

Перевод стандарта IDEF0.

**Федеральный стандарт  
по обработке информации  
Публикация 183**

**21 Декабря 1993**

**Объявление стандарта**

## **МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ (IDEF0)**

Федеральные стандарты обработки информации (FIPS PUBS) публикуются Национальным институтом стандартов и технологий после утверждения министром торговли в соответствии с разделом 111 (d) Закона о федеральной собственности и административных услугах 1949 года с поправками, внесенными Законом о компьютерной безопасности 1987 года, Публичный закон 100-235.

**1. Название стандарта.** Методология функционального моделирования (IDEF0).

**2. Категория стандарта.** Стандарт программного обеспечения, методы моделирования.

**3. Пояснение.**

Данная публикация объявляет о принятии Методологии функционального моделирования (IDEF0) в качестве Федерального стандарта обработки информации (FIPS). Настоящий стандарт основан на программе интегрированной компьютеризации производства научно-исследовательских лабораторий BBC Райт (ICAM), Часть II, том IV

Руководство по функциональному моделированию (IDEF0), Июнь 1981 года.

Настоящий стандарт описывает язык моделирования IDEF0 (семантика и синтаксис), а также связанные с ним правила и методы разработки структурированных графических представлений системы или предприятия. Использование стандарта позволяет создавать модели, описывающие системные функции (действия, поведение, процессы, операции), функциональные связи и данные (информация или объекты), которые поддерживают интеграцию систем.

Этот стандарт является информационно-справочным документом, предназначенным для системных или корпоративных разработчиков моделей, применяющих IDEF0 для определения инструментов при реализации этой методики и других компьютерных специалистов для понимания точных синтаксических и семантических правил стандарта.

**4. Утверждающий орган.** Министр торговли.

**5. Агентство по техническому обслуживанию.**

## **6. Другие документы**

а. Архитектура ICAM, Часть II-том IV Руководство по функциональному моделированию (IDEF0), AFWAL-TR-81-4023, Лаборатория материалов, Авиационные лаборатории BBC Райт, Командование BBC, База BBC Райт-Паттерсон, Огайо 45433, июнь 1981 г.

## **7. Сопутствующая документация**

а. Федеральные правила управления информационными ресурсами, подраздел 201.20.303 «Стандарты» и подраздел 201.39.1002 «Федеральные стандарты».

б. Интегрированная система информационной поддержки (IIS), том V Подсистема общей модели данных, Часть 4 Руководство по информационному моделированию Idef1 Дополненное, декабрь 1985 г.

с. Архитектура ICAM, часть II, том V Руководство по информационному моделированию (IDEF1), AFWAL-TR-81-4023, Лаборатория материалов, Авиационные лаборатории BBC Райт, Командование BBC, База BBC Райт-Паттерсон, Огайо 45433, июнь 1981 г.

д. Управление конфигурацией ICAM, том II Стандарты документации

ICAM при разработке систем (SDM), AFWAL-TR-82-4157, Командование BBC, Авиабаза Райт-Паттерсон, Огайо 45433, октябрь 1983 г.

**8. Цели.** Основными целями настоящего стандарта являются:

а. Предоставить средства для полного и последовательного моделирования функций (действий, поведения, процессов, операций), требуемых системой или предприятием, а также функциональных связей и данных (информации или объектов), поддерживающих интеграцию этих функций;

б. Предоставить методику моделирования, не зависящую от методов или инструментов компьютерной программной инженерии (CASE), но которая может быть использована в сочетании с этими методами или инструментами;

с. Предоставить методику моделирования, обладающую следующими характеристиками:

— Универсальность (для анализа систем различного назначения, сферы применения и сложности);

— Определенность и точность (для производства требуемых, пригодных для использования моделей);

— Краткость и лаконичность (для облегчения понимания, информационного обмена, согласования и проверки);

— Концептуальность (для представления функциональных требований, а не физических или организационных вариантов);

— Гибкость (поддержка нескольких этапов жизненного цикла проекта).

## **9. Применение**

Использование данного стандарта настоятельно рекомендуется если в ходе реализации проектов требуется:

а. применение методики моделирования для анализа, разработки, реинжиниринга, интеграции или приобретения информационных систем;

б. включение метода системного или корпоративного моделирования в методологию анализа бизнес-процессов или разработки программного обеспечения.

Спецификации настоящего стандарта применяются в тех случаях, когда методы системного или корпоративного моделирования используются:

а. в проектах, требующих IDEF0 в качестве метода моделирования;

б. при разработке автоматизированных программных средств, реализующих методику моделирования IDEF0.

Спецификации настоящего стандарта не применяются в проектах, которые требуют использования методов моделирования функций, отличных от IDEF0.

Нестандартные функции метода IDEF0 использовать в том случае,

если необходимая операция или функция не могут быть корректно реализованы с помощью стандартных функций. При этом нестандартные функции могут быть очень полезны. Следует отметить, что использование этих или любых других нестандартных функций может сделать интеграцию моделей более трудной и дорогостоящей.

## **10. Технические характеристики**

Настоящий стандарт принимает Методологию функционального моделирования (IDEF0) в качестве Федерального стандарта обработки информации (FIPS).

## **11. Реализация.**

Реализация стандарта включает в себя два момента: приобретение разработки и интерпретация стандарта.

### **11.1 Приобретение пакета IDEF0.**

Настоящая публикация (FIPS 183) вступает в силу 30 июня 1994 года. Приобретение продукта после этой даты федеральными структурами, разрабатывающими проекты с помощью методологии функционального моделирования IDEF0 или программного обеспечение, реализующего метод моделирования IDEF0, должны соответствовать требованиям FIPS 183.

Соответствие настоящему стандарту должно учитываться независимо от того, приобретается ли проект или программное обеспечение, использующее метод моделирования IDEF0, в рамках закупки системы ADP, или приобретается отдельными закупками, используется ли оно в рамках лизингового соглашения ADP или указано для использования в контрактах на услуги программирования.

Переходный период предоставляет промышленности время для разработки продукции, соответствующей этому стандарту.

Переходный период начинается с даты вступления в силу и продолжается в течение одного (1) года после этого. Положения настоящей публикации применяются к заказам, размещенным после даты настоящей публикации; однако использование метода функционального моделирования, который не соответствует настоящему стандарту, может быть разрешено в течение переходного периода.

### **11.2 Интерпретация настоящего Федерального стандарта обработки информации (FIPS).**

Национальный институт по стандартизации и технологии предусматривает разрешение вопросов, касающихся реализации и применимости настоящего Федерального стандарта обработки

информации. Все вопросы, касающиеся толкования настоящего стандарта, должны быть адресованы:

Директору Лаборатории компьютерных систем

ВНИМАНИЕ: интерпретация FIPS IDEF0

Национальный институт по стандартизации и технологии

Гейтерсбург, Мэриленд 20899

## **12 Отказ от выполнения требований Стандарта.**

При определенных исключительных обстоятельствах руководители федеральных ведомств и агенств могут утверждать отказы от федеральных стандартов обработки информации (FIPS).

Руководители таких учреждений могут передавать данные полномочия только старшему должностному лицу, назначенному в соответствии с разделом 3506 (b) раздела 44 Кодекса Соединенных Штатов. Запросы об отказе удовлетворяются только в том случае, если:

а. Применение стандарта отрицательно скажется на выполнении задачи оператора Федеральной вычислительной системы

б. Соблюдение стандарта может привести к серьезным неблагоприятным финансовым последствиям для оператора, которые не будут компенсированы за счет Государственного бюджета.

Руководители агентства могут утверждать запросы об отказе только на основании письменного решения, в котором разъясняется причина, по которой руководитель агентства сделал данный вывод(ы). Копия каждого такого решения с четко обозначенными причинами или конфиденциальными данными о закупках направляется в адрес:

Директору Лаборатории компьютерных систем

ВНИМАНИЕ: интерпретация FIPS IDEF0

Национальный институт по стандартизации и технологии

Гейтерсбург, Мэриленд 20899

Кроме того, уведомление о каждом утвержденном отказе и делегировании полномочий по утверждению отказов незамедлительно направляется в Комитет по правительственной деятельности Палаты представителей, в Комитет по правительственным делам Сената и публикуется в Федеральном реестре.

Если определение об отказе применяется к закупке оборудования и / или осуществлению услуг, уведомление о решении выдать отказе должно быть опубликовано в Commerce Business Daily как часть уведомления о предложении и приобретении или, если отказ принимается после публикации этого уведомления путем внесения изменений в такое

уведомление. Копия запроса об отказе, любые подтверждающие документы, документ, подтверждающий запрос об отказе, а также подтверждающие и сопроводительные документы с исключениями, которые агентство уполномочено решает оформляются в соответствии с пунктом 5 раздела 552 (b) U. S. C. и являются частью закупочной документации и хранятся агентством.

### **13. Где можно приобрести данный стандарт.**

Экземпляры этой публикации продаются Национальной службой технической информации Министерства торговли США, Спрингфилд, Вирджиния 22161. При оформлении заказа укажите Публикацию 183 «Федеральный стандарт обработки информации (FIPSPUB 183) и название. Оплата может быть произведена чеком, денежным переводом или депозитным счетом.

#### **Вступление:**

В информативном введении рассматриваются предыстория и принцип IDEF0 (произносится как Idef zero). Это помогает читателю ориентироваться в содержании документа и быстрее понять требования нормативных разделов.

Настоящий стандарт состоит из нормативных и информационных разделов. Соблюдение нормативных разделов (Разделы 1-3) обязательно. В информационных разделах (Приложения A D) содержатся дополнительные предложения и рекомендации. Соответствие информационным разделам не является обязательным для соблюдения стандарта, если только такое соответствие не предусмотрено организацией, принимающей данный стандарт.

#### **Предыстория:**

В течение 1970-х годов Программа ВВС США по интегрированной автоматизации производству (ICAM) была направлена на повышение эффективности производства за счет систематического применения компьютерных технологий. Программа ICAM выявила потребность в улучшении методов анализа и коммуникации людей, участвующих в повышении производительности труда на производстве.

В ходе реализации программы ICAM был разработан ряд методов, известных как методы IDEF (iCAM Definition), которые включали в себя следующее:

1. Метод IDEF0, используется для создания „функциональной модели“. Функциональная модель это структурированное представление функций, действий или процессов в моделируемой системе или объекте.
2. Метод IDEF1, используется для создания „информационной модели“. Информационная модель представляет собой структуру и семантику информации внутри моделируемой системы или объекта.
3. Метод IDEF2, используется для создания „динамической модели“. Динамическая модель представляет собой изменяющиеся во времени поведенческие характеристики моделируемой системы или объекта.

В 1983 году программа Интегрированной системы информационной поддержки BBC США усовершенствовала методологию моделирования информации IDEF1, сформировав методологию семантического моделирования данных IDEF1X (расширенный IDEF1).

В настоящее время методологии IDEF0 и IDEF1X широко используются в государственном, промышленном и коммерческом секторах, поддерживая моделирование для широкого круга предприятий и областей применения.

В 1991 году Национальный институт стандартизации и технологии (NIST) получил поддержку от Министерства обороны США, Департамента корпоративного управления информацией (DoD/CIM) для разработки одного или нескольких Федеральных стандартов обработки информации (FIPS) определяющих методологии моделирования. Для функционального моделирования определена методология IDEF0, а для информационного моделирования IDEF1X. Документы FIPS базируются на руководствах IDEF, опубликованных BBC США в начале 1980-х годов.

Подход IDEF0

Методология IDEF0 основана на подходе, разработанном Дугласом Т. Россом в начале 70–х годов и получившем название SADT (Structured Analysis & Design Technique метод структурного анализа и проектирования). В своем первоначальном виде IDEF0 включал в себя как определение языка графического моделирования (синтаксис и семантика), так и описание комплексной методологии разработки моделей.

Методология IDEF0 может быть использована для моделирования широкого спектра автоматизированных и неавтоматизированных систем. Для новых систем IDEF0 может использоваться сначала для определения требований и характеристик функций, а затем для разработки реализации, которая отвечает требованиям и выполняет назначенные функции. Для существующих систем IDEF0 может использоваться для анализа функций, выполняемых системой и регистрации механизмов (средств), с помощью которых они выполняются.

Результатом применения IDEF0 к системе является модель, которая состоит из иерархического ряда диаграмм, текста и глоссария, имеющих перекрестные ссылки друг на друга. Двумя основными компонентами моделирования являются функции (представленные на диаграмме блоками) и данные и объекты, которые связывают эти функции (представлены стрелками).

Как язык функционального моделирования, IDEF0 имеет следующие характеристики:

1. Является всеобъемлющим и выразительным методом, способным графически представлять широкий спектр деловых, производственных и других видов деятельности предприятия на любом уровне детализации.
2. Это последовательный и простой язык, обеспечивающий строгое и точное выражение и способствующий согласованности использования и интерпретации.
3. Улучшает обмен информацией между системными аналитиками, разработчиками и пользователями за счет простоты обучения и делает упор на иерархическую детализацию.
4. Хорошо испытан и проверена в ходе многолетнего использования в Военно-воздушных силах и других правительственных проектах, а также в частном промышленном секторе.



5. Она может быть создана с помощью различных инструментов компьютерной графики; многочисленные коммерческие продукты специально поддерживают разработку и анализ диаграмм и моделей IDEF0.

Помимо определения языка IDEF0, методология IDEF0 также предусматривает процедуры и методы разработки и интерпретации моделей, в том числе для сбора данных, построения диаграмм, циклов обзора и документации. Материалы, относящиеся исключительно к процедурам моделирования, представлены в информационных приложениях к настоящему документу.

## СОДЕРЖАНИЕ

1. Общие сведения
2. Область применения
3. ЦЕЛЬ
4. Определения
5. Модели IDEF0
6. Концепции моделей
7. Синтаксис и семантика
8. Синтаксис
9. Блоки
10. Стрелки
11. Синтаксические правила
12. Семантика
13. Семантика блоков и стрелок.
14. Имена и метки
15. Семантические правила блоков и стрелок.
16. Диаграммы IDEF0
17. Типы диаграмм
18. Контекстная диаграмма верхнего уровня
19. Дочерняя диаграмма
20. Родительская диаграмма
21. Текст и глоссарий
22. Диаграммы-иллюстрации
23. Свойства диаграмм
24. Стрелки как ограничения
25. Активации блока
26. Параллельное функционирование
27. Стрелки как трубы
28. Ветвление стрелок
29. Межблочные соединения

- 30. Граничные стрелки
- 31. ISOM кодирование граничных стрелок
- 32. Стрелки туннелирования
- 33. Стрелка вызова
- 34. Правила построения диаграмм
- 35. Ссылочные выражения (коды)
- 36. Номера блоков
- 37. Номера ноды
- 38. Перечень нод
- 39. Дерево нод
- 40. Ссылки на ноды
- 41. Примечания к модели
- 42. Справочная запись
- 43. Модели
- 44. Описание модели IDEF0
- 45. Контекстные диаграммы
- 46. Контекстные диаграммы высокого уровня
- 47. FEO, текст и глоссарий
- 48. Название модели
- 49. Правила презентации

## ПРИЛОЖЕНИЕ А-КОНЦЕПЦИИ IDEF0

### 1 Предыстория

### 2 Концепции IDEF0

### 3 Обсуждение отдельных концепций IDEF0

#### 3.1 Графика моделирования деятельности

#### 3.2 Обмен информацией путем последовательного введения деталей

#### 3.3 Дисциплинированная командная работа

## ПРИЛОЖЕНИЕ В ПОСТРОЕНИЕ И ИНТЕРПРЕТАЦИЯ ДИАГРАММ IDEF0

## В. 1 Чтение диаграмм IDEF0

### В. 1.1 Подход к разработке модели

### В. 1.2 Порядок чтения диаграммы

### В. 1.3 Семантика блоков и стрелок

#### В.1.3.1 Ограничения не учитываются как и когда

#### В.1.3.2 Несколько вводов, сигналов контроля и выводов

## В. 2 Руководство разработчика по созданию диаграмм IDEF0

### В. 2.1 Основные шаги разработчика

#### В.2.1.1 Выбор контекста, точки зрения и цели

#### В.2.1.2 Создание контекстной диаграммы

#### В.2.1.3 Создание высшей диаграммы

#### В.2.1.4 Создание дочерних диаграмм

#### В.2.1.5 Создание вспомогательного материала

#### В.2.1.6 Выбор блока для детализации

#### В.2.1.7 Деятельность разработчика

В.2.1.7.1 Этап сбора данных

В.2.1.7.2 Этап структурирования

В.2.1.7.3 Этап презентации

В.2.1.7.4 Этап взаимодействия

В.2.2 Черчение диаграмм IDEF0

В.2.2.1 Генерирование функциональных блоков

В.2.2.1 Генерирование функциональных блоков

В.2.2.3 Уровень усилий

В.2.3 Повторное построение диаграмм IDEF0

В.2.3.1 Модифицирующие блоки

В.2.3.2 Связывание (объединение) стрелок

В.2.3.3 Предложение о внесении изменений в контекст

В.2.3.4 Синтаксис ICOM для подключения диаграмм

В.2.4 Графический макет

В.2.4.1 Ограничения на диаграмме

В.2.4.2 Размещение стрелок

В.2.4.3 Макет стрелки

В.2.5 Написание текста

В.2.5.1 Текст и глоссарий

В.2.5.2 Примечания и ссылки

В.3 Сбор данных для моделирования IDEF

В.3.1 Введение

В.3.2 Процесс опроса

В.3.3 Комплекта опроса

ПРИЛОЖЕНИЕ С-ПРОЦЕДУРЫ И ФОРМЫ ЦИКЛА ОБЗОРА

С.1 IDEF Дисциплина командной работы

С.2 Цикл Комплекта IDEF

С.2.1 Роли персонала

С.2.1.1 Разработчик (Автор)

С.2.1.2 Комментатор

С.2.2.1 Руководство для читателей

С.2.2.2 Взаимодействие разработчика с комментатором

С.2.2.3 Рекомендации по проведению совещаний

С.3 Комплекты IDEF

С.3.1 Заполнение Титульного листа Комплекта

С.3.2 Подготовка стандартного Комплекта

С.4 Стандартная форма диаграммы

С.4.1 Рабочая информация

С.4.1.1 Поля „Разработчик / Дата / Проект”

С.4.1.2 Поле “Примечания читателя»

С.4.1.3 Поле «Статус»

С.4.1.4 Поле Читатель/Дата

С.4.1.4 Поле Читатель/Дата

С.4.1.6 Поле «Используется В» (“Used At”)

С.4.2 Поле " Сообщение” (“Message” )

С.4.3 Поле «Нода» (“Node” )

С.4.4 Поле «Название» (“Title” )

С.4.5 Поле «Номер» (“Number”)

С.4.5.1 Поле «Номер» (Большая область)

С.4.5.2 Поле " Номер" ( “Number” )(«Номер страницы Комплекта»  
Малая прямоугольная Область)

С.5 Хранение файлов

С.6 Процедура обзора модели IDEF

Приложение D Информативные определения  
50.

## 1. Общие сведения

### 1.1 Область применения

Данный стандарт описывает язык моделирования (синтаксис и семантика), поддерживающий методологию IDEF0 для разработки структурированных графических представлений системы или объекта. Использование данного стандарта позволяет создавать IDEF0 модели, включающие в себя системные функции (действия, процессы, операции), функциональные взаимосвязи, а также данные и объекты, поддерживающие системный анализ и проектирование, анализ предприятия и реинжиниринг бизнес-процессов.

Данный документ содержит три нормативных раздела, Разделы 1, 2 и 3, которые определяют язык, поддерживающий моделирование IDEF0. Раздел 1, содержит обзор документа. Раздел 2 определяет ключевые

термины, используемые в нормативных разделах. Раздел 3 определяет синтаксис и семантику языка.

В дополнение к трем нормативным разделам настоящий документ содержит также четыре информативных приложения. В приложении А рассматриваются концепции, лежащие в основе IDEF0. В Приложении В приводятся рекомендации по созданию, интерпретации и сбору данных для диаграмм IDEF0. В Приложении С описываются структурированные групповые процедуры (в том числе формы) для обзора и проверки модели IDEF0. В приложении D определяются ключевые термины, используемые в приложениях.

Данный стандарт распространяется на IDEF0 (Руководство по методологии функционального моделирования), разработанный по Программе интегрированной компьютеризации производства BBC США (ICAM), Июнь 1981 года и с тех пор широко применяется многими пользователями IDEF.

## 1.2 ЦЕЛЬ

Основными целями настоящего стандарта являются:

1. Документировать и уточнять методологию моделирования IDEF0 и как правильно ее использовать;
2. Обеспечить средства для полного и последовательного моделирования функций, требуемых системой или объектом, а также данных и объектов, которые связывают эти функции;
3. Предоставить методику моделирования, не зависящую от методов или инструментов компьютерной программной инженерии (CASE), но которая может быть использована в сочетании с этими методами или инструментами;
4. Предоставить язык моделирования, обладающий следующими характеристиками:
  - а) Универсальность (для анализа систем и объектов различного назначения, объема и сложности);
  - б) Определенность и точность (для создания требуемых, пригодных для использования моделей);
  - в) Краткость и лаконичность (для облегчения понимания, информационного обмена, согласования и проверки);



- d) Концептуальность (для представления функциональных требований, не зависящих от физических или организационных вариантов);
- d) Гибкость (поддержка нескольких этапов жизненного цикла проекта).

## 2. Определения

Этот раздел содержит определения, относящиеся к нормативным разделам настоящего документа. Определения информационных приложений см. в Приложении D. Термин, если он определен, сформулирован либо в Разделе 2, либо в Приложении D.

**2.1 Диаграмма А-0:** специальный вид однокомпонентной (контекстной) диаграммы IDEF0, состоящей из одного блока, описывающего функцию верхнего уровня, ее вводы, выводы, контроля, и механизмы, вместе с формулировками цели модели и точки зрения, с которой строится модель.

**2.2 Стрелка:** направленная линия, состоящая из одного или нескольких сегментов, которая моделирует открытый канал или канал, передающий данные или материальные объекты от источника (начальная точка стрелки), к потребителю (конечная точка с «наконечником»). Имеется 4 класса стрелок: стрелка ввода, стрелка вывода, стрелка контроля, стрелка механизма (включает стрелку вызова). (См.: сегмент стрелки, граничная стрелка, внутренняя стрелка).

**2.3 Метка стрелки:** существительное или оборот существительного, связанные со стрелкой или сегментом стрелки и определяющие их значение.

**2.4 Сегмент стрелки:** сегмент линии, который начинается или заканчивается на стороне блока, в точке ветвления или слияния, или на границе (несвязанный конец стрелки)

**2.5 Граничная стрелка:** стрелка, один из концов которой связан с источником или потребителем, а другой не присоединен ни к какому блоку на диаграмме. Отображает связь диаграммы с другими блоками системы и отличается от внутренней стрелки.

**2.6 Блок:** прямоугольник, содержащий имя и номер и используемый для описания функции.

**2.7 Имя блока:** глагол или глагольный оборот, помещенный внутри блока и описывающий моделируемую функцию.

**2.8 Номер блока:** число (0 6), помещаемое в правом нижнем углу блока и однозначно идентифицирующее блок на диаграмме.

**2.9 Ветвление:** Соединение или разделение двух или большего числа сегментов стрелок.

**2.10 Связывание/развязывание:** Объединение значений стрелок в составное значение (связывание в «пучок»), или разделение значений стрелок (развязывание «пучка»), выраженные синтаксисом слияния или ветвления

**2.11 С-номер:** номер, создаваемый в хронологическом порядке и используемый для идентификации диаграммы и прослеживания ее истории; может быть использован в качестве ссылового выражения при определении конкретной версии диаграммы.

**2.12 Стрелка вызова:** вид стрелки механизма, который обозначает обращение из блока данной модели (или части модели) к блоку другой модели (или другой части той же модели) и обеспечивает связь между моделями или между разными частями одной модели.

**2.13 Дочерний блок:** блок на дочерней (порожденной) диаграмме.

**2.14 Дочерняя диаграмма:** диаграмма, детализирующая родительский (порождающий) блок.

**2.15 Контекст:** окружающая среда, в которой действует функция (или набор функций на диаграмме).

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

**2.17 Стрелка управления:** класс стрелок, которые в IDEF0 отображают управления, то есть условия, при выполнении которых выход блока будет правильным. Данные или объекты, смоделированные как элементы управления, могут быть преобразованы функцией, создающей соответствующий вывод. Управляющие стрелки связываются с верхней стороной блока IDEF0.

**2.18 Декомпозиция:** разделение моделируемой функции на функции компоненты.

**2.19 Ссылочное выражение (DRE):** Ссылка (например, номер ноды, C-номер, номер страницы), написанная под нижним правым углом блока IDEF0, чтобы показать, что она детализирована и какая диаграмма ее детализирует.

**2.20 Диаграмма:** часть модели IDEF0, описывающая декомпозицию блока.

**2.21 Номер ноды диаграммы:** та часть ссылки на узел диаграммы, которая соответствует номеру ноды родительского блока.

**2.22 Диаграмма-иллюстрация (FEO):** Графическое описание, используемое, для сообщения специфических фактов о диаграмме IDEF0. При построении диаграмм FEO можно не придерживаться правила IDEF0. В отличие от графической диаграммы IDEF0, диаграмма FEO может не соответствовать правилам IDEF0.

**2.23 Разветвление:** Соединение, на котором сегмент стрелки IDEF0 (идуший от источника к пользователю) делится на два или более сегментов стрелки. Может означать «развязывание пучка»

**2.24 Функция:** действие, процесс или преобразование (моделируемые блоком IDEF0), идентифицируемое глаголом или глагольной формой, которая описывает, что должно быть выполнено.

**2.25 Имя функции:** то же, что и имя блока

**2.26 Глоссарий:** Список определений для ключевых слов, фраз и аббревиатур, связанных с узлами, блоками, стрелками или с моделью IDEF0 в целом.

**2.27 Код ICOM:** Аббревиатура (Input Ввод, Control Управление, Output Вывод, Mechanism – Механизм), код, обеспечивающий соответствие граничных стрелок дочерней диаграммы со стрелками родительского блока; используется для ссылок..

**2.28 Модель IDEF0:** Графическое описание системы, разработанное с определенной целью (см. 3.46) и с выбранной точки зрения. Комплект одной или более диаграмм IDEF0, которые изображают функции системы с помощью графики, текста и глоссария.

**2.29 Выводящая стрелка:** класс стрелок, которые отображают ввод IDEF0-блока, то есть данные или материальные объекты, которые преобразуются функцией в вывод. Стрелки связываются с левой стороной блока IDEF0.

**2.30 Интерфейс:** Разделяющая граница, через которую проходят данные или материальные объекты; соединение между двумя или большим числом компонентов модели, передающее данные или материальные объекты от одного компонента к другому.

**2.31 Внутренняя стрелка:** вводящая, управляющая или выводящая стрелки, концы которых связывают источник с потребителем, являющихся блоками одной диаграммы. Отличается от граничной стрелки.

**2.32 Соединение:** Соединение, на котором сегмент стрелки IDEF0 (идуший от источника к потребителю) сливается с одним или несколькими другими сегментами стрелки, образуя единый сегмент стрелки. Может обозначать связывание значений сегмента стрелки.

**2.33 Стрелка механизма:** Класс стрелок, которые отображают механизмы IDEF0, то есть средства, используемые для выполнения функции; включает специальный случай стрелки вызова. Стрелки механизмов связываются с нижней стороной блока IDEF0.

**2.34 Примечание к модели:** текстовый комментарий, являющийся частью диаграммы IDEF0 и используемый для записи факта, не нашедшего графического изображения.

**2.35 Нода:** Блок, порождающий дочерние блоки; родительский блок. (См.: перечень нод, дерево нод, номер ноды, ссылка ноды, номер ноды диаграммы).

**2.36 Перечень нод:** список, часто ступенчатый, показывающий ноды модели IDEF0 в упорядоченном виде. Имеет то же значение и содержание, что и дерево нод.

**2.37 Номер ноды:** Код, присвоенный блоку для указания его положения в иерархии модели; может использоваться в качестве Подробного ссылочного выражения.

**2.38 Ссылка ноды:** Код, присвоенный диаграмме, для ее идентификации и определения положения в иерархии модели; формируется из сокращенного имени модели и номера ноды диаграммы с дополнительными расширениями.

**2.39 Дерево нод:** Представление отношений между родительскими и дочерними нодами модели IDEF0 в форме древовидного графа. Имеет то же значение и содержание, что и перечень нод.

**2.40 Выводящая стрелка:** Класс стрелок, которые отображают вывод IDEF0 блока, то есть данные или материальные объекты, произведенные

функцией. Выводящие стрелки связываются с правой стороной блока IDEF0.

**2.41 Родительский блок:** блок, который подробно описывается дочерней диаграммой.

**2.42 Родительская диаграмма:** диаграмма, которая содержит родительский блок.

**2.43 Цель:** краткая формулировка причины создания модели.

**2.44 Семантика:** Значение синтаксических компонентов языка.

**2.45 Тильда:** Небольшая ломаная (волнистая) линия, используемая для соединения метки с конкретным сегментом стрелки или примечания модели с компонентом диаграммы.

**2.46 Синтаксис:** Структурные компоненты или характеристики языка и правила, которые определяют отношения между ними.

**2.47 Текст:** Любой текстовый (не графический) комментарий к графической диаграмме IDEF0.

**2.48 Название:** глагол или глагольный оборот, описывающий общую функцию, представленную на диаграмме IDEF0; название дочерней диаграммы соответствует имени ее родительского блока.

**2.49 Стрелка тунелирования:** Стрелка (со специальной нотацией), не удовлетворяющая обычному требованию, согласно которому каждая стрелка на дочерней диаграмме должна соответствовать стрелкам на родительской диаграмме.

