

**КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
ВЫСШАЯ ШКОЛА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ**

Кафедра программной инженерии

И.Р. ПЕТРОВА, Р.Х. ФАХРТДИНОВ, А.А. СУЛЕЙМАНОВА

**МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО
МОДЕЛИРОВАНИЯ IDEF0**

Учебно-методическое пособие по курсу

“Методы анализа и построения информационных систем”

Казань – 2018

УДК 004.436.4

*Рекомендовано Учебно-методической комиссией
Высшей школы информационных технологий и интеллектуальных систем
Протокол № 2 от 18 октября 2018 года*

Рецензент:

Руководитель отдела разработки специального
программного обеспечения (РСПО1) АО «ICL— КПО ВС»
Р.Р. Алтынбаев

Петрова И.Р.

Методология функционального моделирования IDEF0/ И.Р. Петрова,
Р.Х. Фахртдинов, А.А.Сулейманова . – Казань: Казан. ун-т, 2018. – 68 с.

Учебно-методическое пособие «Методология функционального моделирования IDEF0» содержит описание концепции IDEF0, определения основных понятий метода (разделы 1,2). В пособии рассмотрены синтаксис и семантика языка IDEF0 (разделы 3,4), свойства диаграмм, отношения между блоками на диаграммах (разделы 5,6). В разделе 7 рассматриваются правила присвоения ссылочных выражений всем элементам диаграмм. В разделе 8 приводится методика разработки функциональной модели IDEF0 и общие правила построения диаграмм.

Учебно-методическое пособие предназначено для студентов, изучающих курс «Методы анализа и построения информационных систем»

© Петрова И.Р., 2018

© Казанский университет, 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. КОНЦЕПЦИЯ IDEF0	6
2. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ МЕТОДОЛОГИИ И ЯЗЫКА IDEF0	9
3. СИНТАКСИС ГРАФИЧЕСКОГО ЯЗЫКА IDEF0	17
4. СЕМАНТИКА ЯЗЫКА IDEF0	19
Семантика блоков и стрелок	19
Имена и метки	28
Семантические правила блоков и стрелок.....	30
Диаграммы IDEF0	31
Контекстная диаграмма верхнего уровня	32
Дочерняя диаграмма	33
Родительская диаграмма.....	34
Текст и глоссарий.....	36
5. СВОЙСТВА ДИАГРАММ	37
Стрелки как ограничения	37
Параллельное функционирование	37
Ветвление и слияние сегментов стрелок	38
Отношения блоков на диаграммах.....	43
6. ОТНОШЕНИЯ МЕЖДУ БЛОКАМИ ДИАГРАММЫ И ДРУГИМИ ДИАГРАММАМИ	45
Граничные стрелки	46
ISOM - кодирование граничных стрелок	47
Стрелки, помещенные в «туннель».....	48
7. ССЫЛОЧНЫЕ ВЫРАЖЕНИЯ (КОДЫ)	49
Номера блоков.....	49
Узловые номера.....	50
Перечень узлов.....	51
Дерево узлов.....	52
8. МЕТОДИКА РАЗРАБОТКИ ФУНКЦИОНАЛЬНЫХ МОДЕЛЕЙ IDEF 0 ..	52
Общие положения	52
Классификация функций, моделируемых блоками IDEF0	55
Организационно-технические структуры и механизмы IDEF0-моделей.	57
Управление – особый вид процесса, операции, действия	60
Типизация функциональных моделей и IDEF 0– диаграмм.....	61
Методика построения модели	62
Правила построения диаграмм.....	63
ЛИТЕРАТУРА	68

ВВЕДЕНИЕ

Постоянное усложнение производственно-технических и организационно-экономических систем – фирм, предприятий, производств, и др. субъектов производственно-хозяйственной деятельности - и необходимость их анализа с целью совершенствования функционирования и повышения эффективности обуславливают необходимость применения специальных средств описания и анализа таких систем. Эта проблема приобретает особую актуальность в связи с появлением автоматизированных производств и предприятий.

В США это обстоятельство было осознано еще в конце 70-ых годов, когда ВВС США предложили и реализовали Программу интегрированной компьютеризации производства ICAM (ICAM - Integrated Computer Aided Manufacturing), направленную на увеличение эффективности посредством широкого внедрения компьютерных (информационных) технологий.

Реализация программы ICAM потребовала создания адекватных методов анализа и проектирования производственных систем и способов обмена информацией между специалистами, занимающимися такими проблемами. Для удовлетворения этой потребности в рамках программы ICAM была разработана методология IDEF (ICAM Definition), позволяющая исследовать структуру, параметры и характеристики производственно-технических и организационно-экономических систем (в дальнейшем, там, где это не вызывает недоразумений – систем). Общая методология IDEF состоит из трех частных методологий моделирования, основанных на графическом представлении систем:

- IDEF0 используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающие эти функции;
- IDEF1 применяется для построения информационной модели, отображающей структуру и содержание информационных потоков, необходимых для поддержки функций системы;
- IDEF2 позволяет построить динамическую модель меняющихся во времени поведения функций, информации и ресурсов системы.

К настоящему времени наибольшее распространение и применение имеют методологии IDEF0 и IDEF1 (IDEF1X).

Методология IDEF0 основана на подходе разработанном Дугласом Т. Россом в начале 70–ых годов и получившем название SADT (Structured Analysis & Design Technique - метод структурного анализа и проектирования). Основу подхода и, как следствие, методологии IDEF0, составляет графический язык описания (моделирования) систем, обладающий следующими свойствами.

- Графический язык - полное и выразительное средство, способное наглядно представлять широкий спектр деловых, производственных и других процессов и операций предприятия на любом уровне детализации.

- Язык обеспечивает точное и лаконичное описание моделируемых объектов, удобство использования и интерпретации этого описания.

- Язык облегчает взаимодействие и взаимопонимание системных аналитиков, разработчиков и персонала изучаемого объекта (фирмы, предприятия), т.е. служит средством «информационного общения» большого числа специалистов и рабочих групп, занятых в одном проекте, в процессе обсуждения, рецензирования, критики и утверждения результатов.

- Язык прошел многолетнюю проверку и продемонстрировал работоспособность как в проектах ВВС США, так и в других проектах, выполнявшихся государственными и частными промышленными компаниями.

- Язык легок и прост в изучении и освоении.

- Язык может генерироваться рядом инструментальных средств машинной графики; известны коммерческие программные продукты, поддерживающие разработку и анализ моделей - диаграмм IDEF0, например, продукт Design/IDEF 3.7 (и более поздние версии) фирмы Meta Software Corporation.

Перечисленные свойства языка predeterminedелили выбор методологии IDEF0 в качестве базового средства анализа и синтеза производственно-технических и организационно-экономических систем, что нашло свое отражение в упомянутых федеральных стандартах США.

1. КОНЦЕПЦИЯ IDEF0

Function Modeling — методология функционального моделирования. С помощью наглядного графического языка IDEF0 изучаемая система предстает перед разработчиками и аналитиками в виде набора взаимосвязанных функций (функциональных блоков — в терминах IDEF0). Как правило, моделирование средствами IDEF0 является первым этапом изучения любой системы.

Методология IDEF0 позволяет моделировать всю систему как набор чередующихся функций. Простая система обозначений и строгий набор правил построения призван обеспечить точность и ясность при моделировании.

Сущность функционального моделирования

Для любой системы определяющим является ее функциональное содержание, так как оно определяет ее основные свойства. Поэтому в основе функционального моделирования лежит функциональное содержание системы, в качестве отношений между функциями рассматривается информация об объектах, связывающих эти функции.

Методология IDEF0

В основе *IDEF0*-методологии лежат четыре основных понятия:

- функциональный блок;
- интерфейсная дуга (стрелка);
- декомпозиция.

Методология IDEF0 основана на следующих концептуальных положениях.

1. **Модель** – искусственный объект, представляющий собой отображение (образ) системы и ее компонентов. М моделирует А, если М отвечает на вопросы относительно А. Здесь М – модель, А – моделируемый объект (оригинал). Модель разрабатывают для понимания, анализа и принятия решений о реконструкции (реинжиниринге) или замене существующей, либо проектировании новой системы. Система представляет собой совокупность взаимосвязанных и взаимодействующих частей, выполняющих некоторую полезную работу. Частями (элементами) системы могут быть любые

комбинации разнообразных сущностей, включающие людей, информацию, программное обеспечение, оборудование, изделия, сырье или энергию (энергонасосители). Модель описывает, что происходит в системе, как ею управляют, какие сущности она преобразует, какие средства использует для выполнения своих функций и что производит.

2. **Блочное моделирование** и его графическое представление. Основной концептуальный принцип методологии IDEF – представление любой изучаемой системы в виде набора взаимодействующих и взаимосвязанных блоков, отображающих процессы, операции, действия, происходящие в изучаемой системе. В IDEF0 все, что происходит в системе и ее элементах, принято называть функциями. Каждой функции ставится в соответствие блок. На IDEF0 диаграмме блок представляет собой прямоугольник. Интерфейсы, посредством которых блок взаимодействует с другими блоками или с внешней по отношению к моделируемой системе средой, представляются стрелками), входящими в блок или выходящими из него. Входящие стрелки показывают, какие условия должны быть одновременно выполнены, чтобы функция, описываемая блоком, осуществилась.

3. **Лаконичность и точность.** Документация, описывающая систему, должна быть точной и лаконичной. Многословные характеристики, изложенные в форме традиционных текстов, неудовлетворительны. Графический язык позволяет лаконично, однозначно и точно показать все элементы (блоки) системы и все отношения и связи между ними, выявить ошибочные, лишние или дублирующие связи и т.д..

