняют диаграмму IDEF или FEO, обозначаются путем присоединения букв T и G справа к обозначению диаграммы или FEO. Например, FEO, поясняющая блок 3 узла A21, обозначается A213F. Текст, поясняющий FEO A31F, обозначается A31FT. Глоссарий, определяющий термины, использованные на диаграмме A114, должны обозначаться A114G и т.д. Кроме того, если имеется несколько поясняющих FEO или страниц словаря, то справа от букв F и G ставится порядковый номер. Так, две FEO, поясняющие диаграмму A213, должны обозначаться A213F1 и A213F2.

Кроме самого верхнего узла функциональной IDEF-модели, всегда обозначаемой A0, существуют другие узлы высшего уровня, представляющие контекст и окружение. A-0 (A минус ноль) показывают контекст A0. Блок A0 может быть показан в диаграмме A-1 и т.д.

Взаимосвязи между дугами на родительской диаграмме и диаграмме декомпозиции отражаются с помощью ICOM-кодов. Расположение ICOM-кодов представляется подробно на диаграмме-потомке на концах дуг вдоль границ диаграммы. Соответствующий ICOM-код на родительской диаграмме определяется косвенно из положения конца дуги. Буква (I, C, O, M) соответствует стороне блока, к которому присоединяется конец дуги. За ним следует номер, определенный порядком, согласно которому дуга появляется на этой стороне блока (нумеруемые сверху вниз и слева направо). ICOM-коды будут меняться, если дуги на родительском блоке добавляются, удаляются или передвигаются.

## 6.6.2. Семантика

"Семантика" является термином, который применяется к большому ряду факторов, каждый из которых имеет некоторое влияние на "значение" меток, диаграмм и моделей в методологии IDEF0. Для того чтобы установить критерий, используемый для оценки моделей, следует рассмотреть эти аспекты "значения", относящиеся к содержанию и однозначному наименованию данных и функций. Критерий для оценки общего "изложения", "сообщения" или "смысла" диаграммы или модели должен быть выражен в терминах допустимой сложности.

Можно выделить пять типов семантики:

1. ПОЛНОТА.
2. КРАТКОСТЬ.
3. СОГЛАСОВАННОСТЬ.
4. ПРАВИЛЬНОСТЬ.
5. СЛОЖНОСТЬ/ПОНИМАЕМОСТЬ

Каждый из этих критериев рассматривается в следующих коротких разделах.

### **ПОЛНОТА**

Одним из критериев оценки адекватности информационного охвата модели является уточнение. Имеется в виду та степень детализации, при которой вводятся дополнительные детали в диаграмму декомпозиции. В идеале детализация должна быть достаточной для того, чтобы сделать исходную диаграмму более содежательной и информативной, но при этом не перегруженной большим количеством деталей, затрудняющим ее понимание. Между степенью детализации, выражаемой именами функций-блоков, и тем, что выражается с помощью меток дуг данных, должен сохраняться баланс. Диаграмма с очень общими и абстрактными именами функциональных блоков, но с подробно детализированными метками на дугах данных семантически несбалансирована. Степень уточнения может быть низкой и на последующих диаграммах декомпозиции, поскольку место для дальнейшей детализации меток на дугах данных будет ограниченным.

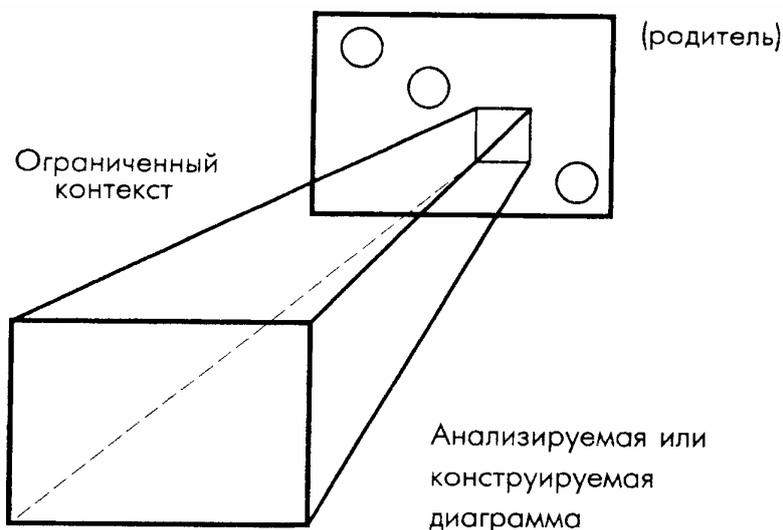
## Определения и показатели

*Определение.*

Достаточность информационного содержания для охвата сути вопроса (контекст).

*Показатели и параметры.*

Уровень диаграммы: Для любой диаграммы, за исключением самой "верхней", степень охвата информации, относящейся к ограниченному контексту на родительской диаграмме.



*Уровень всей модели:*

Степень декомпозиции блоков (меняется в соответствии с целью модели; степень детализации, необходимая для достижения цели).

Степень детализации дуг от вершины до самого нижнего уровня модели (меняется вместе с уровнями детализации и абстракции функциональных блоков).

## КРАТКОСТЬ

"Информационная ценность" меток и названий в методологии IDEF0 определяет, в конечном счете, потенциальное значение содержания диаграммы для читателя. Поэтому авторы должны стараться выбрать метки так, чтобы они были общеупотребительны, особенно если они связаны с технологическими процессами или практикой производственной среды. Термины, используемые на диаграммах и в сопроводительных текстах, должны быть естественными для читательской аудитории. Например, для бухгалтеров будут естественными такие термины, как "косвенный труд", "прямой труд", "G&A численность", а управляющие проектом предпочтут термины "учитываемый труд", "исследования с внутренним финансированием", "поддерживающий труд".

"Степень избыточности" трудно измерить при простом чтении диаграммы или серии диаграмм. Если авторы представляют глоссарий ключевых терминов RI достаточно подробные списки характеристик или особенностей технических терминов, читатели и рецензенты располагают более объективными средствами для принятия решения в случаях, когда две метки являются либо двумя различными названиями одного и того же, либо они именуют разные вещи.

IDEF0-метки являются "кандидатами" для имен классов сущностей и атрибутов в IDEF1; если класс сущностей используется как метка дуги IDEF0, атрибуты класса сущностей помогают в определении точного содержания дуг и облегчают понимание и сравнение дуг. Списки атрибутов для сущностей, поименованных на диаграмме - полезное дополнение к

гlossарию, Они позволяют оценить степень избыточности на основе перекрывающихся наборов атрибутов.

### **Определения и показатели**

#### *Определение.*

Точность информации, содержащейся в диаграмме или модели и/или передаваемой ими. Соответствие терминологии и символов. Отсутствие информации, второстепенной для ориентации модели.

#### *Параметры:*

- Информационное значение меток, названий и имен функций.
- Степень избыточности в именах.
- Естественность терминологии, соответствующая:
  - сути предмета
  - аудитории

## **СОГЛАСОВАННОСТЬ**

Синтаксические проверки служат для подтверждения точности графических нотаций. Они позволяют выявить согласованность меток дуг и провести трассировку дуг данных во всей иерархической структуре модели. Степень согласованности модели определяется рецензентами и экспертами - специалистами в предметной области модели.

Синтаксические проверки согласованности имен функционального блока и обозначений дуг будут проще, если представить список атрибутов как часть глоссария, сопровождающего диаграмму или модель.