**2.50 Точка зрения:** Краткое изложение перспективы модели.

### 3. Модели IDEF0

В этом разделе рассматриваются основные этапы методики моделирования IDEF0, определяются основные компоненты синтаксиса (графический компонент) и семантики (смысл), определяются правила, регулирующие использование метода IDEF0, и описываются типы используемых диаграмм. Хотя компоненты синтаксиса и семантики очень тесно взаимосвязаны, каждый из них обсуждается отдельно без учета фактической последовательности построения.

### 3.1 Концепции моделей

Модель это представление набора компонентов системы или объекта. Модель разрабатывается для понимания, анализа, улучшения или замены системы. Системы состоят из взаимодействующих или взаимозависимых частей, выполняющих некую полезную работу. Частями (элементами) системы могут быть любые комбинации разнообразных сущностей, включающие людей, информацию, программное обеспечение, оборудование, изделия, сырье или энергию (энергоносители). Модель описывает, что происходит в системе, как ею управляют, какие сущности она преобразует, какие средства использует для выполнения своих функций и что производит.

IDEF0 это метод моделирования, основанный на комбинированной графике и тексте, которые представлены организованным и систематическим способом для достижения понимания, поддержки анализа, обеспечения логики для потенциальных изменений, определения требований или поддержки действий по проектированию и интеграции на уровне систем. Модель IDEF0 состоит из иерархического ряда диаграмм, которые последовательно отображают возрастающие уровни детализации, описывая функции и их интерфейсы в контексте системы. Существует три типа диаграмм: графические, текстовые и глоссарий. Графические диаграммы определяют функции и функциональные отношения с помощью синтаксиса и семантики блоков и стрелок. Текстовая диаграмма и глоссарий предоставляют дополнительную информацию, дополняющую графические диаграммы.

IDEF0 является инженерной техникой, предназначенной для выполнения и управления: анализом потребностей, анализом преимуществ, определением требований, функциональным анализом, проектированием систем, техническим обслуживанием и исходными данными для непрерывного совершенствования. Модели IDEF0

предоставляют «схематический чертеж» функций и их интерфейсов, которые должны быть понятны, что позволит принимать логичные, доступные, интегрируемые и реализуемые решения при системном проектировании. Как чертеж устройства говорит нам о том, как различные части взаимодействуют друг с другом, так и модель IDEF0 отражает то, как взаимодействуют и работают функции системы. При использовании системного подхода IDEF0 обеспечивает:

1. Выполнение системного анализа и проектирования на всех уровнях систем, состоящих из людей, машин, материалов, компьютеров и информации всех видов целого предприятия, системы или объекта;
2. Подготовку справочной документации, которая осуществляется одновременно с разработкой, что способствует интеграции новых систем и стимулирует совершенствование существующих систем;
3. Обмен информацией между разработчиками, дизайнерами, пользователями и менеджерами;
4. Достижение консенсуса коалиционной команды на основе единого понимания;
5. Управление крупными и сложными проектами с использованием качественных показателей хода выполнения;
6. Предоставление эталонной архитектуры для корпоративного анализа, информационной инженерии и управления ресурсами.

Дальнейшее обсуждение концепций, философии и ролей участников моделирования проектов с помощью IDEF0 представлено в информационных приложениях к настоящему документу.

## 3.2 Синтаксис и семантика

### 3.2.1 Синтаксис

Структурные компоненты и особенности языка, а также правила, определяющие отношения между ними, называются синтаксисом языка. Компонентами синтаксиса IDEF0 являются блоки и стрелки, правила и диаграммы. Блоки представляют функции, определяемые как деятельность, процесс, операция, действие или преобразование. Стрелки представляют данные или материальные объекты, связанные с функциями. Правила определяют, как следует применять компоненты; диаграммы обеспечивают формат графического и словесного описания

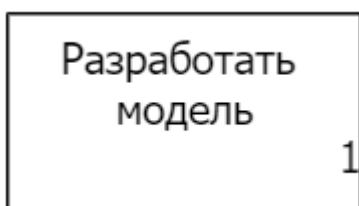


моделей. Формат образует основу для управления конфигурацией модели.

#### 3.2.1.1 Блоки

Блок описывает функцию. Типичный блок показан на рис. 1. Внутри каждого блока указывается его имя и номер. Имя должно быть глаголом совершенного вида, описывающим функцию. Каждая блок на диаграмме имеет номер, который указан в правом нижнем углу. Номера блоков используются для их идентификации на диаграмме и в соответствующем тексте.

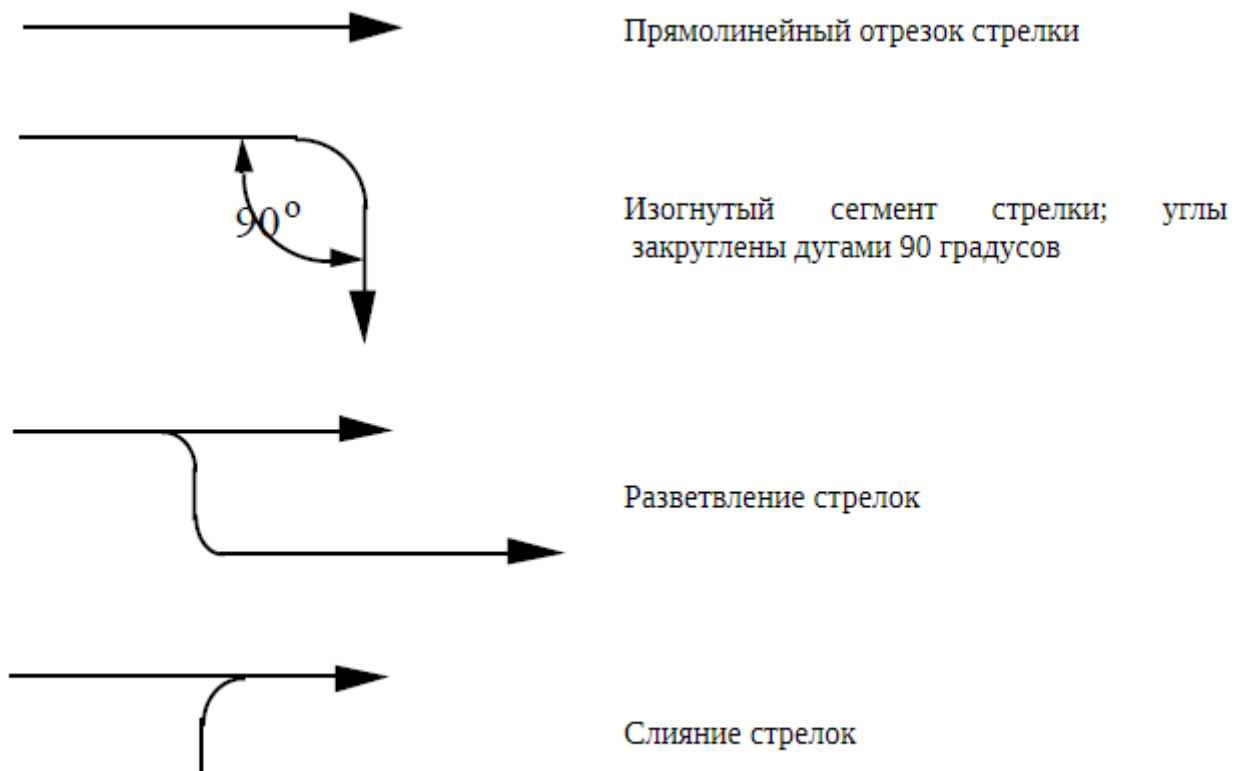
- Имя функции -это глагол или глагольный оборот.
- Отображается номер блока.



**Рисунок 1. Синтаксис блока.**

#### 3.2.1.2 Стрелки

Стрелка состоит из одного или нескольких отрезков линии и наконечника на одном конце. Как показано на рис. 2, сегменты стрелок могут быть прямыми или изогнутыми; в последнем случае горизонтальные и вертикальные отрезки стрелки сопрягаются дугами, имеющими угол  $90^\circ$ , а также могут разветвляться (расходиться или соединяться). Стрелки не представляют поток или последовательность событий, как в традиционных блок-схемах потоков или процессов. Они лишь показывают, какие данные или материальные объекты должны поступить на ввод функции для того, чтобы эта функция могла выполняться.



**Рисунок 1. Синтаксис блока.**

### 3.2.1.3 Синтаксические правила

#### Блоки

1. Размеры блоков должны быть достаточными для указания названия блоков.
2. Блоки должны иметь прямоугольную форму.
3. Блоки должны быть нарисованы сплошными линиями.

#### Стрелки

1. Ломанные стрелки изгибаются только под углом 90 град.
2. Стрелки должны быть нарисованы сплошными линиями.
3. Стрелки должны быть нарисованы вертикально или горизонтально, а не по диагонали.

4. Концы стрелок должны касаться внешнего периметра функционального блока и не должны пересекать его.
5. Стрелки должны присоединяться к блоку на его сторонах. Присоединение в углах не допускается.

### 3.2.2 Семантика

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

#### 3.2.2.1 Семантика блоков и стрелок.

Поскольку IDEF0 является методологией функционального моделирования, имя блока, описывающее функцию, должно быть глаголом или глагольным оборотом. Например, имя блока «Выполнить проверку», означает, что блок с таким именем превращает непроверенные детали в проверенные. После присваивания блоку имени, к соответствующим его сторонам присоединяются вводящие, выводящие и управляющие стрелки, а также стрелки механизма, что и определяет наглядность и выразительность изображения блока IDEF0.

Чтобы гарантировать точность модели, следует использовать стандартную терминологию. Блоки именуются глаголами или глагольными оборотами и эти имена разделяются и группируются в диаграммах декомпозиции. Стрелки и их сегменты, как отдельные, так и связанные в «пучок», помечаются существительными или оборотами существительного. Метки сегментов стрелок являются предписывающими, ограничивающими значение сегмента для применения исключительно к конкретным данным или объектам, которые графически представляет сегмент стрелок. Значения стрелок далее выражаются через синтаксис ветвления и слияния.

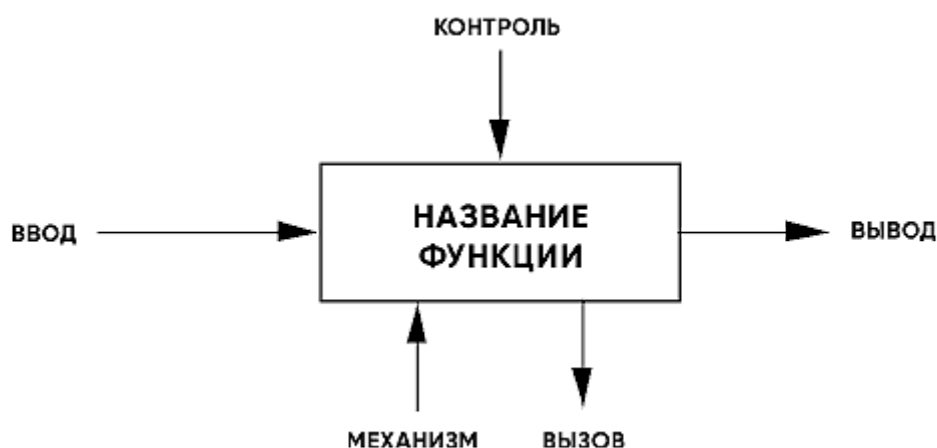
Каждая сторона функционального блока имеет стандартное значение с точки зрения связи блок/стрелка, Сторона блока, к которой присоединена стрелка, однозначно определяет ее роль. Стрелки, входящие в левую часть блока, являются вводами. Вводные данные преобразуются или обрабатываются функцией для получения выводных данных. Стрелки, входящие в блок сверху, являются сигналами контроля. Сигналы

управления задают условия, необходимые функции для формирования правильного вывода. Стрелки, выходящие из блока с правой стороны, являются выводами. Выводы-это данные или объекты, создаваемые функцией.

Стрелки, подходящие к блоку снизу, представляют собой механизмы. Стрелки, направленные вверх, идентифицируют средства, поддерживающие выполнение функции. Другие средства могут быть унаследованы от родительского блока. Стрелки механизма, указывающие вниз, называются стрелками вызова. Стрелки вызова позволяют обмениваться деталями между моделями (связывая их вместе) или между частями одной и той же модели. В вызываемом блоке содержится подробная информация о вызывающем блоке (см. Раздел 3.3.2.10).

Стандартные положения стрелок показаны на Рис.3.

Вспомогательная информация, касающаяся функции и ее назначения, должна быть указана в тексте, связанном с диаграммой. Диаграмма может иметь или не иметь связанный текст. При использовании аббревиатур, сокращений, ключевых слов или фраз в глоссарии должен быть указан полностью расшифрованный термин(ы).



**Рисунок 3. Позиции стрелок и их значение**

#### 3.2.2.2 Имена и метки

Блоки представляют функции, обозначающие что должно быть выполнено. Имя функции должно быть активным глаголом или глагольным оборотом. Примеры имен функций:

произвести детали проверить исполнение разработать детальные чертежи	запланировать ресурсы спроектировать систему изготовить компонент	Провести опрос выполнить техническое обслуживание осмотреть деталь
-------------------------------------------------------------------------------	----------------------------------------------------------------------------------	-----------------------------------------------------------------------------

Стрелки идентифицируют данные или материальные объекты, необходимые для выполнения функции или производимые ею. Каждая стрелка должна быть помечена существительным или оборотом существительного, например:

технические характеристики требования к проектированию инженер-конструктор	протокол испытания детальное проектирование плата в сборе	бюджет директива требования
----------------------------------------------------------------------------------	-----------------------------------------------------------------	-----------------------------------

Пример размещения меток стрелок и названий блоков приведен на Рис.4.

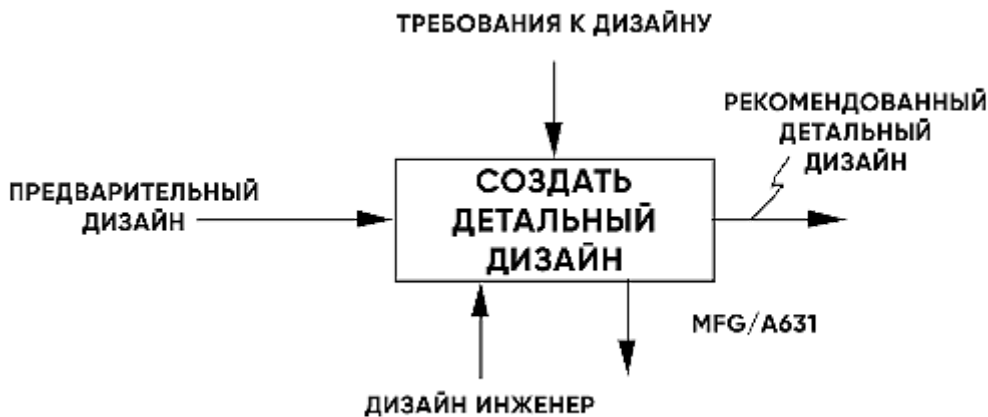



Рисунок 4. Семантика метки и имен.

### 3.2.2.3 Семантические правила блоков и стрелок.

1. Имя блока должно быть активным глаголом или глагольным оборотом.
2. Каждая сторона функционального блока должна иметь стандартное отношение блока/стрелки:
  - а) Вводящие стрелки должны подходить к левой стороне блока;
  - б) Стрелки управления должны подходить к верхней стороне блока.
  - в) Выводящие стрелки должны отходить от правой стороны блока;
  - г) Стрелки механизма (кроме стрелок вызова) должны указывать вверх и подключаться к нижней стороне блока.
  - д) Стрелки вызова механизма должны указывать вниз, подключаться к нижней стороне блока, и помечаться ссылкой на вызываемый блок.
3. Сегменты стрелок, за исключением стрелок вызова, должны помечаться существительным или оборотом существительного, если только единственная метка стрелки несомненно не относится к стрелке в целом.
4. Знак "«Тильда»" (  ) используется для связи стрелки с соответствующей меткой, если только связь между стрелкой и меткой не является очевидной.
5. В метках стрелок не должны использоваться следующие термины: функция, ввод, контроль, вывод, механизм, вызов.

## 3.3 Диаграммы IDEF0

### 3.3.1. Типы диаграмм

IDEF0-модели состоят из трех типов документов: графических диаграмм, текста и глоссария. Эти документы имеют перекрестные ссылки друг на друга. Графическая диаграмма – главный компонент IDEF0-модели, содержащий блоки, стрелки, соединения блоков и стрелок и связанные с ними отношения. Блоки представляют основные функции моделируемого объекта. Эти функции могут быть разбиты (декомпозированы) на составные части и представлены в виде более подробных диаграмм; процесс декомпозиции продолжается до тех пор, пока объект не будет описан на уровне детализации, необходимом для достижения целей конкретного проекта. Диаграмма верхнего уровня в модели дает наиболее общее или абстрактное описание предмета, представленного моделью. За этой диаграммой следует ряд дочерних диаграмм, дающих более подробную информацию об объекте.

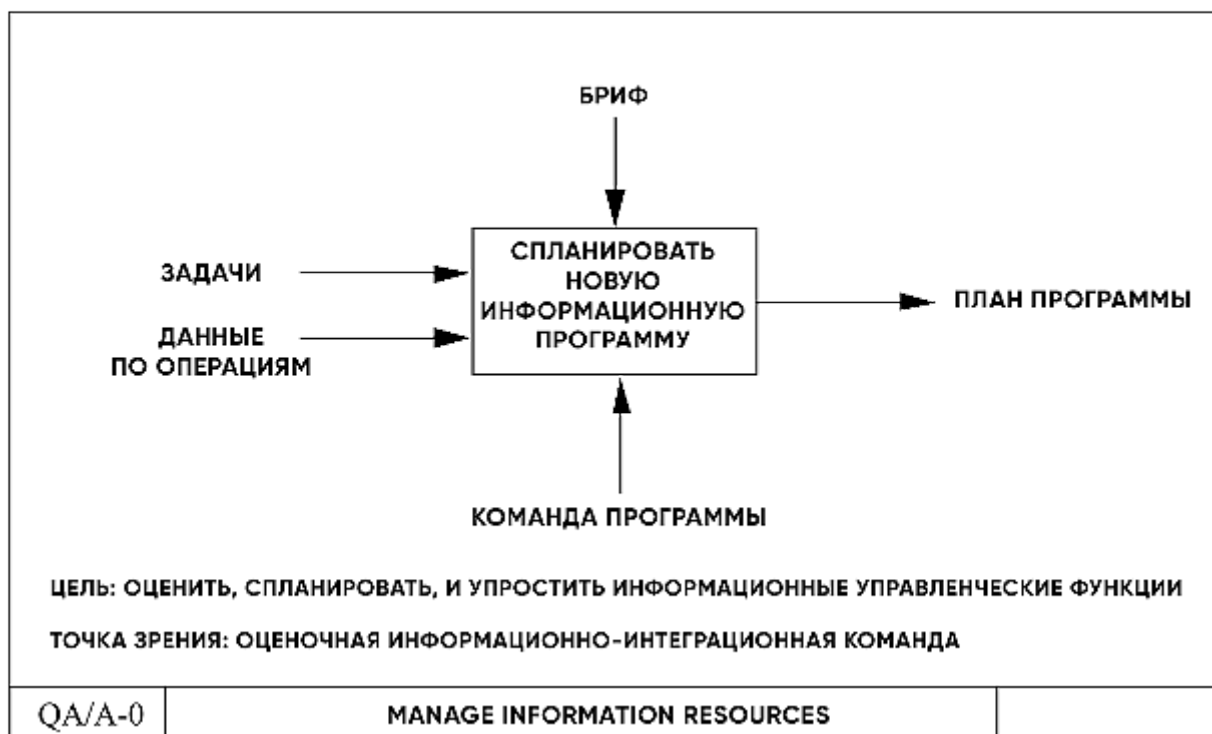
### 3.3.1.1 Контекстная диаграмма верхнего уровня

Каждая модель должна иметь контекстную диаграмму верхнего уровня, на которой объект моделирования представлен единственным блоком с граничными стрелками. Эта диаграмма называется A0 (произносится как минус ноль). Стрелки на этой диаграмме отображают связи объекта моделирования с окружающей средой. Поскольку единственный блок представляет весь объект, его имя – общее для всего проекта. Это же справедливо и для всех стрелок интерфейса, поскольку они представляют полный комплект внешних интерфейсов объекта.

Диаграмма A-0 также задает область действия модели или ее границу и ориентацию. Пример диаграммы A-0 показан на Рис.5.

Контекстная диаграмма A-0 также должна содержать краткие утверждения, определяющие точку зрения и цель модели. Эти утверждения помогают руководить разработкой модели и ввести этот процесс в определенные рамки. Точка зрения определяет, что можно „увидеть“ в контексте модели и с какой точки зрения или „под углом“. В зависимости от аудитории могут быть приняты различные точки зрения, которые подчеркивают различные аспекты объекта. Аспекты, важные с одной точки зрения, могут не появиться в модели, разрабатываемой с другой точки зрения на тот же самый объект.

Формулировка цели выражает причину создания модели, т.е. содержит перечень вопросов, на которые должна отвечать модель, что в значительной мере определяет ее структуру. Наиболее важные свойства объекта обычно выявляются на верхних уровнях иерархии; по мере декомпозиции функции верхнего уровня и разбиения ее на подфункции, эти свойства уточняются. Каждая подфункция моделируется индивидуально блоком, а родительские блоки детализируются дочерними диаграммами на следующем более низком уровне. Все дочерние диаграммы должны находиться в пределах области действия контекстной диаграммы верхнего уровня.



**Рисунок 5. Пример диаграммы верхнего уровня**

### 3.3.1.2 Дочерняя диаграмма

Единственная функция, представленная на контекстной диаграмме верхнего уровня, может быть разложена на основные подфункции путем создания дочерней диаграммы. В свою очередь, каждая из этих подфункций может быть декомпозирована, создавая другую дочернюю диаграмму более низкого уровня. На данной диаграмме некоторые функции, ни одна из функций или все функции могут быть разложены. Каждая дочерняя диаграмма содержит дочерние блоки и стрелки, которые обеспечивают дополнительную детализацию родительского блока.

Дочерняя диаграмма, полученная в результате декомпозиции функции, охватывает ту же область, что и родительский блок, который она детализирует. Таким образом, дочернюю диаграмму можно рассматривать как вложение в родительский блок. Эта структура проиллюстрирована на Рис. 6.

### 3.3.1.3 Родительская диаграмма

Родительская диаграмма это диаграмма, содержащая один или несколько родительских блоков. Каждая обычная (неконтекстная) диаграмма также является дочерней диаграммой, поскольку по



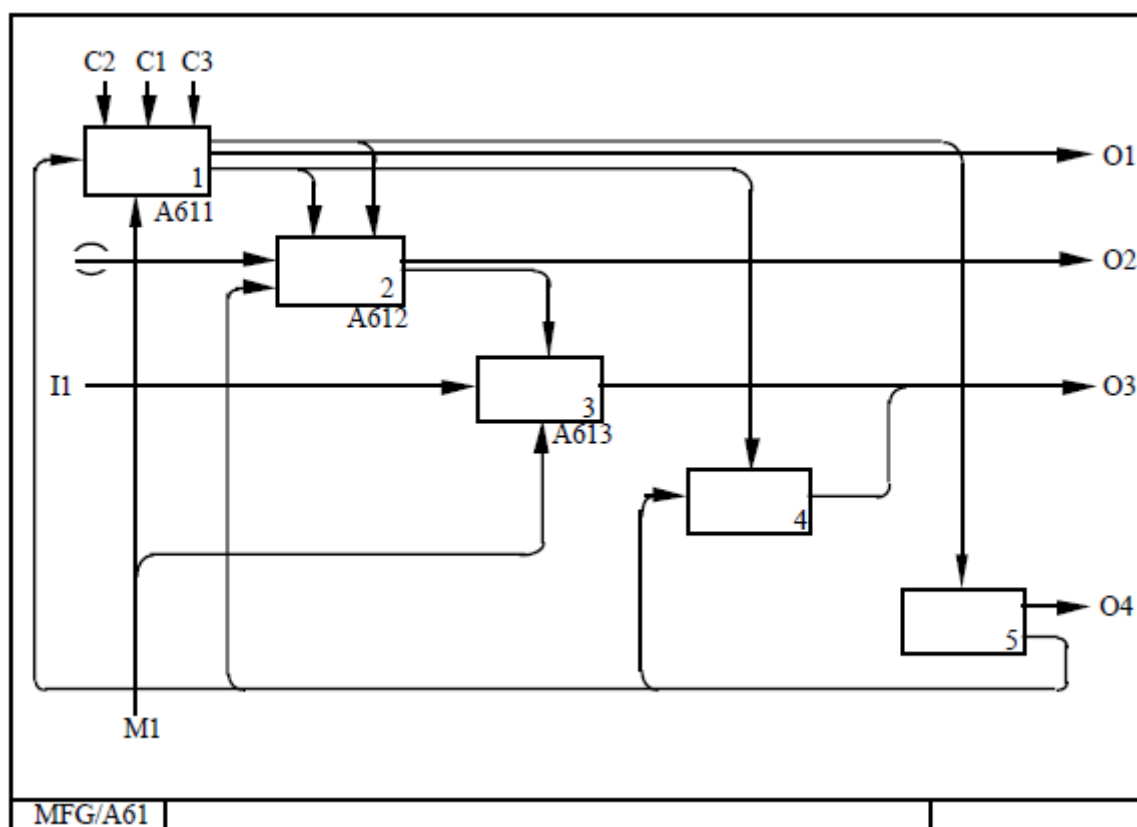
определению она детализирует родительский блок. Таким образом, диаграмма может быть, как родительской диаграммой (содержащей родительские блоки), так и дочерней диаграммой (детализирующей свой собственный родительский блок). Аналогично, блок может быть, как родительским (подробно описываться дочерней диаграммой) так и дочерним (появляющимся на дочерней диаграмме). Основное иерархическое отношение существует между родительским блоком и детализирующей его дочерней диаграммой.

То, что блок является дочерним и раскрывает содержание родительского блока на диаграмме предшествующего уровня, указывается специальным ссылочным кодом. DRE-это короткий код, написанный под нижним правым углом детализированного (родительского) блока, который указывает на его дочернюю диаграмму.

DRE принимает одну из следующих форм:

1. Хронологический номер создания, называемый „С-номером“, который должен однозначно идентифицировать конкретную версию дочерней диаграммы.
2. Номер страницы дочерней диаграммы в опубликованном документе, в котором отображается модель.
3. Номер ноды, ссылающейся на дочернюю диаграмму. Если существует несколько версий дочерней диаграммы, то конкретная версия не может быть указана.
4. Номер примечания к модели, в тексте которого указаны условия выбора конкретной дочерней версии.

Рисунок 7 иллюстрирует использование номеров нод в качестве DRE. На рисунке наличие DRE под блоками 1, 2 и 3 указывает на то, что они были подробно детализированы на указанных дочерних диаграммах.



**Рисунок 7. Использование специального ссылочного кода**

#### 3.3.1.4 Текст и глоссарий

Диаграмма может иметь связанный структурированный текст, который представляет собой краткий комментарий к содержанию диаграммы. Текст используется для выделения характеристик, потоков и межблоковых связей с целью уточнения предназначения элементов и схем, считающихся важными. Текст не должен использоваться для описания и без того понятных блоков и стрелок на диаграммах. Глоссарий содержит определения аббревиатур (акронимов), ключевых слов и фраз, используемых в качестве имен и меток на диаграммах. Глоссарий определяет понятия и термины, которые должны одинаково толковаться всеми участниками разработки и пользователями модели, чтобы правильно интерпретировать ее содержание.

#### 3.3.1.5 Диаграммы-иллюстрации

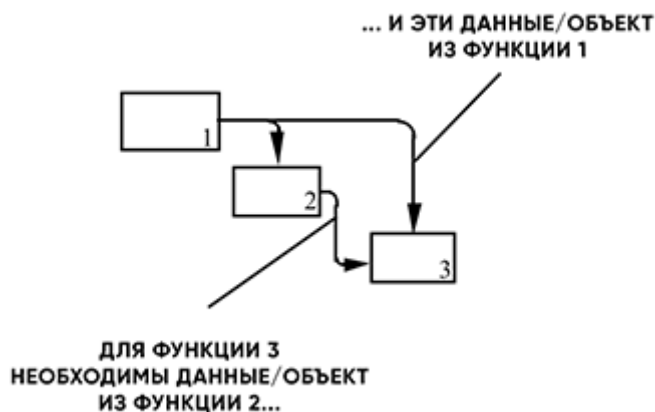
Диаграммы-иллюстрации (FEO, произносится fee-oh) используются в качестве дополнений, поясняющих специфику основных диаграмм для одинакового понимания конкретных областей модели. Дополнительная

детализация не должна быть избыточной и ограничена объемом, необходим для достижения заявленной цели профессионалами. Диаграмма FEO не обязательно должна соответствовать правилам синтаксиса IDEF0.

### 3.3.2 Свойства диаграмм.

#### 3.3.2.1 Стрелки как ограничения

Стрелки на диаграмме IDEF0 представляют данные или объекты и задают определенные ограничения. Только на низких уровнях детализации стрелки могут представлять поток или последовательность, когда моделируемый объект достаточно детализирован для обработки изменений, внесенных в конкретные элементы данных или объекты. Подключение вывода первого блока к вводу, управлению или механизму другого блока показывает, что функция, моделируемая последним блоком, требует и, следовательно, ограничивается наличием соответствующего вывода первого блока. Тип ограничений показан на Рисунке 8. Стрелки, подходящие к блоку, представляют все данные и объекты, необходимые для полного выполнения функции.