4. **Передача информации.** Средства IDEF0 облегчают передачу информации от одного участника разработки модели (отдельного разработчика или рабочей группы) к другому. К числу таких средств относятся:

- диаграммы, основанные на простой графике блоков и стрелок, легко читаемые и понимаемые;
- метки на естественном языке для описания блоков и стрелок, глоссарий и сопроводительный текст для уточнения смысла элементов диаграммы;

- последовательная декомпозиция диаграмм, строящаяся по иерархическому принципу, при котором на верхнем уровне отображаются основные функции, а затем происходит их детализация и уточнение;

- древовидные схемы иерархии диаграмм и блоков, обеспечивающие обозримость модели в целом и входящих в нее деталей.

5. Строгость и формализм. Разработка моделей IDEF0 требует соблюдения ряда строгих формальных правил, обеспечивающих преимущества методологии в отношении однозначности, точности и целостности сложных многоуровневых моделей. Эти правила описываются ниже. Здесь отмечается только основное из них: все стадии и этапы разработки и корректировки модели должны строго, формально документироваться с тем, чтобы при ее эксплуатации не возникало вопросов, связанных с неполнотой или некорректностью документации.

6. Итеративное моделирование. Разработка модели в IDEF0 представляет собой пошаговую, итеративную процедуру. На каждом шаге итерации разработчик предлагает вариант модели, который подвергают обсуждению, рецензированию и последующему редактированию, после чего цикл повторяется. Такая организация работы способствует оптимальному использованию знаний системного аналитика, владеющего методологией и техникой IDEF0, и знаний специалистов – экспертов в предметной области, к которой относится объект моделирования.

7. Отделение «организации» от «функций». При разработке моделей следует избегать изначальной «привязки» функций исследуемой системы к существующей организационной структуре моделируемого объекта (предприятия, фирмы). Это помогает избежать субъективной точки зрения, навязанной организацией и ее руководством. Организационная структура должна явиться результатом использования (применения) модели. Сравнение результата с существующей структурой позволяет, во-первых, оценить адекватность модели, а во-вторых – предложить решения, направленные на совершенствование этой структуры.

2. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ МЕТОДОЛОГИИ И ЯЗЫКА IDEF0

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

Ветвление: разделение стрелки на два или большее число сегментов. Может означать «развязывание пучка».

Внутренняя стрелка: входная, управляющая или выходная стрелка, концы которой связывают источник и потребителя, являющиеся блоками одной диаграммы. Отличается от граничной стрелки.

Входная стрелка: класс стрелок, которые отображают вход IDEF0-блока, то есть данные или материальные объекты, которые преобразуются функцией в выход. Входные стрелки связываются с левой стороной блока IDEF0.

Выходная стрелка: класс стрелок, которые отображают выход IDEF0-блока, то есть данные или материальные объекты, произведенные функцией. Выходные стрелки связываются с правой стороной блока IDEF0.

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

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.

Декомпозиция: разделение моделируемой функции на функции - компоненты.

Дерево узлов: представление отношений между родительскими и дочерними узлами модели IDEF0 в форме древовидного графа. Имеет то же значение и содержание, что и перечень узлов.

Диаграмма А-0: специальный вид (контекстной) диаграммы IDEF0, состоящей из одного блока, описывающего функцию верхнего уровня, ее входы, выходы, управления, и механизмы, вместе с формулировками цели модели и точки зрения, с которой строится модель.

Диаграмма: часть модели, описывающая декомпозицию блока.

Диаграмма-иллюстрация (FEO): графическое описание, используемое, для сообщения специфических фактов о диаграмме IDEF0. При построении диаграмм FEO можно не придерживаться правила IDEF0.

Дочерний блок: блок на дочерней (порожденной) диаграмме.

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

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

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.

Код ICOM: аббревиатура(Input - Вход, Control - Управление, Output - Выход, Mechanism – Механизм), код, обеспечивающий соответствие граничных стрелок дочерней диаграммы со стрелками родительского блока; используется для ссылок.

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

Контекстная диаграмма: диаграмма, имеющая узловой номер A-n ($n \geq 0$), 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.

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

Модель IDEF0: графическое описание системы, разработанное с определенной целью и с выбранной точки зрения. Комплект одной или более диаграмм IDEF0, которые изображают функции системы с помощью графики, текста и глоссария.

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

Перечень узлов: список, часто ступенчатый, показывающий узлы модели IDEF0 в упорядоченном виде. Имеет то же значение и содержание, что и дерево узлов.

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

Родительская диаграмма: диаграмма, которая содержит родительский блок.

Родительский блок: блок, который подробно описывается дочерней диаграммой.

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

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

Семантика: значение синтаксических компонентов языка.

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

Слияние: объединение двух или большего числа сегментов стрелок в один сегмент. Может означать «развязывание пучка».

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

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: входная стрелка, выходная стрелка, управляющая стрелка, стрелка механизма (включает стрелку вызова).

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.

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.

Стрелка, помещенная в туннель (туннельная стрелка): стрелка (со специальной нотацией), не удовлетворяющая обычному требованию, согласно которому каждая стрелка на дочерней диаграмме должна соответствовать стрелкам на родительской диаграмме.

Текст: любой текстовый (не графический) комментарий к графической диаграмме IDEF0.

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

Точка зрения: указание на должностное лицо или подразделение организации, с позиции которого разрабатывается модель

Узел: блок, порождающий дочерние блоки; родительский блок.

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

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

Узловой номер: код, присвоенный блоку и определяющий его положение в иерархии модели; может быть использован в качестве подробного ссылочного выражения.

Управляющая стрелка: класс стрелок, которые в IDEF0 отображают управления, то есть условия, при выполнении которых выход блока будет правильным. Данные или объекты, моделируемые как управления, могут преобразовываться функцией, создающей соответствующий выход. Управляющие стрелки связываются с верхней стороной блока IDEF0.

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

Цель: краткая формулировка причины создания модели.

Пример контекстной функции

Контекстная функция – функция верхнего уровня модели, имеет иерархическую структуру, которая несет имя основного действия выполняемого системой. Изображается на отдельной диаграмме, называемой контекстной (Рис. 1).

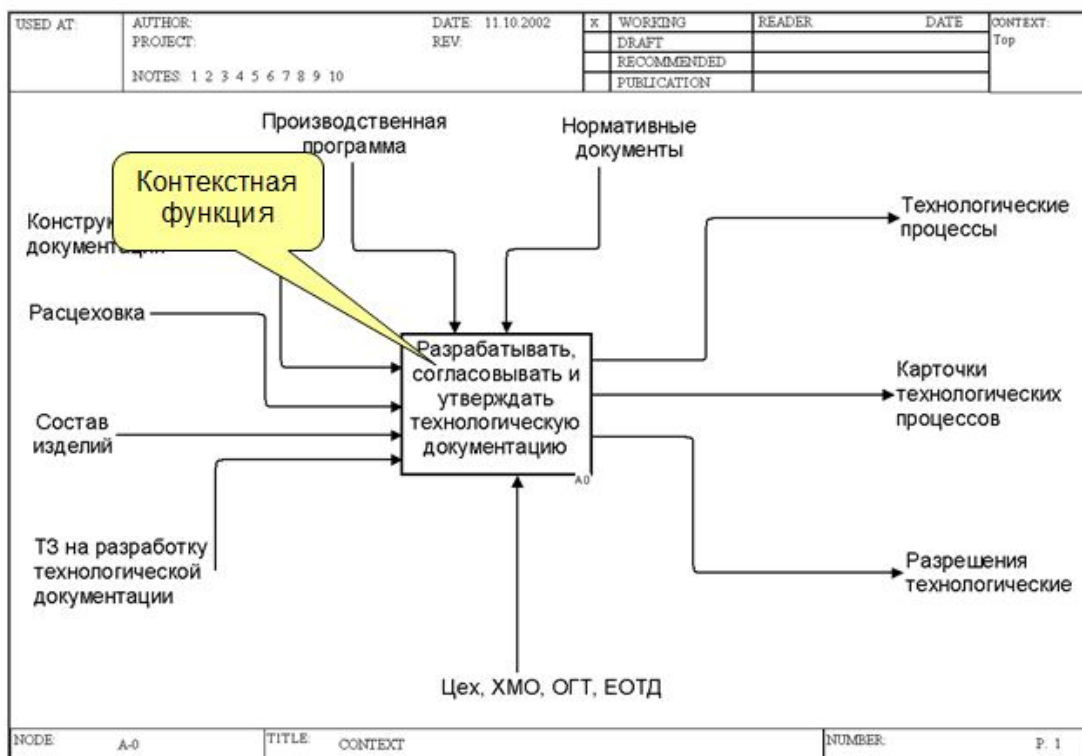


Рис. 1

Пример контекстной диаграммы

На контекстной диаграмме указываются связи системы с внешним миром (Рис. 2).

Потоки на контекстной диаграмме делятся на (Рис. 3):

- входные (то, что перерабатывается системой);
- выходные (результат работы системы);
- управления (регламентирующая и управляющая информации или правила);
- механизма (ресурсы выполняющие работы).

Система преобразует входные потоки в выходные с учетом управления и с использованием механизмов.