### **Определения и показатели**

#### *Определение.*

- Внутреннее: степень единообразия символики и терминологии (внутри модели)
- Внешнее: степень, с которой содержание дает последовательное представление о моделируемой системе (вне диаграммы или модели)

#### *Показатели и параметры*

- Правильность ICOM-меток: размещение и возможность проходить через всю иерархическую структуру.
- Правильность использования туннелей.

## **ПРАВИЛЬНОСТЬ**

Правильность - возможно, наиболее субъективная из семантических характеристик, обсуждаемых в этом разделе. Это объясняется главным образом тем, что окончательный "стандарт" для оценки формируется на основе мнений отдельных специалистов, имеющих наиболее полное представление о моделируемом объекте. Чтобы помочь построению моделей, которые могли бы считаться "правильными" на основе рассмотрения обзора, а также путем оценки техническими экспертами, в методологии IDEF0 разработан "Цикл папки".

## **Определение и показатели**

### *Определение*

- Степень, с которой информация, выраженная в диаграмме или модели, точно описывает моделируемую систему
- Степень, с которой предполагаемые "ограничивающие" отношения и интерфейсы "данные-функция" означают действительно существующие отношения.

### *Показатели и параметры*

- Рассмотрение модели экспертами.
- Проверка по документации компаний и по учебным руководствам терминологии и ее технического употребления.

## **СЛОЖНОСТЬ/ПОНИМАЕМОСТЬ**

"Понимаемость" - важный и трудный критерий оценки качества модели. Это связано главным образом, с его абстрактностью и субъективностью. Другим источником проблем, связанных с оценкой "понимаемости", является тесная взаимосвязь синтаксических и семантических факторов. Мы определяем "понимаемость" как функцию того, "насколько хорошо содержание модели представляется с помощью синтаксиса", поэтому приемлемые "показатели" должны быть получены как из синтаксиса, так и из семантики. Следовательно, ни один отдельный элемент семантики не может служить достаточным показателем, насколько хорошо тот, кто оценивает сложную модель, способен ее понять.

Измерение "сложности", однако - более выполнимая задача. Мы предлагаем рассматривать "понимаемость" и "сложность" как двойственные критерии. Это означает, что некоторые высокосложные фрагменты модели будут, как можно полагать, трудными для понимания и, аналогично, фрагменты, имеющие меньшую сложность, будут, вероятно, намного доступнее для пользователя и легче для понимания.

## **Определения и показатели**

### *Определения*

- Степень, с которой цель модели или диаграммы ясна для оценки
- Степень, с которой дается точное описание через синтаксис в ответ на вопрос "что говорит модель/диаграмма?"

### *Показатели и параметры*

- Легкость нахождения "ОСНОВНОГО ПУТИ"
- Число активации блоков
- Число потенциальных сценариев
- Легкость идентификации активации и сценариев
- Легкость и успех "улучшения размещения"
- Схема оценки связей и сопряжении

Другим критерием, полезным для разработчика IDEF0-модели и рецензента, является легкость, с которой может быть улучшено размещение элементов на диаграмме. "Улучшение размещения" определяется здесь как процедура, необходимая для придания однозначности информации, содержащейся в диаграмме или модели и передаваемой посредством функциональных блоков и интерфейсов "дуги-данные". По мере того, как

модель становится все более сложной и, следовательно, менее понятной, задача улучшения размещения становится более трудной. Правила, касающиеся улучшения размещения элементов на диаграмме с целью уменьшения ее сложности, можно представить следующим образом:

- Пограничные дуги более чем с одним ICOM-кодом должны быть разделены так, чтобы каждый ICOM-код имел собственную дугу.

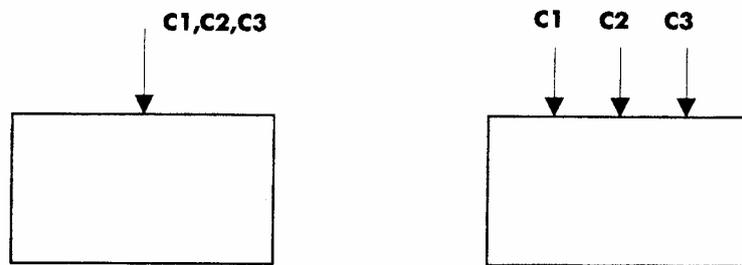


Рис. 6.2. Дуги с более чем одним ICOM—кодом.

- Пограничные дуги, имеющие одинаковые ICOM-коды, должны быть соединены.

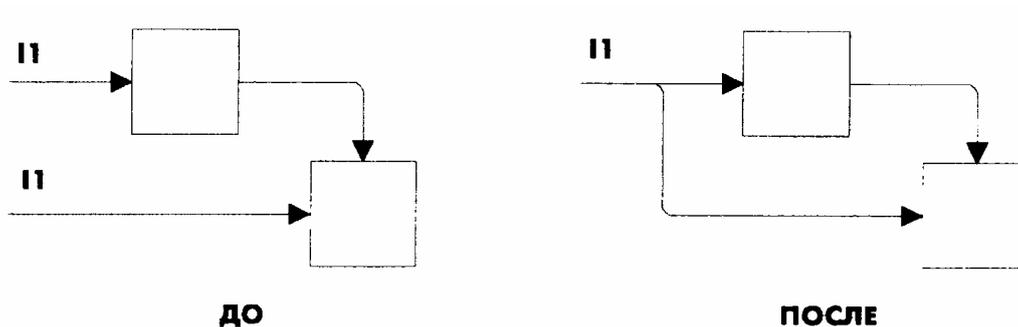


Рис. 6.3. Соединение граничных дуг.

- Имена дуг, которые изменяются при переходе от родительской диаграммы к диаграмме декомпозиции, но представляют те же данные, должны быть унифицированы по всей модели.

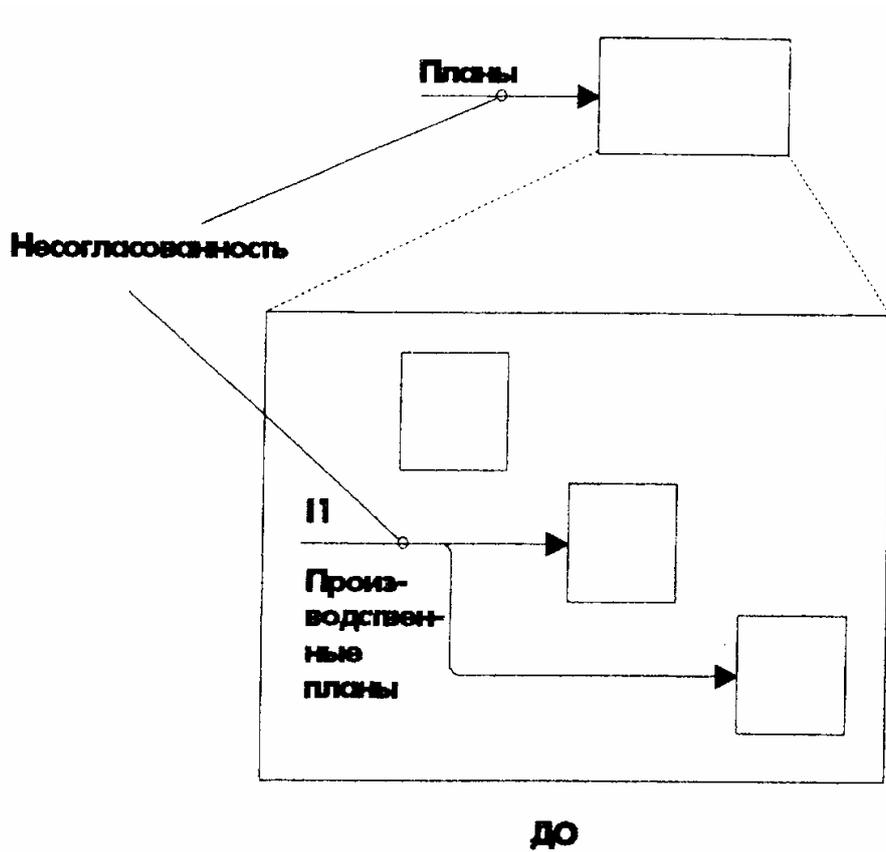


Рис. 6.4. Метки дуг остаются одинаковыми.