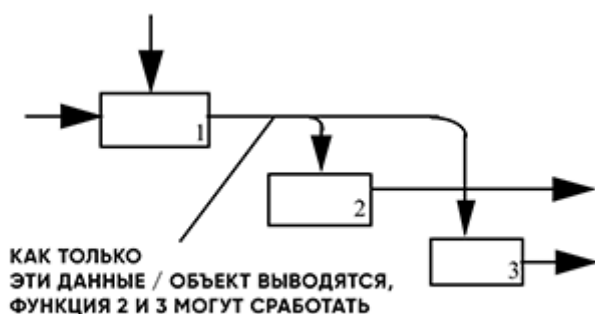
**Рисунок 8. Влияние ограничения**

#### 3.3.2.2 Активации блока

Блок может выполнять определенные части своей функции в зависимости от условий, комбинаций входящего сигнала и контроля и формирует различные выводы. Комбинации условий работы блока носят название активации блока.

### 3.3.2.3 Параллельное функционирование

Несколько функций в модели могут выполняться одновременно, если соблюдаются необходимые ограничения. Как показано на Рис.9, один блок может создать некоторые или все данные или материальные объекты, необходимые для параллельной работы нескольких блоков. Когда вывод одного блока формирует некоторые или все входы, сигналы управления и механизма, необходимые другому блоку, заданная активация последнего блока может зависеть от последовательной работы первого блока. Тем не менее, разные активации одного и того же блока (ов), возможно, с различными требованиями, могут работать одновременно.

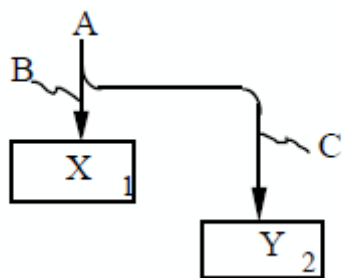


В данном случае функции 2 и 3 могут работать одновременно.

**Рисунок 9. Параллельное функционирование**

### 3.3.2.4 Стрелки как трубы

Стрелки высокого уровня можно сравнить с трубами или каналами. Стрелки высокого уровня имеют общие метки, в то время как стрелки на диаграммах более низкого уровня имеют более конкретные метки. Если стрелка разветвляется, образуя два или более новых сегмента стрелки, каждый сегмент стрелки может иметь более конкретную метку, как показано на Рис.10.

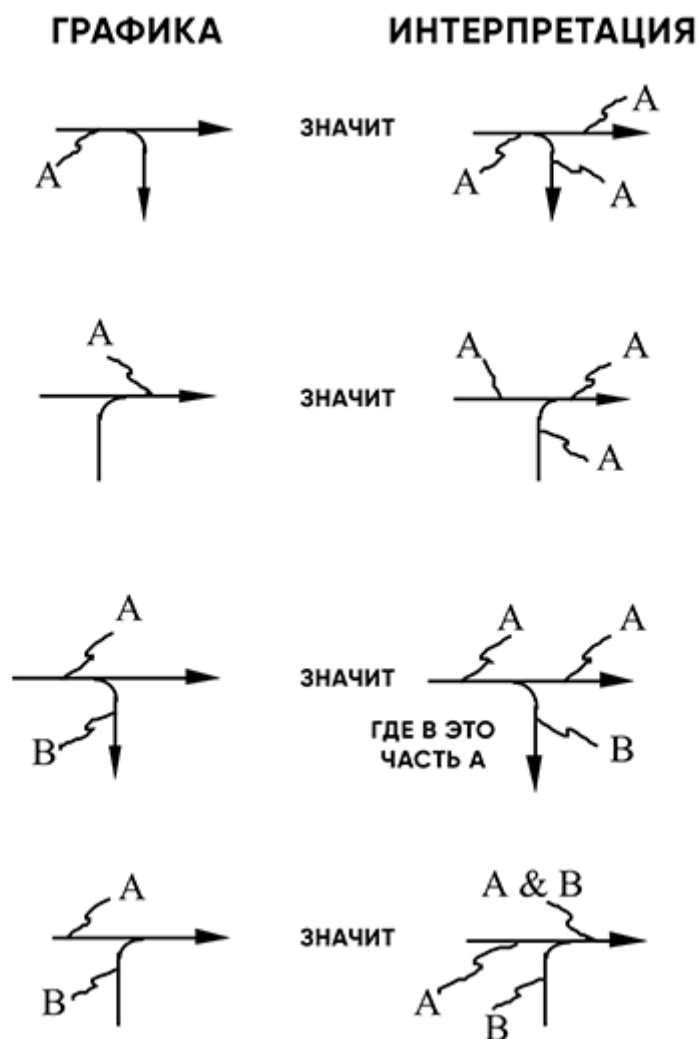


Пучок А разделяется на В и С, чтобы обеспечить управление X и Y.

**Рисунок 10. Стрелка труба с разветвлением**

### 3.3.2.5 Ветвление стрелок

Стрелка может ветвиться (разветвляться или соединяться), указывая, что один и тот же тип данных или объект может быть необходим или создан более чем одной функцией. Ветви могут представлять тот же объект, либо части того же объекта. Поскольку метки указывают, что представляют сегменты стрелок, метки на ветвящихся сегментах стрелок обеспечивают детализацию содержимого стрелок так же, как диаграммы нижнего уровня обеспечивают детализацию родительских блоков. Всё или часть содержимого стрелки может следовать по ответвлению. Разветвленная стрелка может обозначать «разделение» значений, которые были объединены более общей меткой. Соединение двух сегментов стрелки может означать „связывание“, то есть объединение отдельных значений в более общую категорию. Все данные задаются через ветви, если иное не обозначено специальной меткой на каждом сегменте стрелки. Данное правило проиллюстрировано на Рис. 11.



**Рисунок 11. Конструкции разветвления и соединения**

### 3.3.2.6 Межблочные соединения

За исключением одноблоковой контекстной диаграммы A-0, графическая диаграмма содержит минимум три и максимум шесть блоков. Блоки обычно расположены по диагонали от верхнего левого угла до нижнего правого, то есть в конфигурации „лестница“.

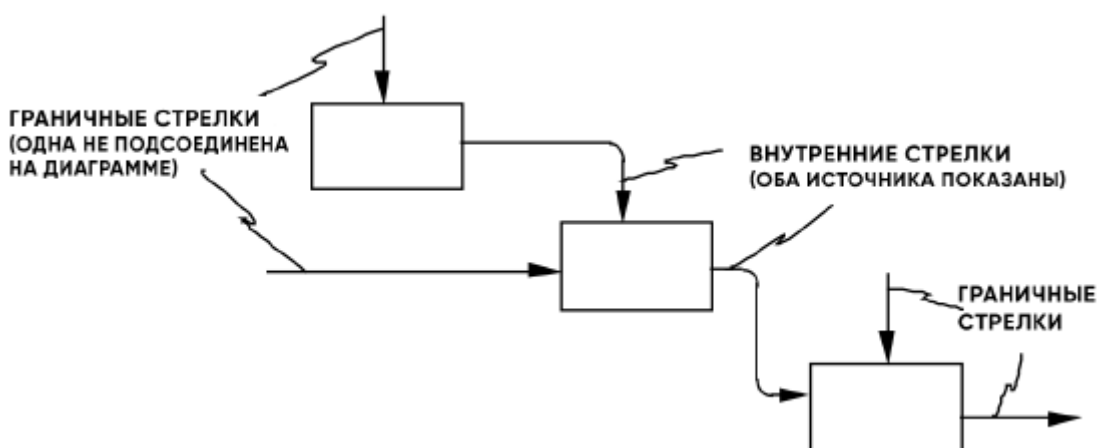
Любая выводящая стрелка может предоставлять некоторые или все данные или объекты ввода, управления или механизма в любой другой блок. Выводящая стрелка может передавать данные или объекты в несколько блоков с помощью механизм разветвления, как показано на Рис.12.



**Рисунок 12. Межблочные соединения**

Если блок на диаграмме детализирован дочерней диаграммой, каждая стрелка, связанная с родительским блоком, должна быть отображена на дочерней диаграмме, если стрелка не туннелируется рядом с ее родительским блоком (см. Раздел 3.3.2.9).

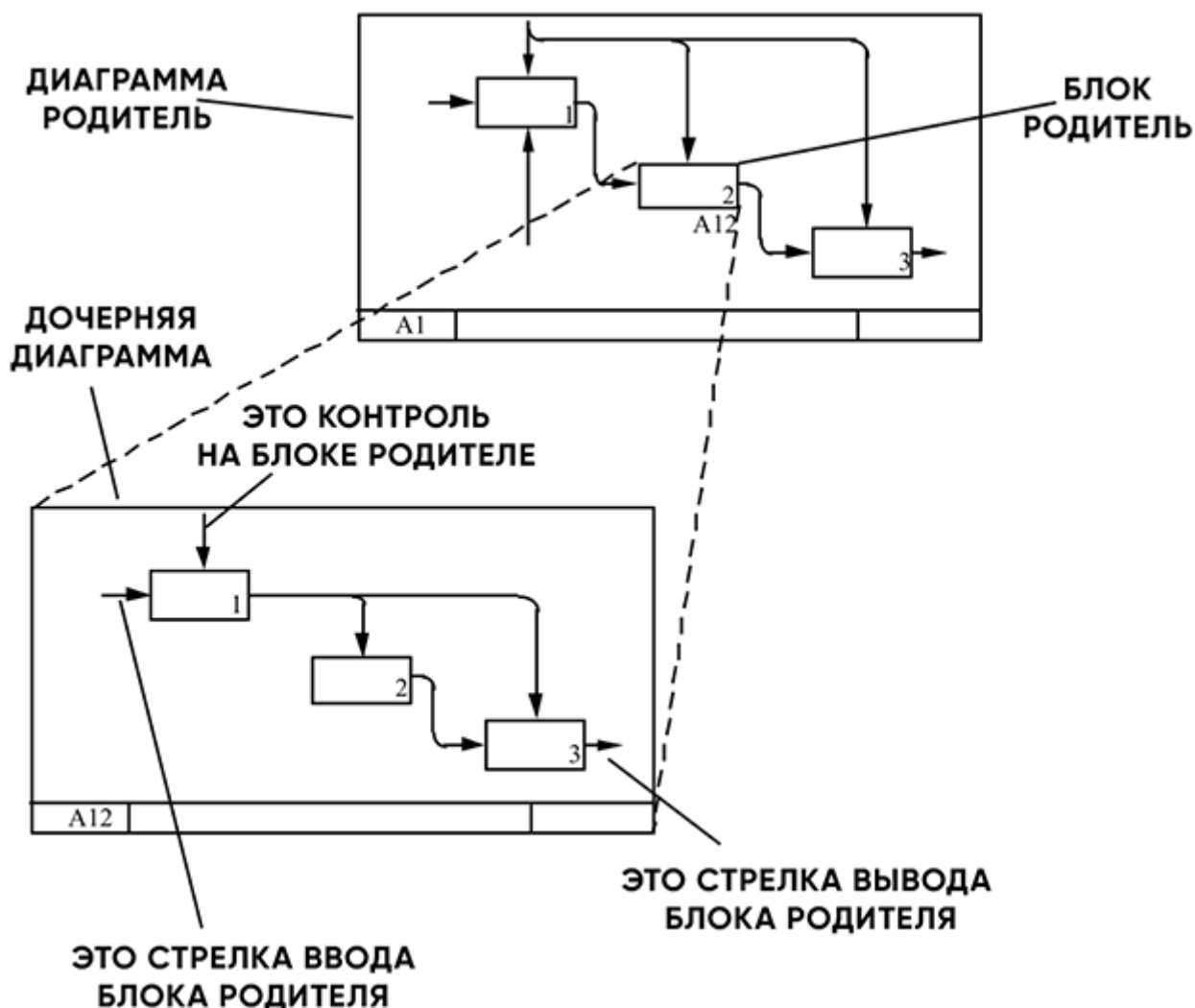
На диаграмме данные или объекты могут быть представлены внутренней стрелкой, причем оба конца (источник и потребитель) соединены с блоками, или граничной стрелкой, имеющей только один подсоединенный конец (источник или потребитель). Внутренние стрелки и граничные стрелки показаны на Рис.13. Граничные стрелки подробно описаны в разделах 3.3.2.7 и 3.3.2.8.



**Рисунок 13. Граничные и внутренние стрелки**

### 3.3.2.7 Граничные стрелки

Граничные стрелки на обычной (неконтекстной) графической диаграмме представляют входы, сигналы управления, выходы или механизмы родительского блока диаграммы. Потребителя или источник граничных стрелок можно найти, только проанализировав родительскую диаграмму. Все граничные стрелки на дочерней диаграмме (за исключением стрелок туннелирования, Раздел 3.3.2.9) должны соответствовать стрелкам, которые соединяются с их родительским блоком, как показано на Рис. 14.



**Рисунок 14. Граничная стрелка**

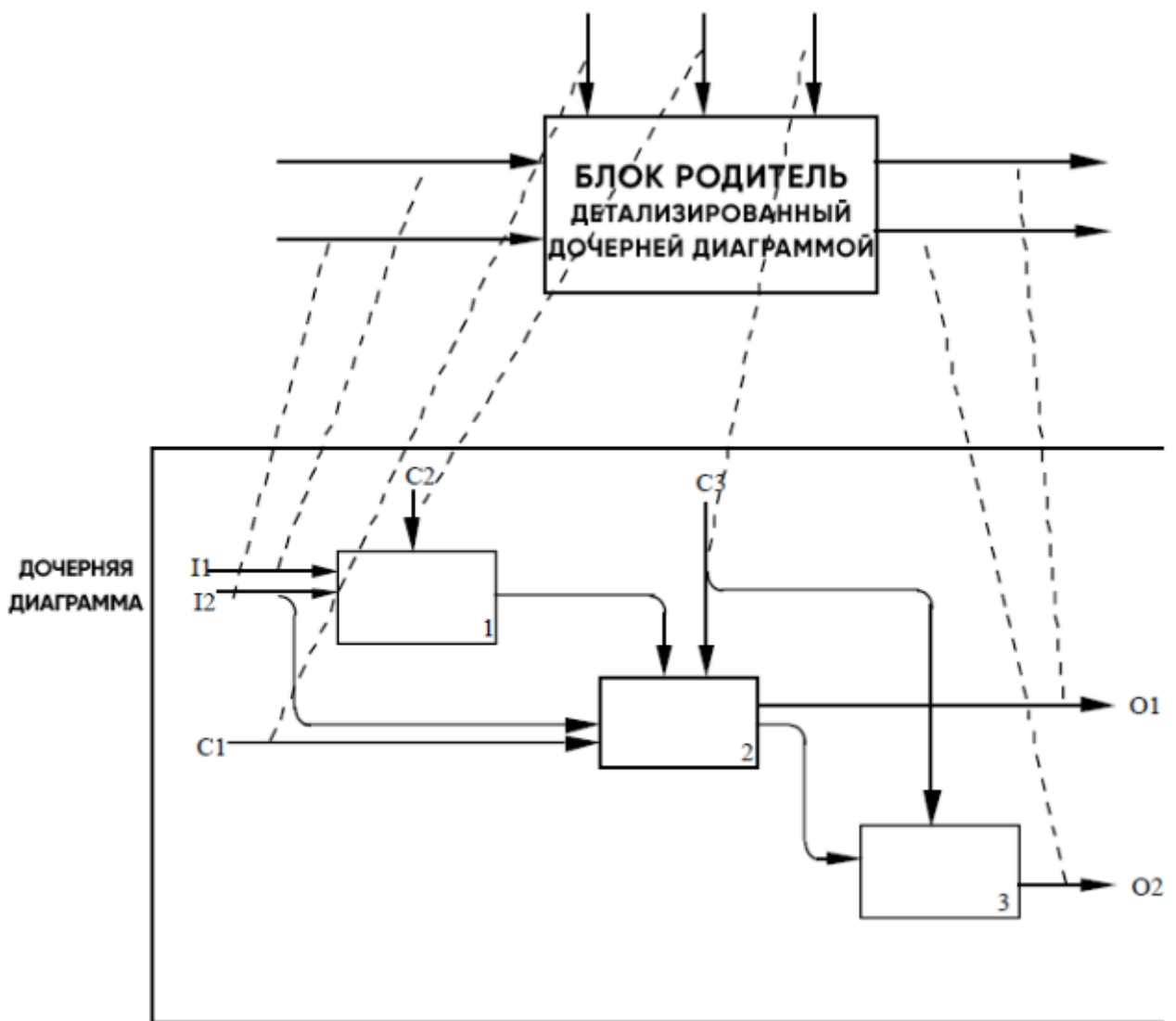
### 3.3.2.8 ICOM кодирование граничных стрелок

Коды ICOM связывают граничные стрелки на дочерней диаграмме со стрелками на родительском блоке. Определенные условные обозначения, называемые кодами ICOM, определяют тип



подсоединения. Буквы I (Input, Ввод), C (Control, Управление), O (Output, Вывод) или M (Mechanism, Механизм) указываются рядом со свободным концом каждой граничной стрелки на дочерней диаграмме. Эта кодировка определяет стрелку на родительском блоке как ввод, сигнал управления, вывод или механизм. За буквой следует цифра, указывающее относительное положение стрелки, подключенной к родительскому блоку, нумерация слева направо или сверху вниз. Например, запись "С3», на граничной стрелке дочерней диаграммы, указывает, что эта стрелка соответствует третьей стрелке управления (слева на право), входящей в родительский блок.

Это кодирование связывает каждую дочернюю диаграмму с ее собственным непосредственным родительским блоком. Если блоки на дочерней диаграмме детализированы на последующих дочерних диаграммах, то на каждой новой дочерней диаграмме назначаются новые коды ICOM, связывающие граничные стрелки этой диаграммы со стрелками на ее собственном непосредственном родительском блоке. Иногда буквенные ICOM коды, определяющие роли граничных стрелок (ввод, управление, механизм), могут меняться при переходе от родительского блока к дочерней диаграмме. На Рисунке 14 показан общий случай, когда роли действительно совпадают, например, вводные данные родительского блока совпадают с вводными данными дочерней диаграммы. В качестве примера изменения ролей управляющая стрелка в родительском блоке может быть вводом на дочерней диаграмме. Аналогично, ввод родительского блока может быть управлением для одного или более дочерних блоков. На Рис. 15 показаны примеры изменения ролей стрелок.



**Примечание:** штриховые линии показывают отношения между граничными стрелками и стрелками родительского блока

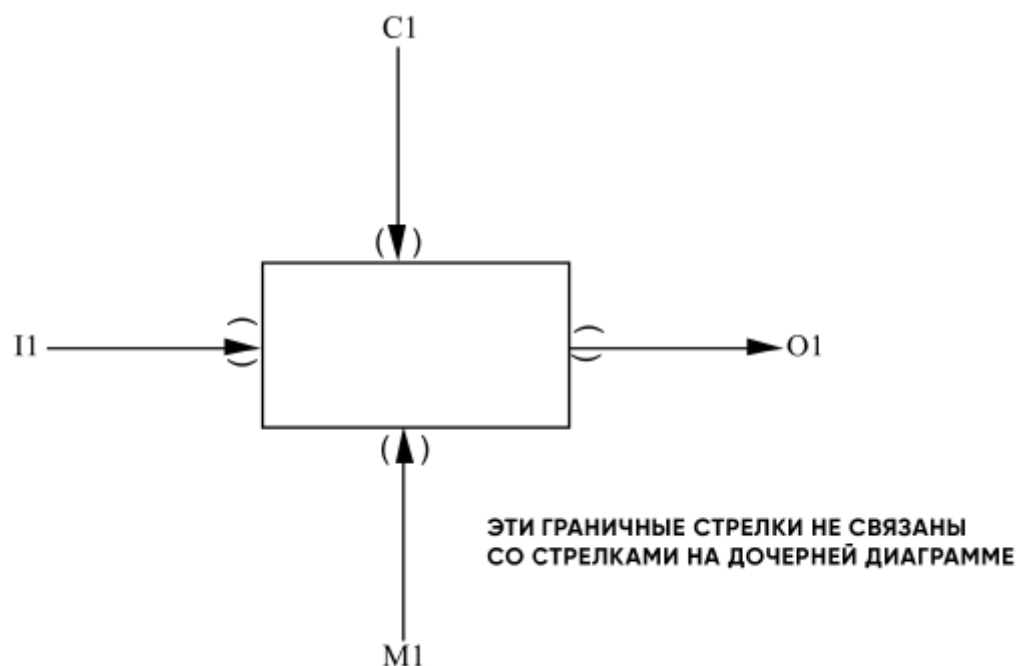
## Рисунок 15. Коды ICOM и изменение ролей стрелок

### 3.3.2.9 Стрелки туннелирования

Стрелка туннелирования используется для предоставления информации на определенном уровне декомпозиции, которая не требуется для анализа на некоторых других уровнях. Стрелка может быть туннелирована на любом выбранном уровне.

Стрелка туннелирования обозначается круглыми скобками в начале и/или конце стрелки (см. Рис. 16). Туннелирование означают, что данные или объекты, выраженные этими стрелками, не рассматриваются на родительской диаграмме и/или на дочерней диаграмме и не обязательны на следующем уровне декомпозиции. Но поскольку эта стрелка соответствует стрелке на родительской диаграмме, ей присваивается код ICOM. Этот код может использоваться в другом месте

модели, например, в ссылочном выражении на диаграмме, где отображается стрелка, чтобы идентифицировать местоположение исходного туннелирования. Стрелка, туннелированная на подсоединенном конце, может быть опущена с одного или нескольких уровней декомпозиции и затем вновь появиться на другом уровне, в одном или нескольких местах, туннелированных на неподсоединенном конце.

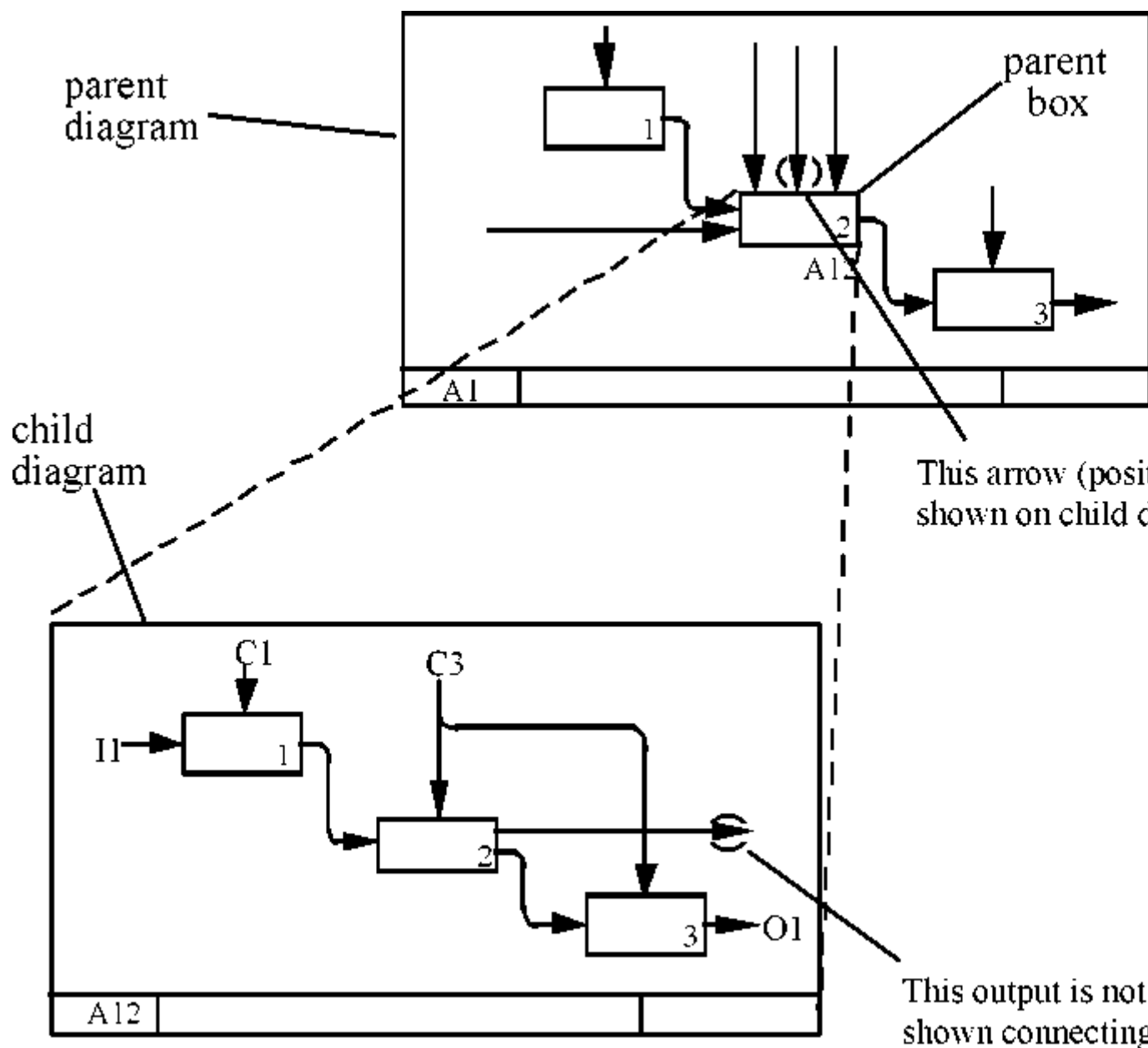


**Рисунок16. Стрелки туннелирования на подсоединенном конце**

Туннелирование стрелки на неподсоединенном конце означает, что данные или объекты не являются необходимыми на следующем более высоком (родительском) уровне и, следовательно, не должны отображаться подключенными к родительскому блоку. Данный случай изображен на Рисунке 17. Поскольку эта стрелка не соответствует стрелке на родительской диаграмме, ей не присваивается код ICOM. Стрелка может иметь прикрепленное примечание к модели, содержащее ссылку на ноду и код ICOM, который определяет местоположение «другого конца» туннеля. Кодирование ICOM для стрелки возобновляется на всех последующих дочерних диаграммах. На Рисунке 18 приведен пример стрелок туннелирования на родительской и дочерней диаграммах.



**Рисунок17. Стрелки туннелирования на неподсоединенном конце**



**Рисунок 18. Пример стрелок туннелирования**

### 3.3.2.10 Стрелка вызова

Стрелка вызова -это частный случай стрелки механизма. Это означает, что вызывающий блок не имеет своей собственной дочерней диаграммы для детализации, и скорее детализируется другим блоком (и его потомками) в той же или другой модели. Несколько вызывающих блоков могут вызывать один и тот же блок.

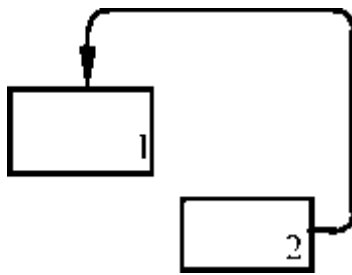
Стрелка вызова помечается ссылкой на узел диаграммы, содержащей вызываемый блок, а также номером вызываемого блока. Вызывающий блок может вызвать только один блок в данной активации. Однако в зависимости от условий, указанных в примечании модели, прикрепленном к стрелке вызова, вызывающий блок может выбрать один

из нескольких возможных вызываемых блоков. В этом случае метка стрелки вызова должна содержать список ссылок на ноды всех возможных вызываемых блоков.

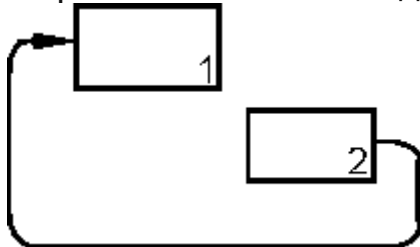
Стрелки вызываемого блока могут не совпадать со стрелками вызывающего блока, как по количеству, так и по значению. В этих случаях примечания к модели, прикрепленные к стрелкам вызова, должны указывать взаимосвязи таким образом, чтобы можно было дать правильную интерпретацию совместно используемым данным и объектам.

### 3.3.3 Правила построения диаграмм

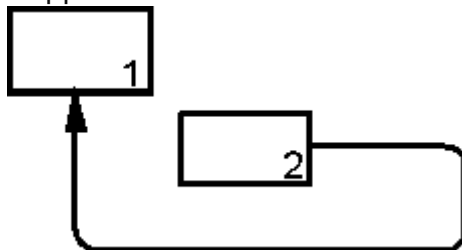
1. Контекстные диаграммы должны иметь номера нод  $A-n$ , где  $n$  больше или равно нулю.
2. Модель должна содержать контекстную диаграмму  $A-0$ , которая включает только один блок.
3. Номер единственного блока на контекстной диаграмме  $A-0$  должен быть 0.
4. Неконтекстная диаграмма должна содержать не менее трех и не более шести блоков.
5. Каждому блоку на неконтекстной диаграмме присваивается номер, который располагается внутри блока в нижнем правом углу. Порядок нумерации от верхнего левого к нижнему правому блоку (номера от 1 до 6).
6. Каждый блок, подвергнутый декомпозиции, должен иметь специальный ссылочный код на дочернюю диаграмму. Ссылка GRE (например, номер ноды, C-номер или номер страницы) помещается под правым нижним углом блока.
7. Стрелки должны быть нарисованы в виде горизонтальных и вертикальных отрезков прямой линии. Отрезки линий, расположенные по диагонали не использовать.
8. Каждый блок должен иметь как минимум одну стрелку управления и одну выводящую стрелку.
9. Блок должен иметь ноль или более вводящих стрелок.
10. Блок должен иметь ноль или более стрелок механизма без вызова.
11. Блок должен иметь 0 или 1 стрелок вызова.
12. Обратные связи по управлению должны быть показаны как «вверх и над»



Обратные связи по входу должны быть показаны как «вниз и под»

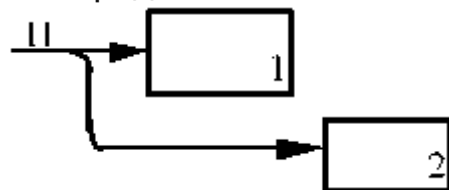


Обратные связи механизма должны быть показаны как «вниз и под»

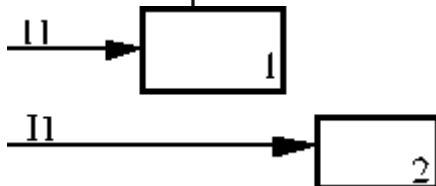


13. Неподсоединенный конец пограничной стрелки должен быть туннелирован или иметь соответствующий код ICOM, указывающий на его соединение с родительским блоком.
14. Открытые граничные стрелки, представляющие одни и те же данные или объекты, должны быть соединены через разветвление, чтобы показать все затронутые места, если это не приведет к потере наглядности и сложности анализа. Несколько источников, поставляющие одни и те же данные или объекты, должны объединяться, образуя единую выводящую граничную стрелку.