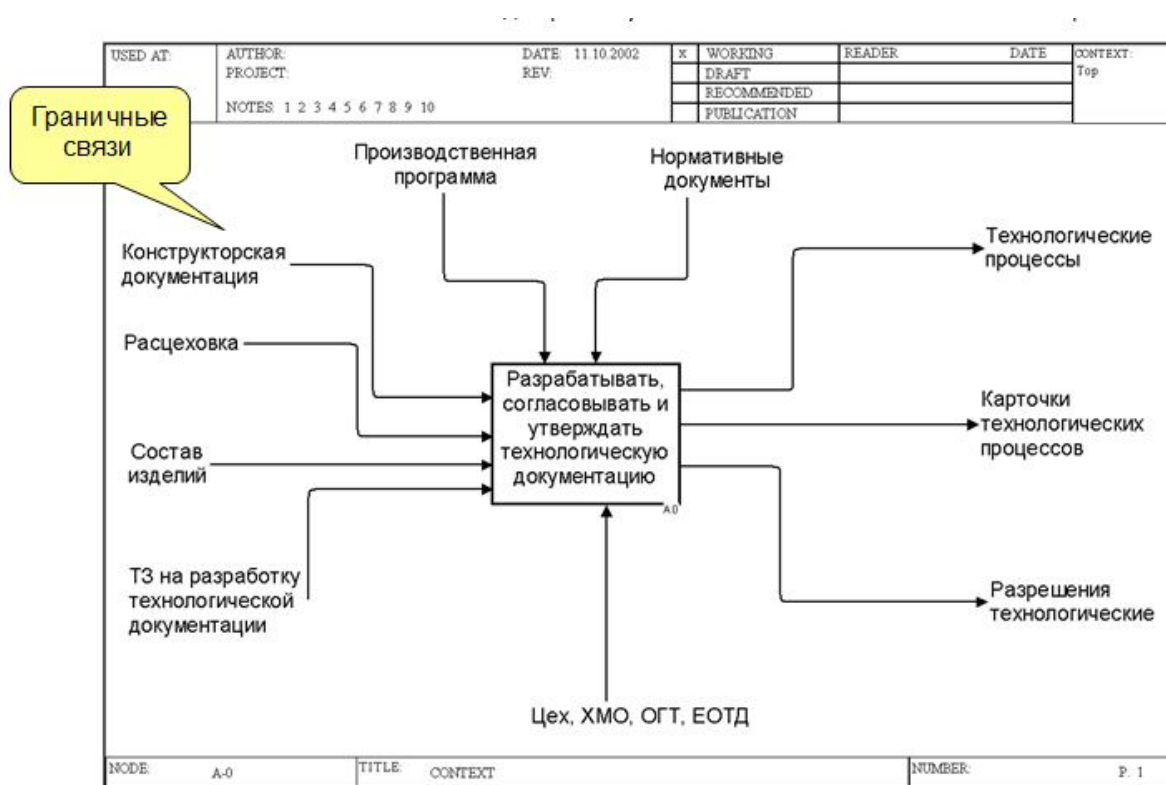


Рис. 2

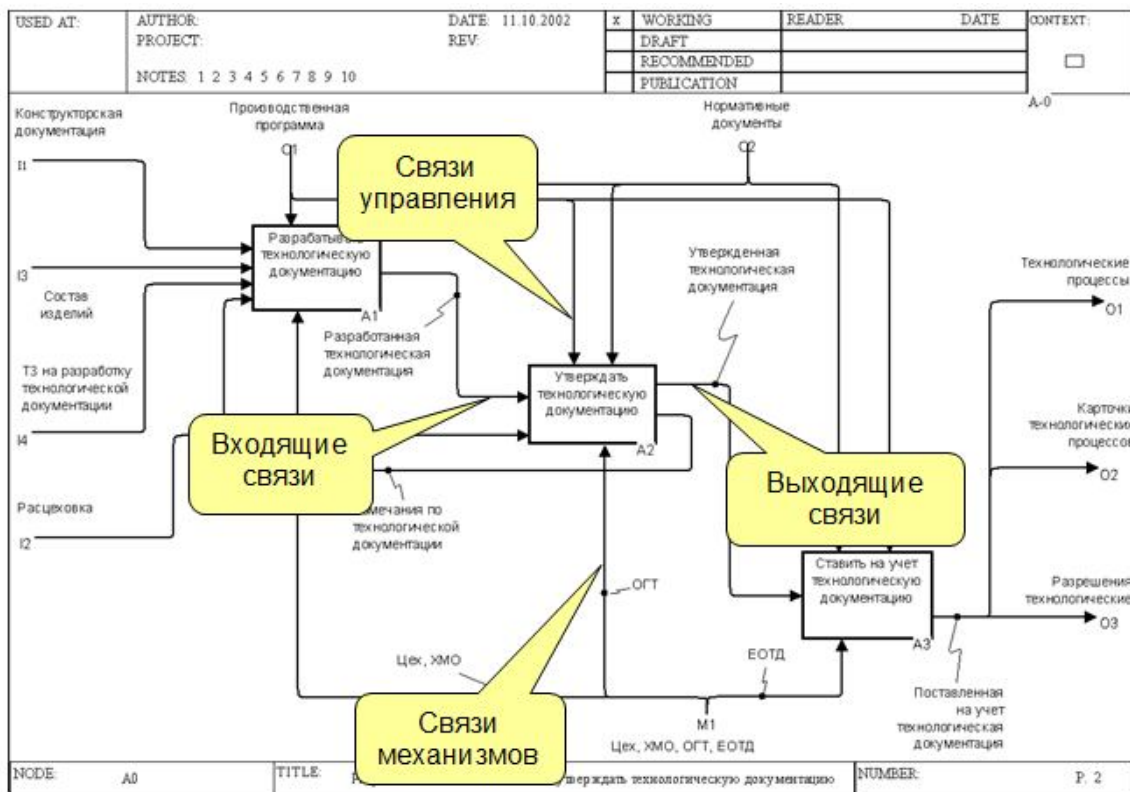


Рис. 3

Пример декомпозиции

Диаграммы декомпозиции содержат вложенное поуровневое описание функций модели. Декомпозиция важна для систем, поэтому им необходимо уделить внимание. Декомпозиция, как процесс расчленения, позволяет рассматривать любую исследуемую систему как сложную, состоящую из отдельных взаимосвязанных подсистем, которые, в свою очередь, также могут быть расчленены на части (Рис. 4).

При декомпозиции связи автоматически продолжают на следующем уровне, как показано на рисунке (Рис. 5).