- Если дуги имеют одинаковые имена, но разный смысл, то автор должен обозначить их уникальными именами:

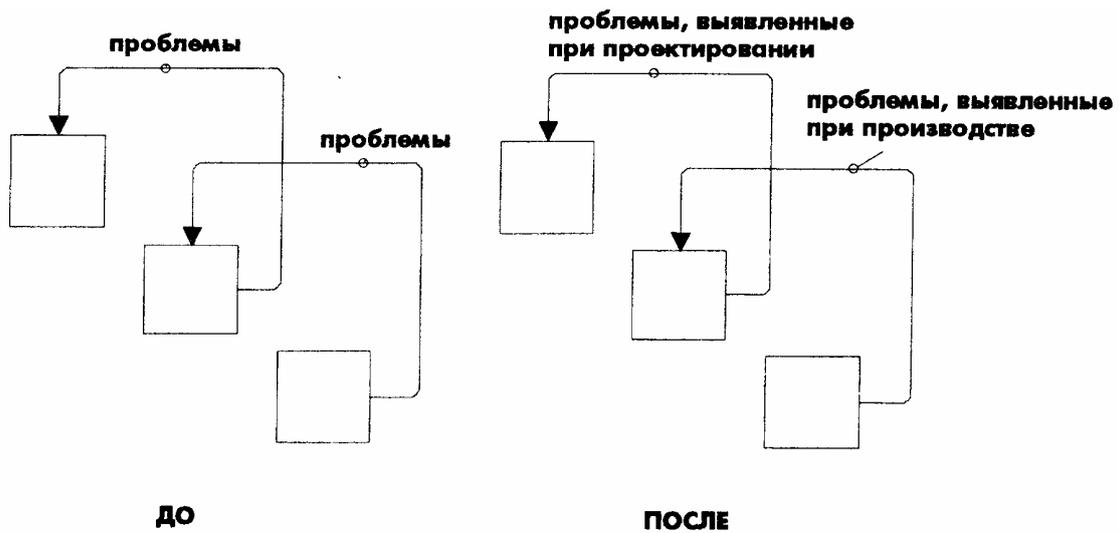


Рис. 6.5. Конкретизированные метки дуг.

- При ветвлении или соединении дуг автор должен дать имена всем неназванным дугам.

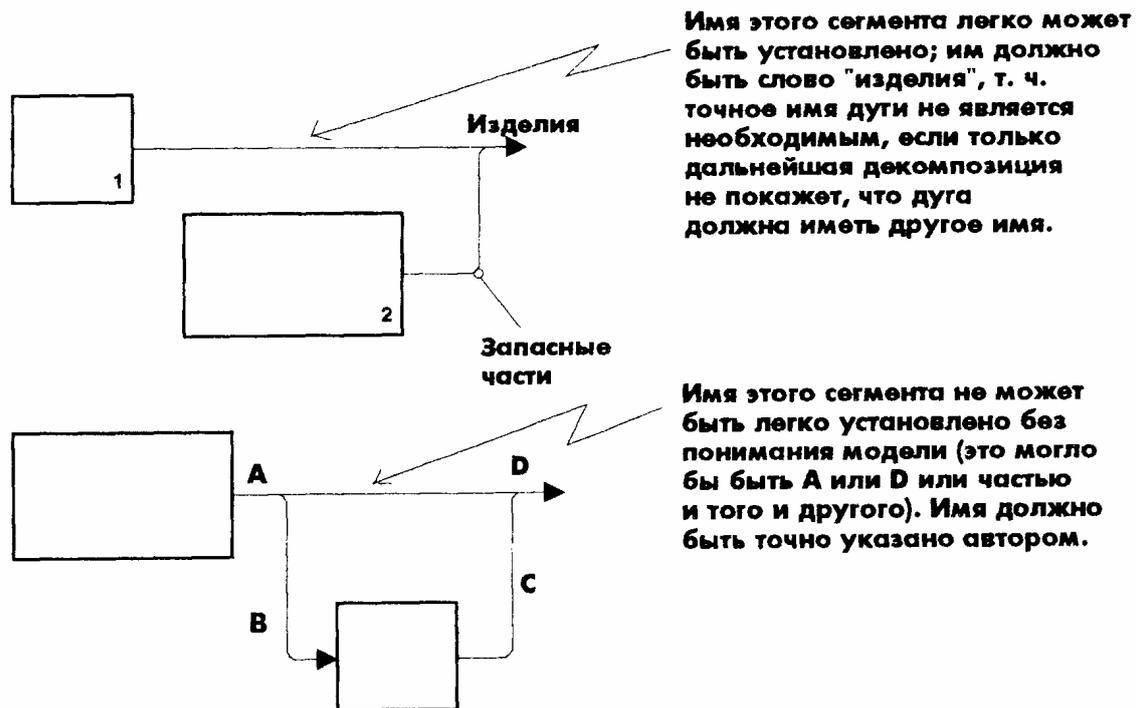


Рис. 6.6. Расщепление меток ветвящихся дуг.

### 6.6.3. Сопряжение и связность

Сопряжение описывает степень стыкуемости компонентов. Для IDEF0 можно описать число и типы взаимосвязей между функциональными блоками на одной диаграмме как взаимосоединения компонентов. Связность описывает степень объединенности отдельных подчастей внутри данной части модели. Таким образом, в IDEF0 связность - весьма значимое свойство для оценки того, насколько удачно сгруппированы функции на одной диаграмме.

Уместность этих показателей для оценки сложности заключается в следующем;

*Если компоненты диаграммы или модели имеют высокое сопряжение друг с другом, модель становится более сложной. Компоненты, составленные из несвязанных элементов, также усложняют модель.*

Таким образом, IDEF0-диаграммы должны становиться более понятны, если их компоненты не тесно состыкованы, но функционально зависимы.

### 6.6.4. Показатели, основанные на сопряжении и связности

Понимаемость модели становится выше, а сложность снижается по мере того, как связность компонентов возрастает, а их сопряжение уменьшается. Из этого следует, что показатели, основанные на различных типах сопряжения и связности, полезны для оценки понимаемости/сложности моделей.

Шкала или код для определения различных типов сопряжения и связности обеспечит авторов инструментом, помогающим создавать более понятные и менее сложные модели.

#### **6.6.4.1. Связь с другими аспектами системного проектирования**

Введение показателей для измерения различных типов сопряжения и связности IDEF-моделей дает дополнительные преимущества. В частности, легче перевести описание системы из состояния "как есть" в состояние "как будет", если сопряжение уменьшают, ограничивая число модульных связей так, чтобы по возможности меньше модулей прямо зависело от одних и тех же допущений.