Так предпочтительнее



Чем такое решение



15. Названия блоков и метки стрелок не должны включать следующие слова: функция, деятельность, процесс, ввод, вывод, контроль или механизм.

### 3.3.4 Ссылочные выражения (коды)

В ссылочных выражениях используются коды, назначенные таким объектам модели, как диаграммы, блоки, стрелки и примечания.

Ссылочные выражения затем могут использоваться в различных контекстах для точного указания на нужный элемент модели.

Основное ссылочное выражение – номер ноды, который появляется там, где выполняется декомпозиция функционального блока и создается его подробное описание на дочерней диаграмме. Все остальные ссылочные коды базируются на номерах нод.

#### 3.3.4.1 Номера блоков

Каждому блоку на диаграмме присваивается номер, помещаемый в нижнем правом внутреннем углу блока. Эта система нумерации необходима для однозначной идентификации блоков в пределах диаграммы и для генерации номеров нод. Эти номера используются также для ссылок на блоки в тексте и глоссарии.

На контекстной диаграмме А-0 единственному блоку присваивается номер 0 (нуль). Номера блоков на всех других графических диаграммах должны принимать значение 1, 2, 3 до максимум 6, чтобы однозначно идентифицировать от трех до шести блоков на каждой такой диаграмме. Для блоков, расположенных по диагонали на диаграмме от верхнего левого угла до нижнего правого угла, блоки нумеруются по порядку, начиная с верхнего левого угла. Если некоторые блоки на диаграмме размещены не по диагонали, то сначала нумеруются «диагональные» блоки (также начиная с левого верхнего блока), а затем – «недиагональные» блоки, начиная с нижнего правого против часовой стрелки

#### 3.3.4.2 Номер ноды

Номер ноды базируется на положении блока в иерархии модели. Обычно номер ноды формируется добавлением номера блока к номеру диаграммы, на которой он появляется. Например, номер ноды блока 2 на диаграмме А25 равен А252. Все номера нод IDEF0 начинаются с заглавной буквы, например, «А». Когда родительский блок подробно описывается дочерней диаграммой, номера нод родительского блока и дочерней диаграммы совпадают.

Контекстные диаграммы и дочерняя диаграмма верхнего уровня



являются исключениями из приведенной выше схемы нумерации нод. Каждая модель IDEF0 имеет контекстную диаграмму верхнего уровня диаграмму A-0. Эта диаграмма содержит единственный «высший блок», который является уникальным родителем всей модели и имеет уникальный номер 0 (нуль) и номер ноды A0. Каждая модель IDEF0 должна также иметь по крайней мере одну дочернюю диаграмму, содержащую декомпозицию блока A0 на 3 ... 6 дочерних блоках. Этим блокам присваиваются уникальные номера нод A1, A2, A3, ... A6. Таким образом, последовательность [A0, A1,..., A2,..., A3,...] начинает нумерацию нод для любой модели.

Например, модель может иметь следующие номера нод:

A-1	Дополнительные контекстные диаграммы более высокого уровня
A-0(содержит высший блок A0)	Дополнительная контекстная диаграмма
Дочерняя диаграмма верхнего уровня A0	Обязательная контекстная диаграмма верхнего уровня
A1, A2, ..., A6	Дочерние диаграммы.
A11, A12, , A16, , A61, ... , A66	Дочерние диаграммы.
A111, A112, ..., Имя A161, A611, ..., A666	Дочерние диаграммы.

Номера нод также могут использоваться в качестве подробных ссылочных выражений для указания детализации родительского блока дочерней диаграммой. Если функция была декомпозирована (разложена на составляющие), то под нижним правым углом родительского блока можно записать номерноды дочерней диаграммы, в которой она подробно описана. На Рисунке 7, DRE (в данном случае номера нод) для блоков 1, 2 и 3 указывают на то, что они были детализированы и идентифицируют дочерние диаграммы.

#### 3.3.4.2.1 Перечень нод

Перечень нод представляет собой информацию о нодах в формате «списка». Все номера нод, наряду с названиями диаграмм или блоков, представляются в виде отступов, отражающих вложенную иерархическую структуру модели. Это позволяет расположить связанные диаграммы в порядке, используемом в обычном оглавлении, как показано на Рис. 19.

A0 Производить продукт

A1 Планировать производство

A11 Выбрать технологию производства

A12 Оценить требуемое время и затраты на производство

A13 Разработать производственные планы

A14 Разработать план вспомогательных мероприятий

A2 Разрабатывать и управлять графиком выпуска и ресурсами

A21 Разработать основной график

A22 Разработать график координации работ

A23 Оценивать затраты и приобретать ресурсы

A24 Следить за выполнением графика и расходом ресурсов

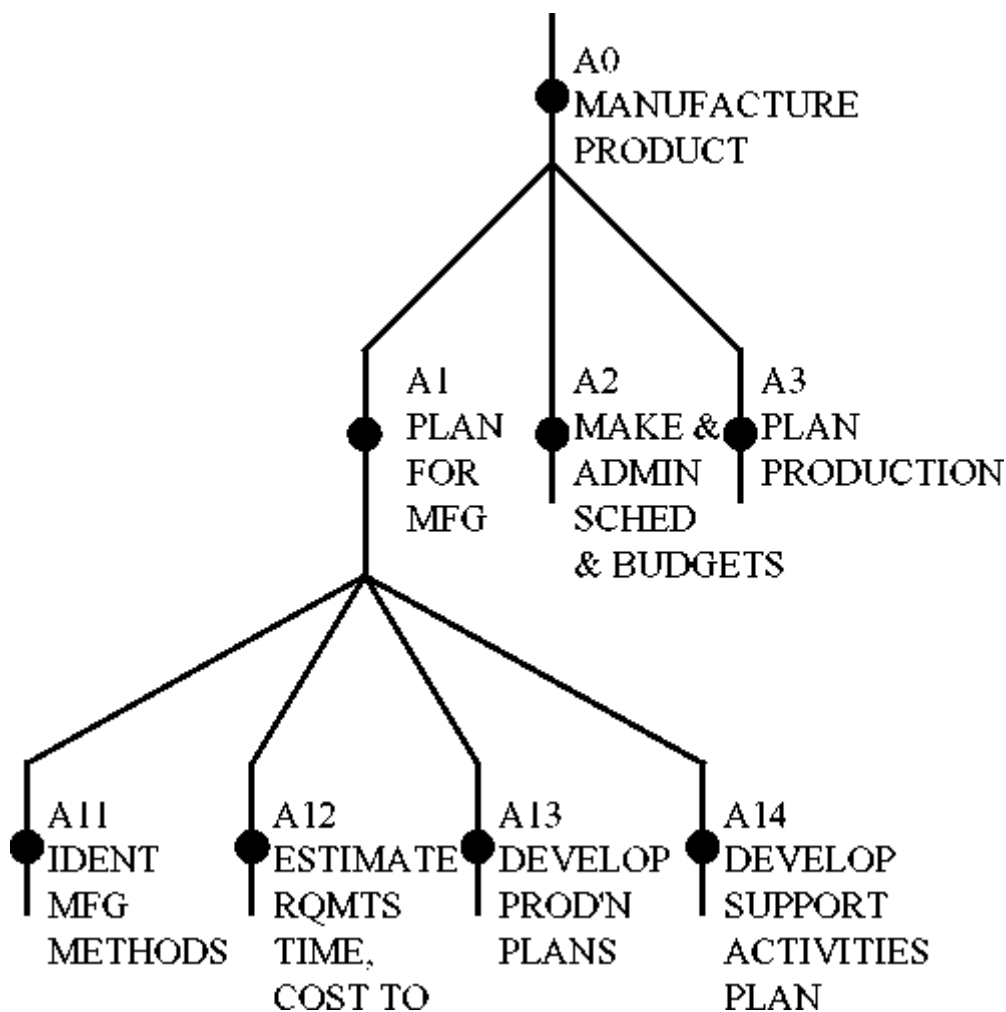
A3 Планировать выпуск продукции

### **Рисунок 19. 3.3.4.2.1 Перечень нод**

#### **3.3.4.2.2 Дерево нод**

Разработанная модель IDEF0 со всеми уровнями структурной декомпозицией может быть представлена на единственной диаграмме в виде дерева нод, дополняющего перечень нод. Использование дерева нод не является обязательным. Содержание дерева нод должно быть идентично содержанию перечня нод или любой части, представляющей интерес.

Не существует стандартного формата для отображения информации об ноде, за исключением того, что иерархия должна быть показана графически в виде дерева, имеющего корень в выбранной ноде (например, A0 для всей модели). На Рисунке 20 представлена иллюстрация.



**Рисунок 20. Типичное дерево узлов**

#### 3.3.4.3 Ссылки на ноды

Каждая диаграмма в модели имеет ссылку на ноду, которая однозначно идентифицирует ее и ее положение в иерархии модели. Ссылка на ноду состоит из сокращенного названия модели (см. раздел 3.4.5) и номера ноды диаграммы (см. раздел 3.3.4.2), разделенных косой чертой (/). Например, модель с именем «Операции обеспечения качества» может иметь аббревиатуру QA, а ссылка на ноду принимает вид QA/A312. В ссылке на диаграмму той же модели можно не указывать аббревиатуру имени модели, а только номер ноды диаграммы.

Ссылка на ноду может также иметь суффикс, например, F (для FEO), T (для текста) или G (для глоссария) и номер страницы. Например, ссылка на ноду для FEO может иметь вид QA / A321F1 (см. Раздел 3.4.4).

#### 3.3.4.4 Примечания к модели

Примечания к модели являются необязательными. Они обозначаются целым числом “n” внутри небольшого квадрата ( n ). Для данной диаграммы, номера примечаний должны образовывать последовательность, начиная с 1. Вертикальные отрезки, расположенные по бокам номера примечания (|n|), могут использоваться в качестве альтернативного обозначения.

Примечания к модели содержат информацию, относящуюся к сообщению диаграммы (и являющуюся его неотъемлемой частью), но не вписывающуюся в синтаксис прямоугольника и стрелки. Если текст примечания относится к нескольким местам или нескольким признакам диаграммы, то номер примечания в рамке (без текста) может быть скопирован и прикреплен с помощью тильды IDEF0 к каждой точке применения, но только на одной диаграмме, на которой появляется текст приложения модели.

#### 3.3.4.5 Справочная запись

При написании текста и примечаний для диаграмм и отдельных частей диаграмм используется стандартная запись. В ее основе лежат номера блоков, узловые номера, коды ICOM и номера примечаний. В следующей таблице приведены примеры справочных записей.

<b>СПРАВОЧНАЯ ЗАПИСЬ</b>	<b>ЗНАЧЕНИЕ</b>
<b>2И1</b>	<b>Блок 2 Ввод 1</b>
<b>О2</b>	<b>Граничная стрелка с кодом ICOM О2</b>
<b>2О2 к 3С1 или 2о2 к 3с1</b>	<b>Стрелка от 2О2 до 3С1 (I, О, С или М могут быть прописными или строчными буквами.)</b>
<b>И2 к 2И3 к 2О2 к (3С1 и 4С2)</b>	<b>От граничной стрелки с кодом ICOM И2 к Вводу 3 Блока 2, через активацию Блока 2, который формирует на Выводе 2 сигналы управления, поступающие через разветвление</b>

	<b>сигнала Управления 1 в Блок 3 и сигнала Управления 2 в Блок 4.</b>
<b>A21.3C2</b>	<b>На диаграмме A21 в этой модели см. Блок 3 Управление 2. Встроенный период означает «смотреть конкретно на».</b>
<b>A42. 3</b>	<b>На диаграмме A42 см. Примечание к модели 3.</b>
<b>A42. 3 </b>	<b>То же самое, что и выше, используя дополнительную нотацию (вертикальные отрезки, вместо примечания к модели в квадратике).</b>
<b>A42.3</b>	<b>На диаграмме A42 в этой модели см. Блок3.</b>
<b>MFG/A42.1</b>	<b>На диаграмме A42 модели, сокращенно MFG, см. Блок 1.</b>

### 3.4 Модели

#### 3.4.1 Описание модели IDEF0

Одной из наиболее важных особенностей IDEF0 как концепции моделирования является то, что она постепенно вводит более высокие уровни детализации через структуру диаграммы, описывающую модель. Таким образом, понимание улучшается за счет предоставления читателю на каждой диаграмме конкретной информации с управляемым количеством деталей для изучения.

Модель IDEF0 начинается с представления всего объекта в виде единого блока – блока с внешними граничными стрелками, связывающими его с функциями и ресурсами вне объекта. Единственный блок называется «высшим блоком» модели. (Высшему блоку

присваивается узловой номер A0.) Поскольку единственный высший блок модели IDEF0 представляет объект в целом, описательное имя в блоке является общим. То же самое справедливо и для внешних стрелок модели, поскольку они представляют собой полный набор внешних граничных условий объекта в целом, включая доступ к механизму, обеспечивающему дополнительные средства выполнения.

#### 3.4.2 Контекстные диаграммы

Диаграмма, в которой имеется высший блок A0, представляет контекст модели и называется контекстной диаграммой. Минимальный контекст для модели это специальная контекстная диаграмма с узловым номером A-0. Контекстная диаграмма A-0 имеет только один высший блок с именем A0 с обозначенными внешними стрелками, а также текстовые определения Точки зрения и Цели модели. (Диаграмма A-0 не имеет кодов ICOM или туннелирования на свободных концах стрелок.)

Иногда, однако, для того, чтобы обеспечить более полное представление о внешнем окружении контекста модели, приводится контекстная диаграмма A-1 (имеющая вид обычной, неконтекстной диаграммы). В контекстной диаграмме A-1 блок A0 занимает место одного из трех-шести пронумерованных блоков (другие блоки сохраняют предназначенные для них номера), поэтому эффект заключается в том, чтобы обеспечить полную родительскую диаграмму (с тремя – шестью блоками) для высшей модели-узлы A1-A6 которые все еще являются дочерними элементами первого поколения. В случае, если используется контекстная диаграмма A-1, контекстная диаграмма A-0 все равно представляется. Диаграмма A-0 по-прежнему имеет только один высший блок с именем A0 с помеченными внешними стрелками, а также текстовыми определениями Точки зрения и Цели модели.

Контекстные диаграммы это диаграммы, имеющие номера нод вида «A-n» (со знаком минус), где n больше или равно нулю. Обычные, неконтекстные диаграммы не имеют знака минус в своих номерах нод. Номер высшего блока модели (представляющего весь моделируемый объект) всегда равен 0. Блок под номером 0 должен присутствовать на обязательной контекстной диаграмме модели A-0, а также на дополнительной контекстной диаграмме A-1 (если таковая имеется), где он занимает место одного из блоков (от 1 до максимум 6) родительской диаграммы A-1 (всей модели). Таким образом, A0 всегда является (общим) номером ноды родительского блока и дочерней диаграммы для всей модели и всегда детализируется блоками с номерами нод A1, A2, A3,... A6.

При наличии только одного блока, A-0 является правильной контекстной диаграммой, но не является (правильной) родительской диаграммой. Правильные диаграммы имеют от трех до шести блоков. Родительский контекст-это то, что предоставляет или обозначает контекст для диаграммы вместо правильной родительской диаграммы. Родительский контекст диаграммы A0 это обязательный A-0, если нет контекстной диаграммы A-1. Если существует контекстная диаграмма A-1, то A-1 является правильным родителем диаграммы A0. Родительский контекст контекстной диаграммы A-0 всегда является «Высшим».

### 3.4.3 Контекстные диаграммы высокого уровня

Контекстные диаграммы высокого уровня имеют номера нод A-n, где n больше единицы. Таким образом, A-1 является контекстной диаграммой, правильным родителем (от A0), но не имеет высокого уровня. Для данного представления модели контекстная диаграмма высшего уровня (наибольшее n) имеет родительский контекст «NONE» (НЕТ), если только контекстная диаграмма высшего уровня не A-0.

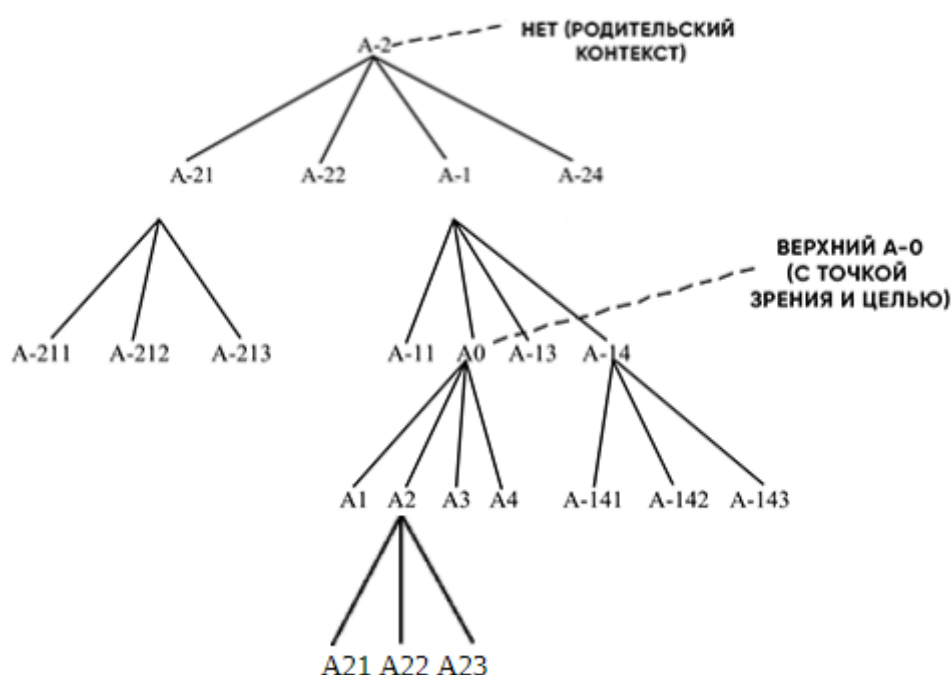
Каждая контекстная диаграмма высокого уровня, A-n, синтаксически является обычной детализированной диаграммой, за исключением того, что один из ее трех-шести блоков имеет номер, замененный на «минус N-1». В этом случае для A-1 этот блок является высшим блоком модели A0, а модель в целом (сам блок A0, родитель дочерних блоков), по-видимому, имеет родителя A-1, прародителя A-2 и т. д.

Предоставляя более полное описание окружающего контекста модели (правда, не во всех отношениях определенное, а только «типичное»), контекстное моделирование (характеризующееся отрицательной нумерацией узлов) обеспечивает более жесткие спецификации граничных условиях диаграммы A0 модели.

Контекстное моделирование выполняется так же, как и обычное детальное моделирование, с той лишь разницей, что отрицательная нумерация (не окончательная, но нормативная интерпретация) сохраняет A0 в качестве «источника» системы координат, основанной на номерах нод для всех ссылок на модель.

Моделирование с отрицательными номерами нод на деле предоставляет все больше и больше информации об источниках и использовании внешних граничных условий. Этот подход может не полностью соответствовать какой-либо конкретной среде. Данные контекстные диаграммы описывают "типичный" контекст.

На рисунке 21 представлена иллюстрация в виде дерева нод, чтобы показать, насколько информативным может оказаться контекст высокого уровня.



**Рисунок 21. Контекст с отрицательным номером ноды.**

#### 3.4.4 FEO, текст и глоссарий

Схема нумерации нод служит основой для согласования терминов FEO (диаграмма-иллюстрация0, текста и глоссария. В процессе разработки важно, чтобы каждый новый элемент информации был связан с нодой, которая инициировала данную информацию.

Для каждой формы (FEO, текст и глоссарий) нотация расширения нумерации нод состоит из одной буквы, добавляемой к



соответствующему номеру ноды. Например, номера нод для FEO должны содержать букву " F " (например, A312F).

Некоторые пользователи IDEF0 записывают определения глоссария в формы диаграмм IDEF0, хотя использование этой формы для глоссариев не требуется. В этом случае страница глоссария должна определять ключевые слова, фразы и аббревиатуры, используемые с определенным, связанной нодой IDEF0. Номера нод таких страниц глоссария должны содержать букву " G " для глоссария (например, A312G).

Аналогичным образом, некоторые пользователи IDEF0 записывают свои текстовые комментарии в формы диаграмм IDEF0, хотя использование этой формы для текста не требуется. В этом случае текстовая страница должна содержать текстовые комментарии для конкретной связанной ноды IDEF0. Номера нод таких текстовых страниц должны содержать букву " T " для текста (например, A312T).

Если существует более одной страницы FEO, глоссария или текста, связанной с данным узлом IDEF0, страницы должны быть обозначены дополнительным номером для уникальной идентификации каждой из них (например, A312F1, A312F2,... A312Г1, A312Г2,..., A312Т1, A312Т2, ...).

#### 3.4.5 Название модели

Каждая модель имеет уникальное описательное имя, которое отличает ее от других моделей, с которыми она может быть связана. Это имя модели обычно сокращается (уникально) для использования в ссылках на узлы. Например, название модели «Производственные операции» может быть сокращенно как MFG. Обсуждение ссылок на узлы см. в Разделе 3.3.4.3.

#### 3.4.6 Правила презентации

1. Когда есть текст, он должен сопровождать соответствующую графическую диаграмму.
2. В неопубликованных моделях глоссарий, относящийся к конкретной графической диаграмме, должен сопровождать

диаграмму и определять только ключевые слова, фразы и аббревиатуры, используемые с конкретным узлом.

3. В опубликованных моделях раздел глоссария определяет для всей модели, расположенные алфавитном порядке ключевые слова, фразы и аббревиатуры.
4. Если для модели предусмотрено оглавление, то оно должно быть представлено в виде дерева нод или перечня нод и содержать номера нод, названия диаграмм и названия блоков.

## ПРИЛОЖЕНИЕ А-КОНЦЕПЦИИ IDEF0

### А.1 Предыстория

Стремление ВВС США сократить расходы и сроки выполнения заказов путем оказания помощи аэрокосмической промышленности в ее усилиях по модернизации подтверждается многочисленными программами “Tech Mod” (Модернизация технологий). Аналогичная цель, но с использованием общепромышленной базы, а не отдельных компаний, была поставлена в рамках программы ICAM (Integrated Computer Aided Manufacturing, Интегрированная автоматизация производства). Целью ICAM была разработка “универсальных подсистем”, которые могли бы использоваться большим числом компаний для обеспечения значительного обновления отрасли в целом. Эти «подсистемы» обеспечивают поддержку таких общих функций, как управление информацией, планирование работы цехов и обработка материалов. Для достижения этой амбициозной цели необходим общий «базовый» коммуникационный аппарат, вокруг которого можно было бы планировать, разрабатывать и внедрять подсистемы в отдельных аэрокосмических компаниях. Исходная база получила название «Архитектура производства», поскольку она должна была предоставить отраслевую архитектуру», показывающую, как сегодня работает промышленность и какие универсальные подсистемы можно планировать, разрабатывать и внедрять.

Для разработки архитектуры необходим “язык”, на котором можно было бы выражать свои мысли и документировать текущие операции в аэрокосмической промышленности. В начале работы над программой ICAM Военно-воздушные силы опубликовали запрос на представление предложений по созданию архитектуры. В качестве понятного и единого языка была определена методика моделирования деятельности (где деятельность определялась как производственная ячейка или операционная единица). Чтобы решать задачи, язык должен был удовлетворять следующим критериям:

- Архитектура должна соответствовать производству и уметь представлять производственные операции естественным и простым способом.

- производственные операции естественным и простым способом. Не смотря на то что рассматриваемый предмет обширен и сложен, он должен предоставлять лаконичные решения и обеспечивать простое и быстрое нахождение ответов на интересующие вопросы.
- Поскольку язык используется различными специалистами, он должен предоставлять возможность общаться с широким кругом сотрудников аэрокосмической промышленности, а также с персоналом Программного офиса ICAM BBC.
- Язык должен служить основой при планировании, разработке и внедрении универсальных подсистем и обеспечивать достаточную строгость и точность для получения упорядоченных и правильных результатов.
- Поскольку исходные условия разрабатываются совместными усилиями представителей большого сегмента аэрокосмической промышленности, то язык как инструмент должен включать методологию (правила и процедуры) его применения, что позволит различным группам разрабатывать архитектурные элементы и широко распространять обзор, критику и одобрение.
- Поскольку исходные условия должны охватывать и учитывать всю аэрокосмическую отрасль, а не какую-либо одну компанию или отраслевой сегмент, то метод должен включать средства отделения «организации» от «функции»; то есть общее соглашение не может быть достигнуто, если организационные различия отдельных компаний не выделены и не выработано и не принято общее функциональное решение.

## A.2 Концепции IDEF0

Первоначальная методология IDEF0 включала в себя основные концепции, которые учитывали каждую из перечисленных выше задач.

Основные концепции IDEF0:

1. Графическое представление моделирования деятельности. Изображение «блок и стрелка» на диаграмме IDEF0 обозначают производственную операцию в виде блока, а взаимосвязи к / от при выполнении операции как стрелки, вводящие/ выводящие из блока. Чтобы иметь возможность отражать реальные производственные операции, блоки должны контактировать с другими функционирующими блоками посредством интерфейсных стрелок, обеспечивающих «ограничения» относительно того, когда и как запускаются и управляются операции.
2. Краткость. Документация производственной архитектуры должна быть краткой, чтобы охватить предмет исследования. Линейная, многословная характеристика текста на обычном языке явно недостаточна. Двухмерная форма, предоставляемая в виде чертежа, имеет желаемую лаконичность, не теряя при этом возможности выражения таких отношений, как интерфейсы, обратная связь и пути ошибок.

3. Обмен информацией. Существует несколько концепций IDEF0, которые предназначены для улучшения обмена информацией:

- Диаграммы, основанные на очень простой графике блоков и стрелок.
- Английский текст для описания назначения блоков (функции) и стрелок (данные или объекты).
- Последовательная экспозиция деталей, отображающая иерархию с основными функциями на верхнем уровне и подфункциями на последующих уровнях, хорошо продуманная детализация.
- Индекс ноды для определения местоположения деталей в иерархической структуре диаграмм.
- Ограничение детализации на каждой последующей диаграмме не более чем шестью подфункциями для удобства чтения.

Диаграммы, сопровождаемые текстом и глоссарием, повышают точность графического представления.

4. Строгость и точность. Правила IDEF0 обеспечивают достаточную строгость и точность для удовлетворения потребностей архитектуры ICAM без чрезмерного ограничения разработчика.

Правила IDEF0 включают:

- Тщательный контроль экспозиции на каждом уровне (правило 3-6 блоков).
- Ограниченный контекст (никаких пропусков или дополнительных неохваченных деталей).
- Синтаксические правила для графики (прямоугольники и стрелки).
- Уникальность имен и меток на диаграмме.
- Связность диаграмм (продуманный специальный ссылочный код [DRE]).
- Взаимодействие данных / объектов (коды ICOM и стрелки туннелирования).
- Разделение ввода и управления (правило определения роли данных или объектов).
- Минимальный контроль функции (все функции требуют по крайней мере одного контроля).
- Ограничение разветвления стрелки (разветвление или соединение) (метки для сегментов стрелки).
- Требования к меткам стрелок (минимальные правила маркировки).
- Цель и Точка зрения (все модели должны иметь формулировку цели и точки зрения).

5. Методология. Пошаговые процедуры предусмотрены для моделирования, анализа и обсуждения задач.
6. Организация и функции. Отделение организации от функции входит в цель модели и осуществляется путем выбора функций и меток стрелок при разработке модели. Постоянный обзор во время разработки модели гарантирует, что организационные точки зрения будут исключены.

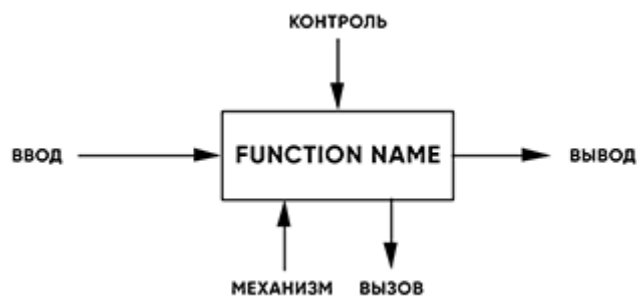
### А.3 Обсуждение отдельных концепций IDEF0

В остальных подразделах приводятся описания некоторых основных концепций, чтобы прояснить их и показать полезность.

#### А.3.1 Графика моделирования деятельности

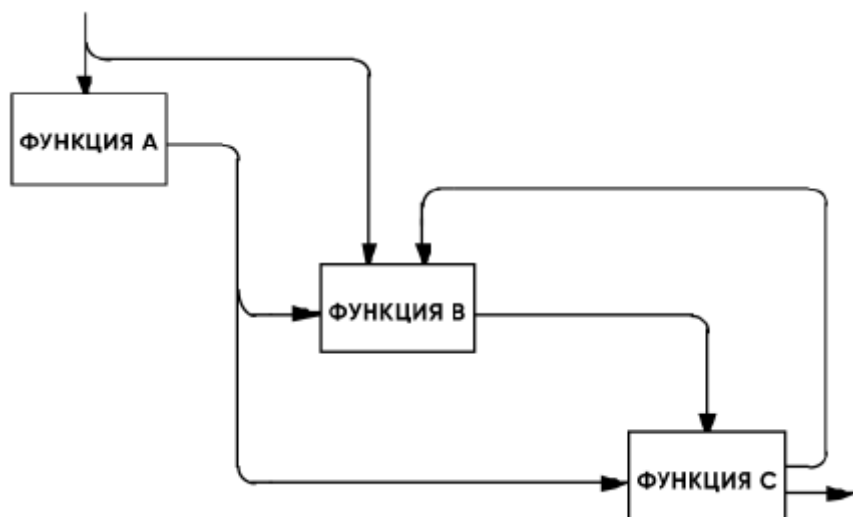
Методология IDEF0 может быть использована для моделирования широкого спектра автоматизированных и неавтоматизированных "систем" или объектов, включая любую комбинацию аппаратных средств, программного обеспечения, машин, процессов или людей. Для новых систем IDEF0 может использоваться сначала для определения требований и характеристик функций, а затем для разработки реализации, которая отвечает требованиям и выполняет назначенные функции. Для существующих систем IDEF0 может использоваться для анализа функций, выполняемых системой, и регистрации механизмов (средств), с помощью которых они выполняются.