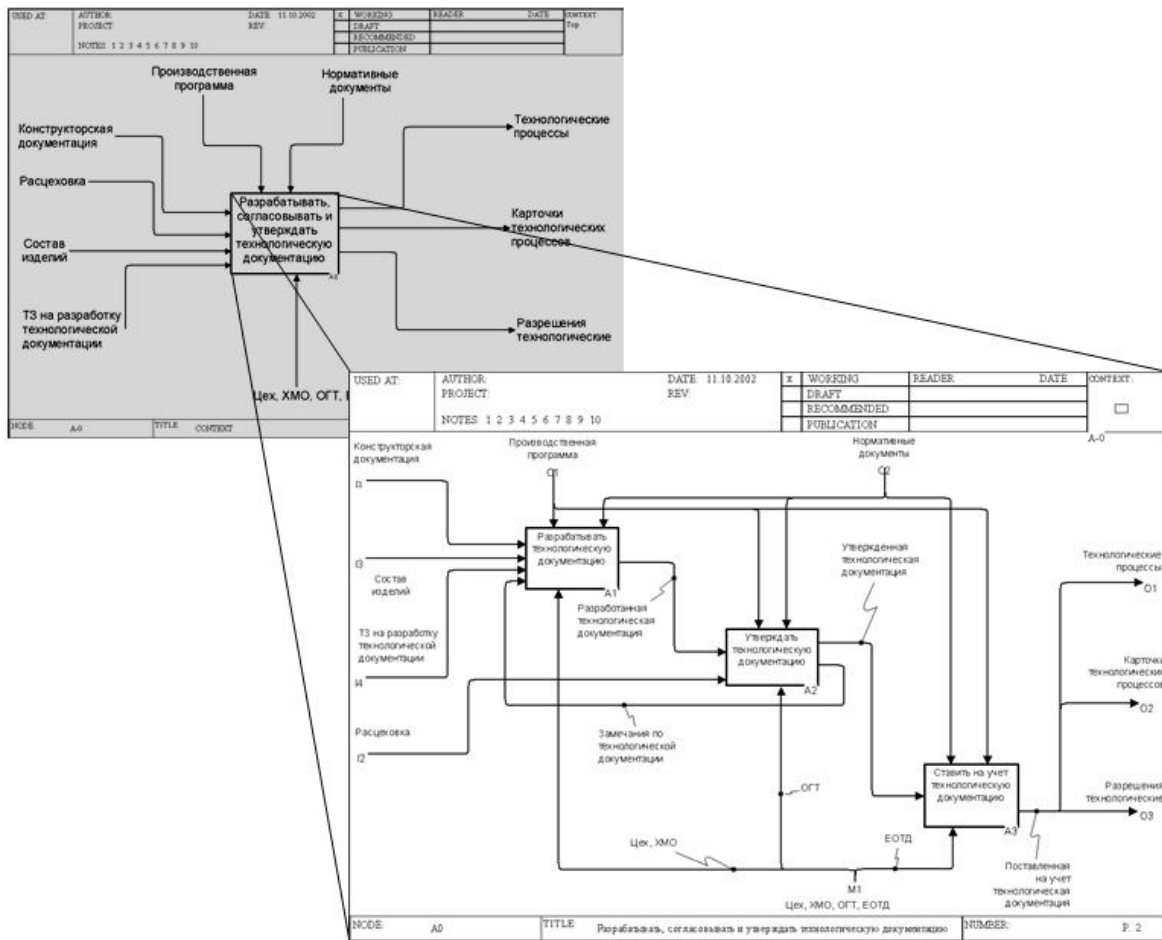


Рис. 4

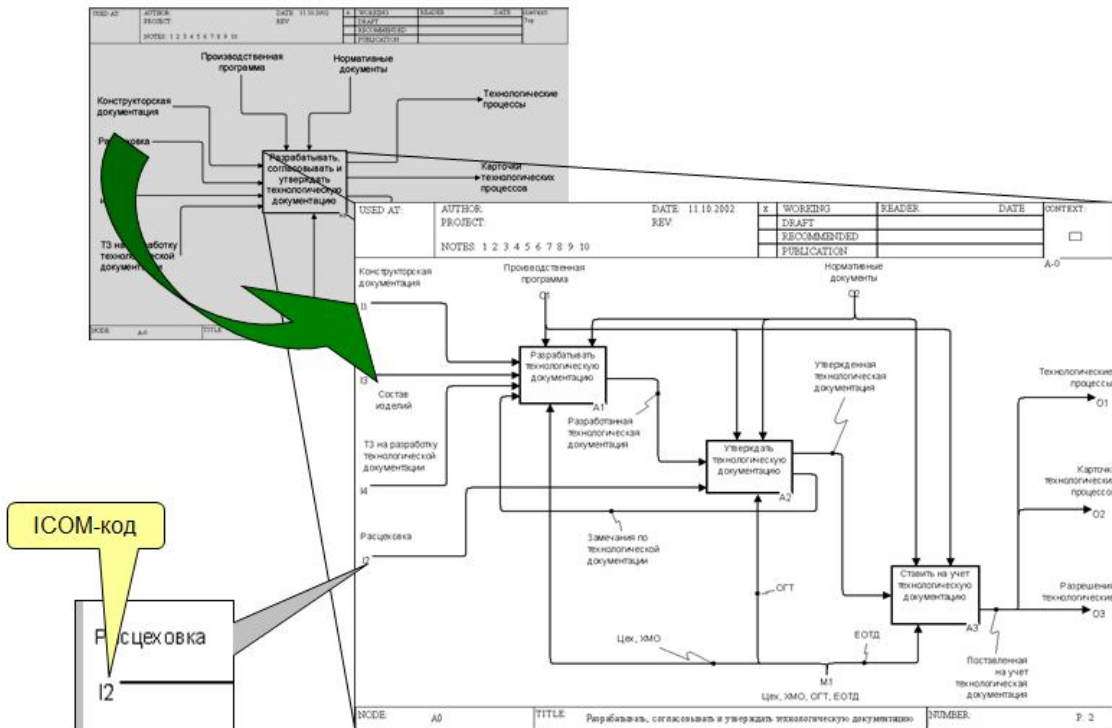


Рис. 5

3. СИНТАКСИС ГРАФИЧЕСКОГО ЯЗЫКА IDEF0

Набор структурных компонентов языка, их характеристики и правила, определяющие связи между компонентами, представляют собой синтаксис языка. Компоненты синтаксиса IDEF0 – блоки, стрелки, диаграммы и правила. 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.

Блок описывает функцию. Типичный блок показан на рисунке **Рис. 6**. Внутри каждого блока помещается его имя и номер. Имя должно быть активным глаголом или глагольным оборотом, описывающим функцию. Номер блока размещается в правом нижнем углу. Номера блоков используются для их идентификации на диаграмме и в соответствующем тексте.

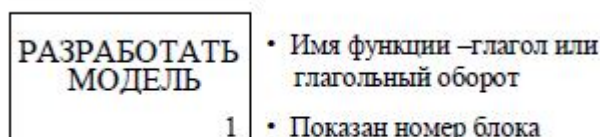


Рис. 6

Стрелка формируется из одного или более отрезков прямых и наконечника на одном конце. Как показано на рисунке Рис. 7, сегменты стрелок могут быть прямыми или ломаными; в последнем случае горизонтальные и вертикальные отрезки стрелки сопрягаются дугами, имеющими угол 90°. Стрелки не представляют поток или последовательность событий, как в традиционных блок-схемах потоков или процессов. Они лишь показывают, какие данные или материальные объекты должны поступить на вход функции для того, чтобы эта функция могла выполняться.

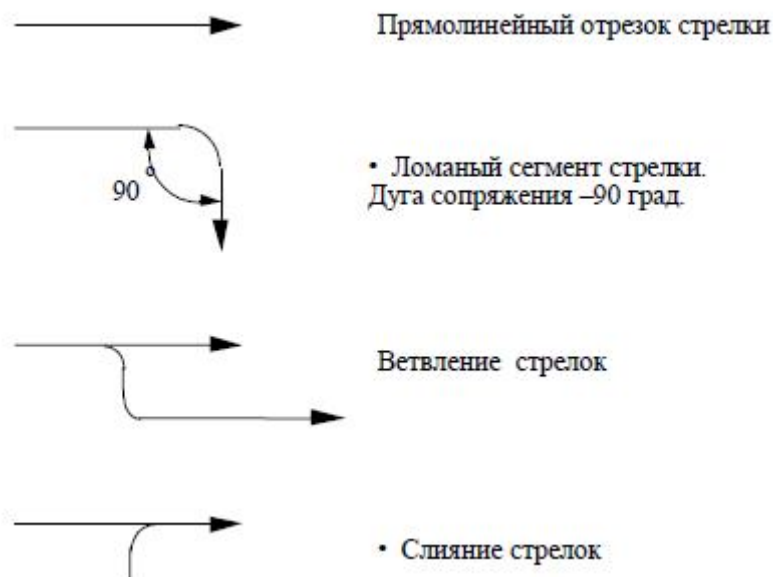


Рис. 7

Синтаксис блоков.

1. Размеры блоков должны быть достаточными для того, чтобы включить имя блока.

2. Блоки должны быть прямоугольными, с прямыми углами.

3. Блоки должны быть нарисованы сплошными линиями.

Синтаксис стрелок.

1. Ломаные стрелки изменяют направление только под углом 90 град.

2. Стрелки должны быть нарисованы сплошными линиями различной толщины.

3. Стрелки могут состоять только из вертикальных или горизонтальных отрезков; отрезки, направленные по диагонали, не допускаются.

4. Концы стрелок должны касаться внешней границы функционального блока, но не должны пересекать ее.

5. Стрелки должны присоединяться к блоку на его сторонах. Присоединение в углах не допускается.

4. СЕМАНТИКА ЯЗЫКА IDEF0

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.

Семантика блоков и стрелок

Поскольку IDEF0 есть методология 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 входные, выходные 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 сохраняются при декомпозиции Стрелки и их сегменты, как отдельные, так и связанные в «пучок», помечаются существительными или оборотами существительного. Метки сегментов позволяют конкретизировать данные или материальные объекты, передаваемые этими сегментами, с соблюдением синтаксиса ветвлений и слияний.

Каждая сторона функционального блока имеет стандартное значение с точки зрения связи блок/стрелки. В свою очередь, сторона блока, к которой присоединена стрелка, однозначно определяет ее роль.

Стрелки, входящие в левую сторону блока - входы. Входы преобразуются или расходуются функцией, чтобы создать то, что появится на ее выходе.

Стрелки, входящие в блок сверху - управления. Управления определяют условия, необходимые функции, чтобы произвести правильный выход.

Стрелки, покидающие блок справа – выходы, т.е. данные или материальные объекты, произведенные функцией.

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).

Стандартное расположение стрелок показано на рисунках Рис. 8, Рис. 9.