Принцип "скрытия" в системном проектировании для той же ситуации заключается в том, что информация, известная только данному набору модулей, скрыта (или изолирована) от других модулей. Преимущество такого скрытия информации, как и уменьшаемого сопряжения, состоит в том, что локальные изменения в модели или проекте системы не повлекут за собой изменения такого количества модулей или компонентов, какое могло бы быть необходимо в другом случае. Таким образом, существует прямая связь между скрытием и сопряжением. Оценка степени сопряжения внутри отдельной модели позволяет получить информацию и о степени скрытия информации.

Как скрытие изоморфно сопряжению, так и локализация изоморфна связности. Локализация - это принцип сведения элементов вместе на основе их физической близости. Смысл этого принципа состоит в минимизации избыточности, которая может возникнуть, если одна и та же информация будет фиксироваться одновременно в нескольких местах. Например, рассмотрите потенциальный объем определений основных производственных подсистем в ICAM-архитектуре "как есть" (функциональная модель производства). Локализация этих функций, связанных с управлением производством и управлением материальными ресурсами, в отдельной функциональной модели, например, облегчит переход к тому, "как (им)- будет" ее окружение, так как соответствующие функции и данные должны быть установлены в функциональной и физической близости и могут, следовательно, рассматриваться как отдельный, но целостный (интегральный) блок. Когда ожидаются изменения в окружающей среде, то значительно труднее проследить все их возможные воздействия на функции, если функции распределены по всей архитектуре.

#### **6.6.4.2. Показатели и типы сопряжения**

Различия типов сопряжения будут обсуждаться в процессе рассмотрения связей между функциями в IDEF0.

##### **6.6.4.2.1. Точка зрения описания**

Традиционно в литературе по системному анализу сопряжения между функциями исследуют с двух точек зрения: 1) рассматривая природу данных в самой связи и 2) определяя отношения среди многих экземпляров или образцов данных в связях.

На рис. 6.7. показан такой анализ с двух точек зрения. Термины, используемые в каждой категории, встречаются в литературе и их техническое значение будет появляться на последующих страницах.

## Природа связи

## Структура связи

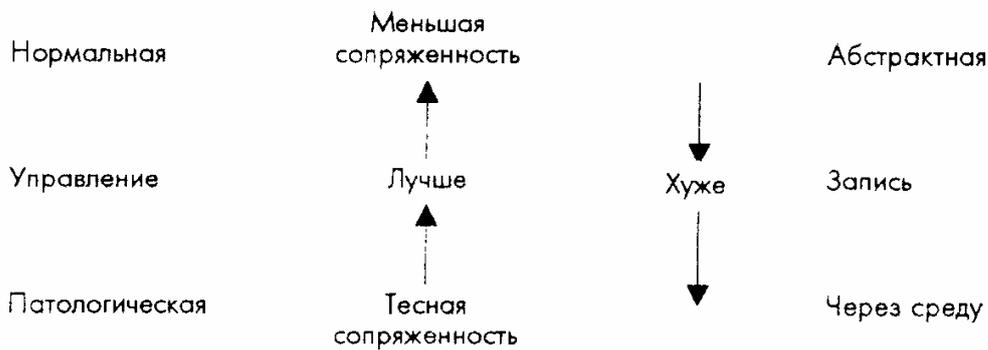


Рис. 6.7. Анализ сопряжения функций.

### 6.6.4.2.2. Природа связей

*Патологическая, связь:* наименее желательна.

О двух функциях говорят, что они имеют патологическую связь, если одна из них ссылается на содержание другой. Такая ситуация возникает, когда не соблюдается принцип скрытия. В качестве примера рассмотрите рис. 6.8.

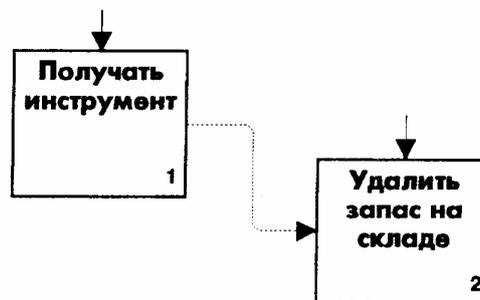


Рис. 6.8. Патологическая связь.

В соответствии с приведенным выше определением дуга данных, соединяющая блоки 1 и 2, является "патологической", а блоки - "содержательно-связанными". Ясно, что блок 2 удаляет те материалы, которые необходимы блоку 1 для работы, и связи такого типа обычно нежелательны, если другие факторы не требуют оговорок на этот счет.

#### *Управляющая связь*

Две функции являются "связанными по управлению", если данные от одной влияют на поток управления другой. Поток управления отличается от управляющей дуги блока. "Управление" в этом контексте относится к виду данных, но не к их положению относительно блока. Управляющая связь возникает тогда, когда функция генерирует управляющее воздействие на поведение другой функции. Этот вид связи может быть помечен записью "функция-имя", которая создает "флажки" на обозначенных функциях, такие, как "Зарегистрировать неудачу/напечатать сообщение об ошибке".

### Нормальная связь

Две функции являются "нормально" связанными, если связи по данным, существующие между ними, или независимы от содержания обеих функций, или если не существует влияния на поток управления от одной функции к другой. Этот тип связи наиболее желательный, как обмен данными между функциями осуществляется через ясно выраженные данные, а не через неявные ссылки к контекстам той или другой функции.

#### 6.6.4.2.3. Структура связи

##### Через среду

Две функции являются сопряженными через среду, если они имеют связи с источником, генерирующим данные, без конкретного определения отношения отдельной части данных к какому-либо блоку. Эта ситуация может быть выражена диаграммой, приведенной на рис. 6.9.

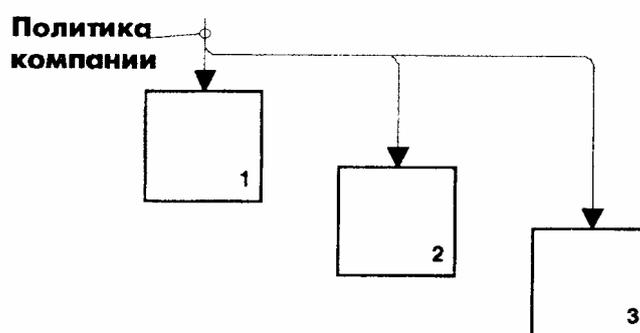


Рис. 6.9. Сопряжение через среду.

Проблемы, возникающие при сопряжении через среду, включают следующее:

1. Принцип скрытости нарушается. Доступ не ограничен только необходимыми данными.
2. Нарушается локализация. Данные связаны не только с их пользователями.
3. Распространение ошибок потенциально увеличивается. Изменения в среде могут непредсказуемо воздействовать на все функции.

##### Запись

Две или более функции являются сопряженными через запись, если они связаны с неглобальным набором данных и не обязательно зависят от того, представлены ли все возможные интерфейсы как сопряжение через среду. Пример для IDEF0 приведен на рис. 6.10.