Результатом применения IDEF0 является модель. Модель состоит из диаграмм, текста и глоссария, имеющих перекрестные ссылки друг на друга. Диаграммы являются основными компонентами модели. Все функции и связи представлены на диаграммах в виде прямоугольников-блоков (функций) и стрелок (интерфейсов данных или объектов). Место, в котором стрелка соединяется с блоком, отражает конкретную роль интерфейса. Сигнал управления подсоединяется к верхней части блока. Ввод, данные или объекты, на которые воздействует операция, подключаются к блоку слева. Выводы операции подсоединены к блоку справа. Стрелки, подключенные к нижней стороне блока, представляют механизмы. Стрелки, направленные вверх, идентифицируют средства, поддерживающие выполнение функции. Другие средства могут наследоваться из родительского блока. Стрелки механизма, направленные вниз, являются стрелками вызова. Стрелки вызова обозначают обращение из данной модели или из данной части модели к блоку, входящему в состав другой модели или другой части модели, обеспечивая их связь, т.е. разные модели или разные части одной и той же модели могут совместно использовать один и тот же элемент (блок). Положения стрелок показаны на Рисунке А1.



### Рисунок А1. Функциональный блок и стрелки данных / объектов

Значения блоков и стрелок используются для организации связи нескольких подфункций на диаграмме, содержащей более общую функцию. Эта диаграмма представляет собой «диаграмму ограничений», на которой отображаются конкретные связи, ограничивающие каждую подфункцию, а также источники и цели связных ограничений (Рис.А2).



### Рисунок А2. Диаграммы ограничений

На рисунке А2 функция В ограничена одним вводом и двумя сигналами управления и формирует один вывод, который ограничивает функцию С. Здесь термин «ограничения» означает, что функция использует данные или объекты, подводимые к блоку, и, следовательно, ограничена в работе интерфейсом; функция не может действовать до тех пор, пока не будет предоставлено содержимое интерфейсной стрелки, а способ работы функции зависит от деталей (значения, числа и т.д.) содержимого интерфейсной стрелки.

#### А.3.2 Обмен информацией путем последовательного введения деталей

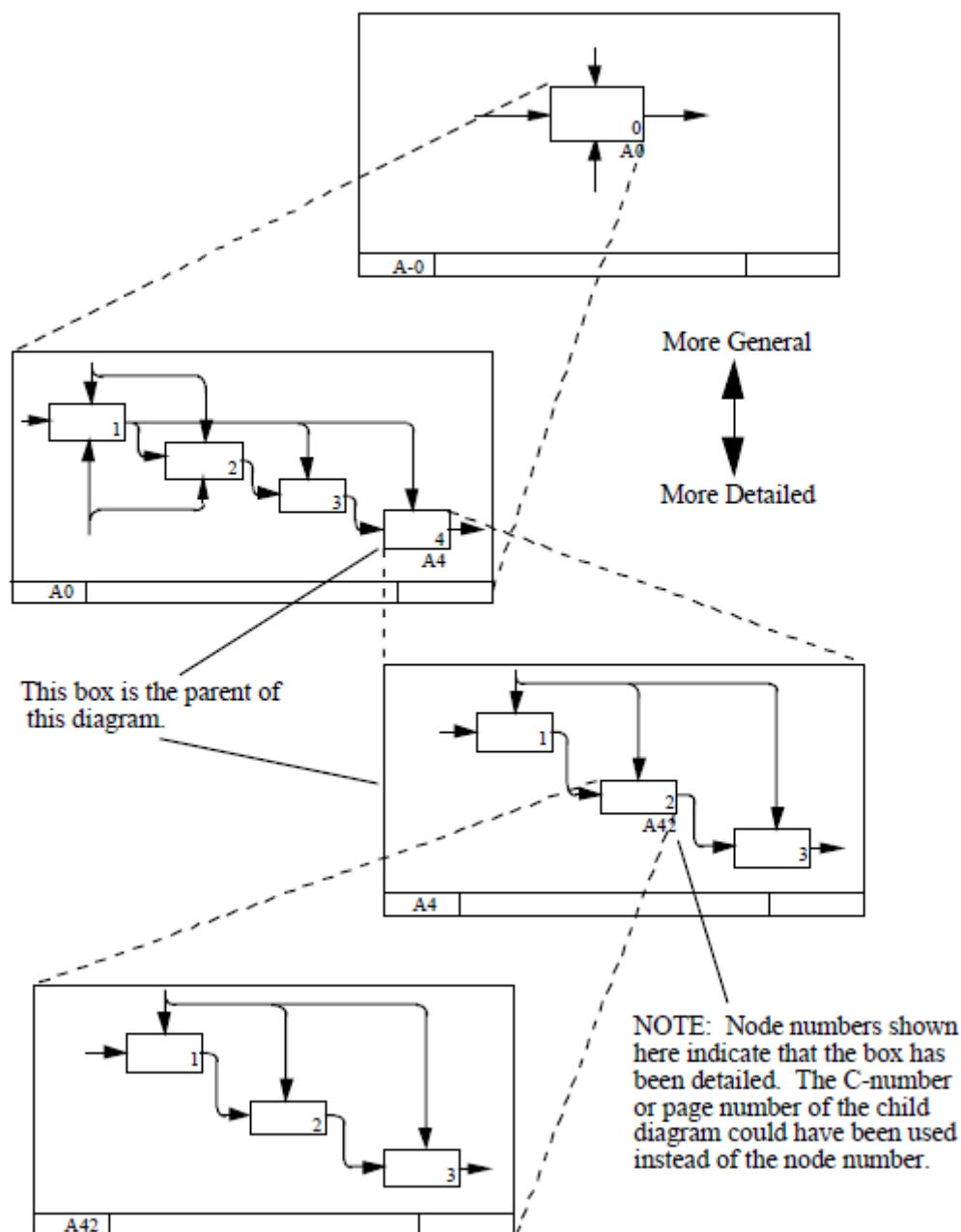
Одной из наиболее важных особенностей IDEF0 как концепции моделирования является то, что она постепенно вводит более высокие уровни детализации через структуру диаграммы, описывающую модель. Таким образом, понимание улучшается за счет предоставления читателю на каждой диаграмме конкретной информации с управляемым количеством деталей для изучения.

Структура модели IDEF0 показана на Рисунке А3. Серия из четырех диаграмм с отношением каждой диаграммы к другим.

Модель IDEF0 начинается с представления всей системы в виде одного блока со стрелками, взаимодействующими с функциями вне системы. Поскольку единственный блок представляет весь объект, его имя – общее для всего проекта. Это справедливо и для всех стрелок диаграммы, поскольку они представляют полный комплект внешних интерфейсов объекта. Диаграмма с единственным блоком называется «контекстная диаграмма» и определяет в тексте Точку зрения и Цель модели

Блок, представляющий систему в виде отдельного модуля, затем детализируется на другой диаграмме блоками, соединенными интерфейсными стрелками. Эти блоки представляют собой основные подфункции одной родительской функции. Декомпозиция отражает полный набор подфункций, каждая из которых представлена в виде блока, границы которого определяются стрелками интерфейса. Каждая из подфункций может быть аналогично декомпозирована (разложена), чтобы отразить еще больше деталей. В IDEF0 мы используем следующую терминологию: функции «декомпозированы», а блоки, представляющие функции, «детализированы».

Блок, если он детализирован, всегда детализируется на дочерней диаграмме не менее чем на три блока, но не более чем на шесть блоков. Верхний предел до шести блоков заставляет использовать иерархию для описания сложных объектов. Нижний предел от трех блоков гарантирует, что введено достаточно деталей, чтобы сделать декомпозицию (детализацию) интересной.



### Рисунок А3. Структура модели IDEF0

Каждая диаграмма в модели показана в точном отношении к другим диаграммам с помощью стрелок взаимосвязи. Когда функция декомпозируется на подфункции, связи между подфункциями отображаются в виде стрелок. Имя каждого блока подфункций плюс обозначенные связи определяют ограниченный контекст для данной подфункции.

Во всех случаях каждая подфункция ограничена только теми элементами, которые находятся в пределах ее родительской функции. Подфункции дискретны и не перекрываются. Кроме того, набор подфункций не может не учитывать какие-либо элементы. Таким образом, как уже указывалось, родительский блок и его интерфейсы обеспечивают контекст для дочерней диаграммы. За исключением



туннельных стрелок, ничто не может быть добавлено или удалено с обозначенной границы.

#### А.3.3 Дисциплинированная командная работа

Методология IDEF0 включает процедуры разработки и критики моделей большой группой людей, а также интеграции вспомогательных подсистем в Архитектуру IDEF0. Дополнительные вспомогательные процедуры, такие как правила и процедуры для библиотек и обзорный цикл (см. Приложение С), также включены в методологию IDEF0.

(Следует отметить, что некоторые из этих правил и процедур, такие как «Цикл Комплекта» или «Цикл критики читателя-разработчика», также используются в других методиках IDEF).

Создание модели IDEF0 является основной процедурой «дисциплинированной командной работы». Создание модели -это динамичный процесс, который обычно требует командной работы. На протяжении всего проекта авторы создают исходные диаграммы, которые распространяются среди участников проекта для рассмотрения и комментариев. Порядок работы требует, чтобы каждый человек, от которого ожидают комментариев к диаграмме, делал их в письменной форме и представлял их автору диаграммы. Автор отвечает также письменно. Этот цикл продолжается до тех пор, пока диаграммы и, в конечном счете, вся модель не будут официально приняты.

IDEF включает в себя процедуры по сохранению письменных отчетов обо всех решениях и альтернативных подходах по мере их реализации в рамках проекта. Копии диаграмм, созданных автором, подвергаются критике со стороны компетентных рецензентов, которые документируют предложения непосредственно на копии. Авторы отвечают на каждый комментарий в письменной форме на том же экземпляре. Предложения принимаются или отклоняются в письменной форме вместе с приведенными аргументами. По мере внесения изменений и исправлений устаревшие версии диаграмм сохраняются в файлах проекта.

Диаграммы корректируются в процессе учета исправлений и комментариев. Более подробная информация добавляется к модели путем создания большего количества диаграмм, которые также пересматриваются и изменяются. Окончательная модель представляет собой соглашение о представлении системы или объекта с согласованной Точкой зрения и обозначенной Целью. Это представление может быть легко прочитано другими участниками, не входящими в первоначальный проект, и использовано для представления определения системы в кратких стенд-ап-брифингах или текущих заседаниях, а также для организации новых проектов и работы над системными изменениями.

# ПРИЛОЖЕНИЕ В СОЗДАНИЕ И ИНТЕРПРЕТАЦИЯ ДИАГРАММ IDEF0

## В.1 Чтение диаграмм IDEF0

Модель состоит из набора диаграмм и связанных с ними материалов, расположенных в иерархическом порядке. Должен быть представлен перечень нод (или оглавление). Размещение диаграмм в иерархическом порядке дает общее представление о модели и позволяет получить доступ к любой ее части.

Чтение выполняется сверху вниз, рассматривая каждую диаграмму как контекст, ограниченный родительским блоком. После чтения диаграмм верхнего уровня читаются диаграммы первого уровня, затем диаграммы второго уровня и т. д. Если требуются конкретные сведения о модели, используется перечень нод для спуска по уровням к требуемой диаграмме.

При публикации модель отображается в формате «пара страниц» и порядке „перечень нод“. Формат „пара страниц“ означает, что каждая диаграмма и весь текст, связанный с ней, отображаются на паре развернутых страниц. (Рисунок В1.)

Helpful text  
discussing the  
current  
diagram.

This is the  
current  
diagram.

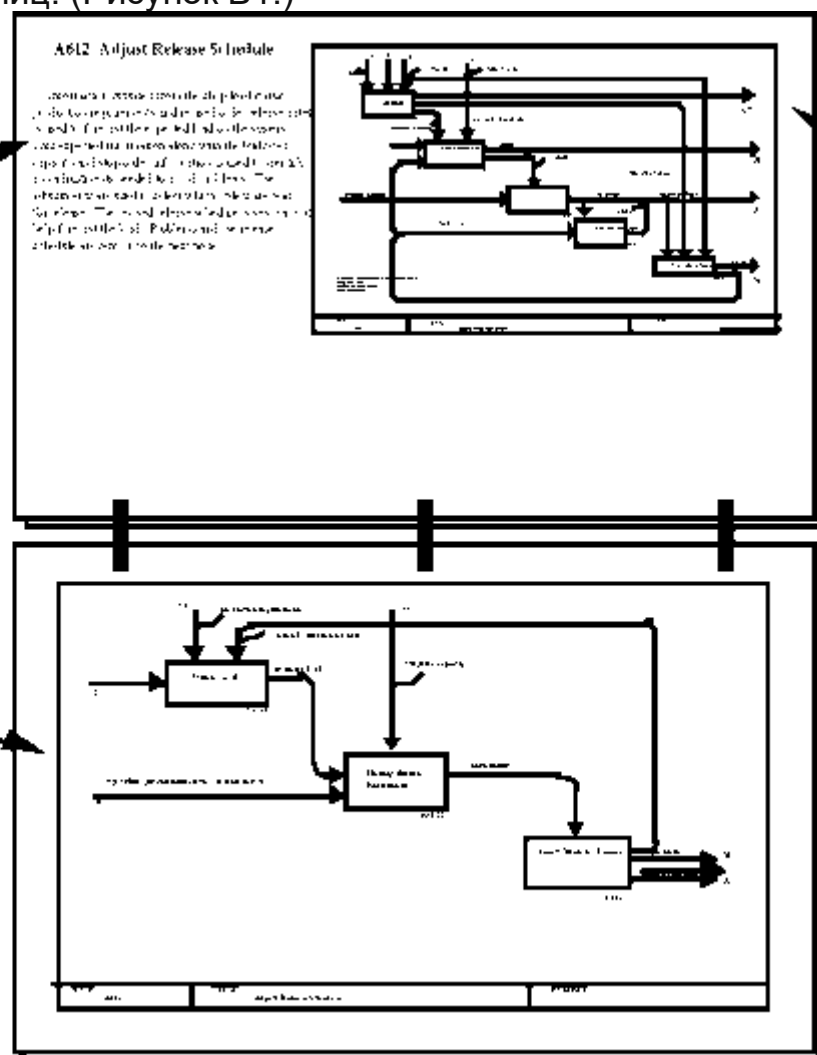


Рисунок В1. Формат « Пара страниц»

Порядок «перечень нод» означает, что все дочерние диаграммы, относящиеся к одному блоку на диаграмме, представляются перед дочерними элементами следующего блока. Это позволяет расположить связанные диаграммы в порядке, используемом в обычном оглавлении, как показано на Рисунке В2.

Меньшая диаграмма является родительской для текущей диаграммы.

A0 Планировать производство

A1 Предположить структуру и метод Mfg.

A2 Оценить требования, стоимость, время для производства

A21 Оценить потребность в ресурсах


A22 Оценить затраты на покупку или изготовление

A23 Оценить сроки запуска и производства

A3 Разработать производственные планы

A4 Разработать план вспомогательных мероприятий

Порядок расположения  
диаграмм в документе



## **Рисунок В2. Перечень узлов, определяющий порядок диаграмм**

### **В.1.1 Подход к разработке модели**

Модели представляют собой краткое описание всей системы или объекта, а также детали конкретного объекта. Чтобы прочитать модель с целью ознакомления, используйте индекс, для поиска диаграмм высокого уровня. (Рисунок В3.)

A0 Производить продукт

A1 Планировать производство

A11 Предположить структуру и метод Mfg.

A12 Оценить требуемое время и затраты на производство

A13 Разработать производственные планы

A14 Разработать план вспомогательных мероприятий

A2 Составить администрирование расписаний и бюджетов

A21 Разработать основной график

A22 Разработать график координации работ

A23 Оценивать затраты и приобретать ресурсы

A24 Следить за выполнением графика и расходом ресурсов

A3 Планировать выпуск продукции

Для подробного ознакомления с моделью воспользуйтесь индексом, чтобы найти все диаграммы, детализирующие интересующую вас тему (Рисунок В4.)

A0 Производить продукт

A1 Планировать производство

A11 Предположить структуру и метод Mfg.

A12 Оценить требуемое время и затраты на производство

A13 Разработать производственные планы

A14 Разработать план вспомогательных мероприятий

A2 Составить администрирование расписаний и бюджетов

A21 Разработать основной график

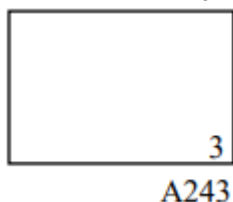
A22 Разработать график координации работ

A23 Оценивать затраты и приобретать ресурсы

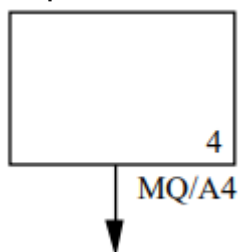
A24 Следить за выполнением графика и расходом ресурсов

A3 Планировать выпуск продукции

Дальнейшую детализацию в модели можно проследить, обратившись к детальному ссылочному выражению (DRE), расположенному чуть ниже номера блока. В нем указаны номер узла, С-номер или номер страницы дочерней схемы, которая детализирует блок. В приведенном ниже примере сведения для графы 3 на диаграмме A24 можно найти на диаграмме с узловым номером A243. Если DRE отсутствует, то блок еще не детализирован.



Детали могут использоваться как внутри модели так и совместно разными моделями. В обоих случаях стрелка вызова (направленная вниз) указывает, где имеется совместно используемая детализация через выражение ссылки, которое может включать в себя уникальное сокращенное название модели. В приведенном ниже примере блок 4 детализирован диаграммой A4 в модели MQ. В этом примере ссылочное выражение является ссылкой на узел диаграммы.



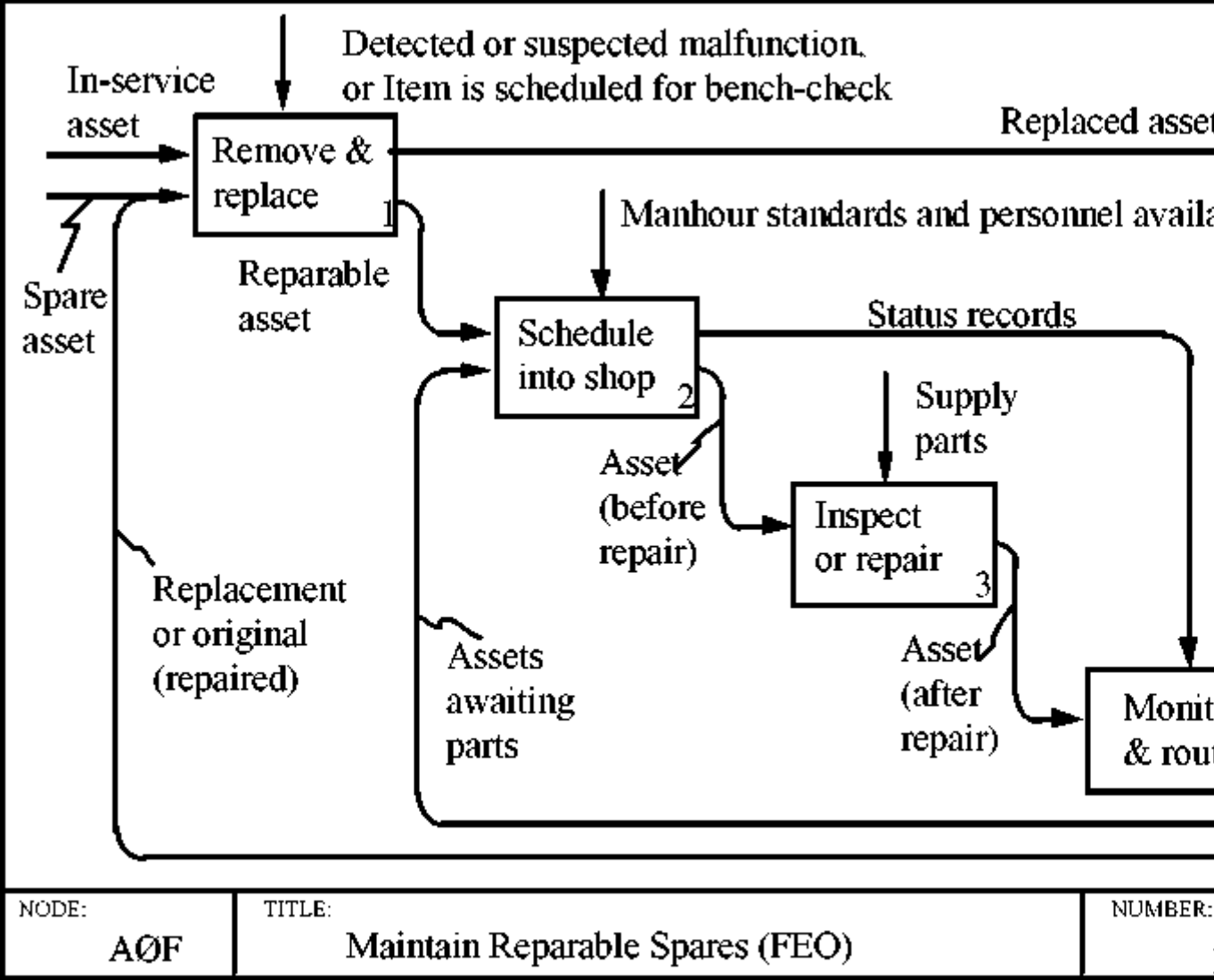
#### В.1.2 Порядок чтения диаграммы

Точная информация о системе содержится в самих диаграммах.

Рекомендуется следующая последовательность чтения:

1. Просмотрите блоки диаграммы, чтобы получить представление о том, что описывается.
2. Вернитесь к родительской диаграмме и обратите внимание на соединения стрелок с родительским блоком. Попробуйте определить "самый важный" ввод, сигнал управления и вывод.
3. Изучите стрелки текущей диаграммы. Попробуйте определить, существует ли основной путь, связывающий "самый важный «ввод или сигнал управления и» самый важный " вывод.
4. Мысленно пройдите по схеме сверху вниз слева направо, используя в качестве ориентира основной путь. Обратите внимание, как другие стрелки взаимодействуют с каждым блоком. Определите, существуют ли другие пути. Проверьте историю, рассказанную диаграммой, рассмотрев, как разрешаются знакомые ситуации.
5. Проверьте, существует ли связанная диаграмма иллюстрация FEO.
6. Наконец, внимательно прочтите текст и глоссарий, если таковые имеются.

Эта последовательность гарантирует, что основные характеристики каждой диаграммы не будут упущены. В тексте обращается внимание на все, что разработчик хочет подчеркнуть. В глоссарии приведено авторское толкование используемой терминологии. Каждая диаграмма имеет центральную тему: от самой важной вводящей граничной стрелки до самой важной выводящей граничной стрелки. Главный путь через блоки и стрелки описывает в общих чертах основную функцию диаграммы. (Рисунок В5.) Другие части диаграммы представляют квалифицирующие или альтернативные условия, которые являются вторичными по отношению к основному пути.



**Figure B5. Example of Main Path**

**Рисунок В1. Формат " Пара страниц"**

Работу системы можно мысленно представить, следуя по главному пути. Конкретные виды вводных данных, обработка ошибок и возможные альтернативные выводы придают объяснительной записке детализацию. Это пошаговое руководство улучшает понимание диаграмм.

В.1.3 Семантика блоков и стрелок

Фундаментальное понятие, которым следует руководствоваться при интерпретации любой диаграммы или набора диаграмм: обязательно подразумевается только то, что явно указано. Это вытекает из самой природы диаграмм ограничений. Неопределенные ограничения не должны учитываться; необходимые ограничения должны быть явными. Отсюда следует: любая дальнейшая детализация, не запрещенная явно, неявно разрешена.

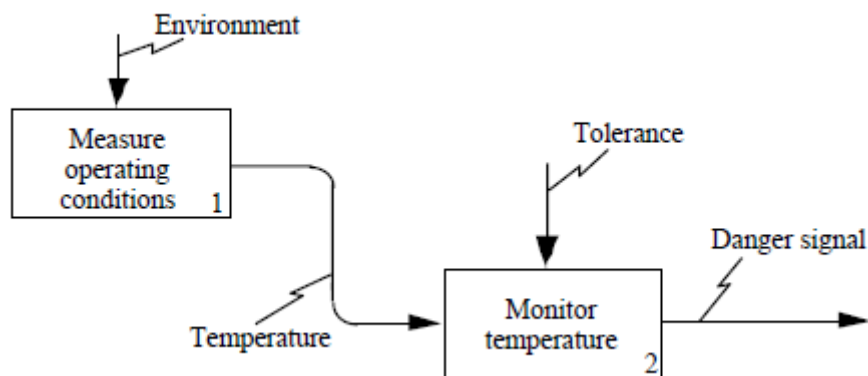


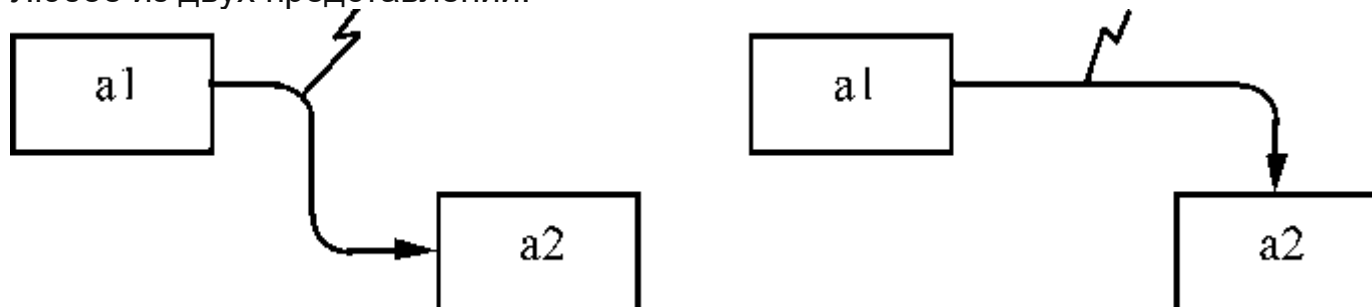
Рисунок В6. Пример ограничений

С помощью рисунка В6 можно сделать предположение, что температура измеряется “достаточно часто” и допуски изменяются “при необходимости”, а температура контролируется в соответствии с допусками “достаточно часто”, чтобы сигнал опасности был получен “достаточно быстро”. Ни одно из этих интуитивных толкований не противоречит последующей детализации, которая показывает, что: температура измерялась путем периодического отбора проб, или текущие допуски были запрошены только тогда, когда температура увеличилась на некоторую фиксированную величину, или показания температуры, полученных с помощью блока 1, хранились в блоке 2, который оценивал характер изменения, чтобы определить, находится ли этот образец в пределах допусков и т. д.

Графические обозначения диаграммы сами по себе абстрактны. Тем не менее, они делают важные фундаментальные различия. Их абстрактная природа не должна умалять предполагаемую широту возможных допускаемых толкований.

В.1.3.1 Как и когда не учитываются ограничения

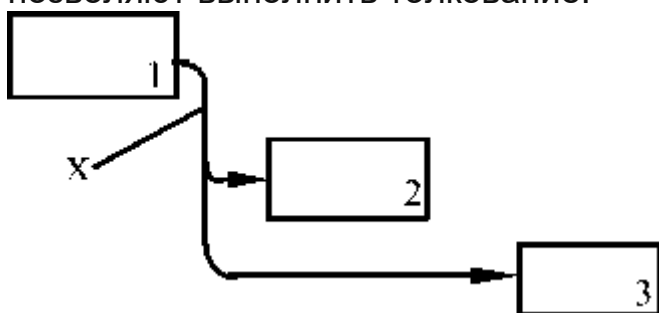
Любое из двух представлений:



говорит, что: действие a2 зависит от «d», которое создано или изменено действием a1.

Каждое представление определяет связь ограничений между двумя блоками. Все, что явно указано промежуточной стрелкой для любого представления, выражается следующим образом: для некоторой активации блока 2 требуется нечто, называемое «d», которое возникает при некоторой активации блока 1.

Часто диаграммы четко указывают на то, что для двух или более блоков может понадобиться содержимое стрелки. Смысл блоков и стрелок, показанных на Рисунке В7, заключается в том, что нечто, произведенное блоком 1, необходимо блоку 2 и блоку 3. Возможно, что активация “источника” стрелки (блок 1) должна предшествовать активации “пункта назначения” (блок 2 или блок 3). Может случиться так, что одной активации источника достаточно для активации любого пункта назначения. Без дополнительной информации, только блоки и стрелки позволяют выполнить толкование.