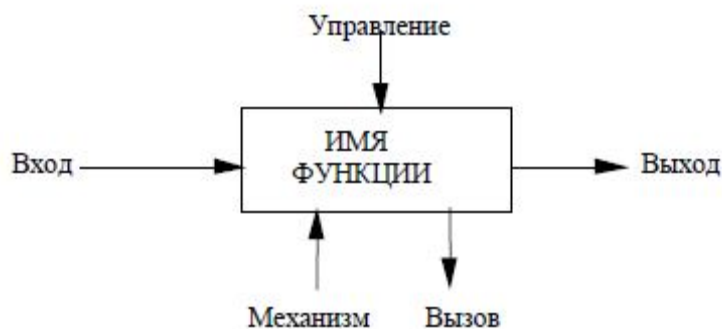


Рис. 8

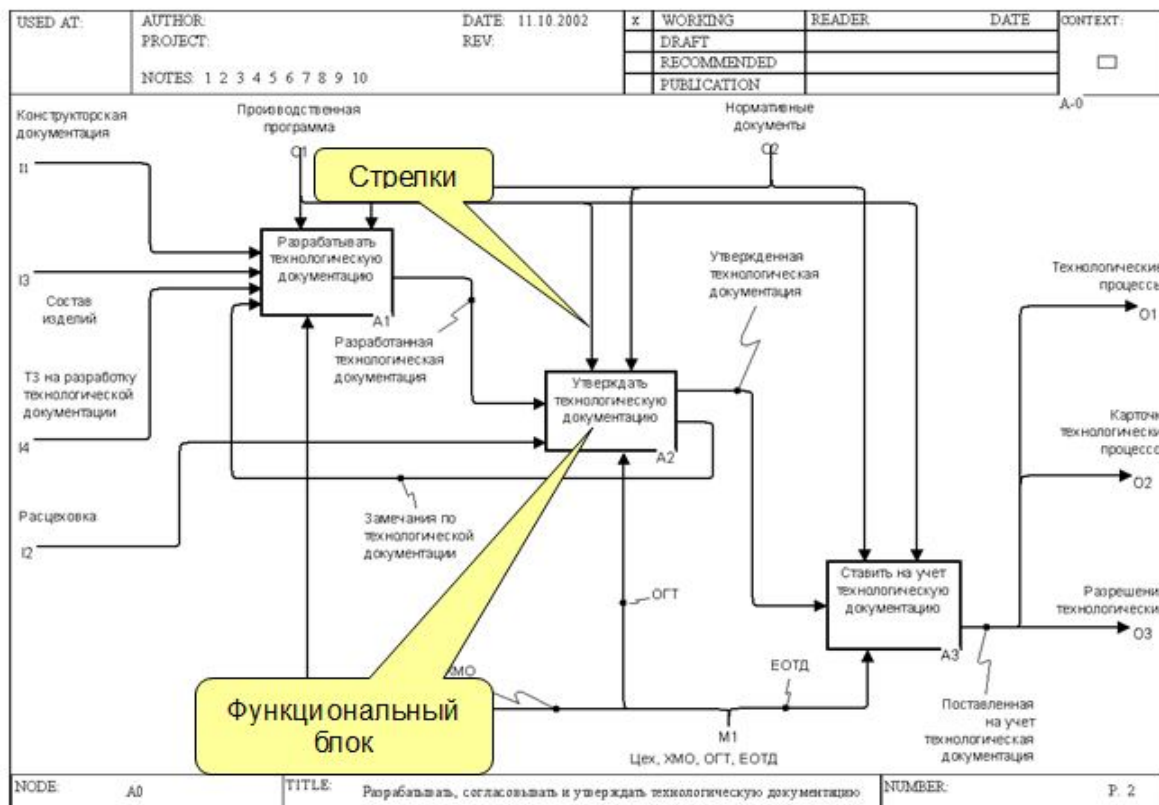


Рис. 9

Применение стрелок

В литературе часто встречается термин ICOM (Input/Control/Output/Mechanism), обозначающий четыре основных типа стрелок: вход, управление, выход, механизм. Механизм и управление не видоизменяются в процессе выполнения функции. Если какой либо поток данных преобразуются функцией, то характер этих изменений должен быть отражен в названии потоков на входе и выходе.

Вход (Input) – материальный или информационный поток который потребляется или преобразовывается функцией чтобы произвести результат работы на выходе. Данная стрелка входит в левую грань блока. Если какой либо поток данных преобразуются функцией, то характер этих изменений должен быть отражен в названии потоков на входе и выходе (Рис. 10).

Управление (Control) – содержит неизменяемые объекты: правила, инструкции, стандарты в соответствии с которым выполняется функция. Их присутствие обязательно, на диаграмме данная стрелка изображается как входящая в верхнюю грань блока (Рис. 11).

Выход (Output) – результат работы функции. Присутствие выходов обязательно. Изображается данная стрелка как выходящая из правой грани (Рис. 12).

Механизм (Mechanism) – неизменяемые ресурсы выполняющие работу функции, например организационные единицы предприятия, отдельные работники, машины и механизмы, вычислительные системы и программные средства. Присутствие обязательно. Изображается такая стрелка как входящая в нижнюю грань (Рис. 13).