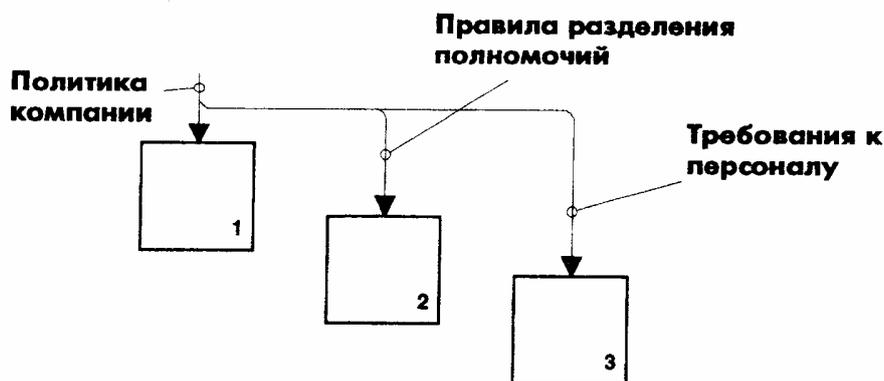


Рис. 6.10. Сопряжение через запись.

Тип интерфейса, показанный здесь, предпочтителен, поскольку определяются отношения конкретных элементов данных к каждому блоку.

#### Абстрактная структура

Два элемента имеют абстрактное сопряжение, если отношение между отдельным элементом и его потенциальными "составляющими" ясно определено. В IDEF0 дуга данных на родительской диаграмме, дробящаяся на отдельные компоненты на диаграмме декомпозиции, представляет собой связь "абстрактное сопряжение". Здесь проявляется скрытость настолько, насколько дуга исходного уровня использует силу абстракции, чтобы обеспечить достижение уровня абстракции, требуемого для более полного понимания контекстов данных. Понятие объединения дуг в IDEF0 соответствует использованию абстрактного сопряжения и должно максимально применяться при конструировании диаграмм.

#### 6.6.5. Показатели и типы связности

В то время как "сопряжение" относится к связи между элементами (функциям и/или данными), "связность" будет использоваться в отношении связывания или внутренней согласованности между объектами диаграммы.

В литературе различаются по крайней мере семь типов связывания.

Относительная значимость		
0	Случайная	<p>Хуже</p> <p>↑</p> <p>↓</p> <p>Лучше</p>
1	Логическая	
2	Временная	
3	Процедурная	
4	Коммуникационная	
5	Последовательная	
6	Функциональная	

Каждый тип будет кратко определен и проиллюстрирован с помощью типичного примера из IDEF0.

(0) *Тип случайной связности*: наименее желательный

Случайная связность возникает, когда конкретная связь между модулями мала или полностью отсутствует. Это относится к ситуации, когда имена данных на IDEF0-дугах в одной диаграмме имеют малую связь друг с другом. Крайний вариант этого случая показан на рис. 6.11.

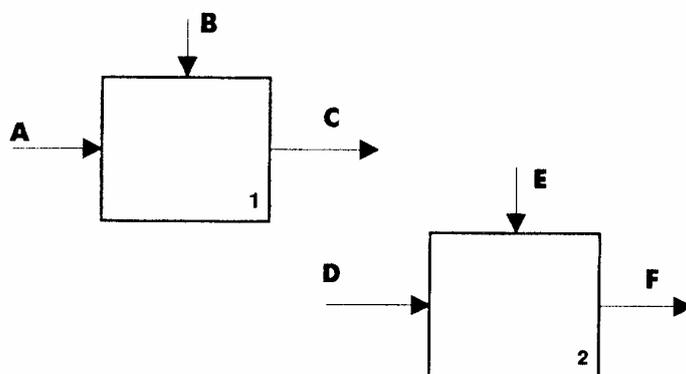


Рис. 6.11. Случайная связность.

(1) *Тип логической связности*

Логическое увязывание происходит тогда, когда данные и функции собираются вместе благодаря тому, что они попадают в общий класс или набор элементов, но необходимых функциональных отношений между ними не обнаруживается. На рис. 6.12. показан типичный случай логической связи.

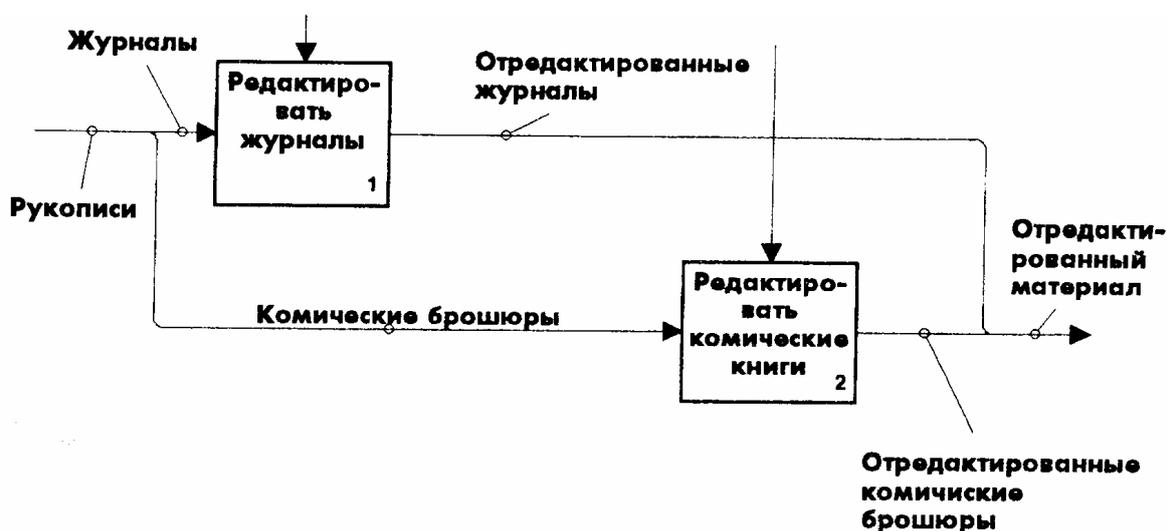


Рис. 6.12. Логическая связность.

(2) *Тип временной связности*

Связанные по времени элементы возникают благодаря тому, что они представляют функции, связанные во времени, когда данные используются одновременно или функции включаются параллельно, а не последовательно. Часто такой пример можно найти на диаграмме, показывающей функции, касающиеся операций инициализации (рис. 6.13.).

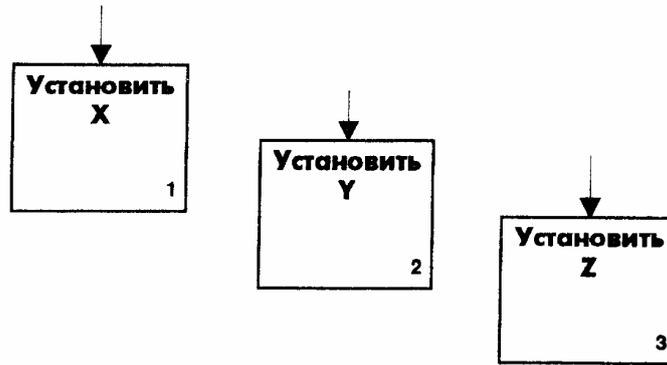


Рис. 6.13. Временная связность.

*(3) Тип процедурной связности*

Процедурно-связанные элементы появляются сгруппированными вместе благодаря тому, что они выполняются в течение одной и той же части цикла или процесса. Общий элемент процесса может быть итерационным или ветвью, зависящей от принятого решения, или линейной последовательностью этапов. Пример процедурно-связанной диаграммы приведен на рис. 6.14.