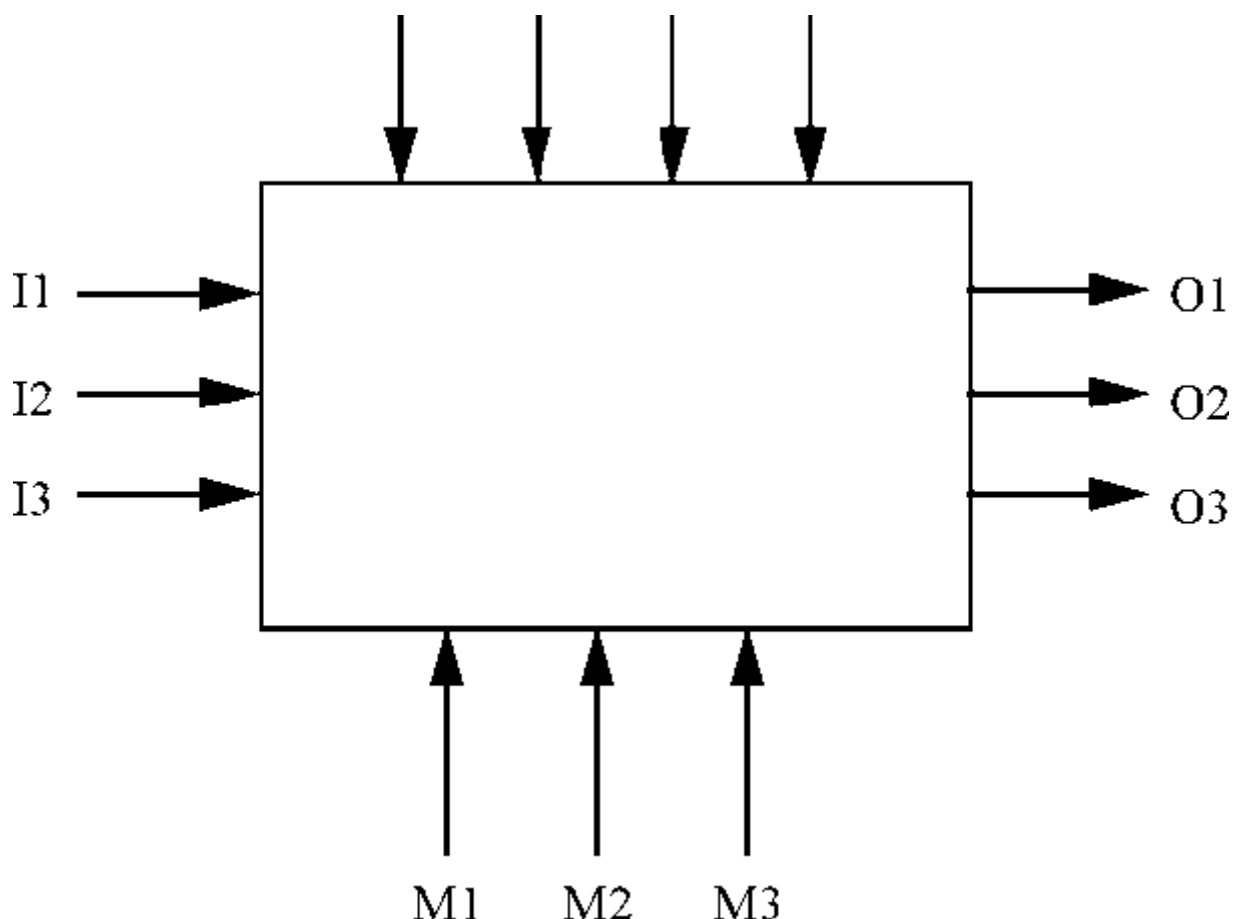


**Рисунок В7. Два блока, использующие содержимое одной и той же стрелки**

В.1.3.2 Несколько вводов, сигналов управления и выводов

Основная интерпретация блока, показанного ниже (Рис.В8), заключается в следующем: для получения любого подмножества данных вывода [O1, O2, O3] может потребоваться любое подмножество данных ввода [I1, I2, I3, C1, C2, C3, C4, M1, M2, M3]. В отсутствие дополнительной детализации нельзя предположить, что:

- а. любой вывод может быть сформирован без данных ввода, или
- б. Для формирования вывода требуются все входные данные

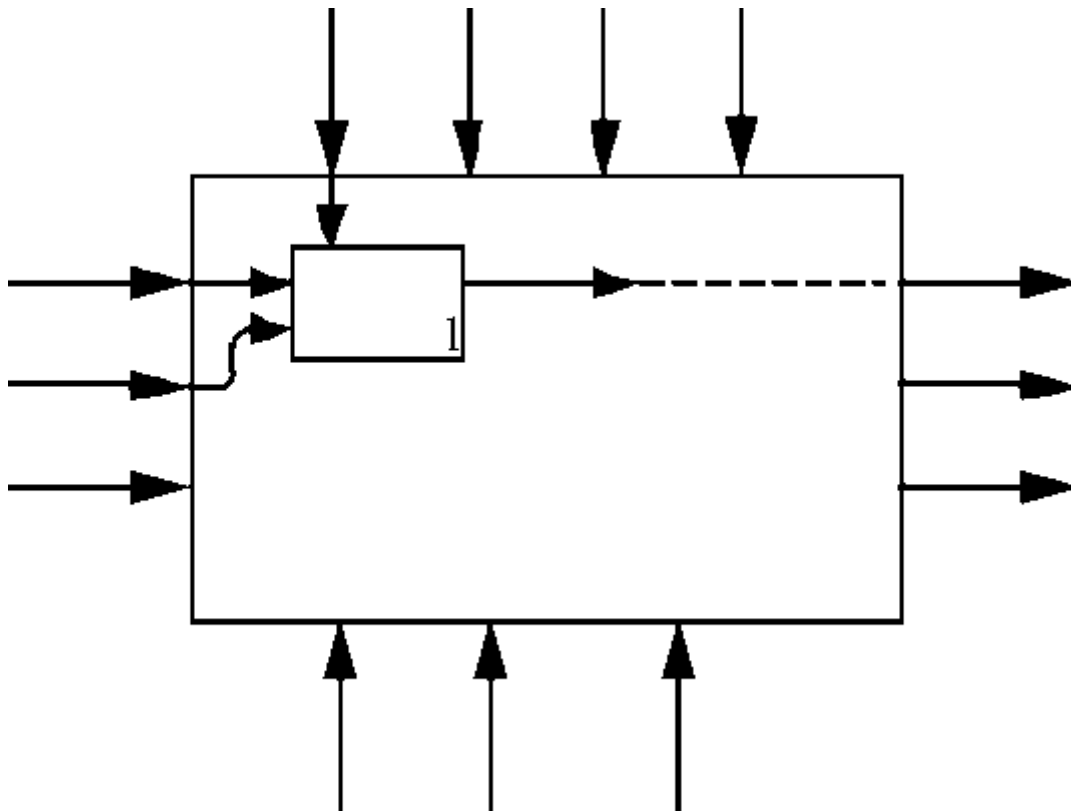


**Рисунок В8. Иллюстрация кодирования ICOM**

Частичная детализация предыдущего блока (показанного на рис. В9, как это может выглядеть на диаграмме иллюстрации FEO) указывает на то, что I3, C2, C3 и C4 не требуются для получения O1. Рисунок В9 иллюстрирует что:

- а. некоторая форма дальнейшей детализации определит точное зависимость выводов от вводов и сигналов управления;
- б. до тех пор, пока не будет представлена подробная детализация, не следует делать ограничительных предположений о выполняемых действиях «внутри» каждого блока;
- с. чтение диаграммы должно быть сосредоточено на стрелках, которые являются явными, а не на именах блоков, которые являются неявными.





**Рисунок В9. Диаграмма-иллюстрация FEO, представляющий детализацию нескольких ICOM**

В.2 Руководство разработчика по созданию диаграмм IDEF0

При создании любой диаграммы IDEF0 необходимо выполнить следующие требования:

- а. её назначение и точка зрения должны соответствовать заявленной цели и точке зрения общей модели;
- б. её граничные стрелки должны соответствовать стрелкам, которые соединяются с родительским блоком;
- с. её содержимое должно точно соответствовать содержимому родительского блока

В.2.1 Основные этапы разработки

Пошаговый порядок разработки позволяет создавать диаграммы и полезные и понятные модели. Порядок действий следующий:

- а. Определить связи объекта более точно, чем предполагает название функционального блока. Это делается с помощью списка данных и объектов, которые подвергаются воздействию или обработке функцией.
- б. Изучить ограниченный набор объектов и сформировать возможные подфункции полной функции.
- с. Определить естественные схемы соединения этих подфункций.
- д. Выполнить разделение или объединение подфункций для создания необходимых блоков.
- е. Нарисовать окончательный вариант диаграммы, обратив внимание на компоновку и ясность.

В 2.1.1 Выбор контекста, точки зрения и цели

Перед началом работы над любой моделью важно определить её ориентацию, то есть контекст, точку зрения и цель. Эти понятия

направляют и ограничивают создание модели. Они могут уточняться в ходе выполнения проекта, но быть последовательными, чтобы ее ориентация оставалась четкой и неискаженной.

Контекст определяет объект модели как часть более крупного целого. Он создает границу с окружающей средой, описывая внешние интерфейсы.

Контекстная диаграмма устанавливает контекст для модели.

Точка зрения определяет, что можно «увидеть „в контексте модели, с какой точки зрения или “под каким углом». В зависимости от поставленной цели могут быть приняты точки зрения, учитывающие различные аспекты объекта. На модель существует только одна точка зрения.

Цель определяет назначение модели или задачи, которые подлежат выполнению и воплощает в себе причину создания модели (функциональная спецификация, дизайн реализации, операции с клиентами и т. д.).

Отправной точкой для любого анализа является привязка контекста.

Решите, что является самым важным, прежде чем будет создан высший блок. Остерегайтесь отклонения от этой тщательно выбранной стартовой области. Каждый шаг должен сверяться с исходной целью.

Решения, которые не подходят, можно отметить для последующего моделирования соответствующих представлений. Ясность проистекает из строгости детализации. Понимание того, как далеко идти, когда остановиться, когда переключать передачи и какие элементы подходят друг к другу, всегда будет зависеть от цели, для которой создается модель.

#### В.2.1.2 Создание контекстной диаграммы

Прежде всего необходимо создать диаграмму A-0. Нарисуйте единое блок, содержащее название функции, которая определяет всю описываемую систему. Используйте стрелки ввода, контроля и вывода, входящие и выходящие из блока, чтобы представить взаимодействие данных и объектов системы с окружением. Эта одноблоковая диаграмма ограничивает контекст всей модели и служит основой для дальнейших усилий по декомпозиции (разбиению). Сформулируйте Цель и Точку зрения по контекстной диаграмме A-0.

Некоторые авторы считают, что проще нарисовать эскиз A0, а затем один блок и стрелки интерфейса, показанные на уровне A-0. Возможно, потребуется несколько раз переключаться при построении диаграмм между A-0 и A0, чтобы получить хорошие условия для декомпозиции. Если диаграмма A-0 начинается со слишком низкого уровня детализации, сделайте блок A-0 основой для новой диаграммы уровня A0. Поднимитесь на один уровень вверх до нового A-0 и вернитесь к утверждениям точки зрения и цели. Повторяйте этот процесс до тех пор, пока не будет достигнуто значение A-0, которое имеет достаточный объем для охвата всех аспектов системы. (Иногда такой более высокий уровень скорее расширит, чем прояснит выбранную точку зрения. Если это так, сделайте многоблоковую контекстную диаграмму A-1 и

сохраните диаграмму A0 в соответствии с первоначальным назначением).

#### В.2.1.3 Создание высшей диаграммы

Все системные функции находятся в пределах одного блока, представленного на диаграмме A-0. Диаграмма ограничивает контекст системы. Диаграмма A0 декомпозирует единственную функцию на диаграмме A-0 на три-шесть основных подфункций.

Реальной «вершиной» модели является диаграмма A0. Ее структура четко представляет, что намерена раскрыть диаграмма A-0. Термины и структура A0 также связывают каждый последующий уровень, потому что диаграмма представляет полное описание выбранного объекта. Нижние уровни разграничивают функции A0 (блоки). Цепочка детализации должна тщательно соблюдаться на каждом этапе, что обеспечит достижение цели модели. Начинать с самой вершины это вызов авторскому мастерству и заставляет разработчика поддерживать уровень абстракции, сохранять глубину модели и переводить детализацию на более низкий уровень.

#### В.2.1.4 Создание дочерних диаграмм

Чтобы сформировать структуру диаграмм, детализируйте каждый блок на диаграмме A0 разбив его на три-шесть основных частей.

Сформируйте новую диаграмму для каждого блока, которая охватывает ту же тему, что и родительский блок, но с большей детализацией.

Чтобы детализировать каждый блок от 3 до 6 дочерних блоков, необходимо получить дополнительную информацию. Создайте первый черновик диаграммы, сначала перечислив все данные и объекты, относящиеся к декомпозированной функции. Обратите внимание на то, чтобы список охватывал всю тему родительского блока, чтобы ни одна часть не была потеряна при декомпозиции. Затем начертите блоки, которые связывают имена подфункций-кандидатов с соответствующими данными и объектами из списка, и нарисуйте стрелки между блоками. Чтобы получить максимально четкую диаграмму, модифицируйте или перерисуйте диаграмму несколько раз до тех пор, пока она не примет оптимальный вид. Разделить (разбить блок на две или более частей) и сгруппировать (объединить две или более части в один блок) до требуемого уровня. Разделяйте и группируйте до тех пор, пока вы не выразите родительскую функцию в трех-шести блоках.

Создайте части более подробных диаграмм уровня, чтобы исследовать места, которые требуют уточнения. Создайте несколько (3 или 4) диаграмм в виде набора, а не по одной диаграмме за раз.

#### В.2.1.5 Создание вспомогательного материала

Каждая диаграмма сопровождается страницей повествовательного текста, глоссарием и, возможно, диаграммой-иллюстрацией. Текст, связанный с диаграммой A-0, должен завершать ориентацию модели и записывается при создании диаграммы A-0. Текст дополняет контекст (выраженный в самой A-0), расширяя изложенную точку зрения и цель модели.

Текст для каждой другой диаграммы (включая A0) совершенно другой. В нем четко и кратко излагается фабула. Текст не дублирует то, что уже сказано на диаграмме, просто описывая каждую функцию блока словами, а скорее объясняет схему. На каждом уровне текст уточняет точку зрения таким образом, что способствует достижению цели.

Диаграмма может иметь или не иметь связанный текст.

Глоссарий объясняет определения, которые автор дает функциям и данным/объектам на диаграмме. Эти определения важны, потому что терминология, используемая в модели, может иметь совершенно иное значение в одной компании, чем значение в другой компании.

Терминология часто различается между подразделениями или дисциплинами в рамках одной и той же компании.

FEО это диаграммы, которые выделяют особенно интересный или тонкий аспект диаграммы. Они не связаны синтаксисом IDEF блоков и стрелок и могут содержать частичные структуры стрелок и примечания, чтобы подчеркнуть их смысл.

#### В.2.1.6 Выбор блока для детализации

Получив полную родительскую диаграмму, «доработайте» более высокие уровни, прежде чем переходить к деталям. То есть, учитывая A0, акцент делается на работу над A1, A2, A3. Декомпозиция A1 на A11, A111 должна выполняться позже. Это позволяет избежать возможных переделок в случае внесения изменений в диаграммы более высокого уровня.

Сохранение равномерной глубины не является строгим правилом.

Размер глубины в любой момент времени зависит от того, позволяет ли большая глубина охвата улавливать смысл лучше, чем одна диаграмма.

Не откладывайте выполнение диаграммы более низкого уровня, например, A111, а делайте эскизы, пока идеи свежи. Важно относиться ко всем таким попыткам как к наброскам, пока не будет подтвержден «горизонтальный» ровный уровень. Будьте готовы переработать материал нижнего уровня, если он конфликтует с более высоким уровнем, например, A1, A2, A3.

Два совета, которые помогут в выборе блока, который нужно детализовать:

1. Начните с "трудной части" части, которая наименее знакома или менее понятна.
2. Выберите блок, детализация которого даст больше информации о других блоках.

Более простые темы легче декомпонировать позже, с меньшим риском ошибок или недосмотра, и они легко поддаются модификации, чтобы соответствовать декомпозиции более сложных проблем.

#### В.2.1.7 Деятельность разработчика

##### В.2.1.7.1 Этап сбора данных

Чтение предыстории: автор собирает информацию об объекте, изучая исходную информацию.

Интервью: Автор лично беседует с экспертом по данному вопросу.

Обдумывание: переварите информацию, полученную при ознакомлении и интервью, прежде чем приступить к фактическому построению диаграмм.

Выбор блока: решите, какой блок подходит для детализации на основе полученной информации.

#### В.2.1.7.2 Этап структурирования

Черчение: включает в себя фактический творческий процесс создания диаграммы. Он не ограничивается только рисованием блоков и стрелок, а также включает в себя перечисление случайных элементов данных, создание эскизов и т. д., которые предшествуют рисованию блоков и стрелок.

Повторное черчение: охватывает стадию построения диаграмм и соответствует редактированию и переработке вербального текста. Здесь деятельность связана не с созданием, а с графическим редактированием и перестановкой для ясности.

Устранение недостатков: это относится к модификации основных чертежей и учету поправок. Это прежде всего механическая операция, сводящая воедино повторные чертежи, часто в ответ на замечания читателя.

#### В.2.1.7.3 Этап презентации

Напишите и отредактируйте текст: текст, сопровождающий любую диаграмму, должен быть точным. Редактирование часто устраняет ненужные детали и избыточность.

Обобщение: соберите любой материал, диаграммы, дерево узлов, глоссарии, текст и т. д., относящийся к данной теме. Добавьте сюда заполненный титульный лист.

#### В.2.1.7.4 Этап взаимодействия

Реагирование: это относится к автору, реагирующему на комментарии. Это сочетание чтения и комментирования реакции на комментарии и ответ читателям.

Собеседование: это время, потраченное на то, чтобы автор и читатель действительно собрались вместе и поговорили о реакции автора на комментарии.

Групповые встречи: это время, потраченное на групповые встречи, анализ прогресса или мозговой штурм следующих шагов. В протоколах заседаний группы определяется обсуждаемая тема.

#### В.2.2 Черчение диаграмм IDEF0

Построение диаграмм это наиболее субъективная и творческая деятельность в процессе моделирования. Данный процесс открыт для широких вариаций между отдельными авторами. Ни один набор шагов не будет одинаково хорош для всех авторов. Шаги, представленные здесь, являются проверенной последовательностью и предназначены для оказания помощи новому автору в рисовании диаграмм IDEF0.

а. Создайте релевантный, но еще не структурированный список данных. Перечислите элементы в контексте родительского блока, которые первыми приходят на ум. Сгруппируйте элементы, если это возможно, чтобы показать сходство.

b. Назовите функции, которые воздействуют на перечисленные данные и начертите блоки вокруг имен.

c. Нарисуйте соответствующие стрелки. Когда блоки нарисованы, оставьте стрелки «заглушки», чтобы сделать блок более информативным. Сделайте полные соединения, когда станет ясно, что раскрывает диаграмма.

d. набросайте макет, который представляет собой наиболее четкое расположение блоков и стрелок. Связывайте стрелки вместе, если структура слишком детализирована. Оставьте только основные элементы и измените диаграмму по мере необходимости.

e. Создайте текст, глоссарий и диаграммы-иллюстрации FEO, если это необходимо, чтобы выделить важные аспекты. При необходимости предложите внести изменения в родительскую диаграмму.

#### B.2.2.1 Генерации функциональных блоков

Функциональные блоки создаются с использованием основных подфункций родительского элемента. Когда имена подфункций написаны, нарисуйте прямоугольники (блоки) вокруг них, чтобы сформировать начало реальной диаграммы. На этом этапе количество блоков не имеет значения. Они могут быть изменены путем кластеризации (объединения) и разделения в соответствии с правилом от трех до шести блоков.

Кластеризация сгруппирует два или более блоков в один блок. Ее цель – объединить связанные функции в единую, более общую функцию. Это устраняет преждевременные детали, которые затемяют сообщение, передаваемое на этом уровне.

Разделение разбивает один блок на две или более частей. Данный процесс является обратным по отношению к кластеризации. Его цель состоит в том, чтобы предоставить более полную информацию и помочь в понимании разбиваемого блока.

Просмотрите получившийся набор функциональных блоков. Ищите оптимальный баланс между выбранными факторами. Посмотрите, можно ли сделать имена более конкретными. Используйте специальные термины и сокращения только в тех случаях, когда это необходимо для продвижения коммуникации с целевой аудиторией, и только на уровне детализированных диаграмм. Не используйте их на самых высоких уровнях (A-0 и A0). Тщательно определите специальные термины в глоссарии.

Во всех случаях используйте в именах функциональных блоков глаголы и глагольные обороты. Всякий раз, когда фразу можно интерпретировать как глагол или существительное, используйте обозначение “(v)” для обозначения предполагаемого использования глагола.

Блоки.

1. В большинстве случаев располагайте блоки по диагонали от верхнего левого угла до нижнего правого. В то время как любая компоновка, в которой четко обозначены намерения автора, является приемлемой, вертикальные или горизонтальные форматы

имеют тенденцию к скоплению стрелок и препятствуют хорошему структурированному стилю анализа.

2. Блоки, расположенные в левом верхнем углу, «доминируют» над блоками, расположенными ниже и справа через управляющие стрелки, которые их связывают. Этот стандартный стиль помогает читателям понять ваш смысл.
3. Пронумеруйте каждый блок в нижнем правом внутреннем углу. Присвойте номера блокам на диаграмме слева направо и сверху вниз. Это поможет определить номер узла для каждого блока. Начальные цифры полного номера узла блока совпадают с номером узла диаграммы. Последняя цифра номера узла это номер блока. Если блок на Рисунке B10 отображается на диаграмме A4, то полный номер узла этого блока будет A42.

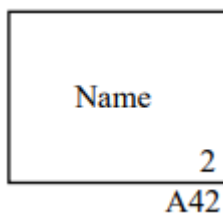


Рисунок B10. Номер блока и номер узла

4. На рабочих или черновых копиях напишите номер узла или C-номер под нижним правым углом любого блока, который детализирован. C-номер DRE указывает на конкретную версию диаграммы, для которой он предназначен.

5. Любая диаграмма не может содержать более шести блоков.

#### B.2.2.2 Создание стрелок интерфейса

Начертите интерфейсные стрелки, подсоединенные к каждому блоку. Соедините концы стрелок, чтобы показать, какие выводы питающие, какие являются вводными и управления.

Следует помнить, что вводные данные/объекты преобразуются функцией для получения выводных данных. Если стрелка содержит как вводные, так и управляющие данные то данная стрелка отображается как стрелка управления. Если вы не уверены, является ли стрелка управлением или стрелкой ввода обозначьте ее как стрелка управления. Если неясно, нужна ли вообще конкретная стрелка данных или объекта, не указывайте ее (если это не единственный элемент управления). Выводные стрелки показывают результаты возможных активаций функции. Синтаксис выводных стрелок не определяет, какие шаблоны выводных стрелок могут применяться и при каких обстоятельствах. Если последовательность представляет особый интерес, нарисуйте диаграмму FEO, иллюстрирующую деталь. Не беспокойтесь о последовательности. Просто убедитесь, что все важные случаи учтены диаграммой.

Группируйте связанные стрелки там, где это возможно. Самая распространенная ошибка при создании стрелок сделать структуру

стрелок или метки стрелок слишком детализированными. Уровень детализации стрелок должен соответствовать уровню детализации блоков. На высоких уровнях и названия блоков, и метки стрелок будут общими.

В качестве окончательной проверки сравните все стрелки со списком данных, чтобы убедиться, что каждый корректный элемент данных отображается. Элементы, которые не отображаются, либо не подходят для этого уровня детализации, либо были пропущены при создании стрелок.

Думайте о контроле и ограничении, а не о потоке.

Основное правило при компоновке структуры стрелки "ограничивайте, не упорядочивайте." Т.е. сделать так, чтобы структура диаграммы отражала взаимосвязи, которые будут верны независимо от того, какая последовательность соблюдается.

Несмотря на то, что для достижения желаемого конечного результата что-то должно улучшаться от этапа к этапу, постарайтесь установить ограничения, которые должны соблюдаться или инвариантные свойства, которые должны быть истинными, а не используйте какую-то одну конкретную последовательность шагов, которая приведет к результату.

Причина в том, что все блоки могут функционировать одновременно.

Таким образом, последовательность не имеет никакого значения.

Всегда эффективнее ограничивать, чем упорядочивать. Везде, где это возможно, должны быть созданы диаграммы, которые «говорят правду» независимо от того, какие шаги сделаны первыми. Ясно, что это лучше, чем ограничиваться только одной из возможных последовательностей.

Зачастую проще продумать действия подфункции в определенной последовательности, чтобы получить вариант и закрепить это на бумаге.

Это может быть хорошим способом начать работу, но всегда перерабатывайте первую попытку в структуру ограничений.

Тщательно маркируйте.

Второочередность ненужных деталей выделяет значимые элементы. Не загромождайте свои диаграммы излишней информацией и слишком большим количеством стрелок. Не закливайтесь на одном уровне. Не нужно представлять все сразу, чтобы избежать незаконченности и недопонимания. Смысл работы в том, чтобы в конечном итоге донести все что необходимо.

Кроме того, если в диаграмме слишком много элементов, она теряет гибкость. Это приводит к потере эффективности. Сила исходит от структуры. Этого можно добиться, оставив детали подфункциям.

Выполните итерацию между диаграммами высокого уровня и подфункциями, которые раскрывают детали.

Сомнительные стрелки.

Часто бывает трудно определить, показывать стрелку или нет. Самый простое решение: «Когда сомневаешься, пропусти стрелку.» Если стрелка несущественна для основной магистрали, если есть вопросы по ней, то, по всей видимости, неправильно ее включать.



Неправильное удаление сомнительной стрелы в данной ситуации не повлечет за собой необратимых повреждений. Необходимость в стрелке прояснится при рассмотрении подфункций, и дисциплина ICOM заставит стрелку вернуться на этот уровень. Тогда не будет никаких вопросов.

#### В.2.2.3 Уровень усилий

Исходной целью при построении диаграммы должна быть четкая диаграмма, представляющая определенное сообщение и не нарушающая никаких синтаксических правил. После построения диаграммы, критические замечания, высказанные экспертами, могут использоваться для улучшения первой версии. Большинство диаграмм можно изменить, чтобы сделать вторую версию, лучше первой. Первая редко бывает оптимальной.

По мере совершенствования мастерства первые диаграммы становятся лучше, и авторы чувствуют себя более уверенно, используя IDEF0.

Переработка диаграмм всегда является необходимой частью процесса.

Ключевая идея состоит в том, чтобы использовать цикл пересмотра для достижения прогресса на бумаге. При последовательном упорядоченном продвижении все важные аспекты должным образом обрабатываются.

IDEF0-это методология формирования мыслей, а не просто практика по построению диаграмм. Изложение мыслей на бумаге и кропотливый труд-это путь к успешному разрешению проблемы. Лучше задавайте хорошие вопросы, а не на представляйте “идеальные” ответы.

#### В.2.3 Повторное построение диаграмм IDEF0

##### В. 2.3.1 Модифицирующие блоки

При первом создании диаграммы представляются 3-6 функциональных блоков примерно одинакового уровня детализации. Кластеризация и разделение предоставляют границу, которая более понятна и обеспечивает более простое взаимодействие между функциональными блоками.

Чаще всего кластеризация и разделение работают вместе. Блоки разделяются и полученные фрагменты группируются в новые блоки, которые более точно передают предполагаемое сообщение.

Охватывается один и тот же объект, но части группируются более понятным образом.

Разделение и перестройка.

Важно, чтобы все блоки диаграммы имели согласованную форму.

Изменения, внесенные в другом месте, не должны оказывать существенного влияния на подфункцию. Разделите и перестройте, чтобы восстановить баланс.

Иногда направление перемещения продукта в блоке не совпадает с другими блоками текущей диаграммы. Часто проблема заключается в том, что другие аспекты претерпели изменения и уточнения. То, что раньше было хорошей идеей, теперь не актуально или неправильно. Разделите проблемный блок на две или более частей, одна из которых все еще содержит суть первоначальной идеи, но более уместно и кратко. Ожидайте изменения описания блока (или блоков). С разделением,

новые идеи становятся более понятными и теснее увязываются с соответствующими блоками.

Кластеризация и замена.

Твердая абстракция одновременно и яснее, и мощнее, чем преждевременная детализация. Сгруппируйте связанные блоки и замените их одним охватывающим блоком.

Зачастую достаточный уровень абстракции можно улучшить, сгруппировав несколько блоков в более общий вид, отложив детализацию на следующий более низкий уровень. Нарисуйте линию вокруг группируемых блоков и замените их одним блоком с подходящим названием. Дополнительный уровень не является препятствием. Это лучшее представление, потому что структура представлена более четко. Это признак часто появляется при разделении и является одним из самых мощных способов объяснения функций.

В.2.3.2 Связывание (объединение) стрелок

Стрелки и блоки должны иметь соответствующий уровень абстракции на диаграмме.

Этого можно достичь двумя способами:

1. Объедините стрелки, связанные с одним и тем же источником и пунктом назначения под одной более общей меткой и сделайте одну стрелку.
2. Переименуйте некоторые блоки (используя Разделение и Кластеризацию), чтобы лучше распределить подфункции и повторно обозначьте вновь полученные стрелки.

Редко избыток стрелок указывает на ошибку. Может быть, потому что они точны и наносятся аккуратно. Но всегда верно то, что избыток стрелок мешает рассмотреть сообщение. Способность потребителя правильно оценивать то, что представлено, должно определять количество используемых стрелок.

В.2.3.3 Предложение о внесении изменений в контекст

Более детальное представление при построении новой диаграммы, может выявить ошибки или упущения в родительской диаграмме.

Родительская модификация-это естественное и ожидаемое событие.

При создании структуры стрелки правило гласит “ » если есть какие либо сомнения в том, нужна ли стрелка функциональному блоку, оставьте их-более поздняя детализация покажет, действительно ли стрелка нужна.” Это тот момент, когда такие вопросы решаются по мере поступления, а не с помощью ранних рассуждений.