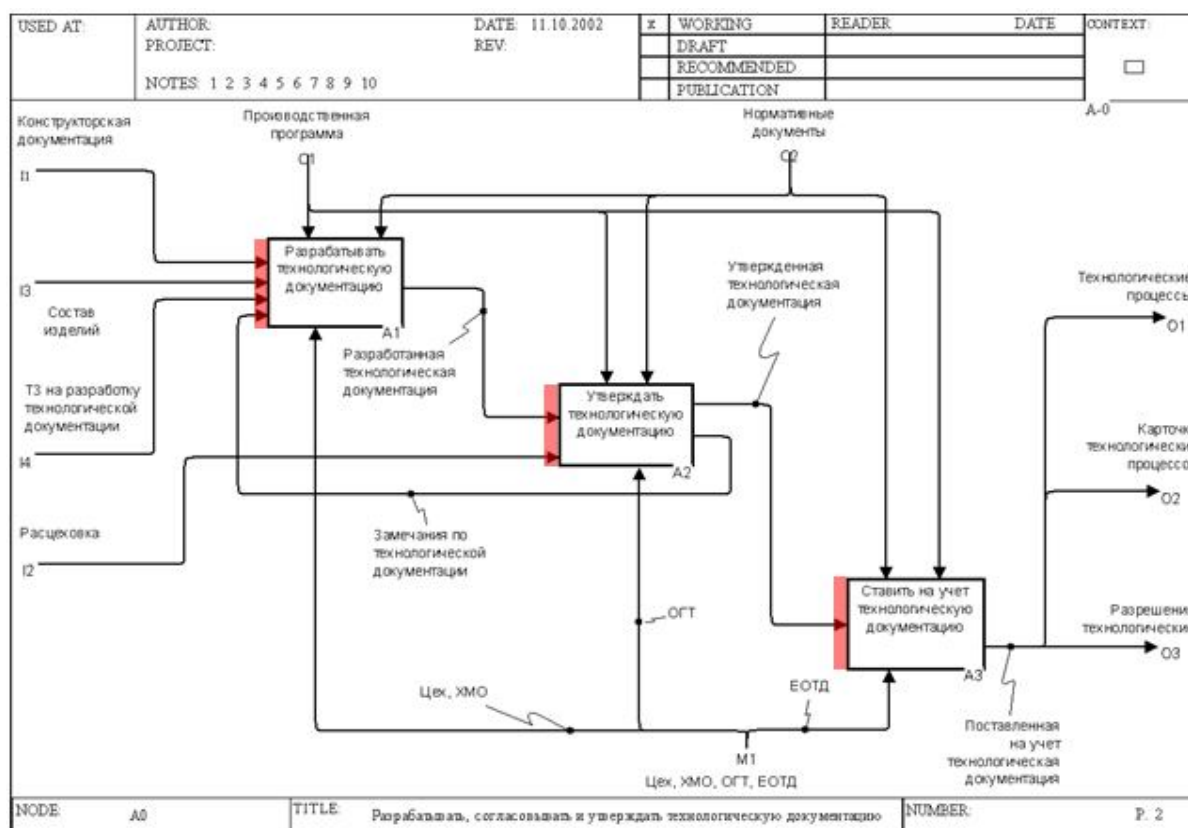


Рис. 10

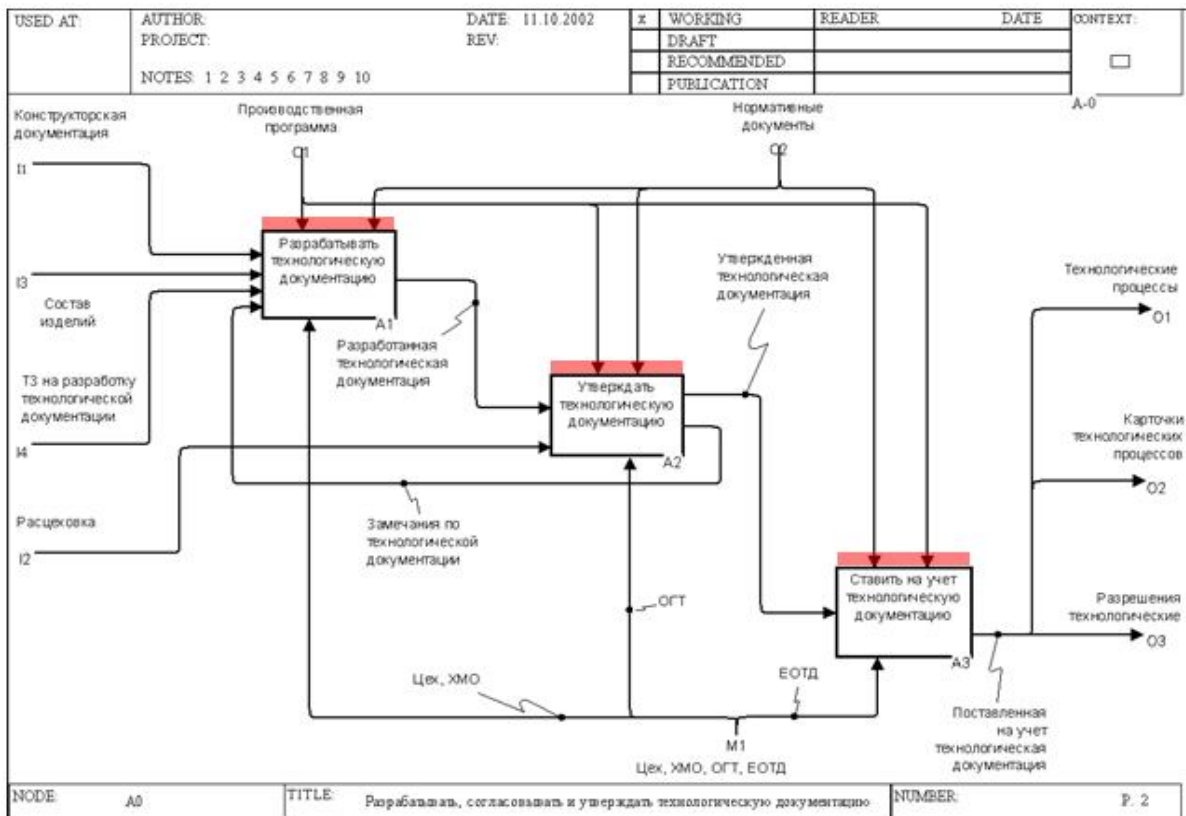


Рис. 11

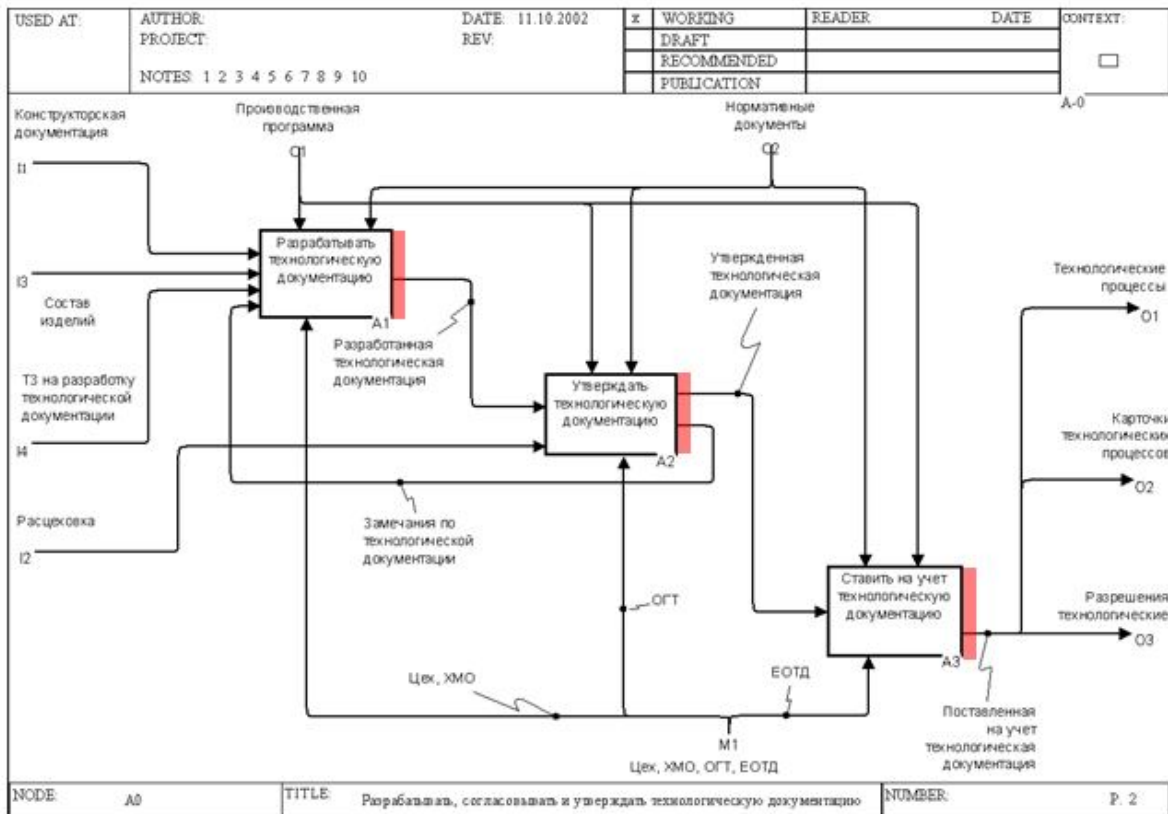


Рис. 12

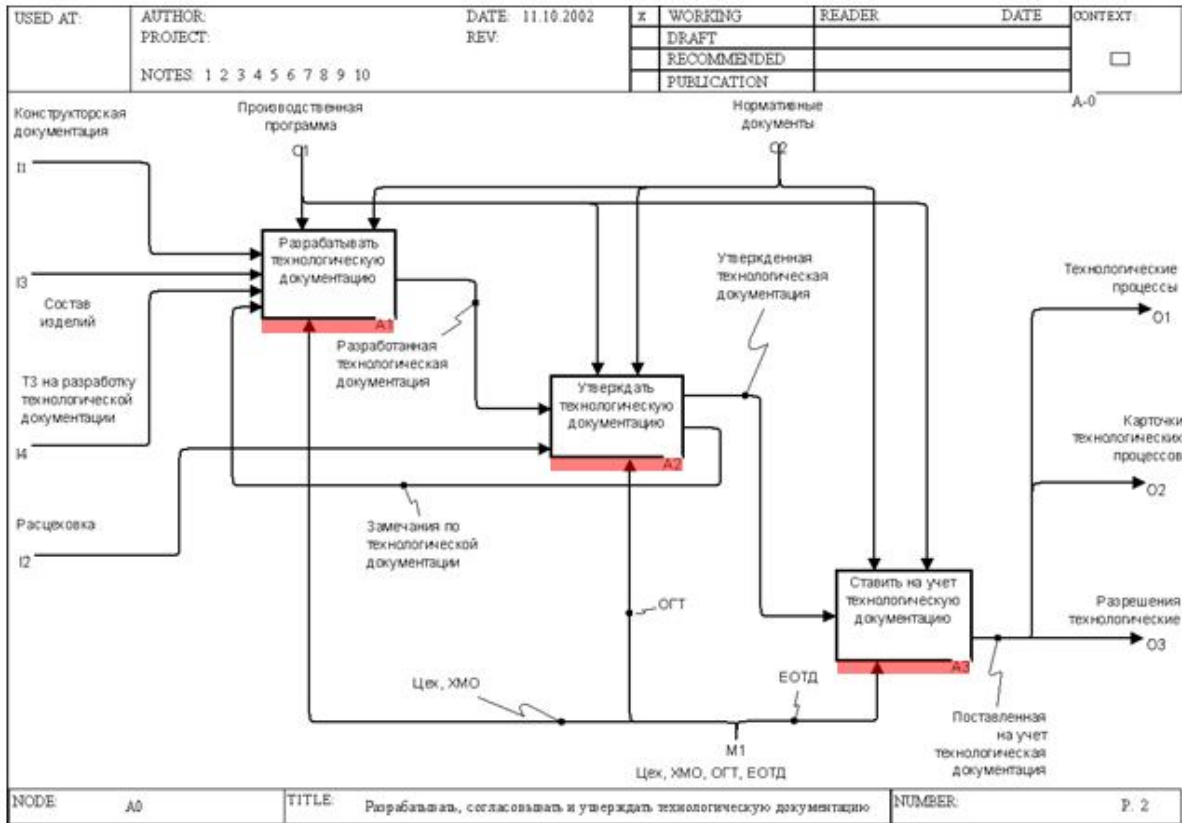


Рис. 13

Граничные стрелки начинаются от границ диаграммы и заканчиваются у функции или наоборот. Связывают функции диаграммы с внешним миром. Определяются на родительской диаграмме. Для идентификации граничных стрелок используются ICOM-коды (Рис. 14).

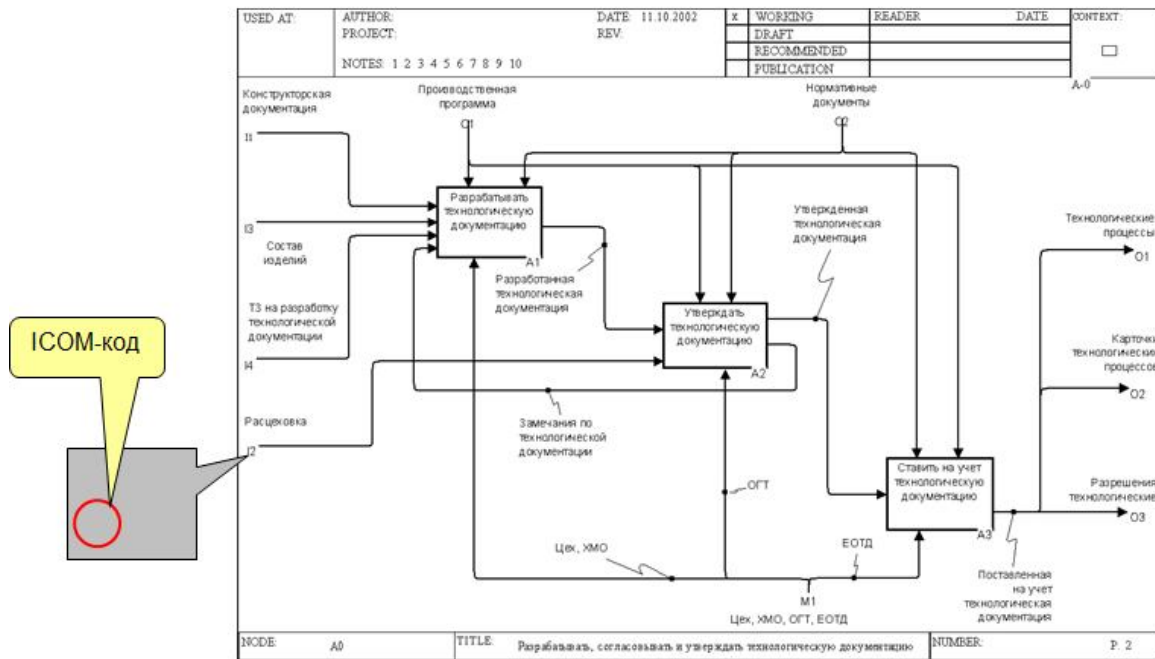


Рис. 14

Кроме того, выделяют внутренние связи, которые не касаются границ диаграммы.