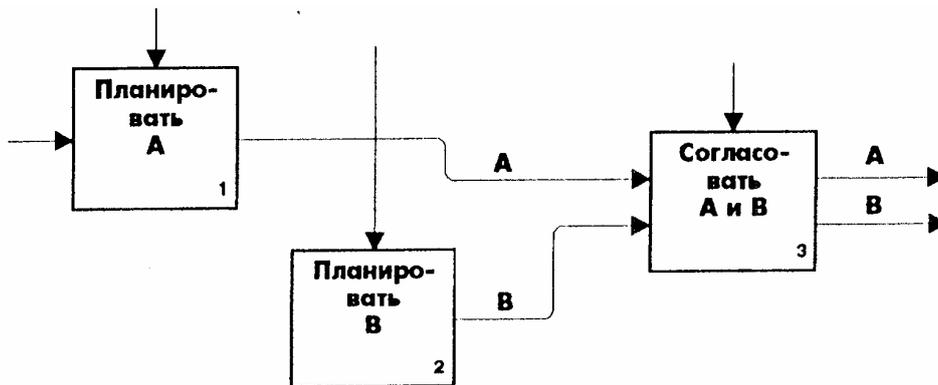


Рис. 6.14. Процедурная связность.

В этой диаграмме блоки собраны вместе, поскольку оба выхода (А и В) необходимы для выполнения последующей функции. Заметьте, что блоки 1 и 2 не влияют друг на друга и они, следовательно, объединены своим отношением к процессу в блоке 3.

*(4) Тип коммуникационной связности*

Диаграммы демонстрируют коммуникационные связи, когда блоки группируются благодаря тому, что они используют одни и те же входные данные и/или производят одни и те же выходные данные. Это первый тип связи, обсуждаемый далее, который представляет предпочтительный уровень связи для IDEF0. Важным доводом является то, что ни один из уровней связности, обсуждавшихся выше, не привязан достаточно тесно к конкретной проблеме. Коммуникационная связность является самым нижним уровнем, на котором мы

встречаем связи между функционирующими элементами, являющимися по своей сути проблемно-зависимыми.

На рис. 6.15. показан типичный случай.

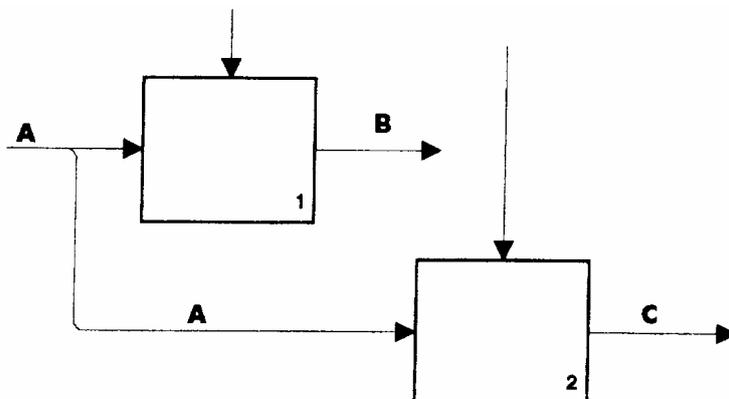


Рис. 6.15. Коммуникационная связность.

#### (5) Тип последовательной связности

На диаграммах, имеющих последовательные связи, выход из одной функции служит входными данными для следующей функции. Связь между элементами на диаграмме является более тесной, чем на рассмотренных выше уровнях связей, поскольку моделируются причинно-следственные зависимости. Следовательно, это высокая степень связности. На рис. 6.16. показана последовательная связность.

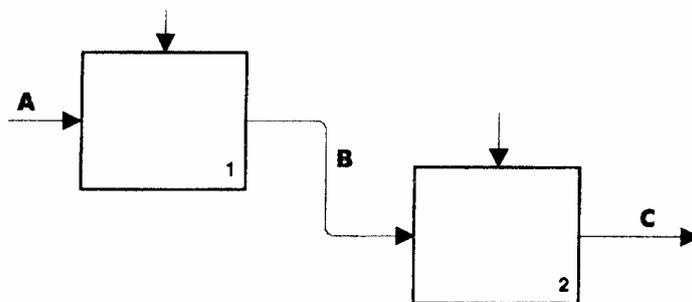


Рис. 6.16. Последовательная связность.

#### (6) Тип функциональной связности

Диаграмма отражает полную функциональную связность, когда все элементы функции влияют на выполнение одной единственной функции или результата. Диаграмма, которая является чисто функциональной, не содержит чужеродных элементов, относящихся к последовательному или более слабому типу связности. Одним из способов определения функционально-связанных диаграмм является рассмотрение двух блоков, связанных через управляющие дуги, как показано на рис. 6.17.

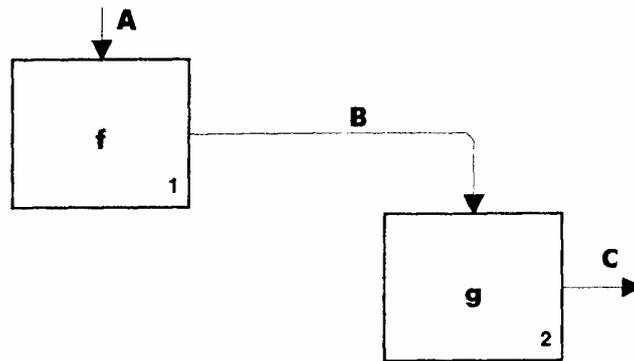


Рис. 6.17. Функциональная связность.

В математических терминах необходимое условие для простейшего типа функциональной связности, показанной на рис. 6-17, имеет следующий вид:

$$C = g(B) = g(f(A))$$

Другие методы определения функциональной связности основаны на критериях исключения. Приведенные ниже рекомендации помогут авторам и рецензентам отличать функциональные связности от нефункциональных.

- Если существует единственно приемлемый путь описания диаграмм с использованием сложного предложения, предложения, содержащего запятую, или предложения, содержащего более одного глагола, то менее вероятно, что диаграмма является функциональной. Она, видимо, последовательная, коммуникационная или логическая в терминах связности.
- Если описывающее предложение содержит такие временные ориентиры, как "первый", "следующий", "после", "затем", "начать", "когда", "до тех пор, пока" или "для всех", то диаграмма имеет временную или процедурную связность; реже такие слова обозначают последовательную связность.
- Если утверждение описывающего предложения не содержит одного конкретного объекта, следующего за глаголом, диаграмма является, вероятно, логически связанной.

### *Заключение*

На рис. 6.18. представлены все типы связей, рассмотренные в предыдущих разделах. Важно отметить, что уровни 4-6 устанавливают типы связностей, которые разработчики считают важнейшими для получения диаграмм хорошего качества. Авторы должны стремиться к максимальному количеству связей этих типов.