Изменения родительской диаграммы могут представлять разную степень сложности. Если изменение адаптируется с помощью ревизии только самого родителя, это проще, чем ревизия, которая затрагивает более удаленные диаграммы. Предлагая изменение, тщательно продумайте его и оцените сложность. Замена простых изменений на те, которые вызывают сложности может нанести ущерб качеству декомпозиции. Когда коррекция завершена, проверьте все граничные соединения, чтобы убедиться, что коды ICOM отображены правильно. При внесении

изменений информируйте других авторов, работающих над данной диаграммой.

Всегда возвращайтесь к родительской диаграмме при детализации блока. Это поможет вашему творчеству. Если детализированная диаграмма не соответствует контексту, то текущая работа или контекст выполнены неправильно. Откорректируйте контекст или текущую работу. Все должно соответствовать.

#### В.2.3.4 Синтаксис ICOM для подключения диаграмм

Важным аспектом для понимания диаграмм является способность находить и правильно оценивать необходимые сведения. Узловые номера отражают структуру декомпозиции функции (детализация блока). Набор стрелок обеспечивает интерфейсные соединения.

Коды ICOM указываются на всех стрелках, имеющих на диаграмме один не соединенный конец. Граничные стрелки соединяют сеть стрелок между диаграммами. Каждой граничной стрелке присвоен код ICOM с информацией о соединении стрелки с родительским блоком.

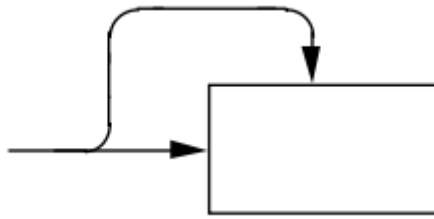
#### В.2.4 Графический макет

Разместите блоки по диагонали в соответствии со структурой ограничений, от верхнего левого угла до нижнего правого. Управляющие обратные связи идут вверх и влево, а стрелки выводных обратных связей (которые моделируют память) идут вниз и влево. На этом этапе пронумеруйте блоки слева направо.

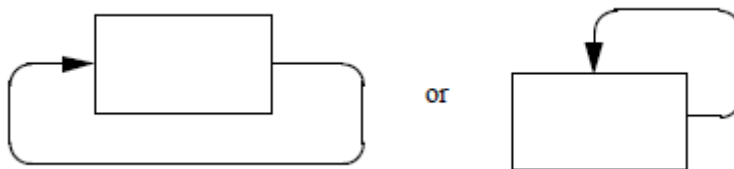
Рекомендуется начать построение макета с нанесения наиболее часто применяемых стрелок ограничений, добавляя позже менее используемые пути. Это подмножество стрелок позволит определить положение блока. Нанесите все граничные стрелки, показанные на родительской диаграмме, а затем добавьте оставшиеся стрелки.

##### В.2.4.1 Ограничения на диаграмме

1. Функциональные блоки всегда должны иметь управляющие стрелки, при этом вводящие стрелки могут отсутствовать.
2. Если вводящая стрелка выполняет функции управления и ввода, отобразить ее как стрелку управления. При наличии сомнений по данному поводу всегда выбирать стрелку управления. Стрелка, отображаемая на родительском блоке в виде стрелки управления, может появиться на следующем уровне в качестве стрелки управления или стрелки ввода или того и другого, в зависимости от ее отношения к подфункциям на этом уровне.
3. В общем случае не разделяйте стрелку, на вводящую и управления при подключении к одному и тому же блоку. Этот момент лучше всего заметен на диаграмме нижнего уровня, где показано назначение каждой ветви и причина разделения. Если все таки необходимо разделить то выполните это на родительском уровне, промаркировав ветви, что сделает ваше решение более весомым.



4. Итеративная деятельность (хранение в памяти или обратная связь) может быть показана как:



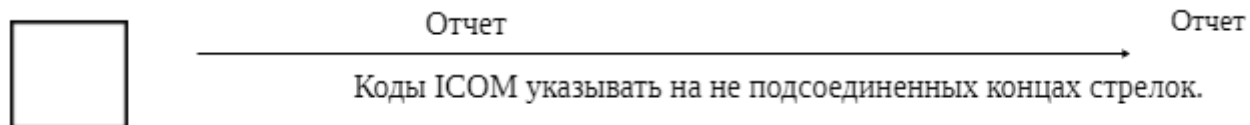
5. Старайтесь избегать избыточности, например:



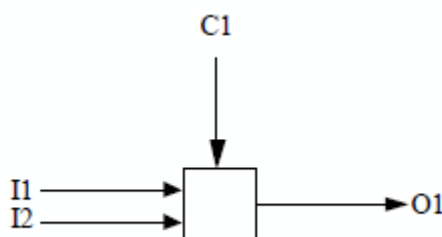
В этих случаях тривиальные названия блоков просто повторяют сообщение, передаваемое размещением стрелок. Небольшое дополнительное размышление обычно приводит к более информативным названиям блоков.

#### В.2.4.2 Размещение стрелок

1. Стрелки наносятся только в виде горизонтальных и вертикальных отрезков, а не по диагонали или в виде кривых (за исключением углов).
2. Расположите углы стрелок, пересечения и метки на разумном расстоянии от блоков.
3. Не используйте ключевые слова, такие как «данные», «функция», «ввод», «вывод», «контроль» или механизм в названиях или метках, если это не является абсолютно необходимым.
4. Если стрела длинная, пометьте ее дважды.



5. Коды ICOM указывать на не подсоединенных концах стрелок.



6. Соедините открытые граничные стрелки, чтобы показать места, находящиеся под воздействием. В противном случае пользователь может пропустить соединения.



7. Разместите параллельные стрелки адекватно. За ними трудно уследить визуально, если они длинные и расположены близко друг к другу.



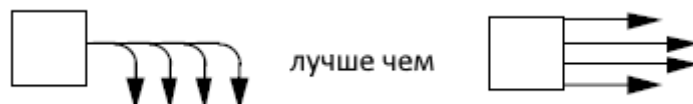
8. Нарисуйте дополнительные наконечники стрел вдоль стрел, где это необходимо для ясности.

#### В.2.4.3 Макет стрелки

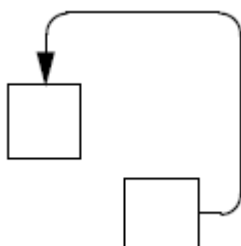
1. Объедините стрелки, имеющие одним и тот же источник и пункт назначения, если только стрелка не имеет такой важности, что включение ее в пучок уменьшит ясность понимания.

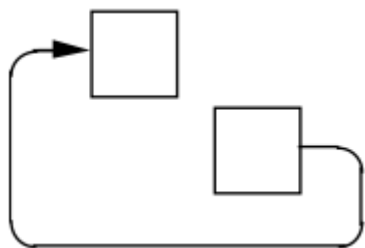


2. Используйте как можно меньше стрелок на любой стороне блока. Если их слишком много, скомпонуйте, присвойте соответствующую метку и разведите ветви по назначению.

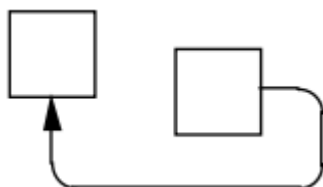


3. Обратные связи по управлению должны быть показаны как «вверх и над»





**Обратные связи по вводу должны быть показаны как «вниз и под»**



**Обратные связи по вводу должны быть показаны как «вниз и под»**

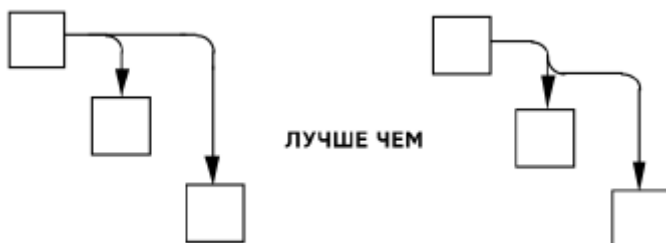
4. Если стрелка разветвляется и подключается к нескольким блокам, нарисуйте ее в одном и том же относительном положении ICOM на каждом блоке, если это возможно.



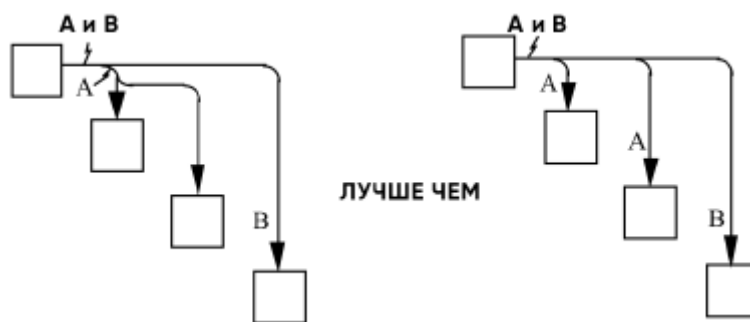
5. Расставьте стрелки так, чтобы свести к минимуму пересечения.



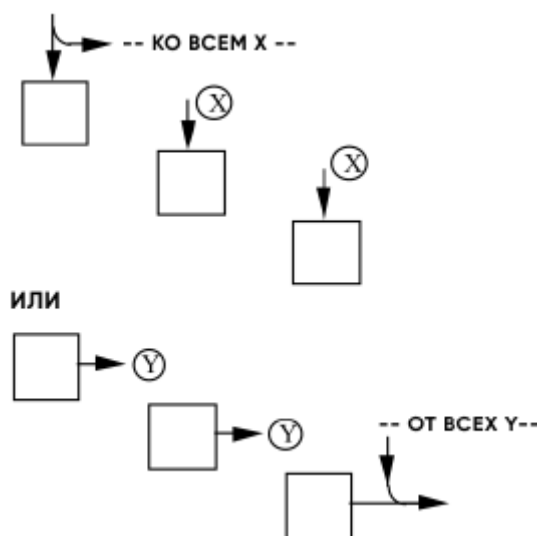
6. Сведите к минимуму кривые и углы, сохраняя при этом помеченные сегменты четкими



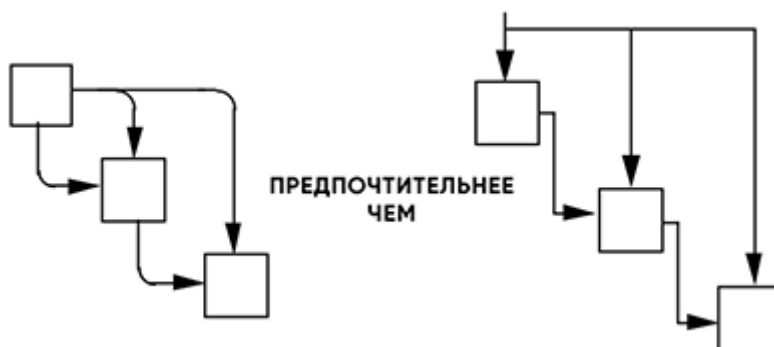
7. Используйте выразительный потенциал ветвящихся стрелок, когда и если это уместно.



8. Чтобы избежать нагромождения при показе стрелки, которая применяется идентично или получена идентично из каждого блока на диаграмме, используйте форму «ко всем»:



9. Все стрелки должны иметь изогнутые углы на стыках, разветвлениях и изгибах.



## В.2.5 Написание текста

### В.2.5.1 Текст и глоссарий

Текст, который сопровождает каждую диаграмму, представляет собой краткий интегрирующий обзор диаграммы с указанием важных связей, шаблонов или взаимодействий между блоками и стрелками.

Желательно, чтобы текст был меньше страницы в длину. Он выделяет особенности, которые, по мнению разработчика, представляют особый интерес или значение, знакомя пользователя с основными идеями диаграммы. Он не дублирует каждую деталь, показанную на самой диаграмме.

Пишите текст только после того, как диаграмма получила достаточно высокий уровень рассмотрения и одобрения. До написания текста диаграмма способствует правильной передаче намеченного сообщения. Текст, основанный на тщательно проработанных диаграммах, будет так же структурирован и организован, как и сама диаграмма.

Используйте определения глоссария, чтобы обобщить специальные значения, которые могут потребоваться для ключевых терминов, слов, фраз и аббревиатур, применяемых в диаграмме. Слово или фраза могут иметь разные коннотации в зависимости от пользователя диаграммы. Попробуйте создать хороший текст, без добавления диаграмм-иллюстраций FEO. Диаграммы FEO следует использовать для иллюстрации не совсем очевидных аспектов, которые проясняют намерение диаграммы, но которые загромождали бы диаграмму, если бы были включены. Если FEO необходима, то текст, который сопровождает ее, должен ссылаться на соответствующую диаграмму.

#### В.2.5.2 Примечания и ссылки

Есть два вида примечаний: примечания к модели и примечания читателя. Примечания к модели обсуждаются в нормативном разделе. В отличие от примечания к модели, которые являются частью диаграммы, на которой они появляются, примечания читателя явно находятся на диаграмме, но не являются ее частью. Таким образом, они не могут изменить значение синтаксиса или семантики диаграммы. Как и для примечания к модели, если текст примечания читателя относится к нескольким местам или нескольким признакам диаграммы (включая другие примечания к модели или читательские примечания), то номер примечания в кружочке (без текста) может быть скопирован и прикреплен с помощью тильды IDEF0 к каждой точке применения, но только на одной диаграмме, на которой появляется текст приложения читателя.

По определению, примечания читателя носят строго информативный, а не нормативный характер. Они могут быть о чем угодно, в отношении диаграммы или модели. На практике примечания читателя могут быть посвящены моделируемому объекту или его представлению, включая подбор слов, компоновку, точность и т. д.

Примечания читателя обозначаются цифрой " n " внутри небольшого круга: n. Для любой конкретной диаграммы (и, следовательно, номера ноды) числа "n" образуют числовую последовательность, начинающуюся с "1". Все читатели (включая разработчика, когда он комментирует что-то) должны использовать одну и ту же последовательность примечаний для диаграммы. Номера примечаний читателя для ноды никогда не должны использоваться повторно или переназначаться. Таким образом, последовательность, созданная в контексте номера ноды является постоянной записью обсуждения читателя/разработчика относительно этой ноды.

При написании текста и примечаний для диаграмм и отдельных частей диаграмм используется стандартная запись. Содержание ссылки



базируется на номерах блока, кодах ICOM, номерах нод и номерах примечаний.

Например:

O2	означает	Граничная стрелка с кодом ICOM O2
2I1	означает	Блок 2 Ввод 1
2O2 к 3C1	означает	Стрелка от 2O2 до 3C1
n	означает	Примечание к модели n
n	означает	Примечание к модели n (альтернативная нотация)
(n)	означает	Примечание читателя n (альтернативная нотация)
n	означает	Примечание читателя n
QA/A21.3C2	означает	В модели "QA" на диаграмме A21, см. Блок 3 Управление 2
A42. 3	означает	На диаграмме A42 в этой модели см. примечание читателя 3.
A42. 3	означает	На диаграмме A42 в этой модели см. примечание к модели 3.
A42.3	означает	На диаграмме A42 в этой модели см. Блок3.

Данные элементы могут использоваться отдельно, если они относятся к текущей диаграмме (например, в примечаниях к модели или тексте). В противном случае им должен предшествовать номер ноды, а при необходимости и название модели. Период ".«используется для обозначения „смотреть“ определенный предмет на указанной диаграмме.

Например:

MFG/A21. 3C2 означает: В модели » MFG", на диаграмме A21, см. Блок 3  
Управление 2.

Каждая ссылка нуждается только в таком количестве полей, которое необходимо для полной однозначности.

Наиболее полная форма следующая:

АССТ/(A21=BT56).1O2 к 4C3

что означает в модели «АССТ», диаграмма A21, С-номер BT56," см. " стрелку от Блока 1 Вывод 2 к Блоку 4 Управление 3.

Добавляйте аккуратные и полные ссылки в формулировку текста, глоссария и страниц ФЕО. Например: «Влияние (2o3-1c2 и 3c2) этой стоимости на конечную цену (o2) вызывает озабоченность.» Читается следующим образом: третий вывод Блока 2 через блоки 1 и 3 к граничной стрелке O2. После написания текста, пройдите по нему и добавьте ссылки с номерами блоков, чтобы точно привязать текст к диаграмме.

В большинстве случаев достаточно простого номера блока("Блок3") или ссылки на пару стрелок ("Блок 3, O1 и C2"). Когда читателю

действительно нужно «увидеть» другую диаграмму, используйте". «из справочного языка.

Подобно тому, как кодирование ICOM естественным образом расширяет схему ссылок, основанную на нумерации нод, примечания к модели и примечания читателя позволяют ссылаться на них.

Например:

A21.3C2 |2| (5) ответ

переносит предыдущий пример ("на диаграмме A21, см. Блок 3 Управление 2») на случай, когда читатель ссылается на ответ разработчика диаграммы на примечание читателя № 5 (возможно, добавленное вторым читателем диаграммы), который прокомментировал модель, авторское примечание к модели № 2, касающееся рассматриваемого Управления! В этом примере "ответ " демонстрирует использование кратких выражений естественного языка в справочном языке.

Этот пример также иллюстрирует использование сокращений n (примечания к модели) и (n) для n (примечание читателя) в качестве текстовых ссылок. Эти сокращения облегчают подготовку ссылок на примечания с помощью стандартных текстовых процессоров. Ссылки на примечания в тексте IDEF0 и письменная переписка являются примерами текстовых ссылок.

### V.3 Сбор данных для моделирования IDEF

#### V.3.1 Введение

При анализе или проектировании любой системы может возникнуть необходимость в получении или проверке фактов о данной системе или объекте исследования. Существует множество источников фактической информации.

Можно выбрать следующий путь:

- Прочитать существующие документы, используя оглавление и перечень, чтобы найти необходимую информацию.
- Понаблюдать за системой в действии, если она уже существует.
- Опросить большую группу людей с помощью анкет или других подобных средств.
- Поговорить с одним или несколькими экспертами, которые обладают нужными знаниями.
- Использовать то, что уже известно разработчику.
- Предложите или придумайте гипотетическое описание и попросить читателей помочь приблизить его к реальности.

Из всех этих методов наиболее важным является личное взаимодействие с экспертом. Редко вся существующая информация представлена в документах. Предвзятые представления, отраженные в анкетах, часто ошибочны.

Ключевой частью интервьюирования является запись полученной информации. Это может быть сделано либо в виде неофициальных

заметок, либо в виде списков действий и данных/объектов, либо в виде формализованной матрицы функций, либо в виде эскизов диаграмм.

### В.3.2 Процесс опроса

Цель опроса состоит в сборе информации от лица, обладающего экспертными знаниями, которые считаются важными для аналитической работы. Существует четыре типа опроса, которые могут быть проведены в ходе выполнения этапа анализа проекта IDEF.

Поиск фактов для понимания текущих операций. Этот тип опроса используется для установления содержания текущей операционной модели или для того, чтобы понять существующую окружающую среду.

Выявление проблем для оказания помощи в установлении будущих требований. Этот тип опроса используется для проверки текущей операционной модели и обеспечения основы для будущей операционной модели.

Обсуждение решений относительно будущих возможностей системы. Этот тип опроса используется для определения содержания будущей операционной модели.

Авторский/редакторский семинар IDEF. Этот тип опроса используется для решения проблем, возникших при построении модели IDEF.

Причина для определения типов опроса заключается в том, что в ходе выполнения фактического опроса появляются компоненты каждого типа.

Респондент может рассказать интервьюеру факты о данной системе с точки зрения проблем. Кроме того, респондент может определить проблемы с точки зрения их решения. Постоянно классифицируя высказывания респондентов, интервьюер может лучше оценить точку зрения эксперта.

### В.3.3 Комплект опроса

Для документирования опроса можно использовать “стандартный” набор для опроса. Он может быть сохранен в файле опроса и распространен среди соответствующих сотрудников проекта. Список лиц, до которых доводятся результаты опросов может включать в себя других членов аналитической группы или даже респондента опроса для исправления, добавления и удаления. Комплект для опроса может содержать следующее:

1. Титульный лист (обложка комплекта)
2. Опрос и запись последующих действий
  - (a) Имя проводящего опрос (IDEF имя автора)
  - (b) Дата опроса (дата диаграммы IDEF)
  - © Продолжительность опроса (Время начала, Время окончания)
  - (d) Имя респондента
  - (e) Должность респондента и организационная ответственность
  - (f) Номер телефона респондента и другие контактные данные
  - (g) Идентифицированные дополнительные источники информации
    1. Документы-название и местонахождение
    2. Другие опрошенные—Имя, Должность, Организационная Ответственность, Адрес, Номер телефона

- (h) Основные элементы информации краткое изложение ключевых моментов, затронутых в опросе.
  - (i) Последующие вопросы и / или проблемные области, которые не были затронуты во время опроса или отложены
  - (j) Новые термины для глоссария проекта
3. Список действий и данных / объектов
  4. Программа опроса (разрабатывается при подготовке к опросу).
  5. Примечания к опросу и черновые схемы

## ПРИЛОЖЕНИЕ С-ПРОЦЕДУРЫ И ФОРМЫ ЦИКЛА ОБЗОРА

### С.1 IDEF Дисциплина командной работы

Создание модели -это динамичный процесс, который обычно требует командной работы. На протяжении всего проекта черновые фрагменты модели создаются авторами и распределяются между другими участниками проекта, экспертами по объекту, менеджментом и т. д. для обзора и комментариев. Черновые фрагменты модели называются Комплектами и могут содержать диаграммы, текст, глоссарий или любую другую информацию, которая, по мнению автора, имеет отношение к разработке модели.

Для правильного чтения и понимания моделей IDEF0 требуется непродолжительное обучение и скромный опыт. Глубокое понимание темы необходимо для обеспечения гарантии качества IDEF-моделирования, предоставляемое командой. Каждый, кто достаточно глубоко продвинулся в правильном чтении, называется «читателем».

Дисциплина командной работы IDEF определяет всех лиц, заинтересованных в обзоре модели, в качестве рецензентов.

Рецензенты, которым поручено изучить и сделать в письменном виде свои замечания по поводу Комплекта, называются комментаторами.

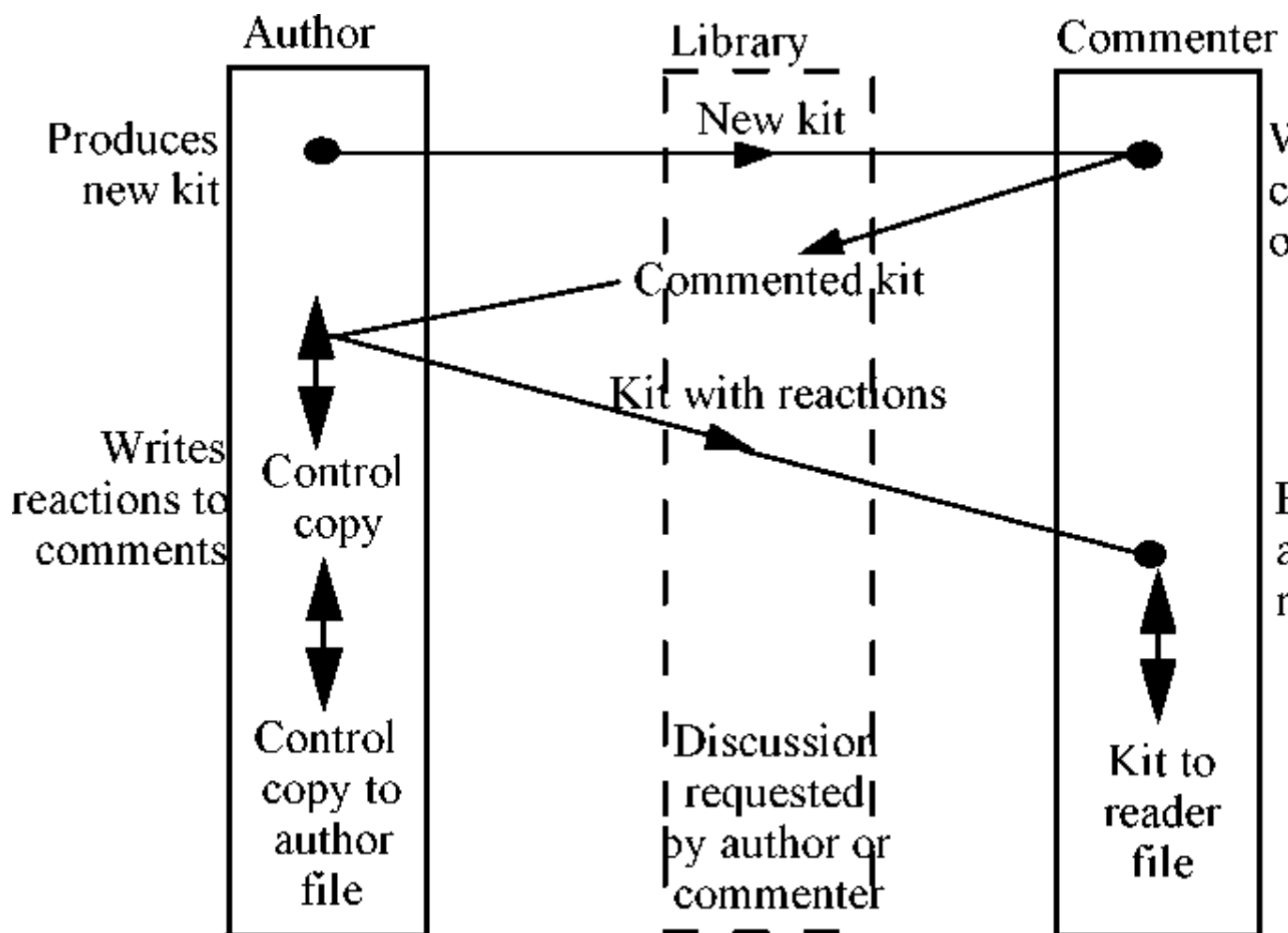
Рецензенты, которые получают Комплект только для информации, а не для письменных комментариев и называются читателями. Автор и комментаторы разделяют ответственность за качество модели. Приняв согласованный результат, другие рецензенты разделяют ответственность за результат.

Порядок работы требует, чтобы каждый человек, от которого ожидают комментариев к диаграмме, делал их в письменной форме и представлял их автору диаграммы. Написав на читательском экземпляре, автор отвечает на каждую заметку в письменной форме (простая галочка, для согласования; в противном случае-заметка в ответ). Этот цикл продолжается, охватывая все Комплекты, относящиеся к конкретной модели, до тех пор, пока модель не будет завершена и рекомендована к публикации.

Через регулярные промежутки времени в ходе эволюции модели в библиотеку помещается мастер-копия последней версии, а копия распространяется в виде Комплекта, который рассылается читателям для оказания им помощи в поддержании актуальной информации о модели. По мере того, как комментарии к каждому Комплекту рецензируются автором, последний вносит коррективы в мастер-копию модели для включения в нее исправлений и изменений. Еще один Комплект, который включает в себя последние изменения, затем распространяется по списку читателям. Более подробная информация добавляется путем создания большего количества диаграмм, текста и глоссария. Больше комментариев, больше изменений. Конечным результатом этого процесса для организованной командной работы является высокая уверенность в том, что окончательные модели IDEF являются эффективными, хорошо выраженными и что консенсус был достигнут с группой читателей, которые были вовлечены в цикл обзора.

## C.2 Цикл Комплекта IDEF

При создании документа материалы, написанные или собранные автором, распространяются в виде стандартного Комплекта среди комментаторов, рецензентов и других читателей. Комментаторы просматривают материал и пишут о нем комментарии. Комментаторы возвращают Комплект автору, который реагирует на комментарии и может использовать комментарии для пересмотра или расширения материала. Комплект возвращается комментатору вместе с ответами автора. Читатели также могут возвращать комментарии автору, но это не обязательно. Данный процесс называется Циклом комплекта (Рисунок C1).



**Рисунок С1. Цикл Комплекта**

Этапы цикла Комплекта следующие:

- Разработчик собирает материал для рецензирования в Стандартный Комплект. Заполняется титульный лист. Экземпляры Комплекта раздаются каждому читателей и автору. Оригинал предоставляется для справки.
- В течение указанного времени отклика комментатор ознакамливается с Комплектом и пишет комментарии непосредственно на копии в виде примечаний читателя, по возможности красным цветом. Комплект возвращается автору.
- Разработчик отвечает в письменной форме непосредственно на экземпляре каждого комментатора, по возможности синим цветом. Автор может согласиться с комментарием, поставив на нем галочку и отметив его на своем рабочем экземпляре, включить в следующую версию модели. В случае несогласия автор пишет ответную записку, приложенную к примечанию читателя (без нового номера примечания). Независимо от того, есть ли разногласия, Комплект возвращается читателю, завершая один цикл Читатель/Автор. (См. С. 4. 1. 2 относительно нумерации Примечаний читателя.)
- Читатель изучает ответы автора и, если он удовлетворен, архивирует комплект. (Комментарии к Комплектам всегда

сохраняются у читателя.) Если назначенный комментатор не согласен с ответами автора, организуется встреча с автором для устранения разногласий. Если это невозможно сделать, то перечень вопросов передается в соответствующий орган для принятия решения. Автор не обязан разрешать разногласия с каждым читателем, но комментаторы лишаются права голоса, если их проблемы не разрешаются.

Этот цикл продолжается до тех пор, пока не будет создан документ, представляющий собой тщательное рассмотрение замечаний всех участников проекта. Кроме того, сохраняется полная история этого процесса.

Результаты Цикла комплекта представляет собой документ, в который внесли свой вклад авторы и комментаторы, а также, при необходимости, перечень вопросов, требующих реакции руководства.

На протяжении всего цикла библиотекарь проекта занимается копированием, распространением, подшивкой и передачей Комплектов между авторами, комментаторами, рецензентами и читателями.