Выход-вход замыкает выход одного функционального блока на вход другого, как показано на рисунке (Рис. 15)

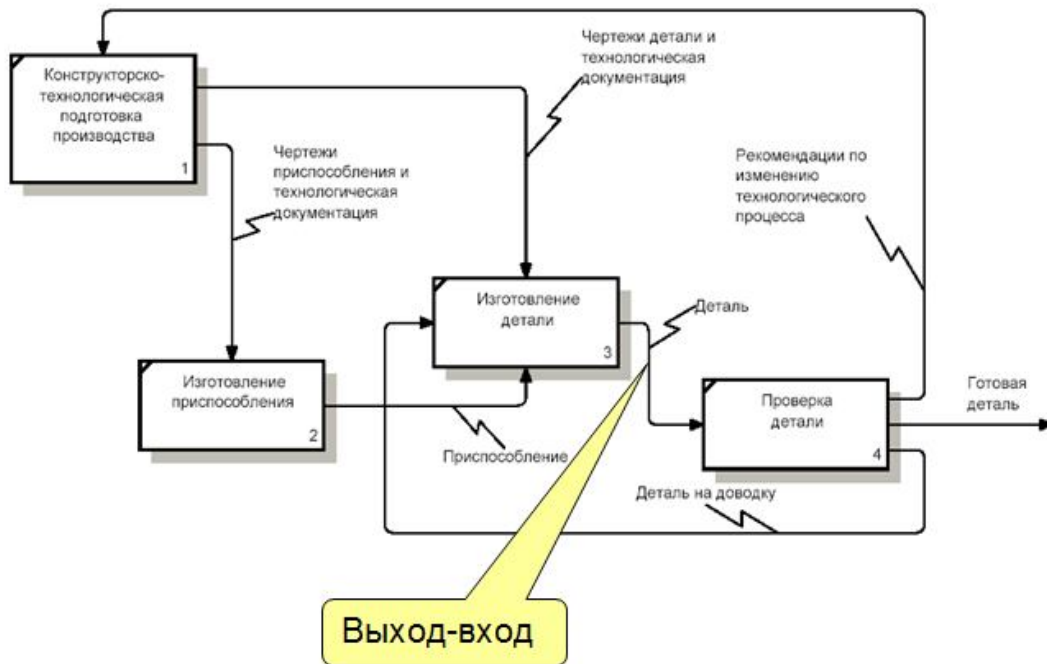


Рис. 15

Выход-управление показывает, что предшествующая функция управляет выполнением последующей (Рис. 16).



Рис. 16

Когда одна функция выбирает или создает ресурс, который будет являться исполнительным механизмом другой функции, такую связь называют **выход-механизм** (Рис. 17).

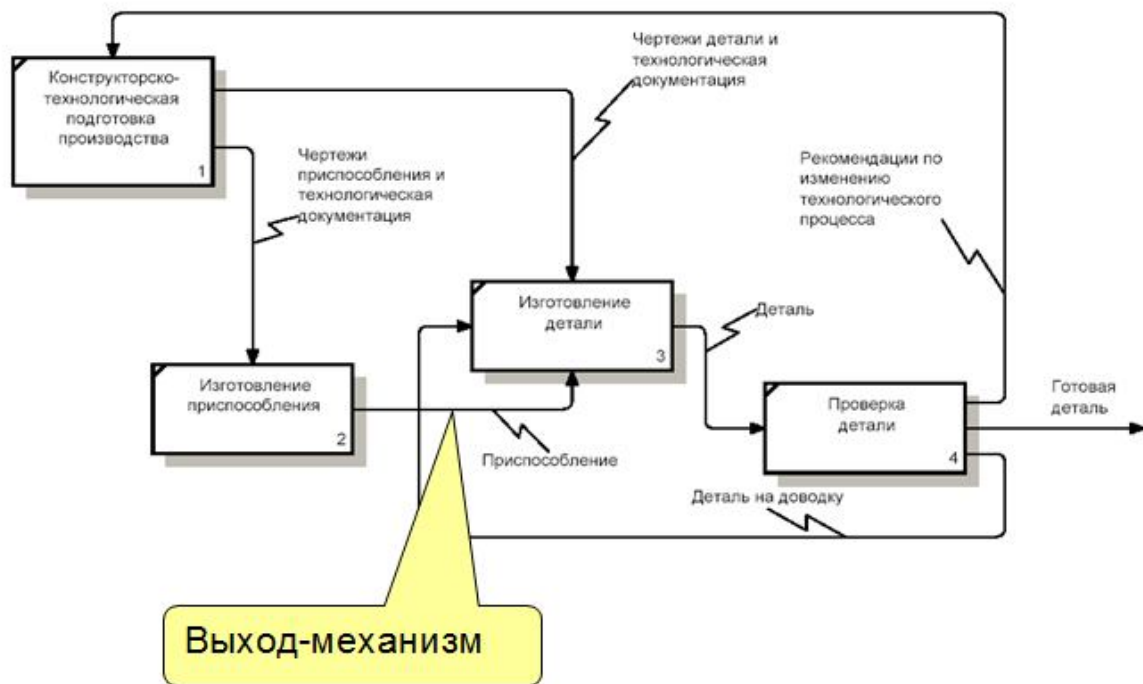


Рис. 17

Обратная связь по входу перенаправляет выход последующей функции на вход предыдущей. Используется для описания возможности повторной обработки потока объектов или для описания циклических действий над потоком (Рис. 18).

Обратная связь по управлению имеет место, когда выход последующей функции направляется на управление одной из предыдущих функций. Например, она может описывать обратную связь, регулирующую производственные процессы, что может повлиять на качество выпускаемого изделия (Рис. 19).



Рис. 18

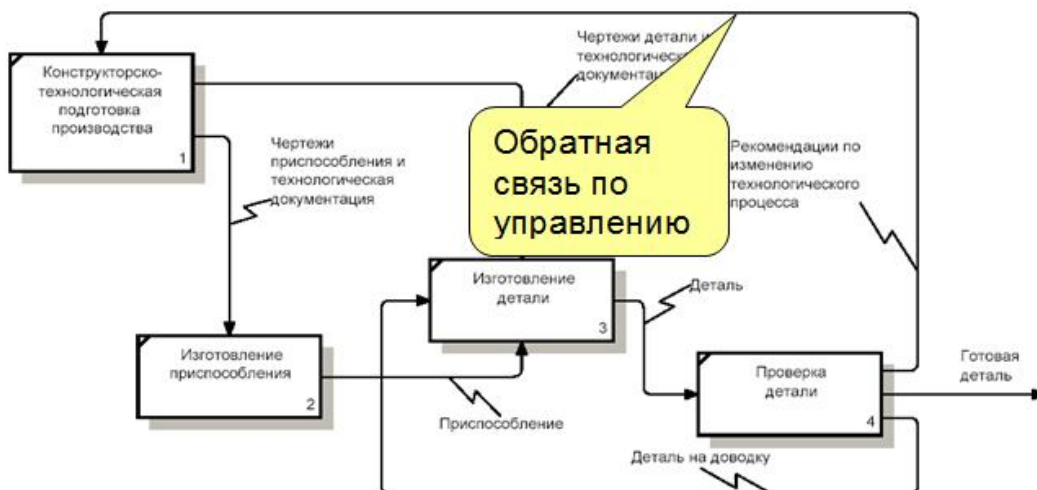


Рис. 19

Имена и метки

Имена функций – глаголы или глагольные обороты (Рис. 20).

Примеры таких имен :

производить детали, планировать ресурсы, наблюдать за выполнением, проектировать систему, эксплуатировать, разработать детальные чертежи, изготовить компонент, проверять деталь

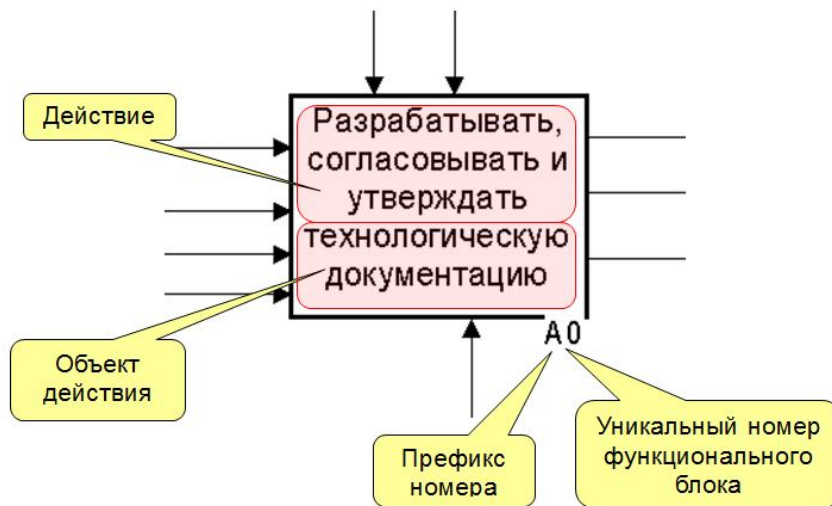


Рис. 20

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

спецификации, отчет об испытаниях, бюджет, конструкторские требования, конструкция детали, директива, инженер-конструктор.

Пример размещения меток стрелок и имени блока показан на Рис. 21.