Значимость	Тип связности	Для функций	Для данных
0	Случайная	Случайная	Случайная
1	Логическая	функции одного и того же множества или типа (например, "редактировать все входы" )	Данные одного и того же множества или типа
2	Временная	Функции одного и того же периода времени (например, "операции инициализации")	Данные, используемые в каком-либо временном интервале
3	Процедурная	Функции, работающие в одной и той же фазе или итерации (например, "первый проход компилятора")	Данные, используемые во время одной и той же фазы или итерации
4	Коммуникационная	Функции, использующие одни и те же данные	Данные, на которые воздействует одна и та же деятельность
5	Последовательная	функции, выполняющие последовательные преобразования одних и тех же данных	Данные, преобразуемые последовательными функциями
6	Функциональная	функции, объединяемые для выполнения одной функции	Данные, связанные с одной функцией

## 7. Сбор информации для IDEF-модели

### 7.1. Вступление

При анализе или в процессе разработки любой системы может возникнуть необходимость в получении ответов на вопросы, появляющиеся по ходу дела, или проверке отдельных фактов о системе. Существует немало источников получения такой информации. Например, вы можете:

- Ознакомиться с имеющейся документацией, используя оглавления и предметные указатели.
- Изучить саму систему в действии, если она уже существует.
- Провести опрос большой группы людей с помощью вопросников или других способов сбора информации.
- Провести беседу с одним или более "экспертов", располагающих необходимой информацией,
- Использовать все сведения, которые уже известны автору.
- Придумать или изобрести гипотетический вариант и обратиться к читателям с просьбой помочь приблизить его к действительности.

Из всех этих методов наиболее важный - личное общение с экспертом, Крайне редко случается, что вся существующая информация по той или иной проблеме имеется в письменной форме. Составленные же заранее ответы на поставленные в вопроснике вопросы зачастую оказываются ошибочными.

Получение информации от эксперта осуществляется в форме *опроса*. Поэтапный процесс позволяет проводить опрос без риска оказать нежелательное давление на эксперта, что может иметь место, если проводящий опрос использует полученную ранее информацию,

Ключевым моментом опроса является регистрация полученной информации. Это может быть сделано путем неформальных записей, путем заполнения листов данными о выполненных операциях или таблиц соответствующих функций, а также путем создания эскизов диаграмм.

### 7.2. Процесс опроса

Целью опроса является получение информации от специалиста, обладающего знаниями, которые представляют интерес для аналитика. Существует четыре типа опросов, которые можно проводить на аналитической стадии разработки IDEF-проекта.

Четыре типа опросов

а) *Сбор фактов* для понимания текущей ситуации. Этот тип опроса используется для создания модели текущих операций, а также для лучшего понимания существующей обстановки.

б) *формулировка проблемы* для определения будущих требований. Этот тип опроса используется для проверки достоверности модели текущих операций и при создании модели будущих операций.

в) *Обсуждение решений*, касающихся возможностей будущих систем. Этот тип опроса используется при создании модели будущих операций.

г) *Встреча IDEF-автора и заинтересованных лиц*. Этот тип опроса используется для разрешения проблем, которые возникли в ходе создания IDEF-модели.

Причина идентификации типа опроса объясняется тем, что в ходе проведения опроса возникает необходимость использовать схемы различных типов опросов. Респондент может

сообщить лицу, проводящему опрос, факты о данной системе с точки зрения проблемы в целом. Респондент может также рассматривать проблему с точки зрения ее решения. Тем самым человек, проводящий опрос, постоянно классифицируя ответы респондента, может получить в ходе опроса максимум полезной информации.

### 7.3. Папка опроса

Рекомендуется использовать для записи данных опроса "стандартную" папку опроса. Она может храниться в файле опроса и распределяться среди соответствующих лиц. Такое распределение может распространяться и на других членов группы аналитиков и даже на самих респондентов для внесения ими поправок, дополнений и изменений. В папку опроса входят:

1. Обложка (обложка папки);
2. Опрос и его запись:
  - а) *Имя лица, проводящего опрос, (имя IDEF-автора);*
  - б) *Дата опроса (дата создания IDEF-диаграммы);*
  - в) *Продолжительность опроса (время его начала и конца);*
  - г) *Имя респондента;*
  - д) *Звание респондента и его должность;*
  - е) *Номер телефона и добавочный номер;*
  - ж) *Дополнительные источники получения информации:*
    - 1). *Документы - название и месторасположение;*
    - 2). *Другие респонденты - имя, звание, должность, адрес, номер телефона.*
  - з) *Существенные элементы информации - резюме по ключевым проблемам, обсуждаемым в ходе опроса;*
    - и) *Дополнительные вопросы и те важные проблемы, которые либо не были затронуты во время опроса, либо были отложены;*
  - к) *Новая терминология для глоссария проекта:*
    - 1). *Список операций и полученных данных;*
    - 2). *Порядок проведения опроса (разрабатывается в ходе подготовки к опросу - подробнее в последующем разделе);*
    - 3). *Записи и наброски диаграмм, сделанные в ходе опроса.*

### 7.4 Правила проведения опроса

Проведение опроса разбивается на пять стадий, каждая из которых рассчитана на получение максимального количества информации при затрате минимального времени на запись результатов:

- Подготовка
- Вводная стадия
- Проведение опроса
- Окончание опроса
- финальная стадия опроса

Каждая стадия требует определенных действий, которые в той или иной мере должны быть выполнены в ходе опроса. При этом на каждой стадии следует прибегать и к средствам психологического воздействия, которые призваны помочь установлению атмосферы профессионализма и доверия с респондентом.

### 7.4.1 Подготовка к опросу

Тщательно продумав ключевые положения предстоящего опроса, вы можете добиться, тем самым, более организованного и эффективного диалога. На стадии подготовки опроса следует выполнить следующие действия:

1. Отобрать респондентов

*а) Из наиболее важных подразделений;*

*б) Из кандидатур, рекомендуемых другими респондентами;*

*в) Из разных уровней организационных структур: высший уровень - источник "общей картины процесса", низший уровень - источник детализированной информации, а средний - ликвидирует разрыв между ними,*

2. Договориться о встрече

*а) Она должна быть непродолжительной - от получаса до часа;*

*б) Ее не следует проводить непосредственно перед обедом или перед окончанием рабочего дня;*

*в) Определите цель опроса;*

*г) Объясните роль того, кто проводит опрос,*

3. Предварительно сформулировать предполагаемые вопросы:

*а) Касающиеся главных сфер производства - они используются как основа опроса и помогают определить "широкий спектр общих вопросов";*

*б) Специфические вопросы,*

4. Ознакомиться с уже имеющейся информацией,

5. Ознакомиться с соответствующей терминологией.

6. Обеспечить координацию с данными других опросов.

Проверьте занесенные в файлы данные других опросов с тем, чтобы убедиться, участвовал или не участвовал респондент ранее в опросах. Если участвовал, то ознакомьтесь с результатами предыдущего опроса.

7. Выпишите из записей опросов нужную вам информацию.

8. Составьте план опроса.

### 7.4.2. Вводная стадия

Эта стадия имеет целью установление взаимопонимания между лицом, проводящим опрос, и респондентом. Процедура знакомства перед началом опроса, как правило, весьма непродолжительна. Это время очень важно использовать для того, чтобы объяснить респонденту мотивы, побудившие прибегнуть к его помощи.

На этой стадии опроса следует придерживаться следующих правил:

1. Приготовьте для знакомства с респондентом какой-нибудь официальный документ, лучше всего визитную карточку (это устранит для респондента сложности, связанные с написанием и произношением вашей фамилии и исключит тем самым повод для возможных затруднений).

2. Объясните цель опроса:

*а) Изложите информацию, полученную при первоначальном контакте;*

б) Объясните вашу точку зрения на данный опрос. За основу возьмите 1, 2, 3 или 4 тип опроса;

в) Объясните цель опроса, даже если он является повторным.