#### С.2.1 Роли персонала

Роли и функции вовлеченных людей заключаются в следующем:

- Разработчики (Авторы)  
Разработчики (авторы) модели лица, создающие IDEF0 –модели.
- Комментаторы (эксперты или другие разработчики)  
Люди, хорошо осведомленные о моделируемом объекте, от которых авторы могли получить информацию посредством беседы, и имеющие достаточную подготовку в технике IDEF, чтобы предлагать структурированные комментарии в письменной форме. Люди, которым поручили сделать письменный комментарий Комплекта.
- Читатели (эксперты или все, кто хочет быть в списке читателей)  
Люди, хорошо осведомленные о моделируемом объекте, от которых разработчики, возможно, получили информацию посредством беседы или опроса, которые просматривают документацию для получения информации, но не делают письменных комментариев.
- Библиотекарь  
На данное должностное лицо возлагается обязанность вести архив документов, делать копии, распространять Комплекты и вести учет.

Лица, исполняющие данные функции, должны быть обученными и опытными читателями IDEF0, чтобы они могли уверенно выполнять возложенные на них задачи.

«Роль» не имеет ничего общего с должностью, и одного и того же человека могут попросить выполнять несколько ролей. Таким образом, участие каждого индивидуума, по сути, уникально и зависит от исполняемого Комплекта.

#### С.2.2 Руководство для разработчиков (авторов), читателей и комментаторов

В этом разделе рассматриваются общие рекомендации для читателей и разработчиков. Читатели могут иметь дополнительные задачи, то есть выступать в качестве комментаторов и рецензентов, но здесь данный вопрос не рассматривается.

#### С.2.2.1 Руководство для читателей

Единый набор вопросов и правил не может быть предложен для комментирования, так как тематика, стиль и техника сильно отличаются друг от друга. Однако существуют руководящие принципы, обеспечивающие повышения качества. Основными критериями качества являются: будет ли документ составлен и оформлен так, чтобы быть понятным целевой аудитории? Достигает ли он своей цели? Является ли он правильным и точным, учитывая ограниченный контекст? Общие руководящие принципы при комментировании таковы:

- Делайте заметки краткими, точными и конкретными. Если разработчик понимает, что понятные вопросы для лаконичности опускаются, то это облегчает общение и уменьшает сумятицу.
- Обозначение *n* в кружочке (примечание читателя) определяет комментарий. Чтобы написать *n*-примечание, определите порядковый номер примечания из списка ПРИМЕЧАНИЯ ЧИТАТЕЛЕЙ, присвойте примечанию номер, обведите его кружком и подключите примечание соответствующей тильдой. (См. Раздел С. 4 Стандартная форма диаграммы.)
- Критика должна быть конструктивной. Старайтесь предлагать решения или четко указывать на источники проблем.
- Найдите время, чтобы ознакомиться и обобщить все комментарии. Они могут быть размещены на обложке или на отдельном листе. (Не указывайте то, что уже размещено на других страницах.) Вопросы, которые должны обсуждаться на встречах Разработчиков / комментаторов могут быть обобщены. Список обсуждаемых вопросов должен быть конкретным.

Продолжительность времени, затрачиваемого на критику, зависит от множества факторов: знакомства с тем, что описывается, количества раз, когда данная тема уже рассматривалась, опыта читателя и разработчика и т. д. Комплект, возвращенный разработчику без каких-либо комментариев, кроме подписи читателя и галочки, означает, что читатель полностью согласен с разработчиком. Читатель должен осознавать, что за качество документа он несет общую ответственность с разработчиком.

#### С.2.2.2 Взаимодействие разработчика с комментатором

Когда читатель возвращает Комплект, разработчик отвечает, ставя "А" или "Х" на каждое *n*-примечание (Примечание читателя). "А" означает, что разработчик согласен с комментатором и учтет комментарий в следующую версию Комплекта. "Х" означает, что разработчик не согласен. Разработчик должен указать причину несогласия в письменном виде, так где помещен комментарий. После того, как разработчик



ответил на все замечания, Комплект возвращается для сохранения читателю.

#### С.2.2.3 Рекомендации по проведению совещаний

До тех пор, пока комментарии и ответы не будут представлены в письменном виде, читателям и разработчикам не рекомендуется вступать в диалог.

Назревшее совещание организуется следующим образом:

1. Каждое совещание должно быть ограничено по времени.
2. Каждое заседание должно начинаться с обсуждения определенных вопросов, которые относятся к одному или нескольким комментариям и ответам разработчиков, и заседание должно стремиться рассматривать именно эти вопросы.
3. Каждое заседание должно заканчиваться, когда участники соглашаются с тем, что уровень производительности снизился и индивидуальные усилия будут более полезны.
4. Каждое заседание должно заканчиваться согласованным перечнем пунктов повестки дня, которые могут включать в себя расписание последующих заседаний с конкретными повестками дня.
5. На каждом заседании должен быть назначен "секретарь", который обязан вести протокол и записывать предложения, решения и темы.
6. 6. Серьезные неразрешенные проблемы следует решать профессионально, документируя обе стороны, участвующие в споре.

Итогом совещания должно стать письменное решение вопросов или перечень вопросов, подлежащих разрешению соответствующим органом управления. Решение может быть представлено в форме дополнительного изучения любым из участников.

#### С.3 Комплекты IDEF

Комплект это технический документ. Он может содержать диаграммы, текст, глоссарии, краткое изложение решений, справочную информацию или что-либо представленное для рассмотрения и комментариев.

В соответствующем Титульном листе представлен материал в виде Комплекта. Титульный лист содержит поля для разработчика, даты, проекта, номера документа, названия, статуса и примечаний.

Существует два вида Комплектов IDEF

Стандартный комплект предназначен для комментариев. Он считается «рабочим документом», призванным помочь разработчику в уточнении его общей модели.

Комплект обновления содержит последнюю версию модели. Он рассылается только для информации и предназначена для помощи в сохранении текущей информации об общей модели в то время как часть модели обрабатывается в Цикле комплекта. Комплект обновления может включать только те страницы, которые были изменены с момента предыдущего обновления.

Комплекты обновления представляются через регулярные промежутки времени. Эти Комплекты включают в себя последнюю версию модели. От получателей комплектов обновления не предполагается получение комментариев, хотя они могут сделать это по своему усмотрению. Комплекты обновления хранятся получателями в архиве.

В соответствующем Титульном листе представлен материал в виде Комплекта. Титульный лист содержит поля для разработчика, даты, проекта, номера документа, названия, статуса и примечаний.

- Рабочая информация
  - Разработчик или команда, создающие модель
  - Название проекта и номер задачи
  - Дата сдачи оригинала в библиотеку
  - Даты всех опубликованных редакций
  - Статус модели (рабочая, черновая, рекомендована к принятию или публикации в качестве окончательной модели)
- Информация рецензента
  - Подача и копирование информации
  - Список рецензентов Комплекта
  - Запланируйте дату различных этапов Цикла комплекта
- Информация о содержании
  - Оглавление Комплекта
  - Состояние каждого раздела комплекта
  - Комментарии или специальные инструкции для библиотекаря
- Идентификационная информация
  - Имя модели (в поле Узла)
  - Название модели
  - С-номер

```
2da828f359656474706450726f00010001000000110001000000000000000000
```





- Форма диаграммы имеет стандартный размер для удобства передачи и копирования. Форма разделена на три основных раздела:

- Форма диаграммы должна использоваться для информации, представляемой в письменном виде.

7075626c017800020000036d616300000000000000000000000000000000  
000000000000000000000000a8  
57368342440001000001420946696775726520433400000000000000000000  
0000000000000000  
00000000000000000000000000000000000000000000000000000000000000  
0000000000000000000b7  
2ba82916c0656474706450726f000100010000001100010000000000000000  
0000000000000000a6669  
6e616c2d666970730001000400000142000200186d61633a66696e616c2d66  
6970733a46696775  
7265204334000900a700a76166706d00000000000200180039005900750095  
009e012a00000000  
00000000000000000000000000000000000000000000000000000000000773  
7065636b6c6500000000  
0000000000000000000000000000000000000000000000000036d6163000000000000  
0000000000000000  
00000000000000000000467756e6e000000000000000000000000000000000  
00000000000000000000000000000000  
0000000000000000000000000000000000000000ffff000001010000a8  
2932fa000020bf800000  
02005c2d20000000000005c2d30000188a400000000

# СТАНДАРТНАЯ ФОРМА ДИАГРАММЫ

ИСПОЛЬЗУЕТСЯ ПРИ:	РАЗРАБОТАЛ:	ДАТА:	РАБОЧИЙ	КОНТЕКСТ:
	ПРОЕКТ:	РЕД.	ПРОЕКТ	
	ПРИМЕЧАНИЯ 1 2 3 4 5 6 7 8 9 10		РЕКОМЕНДУЕМЫЙ	
			ПУБЛИКАЦИЯ	
НОДА	НАЗВАНИЕ			НОМЕР
				СТРАНИЦА:

**Рисунок С4. Стандартная форма диаграммы**

Форма составлена таким образом, что Рабочие информационные поля, расположенные в верхней части формы, могут быть убраны после заполнения окончательной версии «одобренной к публикации».

Идентификационные поля располагаются вертикально друг под другом, когда формы (с необрезанной верхней частью) прикрепляются кнопками, в порядке сверху вниз к доске или стене. Открытые полосы идентификационного поля выступают в качестве буквенные указатели, расположенные в порядке индекса ноды. Если соблюдается формат двусторонней публикации «пара страниц», когда нижняя часть страницы с текстом и контекстом обращены к переплету, то поле сообщения на странице становится видимым, при поднимании вверх каждой предшествующей страницы, закрепленной на стене.

## С.4.1 Рабочая информация

### С.4.1.1 Поля " Разработчик / Дата / Проект"

В данных полях приводится информация о том кто первоначально создал диаграмму, дата, когда она была впервые начерчена и название проекта, под которым она была создана. Поле «дата» может содержать дополнительные даты, написанные ниже исходной даты. Эти даты представляют собой изменения в листе оригинала. Если лист

переиздается без каких-либо изменений, то дата пересмотра не добавляется.

#### С.4.1.2 Поле «Примечания читателя»

В этом поле читатель оставляет свои примечания после ознакомления с диаграммой. При появлении комментария на странице соответствующий номер примечания зачеркивается. Этот процесс гарантирует, что каждому примечанию читателя на диаграмме присваивается уникальный номер и что номера на каждой диаграмме являются последовательными.

#### С.4.1.3 Поле «Статус»

Классификации статусов указывают на стадии утверждения, а именно:

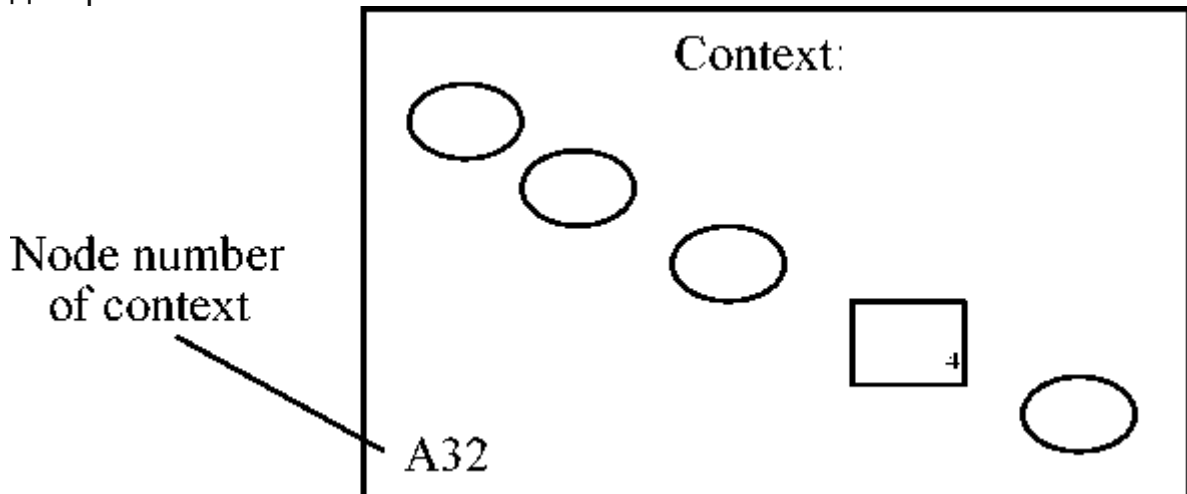
РАБОЧИЙ	Диаграмма подвергается серьезным изменениям, перезапускается процедура утверждения. Как правило, прежде чем будет сменен статус выполняется несколько редакций рабочего проекта. Новый разработчик диаграммы обычно устанавливает статус рабочий.
ПРОЕКТ	Диаграмма подвергается незначительным изменениям по сравнению с предыдущей диаграммой и имеет определенный уровень согласования, принятый группой читателей. Диаграмма в стадии «Проект» -это диаграмма, предложенная руководителем задания (который может быть разработчиком). Диаграммы в стадии «Проект» могут подвергаться дальнейшему пересмотру, если они включены в текущий комплект вместе с другими диаграммами или возвращены к рассмотрению из файла какого-либо читателя, даже если они не входят в текущий комплект. Статус проекта остается в силе до тех пор, пока диаграмма не будет принята на совещании технического комитета или совета.
РЕКОМЕНДУЕМЫЙ	Диаграмма и сопровождающий ее текст рассмотрены и одобрены совещанием технического комитета или совета, и считается, что эта диаграмма не изменится.
ПУБЛИКАЦИЯ	Документ может быть отправлена для окончательной печати и публикации.

#### С.4.1.4 Поле «Читатель/Дата»

В этой области читатель должен указать свою фамилию и дату рассмотрения каждой формы.

#### С.4.1.5 Поле " Контекст"

Эскиз макета блока родительской диаграммы с выделенным родительским блоком. Номер ноды родительской схемы записывается в левом нижнем углу контекстного поля (Рисунок С5). Номер родительского блока может быть записан в выделенном блоке, хотя он также является последней цифрой в записи поля ноды дочерней диаграммы.



**Рисунок С5. Иллюстрация контекстного поля**

Возникают следующие особые случаи: 1) Контекстное поле требуемой формы контекстной диаграммы A-0 «ТОР», записанное в центре поля. 2) Контекстное поле дополнительной контекстной диаграммы высокого уровня A-1 это A-2, эскиз, если имеется такая контекстная диаграмма более высокого уровня, и аналогично для A-n, где n = 2 или более. 3) Контекстное поле для контекстной диаграммы высшего уровня, A-n, для самого большого n (= 1 или больше) «NONE». Иллюстративный пример см. на рис.21.

#### С.4.1.6 Поле «Используется в» ("Used At")

Это список диаграмм, помимо родительского контекста, которые каким-либо образом используют или ссылаются на данную страницу формы диаграммы.

Чаще всего используется перечисление одной или нескольких ссылок нод на подмодели, для которых родительский блок этой дочерней диаграммы обеспечивает поддержку вызовов при детализации этого дочернего элемента. При необходимости (случай n=1 принимается условно), выражение "node\_reference\_expression" (которое начинается с «model\_name/», за которым следует номер узла) заканчивается «Mn», n больше или равно 2, превращая ссылку в полную ICOM-кодovou ссылку на конкретную стрелку механизма высшего блока поддерживаемой подмодели. Например, " TLS/A34M2", записанное в поле Used At для дочерней диаграммы " MN/A211", утверждает, что родительский блок " MN/A21 "обеспечивает поддержку механизма через вторую стрелку механизма его верхнего блока для подмодели " TLS/A34."

При такой поддержке удаленной субмодели, указанной в поле «Used At», любой дочерний блок на дочерней диаграмме (а более обще, родительский блок, предоставляющий поддержку, или любые дочерние блоки) могут быть призваны поставлять деталей для некоторого



доступного дочернего блока (рассматриваемого как верхний блок субмодели), чья ссылка на ноду появляется в поле «Used At.» Доступный дочерний блок это блок, доступный по стрелке поддержки механизма ветвления, источником которого является ICOM-код, подключенный к поддержке механизма.

Если верхняя часть диаграммы отрезана для публикации, то содержимое поля Used At должно быть скопировано в поле Message в виде n -note (примечание к модели).

#### С.4.2 Поле "Сообщение" ("Message")

Поле «Сообщение» содержит основное сообщение, которое должно быть передано. Это поле обычно используется для построения диаграмм с помощью графического языка IDEF. Однако оно может быть использовано для других целей: глоссарий, контрольные списки, заметки, эскизы и т. д. Участники проекта не должны использовать никаких других документов, кроме форм диаграмм, чтобы система регистрации на основе ссылочных номеров могла обеспечить полную запись проекта. Этому может способствовать поддержка инструмента IDEF, включающего в себя электронную почту и доску объявлений, с автоматическим присвоением каждому участнику С-нумерации и «предпочтений».

#### С.4.3 Поле «Нода» ("Node" )

Это поле содержит полную ссылку на ноду для листа (включая model\_name, косую черту, номер ноды и "F "(для FEO)," T "(для текста) или" G "(для глоссария) —с номером страницы" 1 "или" 2 " и т. д. и прилагается в конце, чтобы указать переполнение страниц, если это необходимо), так что лист однозначно определен для справочных и других целей.

#### С. 4. 4 Поле «Название» ("Title")

Поле «Название» содержит название материала, представленного на Форме диаграммы. Если поле «Message» содержит диаграмму, то содержимое поля «Title» должно точно соответствовать имени, написанному в родительском блоке.

#### С. 4. 5 Поле «Номер» ("Number")

##### С. 4. 5. 1 Поле «Номер» (Большая область)

Большая область содержит С-номер. С-номер состоит из двух или трех букв инициалов разработчика (выбранных для того, чтобы быть уникальными среди участников проекта), за которыми следует номер, присвоенный разработчиком. С-номер помещается в левом нижнем углу поля "Number" и является основным средством ссылки на сам лист, поскольку используемый лист как форма диаграммы может быть создан только один раз. Каждая форма диаграммы, используемая разработчиком, получает уникальный С-номер, который обычно должен быть первой меткой, указанной на форме.

Если в проекте принято решение отслеживать историю версий, то С-номер формы диаграммы, для которой этот лист является измененной версией, должен быть записан в скобках (пробел необязательный) после записи С-номера разработчика в поле «С-номер». Например, "AB34

(CD123) "означает, что этот лист ("AB34") предназначен для замены уже существующего"CD123".

При публикации модели номер C может быть заменен стандартным порядковым номером страницы (например, «стр. 17»).

С. 4. 5. 2 Поле " Номер" ("Number") («Номер страницы Комплекта» Малая прямоугольная Область)

Номер страницы Комплекта записывается библиотекарем в правой части поля "Number" внутри небольшого прямоугольника. Он состоит из номера документа, за которым следует буква, идентифицирующая лист внутри документа.

#### С. 5 Хранение файлов

Каждый официально назначенный участник проекта ведет досье полученных документов «Читатель/Разработчик». Библиотекарь должен отслеживать официальные основные и справочные файлы проекта, архивируя каждый Комплект, представленный в ходе проекта.

Изменения в процессе подачи могут происходить в зависимости от индивидуальных предпочтений, но следующие файлы, хранящиеся в алфавитно-отсортированном порядке ссылочных номеров в качестве основной организационной структуры подачи, являются минимальными:

- Стандартный Комплект Файлов  
Обрабатывается разработчиками, комментаторами и, возможно, читателями. Титульные листы комплекта файлов в хронологическом порядке, как главный журнал, но извлекают и хранят большинство C-страниц в соответствующих моделях в порядке ссылок на узлы. Если вы сомневаетесь, оставьте лист в поданном Комплекте, возможно, добавив перекрестные ссылки на Примечания читателя на уже подшитых формах диаграмм в выбранных других местах подачи. (Каждый лист является собственностью читателя, и нумерация заметок читателя перезапускается заново для каждого листа с номером C, поэтому добавленные личные примечания читателя не могут причинить никакого вреда.)
- Обновленные файлы текущей модели:  
Обрабатываются разработчиками, комментаторами и читателями из полученных Обновленных Комплектов. Может быть отдано предпочтение версиям Project Master по мере развития проекта.
- чиеРабо файлы:  
Обрабатываются разработчиками и любым читателем, который инициирует специальный обмен между участниками, не имеющими официального назначенного разработчика. Титульный лист Комплекта для темы, начинаемой читателем, должен иметь предлагаемое название, чтобы упорядочение содержания комплекта в алфавитном порядке соответствовало официальным рабочим файлам моделей и т. Д.

- Файлы проекта:  
Обрабатываются библиотекарем в соответствии со стандартами, установленными руководством проекта.

### С. 6 Процедура обзора модели IDEF

В дополнение к Циклу комплектов была разработана пошаговая процедура в качестве руководства для представления информации о модели группе «рецензентов» (возможно, не имеющих опыта самостоятельного анализа моделей IDEF0 ). Она не заменяет процесс обзора цикла «Читатель/Разработчик», который является центральным для обеспечения качества моделирования IDEF0 (поясняется в С.2), но может быть полезна при периодическом использовании проекта на техническом уровне, чтобы предоставить возможность всем участникам обмениваться или разрабатывать общие интерпретации, которые могут не проявляться в индивидуальных обменах на основе Комплекта.

1. Представьте модель, подлежащую анализу, с помощью перечня нод Это оглавление модели. Предоставляет краткий обзор того, что должно произойти.
2. Представьте выбранные термины глоссария. Поощряйте каждого рецензента заменять собственные толкования слов теми, которые выбрала команда презентации. На данном этапе значения не должны подвергаться сомнению. Изменение значения теперь потребует значительных изменений в диаграммах.
3. Представьте каждую диаграмму для рассмотрения.

Обзор диаграммы-это упорядоченный, пошаговый процесс, в ходе которого можно задавать вопросы, которые помогут выявить потенциальные слабые места в диаграмме или ее тексте. Ниже приведены шесть шагов структурированного пошагового обзора. Поправки к диаграмме могут быть предложены на любом этапе. Исправления могут быть приняты к исполнению немедленно или позже.

Шаг 1: Сканирование диаграммы.

Этот шаг позволяет читателю получить общее представление о содержании диаграммы. Как правило, читатель просматривает родительскую диаграмму, которая отображает текущую диаграмму как один из ее блоков. Читатель изучает, как автор декомпозировал функцию.

Критерии принятия:

1. Декомпозиция полезна для достижения назначенной цели и завершена в контексте родительского блока. Все функции нижнего уровня можно четко разделить на категории в каждом из блоков.
2. Диаграмма отражает, по мнению рецензента, соответствующую точку зрения, основанную на цели модели.
3. По мнению рецензента, предоставленной новой информации достаточно, чтобы улучшить понимание родительского блока. Применено не так много деталей на диаграмме, чтобы она казалась сложной для чтения.

Если проблема недостаточно очевидна, критический разбор может быть отложен до следующего шага 3. Однако первое впечатление должно быть сформировано. Проблемы могут быть указаны на магнитно-маркерной доске пока не будут решены

Шаг 2: Анализ родителя.

Как только читатель поймет декомпозицию текущей диаграммы, родительская диаграмма должна быть пересмотрена для обеспечения совместимости.

Критерии принятия:

1. Декомпозиция охватывает все точки, которые рецензент ожидал увидеть при чтении родительской диаграммы.
2. Теперь, когда декомпозиция этой части родительской диаграммы продемонстрирована, детализация, которую рецензент предусмотрел для этого блока все еще считается правильной. Если нет, обратите внимание на недостающие детали.

На этом этапе может быть важно на короткое время вернуться к родительской диаграмме и добавить новые n-примечания (примечания читателя) или доработать существующие, основываясь на дополнительном понимании, полученном при изучении декомпозиции.

Шаг 3: Объедините родительский блок и детализированную диаграмму. На этом шаге проверяются подсоединения стрелок интерфейса от родителя к дочерним элементам.

Критерии принятия:

1. Нет ни пропущенных, ни лишних стрелок интерфейса.
2. Граничные стрелки помечены соответствующими кодами ICOM.
3. Метки дочерних стрелок это то же самое, что и метки со стрелками родителей. Метки передают правильное и полное содержание стрелок.
4. Изучение соединительных стрелок не выявило никаких проблем в родительской диаграмме. (Добавленный интерфейс может привести к неправильному пониманию сообщения, переданного родителем).

Выполните обход по часовой стрелке всех четырех сторон родительского блока, проверяя каждую стрелку, что обеспечит выявления соответствия граничных стрелок кодов ICOM родительским стрелкам.

Шаг 4: Проанализируйте шаблон внутренней стрелки.

Шаблон из блока и стрелок является основным представлением создаваемой модели.

Каждый блок необходимо рассмотреть в порядке номеров нод и размещение стрелок в порядке ICOM для каждого блока. Когда этот процесс завершен, рецензенты должны пройти по диаграмме, изучить последствия ситуаций, с которыми они знакомы, и проверить способность диаграммы моделировать известные взаимосвязи.

Критерии принятия:

1. Диаграмма не выглядит загроможденной. Количество пересечений и изгибов стрелок сведено к минимуму.
2. Блоки сбалансированы с учетом деталей. В каждом блоке должно быть одинаковое количество деталей. Однако компромиссы по этому критерию приемлемы ради ясности.
3. Диаграмма должна соответствовать опыту и знаниям рецензента в данной области. Обратная связь и условия возникновения ошибок должны быть показаны так, как понятно рецензенту.
4. Уровень детализации стрелок должен соответствовать уровню детализации блоков. Следует рассмотреть возможность объединения стрелок в более общие стрелки.

Шаг 5: Ознакомьтесь со вспомогательной документацией.

На этом этапе рассматриваются вопросы, которые разработчик выделил в тексте, глоссарии и диаграмме-иллюстрации FEO.

Критерии принятия:

1. Текст подтверждает интерпретацию, полученную при изучении самой диаграммы.
2. Нормальные пути, обратная связь, обработка ошибок и другие функции, предлагаемые текстом, находятся на диаграмме или в диаграмме FEO (только для экспозиции).
3. Существенные особенности диаграммы, выявленные на этапах 1-4, отражены в тексте, глоссарии или FEO.
4. Ссылки на диаграмму достаточно подробны, чтобы связать текст, глоссарий или FEO с конкретными частями диаграммы.

Шаг 6: Установите статус диаграммы.

Установите состояние диаграммы (как определено ранее) на:

- Рабочий
- Проект
- Рекомендуемый
- Публикация

## Приложение D Информативные определения

Этот раздел содержит определения, относящиеся к нормативным разделам настоящего документа. Определения нормативных разделов приведены в Разделе 2. Термин, если он определен, указан в Разделе 2 или в Приложении D, но не в обоих местах одновременно.

D.1 Уровень утверждения:

одно из следующих четырех состояний, присвоенных модели IDEF для обозначения степени рассмотрения и утверждения:

D.2 Разработчик (Автор):

лицо, которое разрабатывает и несет ответственность за любую конкретную модель IDEF или диаграмму.

Рабочий	(Самый низкий уровень)
Проект	(Следует после самого низкого уровня)
Рекомендуемый	(Следует после высокого уровня)
Публикация	(Самый высокий уровень)

### D.3 Кластеризация:

Блоки на диаграммах группируются и разделяются. При кластеризации два или более блоков группируются в один блок. Цель кластеризации объединить связанные функции в единую, более общую функцию.

### D.4 Комментатор:

читатель, имеющий достаточную подготовку в технике IDEF, чтобы предлагать конкретные комментарии, используя систему нумерации примечаний читателя и (часто) ссылаясь на недостатки в применении самой техники.

Назначенный читатель, который разделяет ответственность с автором за качество Комплекта IDEF, модели, диаграммы или другого результата IDEF.

### D. 5 Проект:

см. Уровень утверждения.

### D.6 Эксперт:

человек, знакомый с частью моделируемой системы (или объекта).

Может служить источником информации или рецензентом части модели.

### D.7 Роль IDEF:

Должность в проекте IDEF. См. термины разработчик, эксперт, комментатор, читатель и библиотекарь.

### D.8 Комплект:

Стандартизированный пакет диаграмм, содержащий части или законченные современные модели, подлежащие обзору. Смотрите Цикл Комплекта.

### D.9 Титульный лист комплекта:

Специальная форма, используемая для контроля маршрутизации комплекта через Цикл комплекта.

### G.10 Цикл комплекта:

Формальная процедура цикла читатель / разработчик, в которой используются Комплекты для получения экспертной оценки во время разработки модели.

D.11 Библиотекарь:

Лицо, ответственное за маршрутизацию и отслеживание комплектов, а также за упорядоченное хранение проектных файлов и архивов.

D.12 Поле проекта:

Поле в форме диаграммы IDEF, в котором записывается название задачи, для решения которой разрабатывается модель IDEF.

D. 13 Публикация:

см. Уровень одобрения.

D.14 Читатель:

человек с (ограниченной) подготовкой в технике IDEF, достаточной для точной интерпретации синтаксиса и основных значений, а также для чтения и записи примечаний читателя, который рассматривает часть или всю модель.

D.15 Цикл Читатель / Разработчик:

Процедура, использующая Примечания читателя для получения экспертной оценки или экспертного заключения в ходе разработки модели.

D.16 Примечание читателя:

текстовый комментарий читателя к диаграмме IDEF0. Примечания читателя не публикуются как часть диаграммы, а скорее используются для общения во время цикла Читатель/Разработчик.

D. 17 Рекомендуются:

См. Уровень утверждения.

D.18 Рецензент:

Рецензент несет ответственность за целесообразность и правильность Комплекта IDEF, модели, диаграммы или других документов IDEF.

Некоторым рецензентам не хватает читательской подготовки, но они участвуют на определенных этапах разработки проекта.

D.19 Разделение:

Блоки на диаграммах группируются и разделяются. Когда родительский блок детализируется на дочерней диаграмме, родительский блок разбивается на части, некоторые из которых затем могут быть сгруппированы, чтобы сформировать от 3 до 6 блоков на дочерней диаграмме.

D. 20 Роль:

То же самое, что и роль IDEF.

D. 21 Работа:

см. Уровень одобрения.