3. Получите согласие на запись беседы. Респондент может потребовать сохранения конфиденциальности,
4. Установите отношения доверия между экспертом и автором -устраните опасения, что опрос будет использован для того, чтобы объяснить респонденту, как ему следует работать, или что его работа находится на грани срыва.
5. Начните опрос с общих вопросов, которые помогут "разговорить" респондента, - эти вопросы должны основываться на тематических проблемам, обозначенных в разработанной заранее программе.
6. Оцените возможности респондента предоставить нужную информацию - если его информация носит чрезмерно общий характер или, наоборот, чересчур детальный для той стадии, на которой находится подготовка IDEF-модели, пересмотрите возможности сотрудничества с респондентом. При необходимости прервите опрос - он может стать потерей времени как для вас, так и для респондента.
7. Приступайте к конкретным вопросам из программы опроса.
8. Пишите, как можно меньше говорите.

### 7.4.3. Проведение опроса

Нецелесообразно определять вопросы непосредственно в ходе опроса, однако представляется вполне возможным наметить те основные принципы, которыми следует руководствоваться при его проведении. Первый набор таких принципов касается качества получаемой информации. Второй набор охватывает проблемы, связанные со стимуляцией информационного потока.

*Качество информации.* Человеческий разум устроен таким образом, что он по-разному воспринимает процесс разговора. Опасность в ходе опроса заключается в том, что такой подход вынуждает респондента думать не только о том, что он говорит, но и о том, какого от него ждут ответа.

С тем чтобы помочь лицу, проводящему опрос, уяснить смысл сказанного, следует прибегнуть к набору вопросов, которые позволят ему воспринимать нужным образом представляемую информацию:

- Какие факты были приведены в поддержку основных пунктов обсуждаемой проблемы?
- Насколько нова данная информация?
- Насколько она исчерпывающа?
- Действительно ли я понял смысл сказанного?
- Отвечает ли моим целям уровень детализации в представленной информации ?
- Не упущено ли что-то?
- Была ли данная информация предварительно обсуждена с кем-либо другим?
- Какую ценность представляет данная информация?
- В достаточной ли мере обсуждены дополнительные аспекты?
- Изменилась ли моя точка зрения после проведения опроса?

*Как стимулировать информационный поток.* Следующий набор руководящих принципов может быть использован для того, чтобы стимулировать респондента на предоставление максимального количества надежной информации.

- Сократите до минимума не относящиеся к делу замечания и разговоры. Цель опроса - получение информации, а не завязывание дружеских отношений и изложение своих взглядов.

- Не пропустите ситуацию, когда респондент оказывается неспособен четко определить сложности, возникающие при решении обсуждаемой проблемы. Это может быть признаком того, что респонденту трудно найти общий язык с лицом, проводящим опрос.
- Предоставьте респонденту время обдумать вопрос. Не предлагайте своих ответов и не задавайте дополнительных вопросов, Пауза в ходе опроса весьма полезна, она позволит респонденту восстановить в памяти важные аспекты информации,
- Постарайтесь избежать каких-либо внешних отвлекающих моментов, которые могут нарушить ход мыслей респондента. Если возможно, проводите опрос вне естественной среды респондента.
- Обратите внимание на отвлекающие моменты в душевном состоянии респондента, на признаки того, что он испытывает беспокойство или не может найти общий язык с лицом, ведущим опрос.
- Постарайтесь определить, является ли полученная информация фактом или лишь мнением респондента.
- Помогайте более глубокому осмыслению вопроса, предложив респонденту перефразировать ответ или изложить резюме предоставленной информации.
- Выясните степень подготовки респондента и его участия в процессе, который стал предметом опроса. Зная его отношение к организации и существующим системам, можно придти к более глубокому пониманию ответов респондента.
- Не прибегайте к сарказму и юмору.
- Не упоминайте и не обсуждайте результаты опросов, проведенных с другими респондентами.
- Фиксируйте все вопросы, которые задает респондент. Лицо, проводящее опрос, обязано ответить на все заданные вопросы, за исключением тех, которые касаются клиента, - руководства организации, планов, отдельных личностей.
- Проявляйте интерес к тому, что говорит респондент.
- Сосредоточьте внимание на неизвестных и сложных аспектах обсуждаемой проблемы. Избегайте общих мест.
- Будьте внимательны к несообразному или неправильному употреблению слов. Для уточнения незнакомого или спорного термина попросите респондента дать его точное значение. Внесите это значение в глоссарий проекта.
- Не противоречьте респонденту даже в том случае, когда факты не подтверждают им сказанного. Для разрешения таких конфликтов используйте цикл папки.
- Будьте скромны в своем поведении. Экспертом является респондент, а не лицо, проводящее опрос.
- Отложите проблемы, обсудить которые в полной мере вы не успели на другое, заранее отведенное для опроса время. Не пытайтесь продлить опрос, лучше договоритесь о новой встрече.
- Принимайте во внимание разные мнения по одному и тому же вопросу. Используйте IDEF-модели для демонстрации этих мнений и разрешения конфликта.
- Стимулируйте респондента, задавая четкие, законченные, относящиеся к делу вопросы.

#### 7.4.4. Окончание опроса

Опрос следует закончить при наличии любой из четырех следующих причин;

*а) Информация, получаемая в ходе опроса, не отвечает поставленным требованиям;*

*б) Исчерпан лимит времени;*

*в) Лицо, проводящее опрос, получило в полном объеме необходимую информацию;*

*г) Между тем, кто проводит опрос, и респондентом возникла конфликтная ситуация.*

В зависимости от причины, диктующей необходимость завершения опроса, следует придерживаться следующих правил при его окончании:

- Опрос не следует прекращать внезапно, в конце его следует провести несколько минут за неофициальным разговором.
- Следует подытожить основные положения опроса.
- Следует выделить те важные проблемы, которые были либо отложены, либо не были вовсе затронуты в ходе опроса.
- В случае необходимости следует договориться о дополнительной встрече.
- Следует попросить респондента порекомендовать других людей для проведения опроса.
- Если записи, сделанные в ходе опроса, подлежат просмотру респондентом до их распространения, следует оговорить это обстоятельство перед окончанием опроса.
- Следует поблагодарить респондента за потраченные им время и усилия.

#### 7.4.5 Финальная стадия

Эта стадия опроса наступает после того, как вы убедитесь в том, что полученная в ходе опроса информация должным образом зафиксирована и доведена до сведения всех членов группы проекта. Основным средством, используемым для завершения финальной стадии, является папка опроса. Если респондент высказал несогласие со сделанными в ходе опроса заметками, лицу, проводящему опрос, следует по его окончании немедленно изложить наиболее существенные пункты, обсуждавшиеся в ходе опроса. На финальной стадии опроса следует сделать следующее:

- а) Определить дополнительные источники информации;*
- б) Подытожить основные элементы полученной информации;*
- в) Выделить новые термины для глоссария проекта;*
- г) Перечислить те дополнительные вопросы и важные проблемы, которые были либо отложены, либо не были вовсе затронуты в ходе опроса;*
- д) Составить список операций и полученных данных;*
- е) Дополнить записи, включив в них ту информацию, которая припомнилась вам при их просмотре;*
- ж) Приготовить черновые IDEF-диаграммы, отражающие полученную информацию;*
- з) Выделить сделанные в ходе опроса предположения или спорные вопросы;*
- и) Напечатать материалы для папки опроса и распространить ее;*
- к) Указать имена, место работы экспертов, номера телефонов и адреса лиц, упомянутых в ходе опроса.*