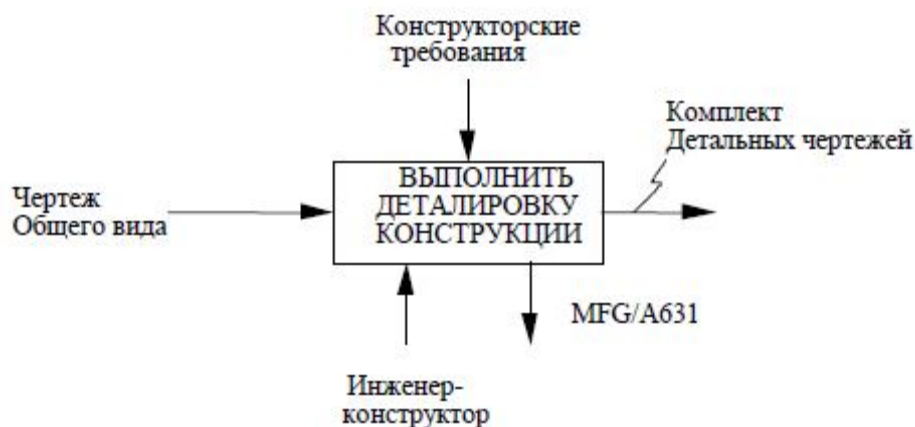


Рис. 21

Расположение блоков на диаграмме имеет значение. На рисунке (Рис. 23) представлено изображение где блок А1 доминирует над блоком А2, блок А2 доминирует над блоком А3.

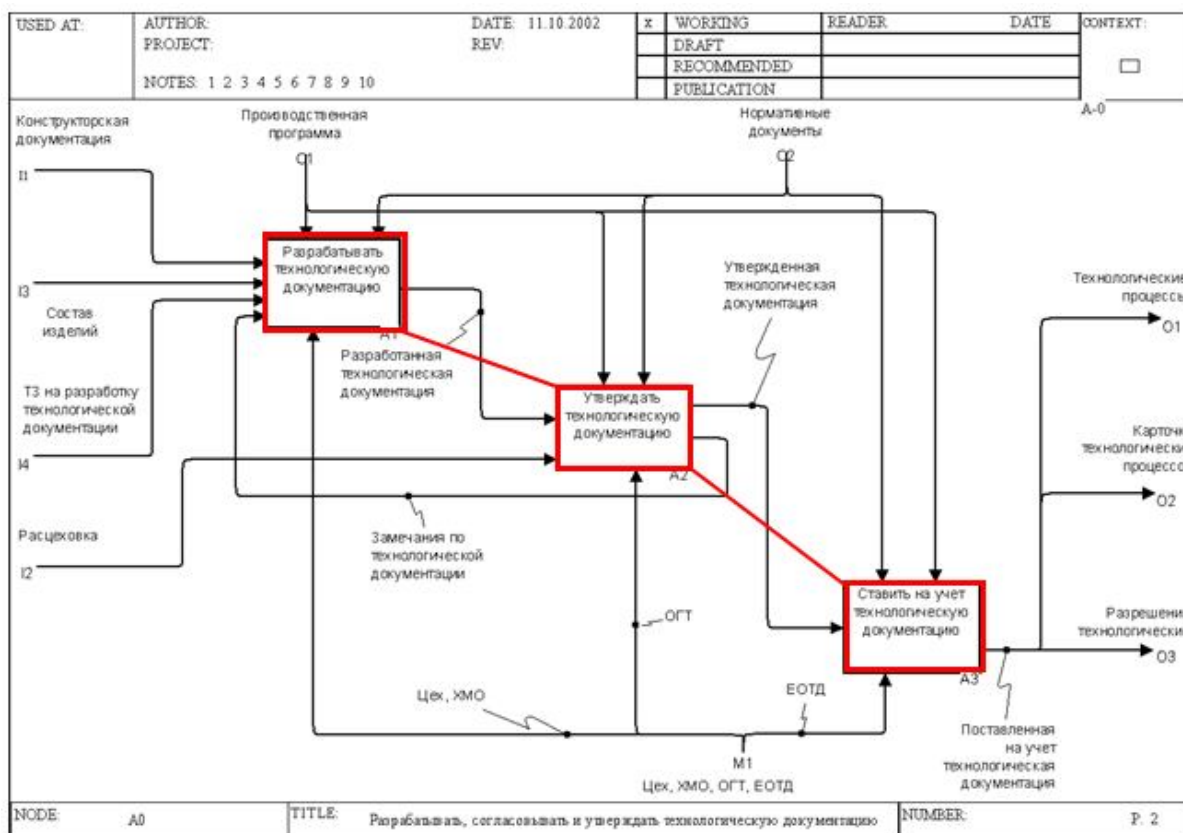


Рис. 23

Диаграммы IDEF0

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 ассоциированные с ними отношения. Блоки представляют основные функции моделируемого объекта. Эти функции могут быть разбиты (декомпозированы) на составные части и представлены в виде более подробных диаграмм; процесс декомпозиции продолжается до тех пор, пока объект не будет описан на уровне детализации, необходимом для достижения целей конкретного проекта. Диаграмма верхнего уровня обеспечивает наиболее общее или абстрактное описание объекта

моделирования. За этой диаграммой следует серия дочерних диаграмм, дающих более детальное представление об объекте.

Контекстная диаграмма верхнего уровня

Каждая модель должна иметь контекстную диаграмму верхнего уровня, на которой объект моделирования представлен единственным блоком с граничными стрелками. Эта диаграмма называется А-0 (А минус нуль). Стрелки на этой диаграмме отображают связи объекта моделирования с окружающей средой. Поскольку единственный блок представляет весь объект, его имя – общее для всего проекта. Это же справедливо и для всех стрелок диаграммы, поскольку они представляют полный комплект внешних интерфейсов объекта. Диаграмма А-0 устанавливает область моделирования и ее границу. Пример диаграммы А-0 показан на рисунке Рис. 24.

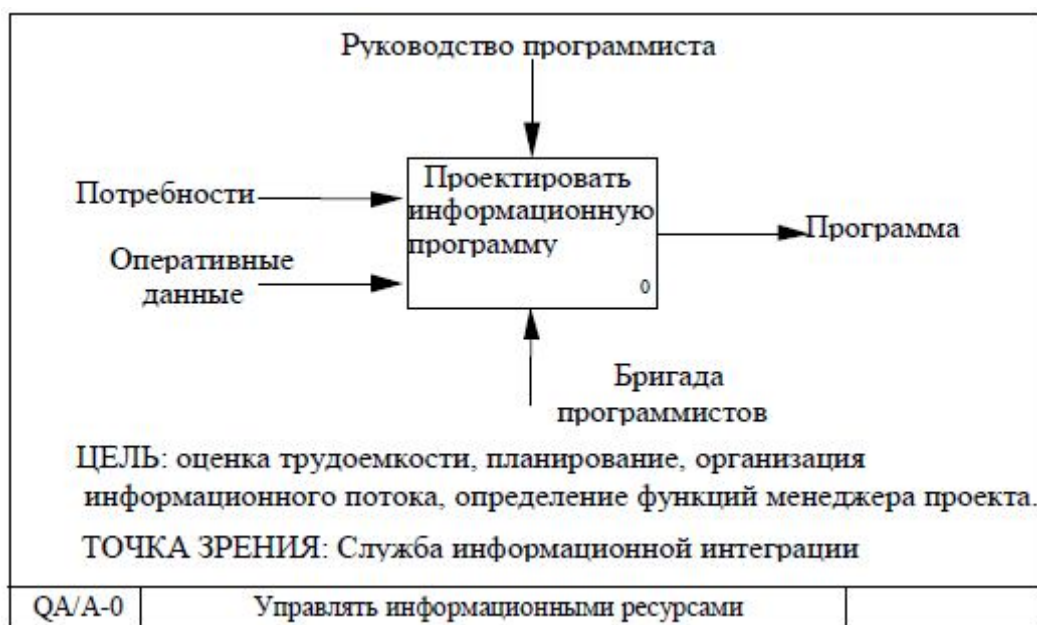


Рис. 24

Контекстная диаграмма А-0 также должна содержать краткие утверждения, определяющие точку зрения должностного лица или подразделения, с позиций которого создается модель, и цель, для достижения которой ее разрабатывают. Эти утверждения помогают руководить разработкой модели и ввести этот процесс в определенные рамки. Точка зрения определяет, что и в каком разрезе можно увидеть в пределах контекста модели. Изменение точки зрения,

приводит к рассмотрению других аспектов объекта. Аспекты, важные с одной точки зрения, могут не появиться в модели, разрабатываемой с другой точки зрения на тот же самый объект.

Формулировка цели выражает причину создания модели, 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, в свою очередь, декомпозируется на элементы следующего уровня, и так происходит до тех пор, пока не будет получена релевантная структура, позволяющая ответить на вопросы, сформулированные в цели моделирования. Каждая подфункция моделируется отдельным блоком. Каждый родительский блок подробно описывается дочерней диаграммой на более низком уровне. Все дочерние диаграммы должны быть в пределах области контекстной диаграммы верхнего уровня.

Дочерняя диаграмма

Единственная функция, представленная на контекстной диаграмме верхнего уровня, может быть разложена на основные подфункции посредством создания дочерней диаграммы. В свою очередь, каждая из этих подфункций может быть разложена на составные части посредством создания дочерней диаграммы следующего, более низкого уровня, на которой некоторые или все функции также могут быть разложены на составные части.

Каждая дочерняя диаграмма содержит дочерние блоки и стрелки, обеспечивающие дополнительную детализацию родительского блока.

Дочерняя диаграмма, создаваемая при декомпозиции, охватывает ту же область, что и родительский блок, но описывает ее более подробно. Таким образом, дочерняя диаграмма как бы вложена в свой родительский блок. Эта структура иллюстрируется на рисунке (Рис. 25).

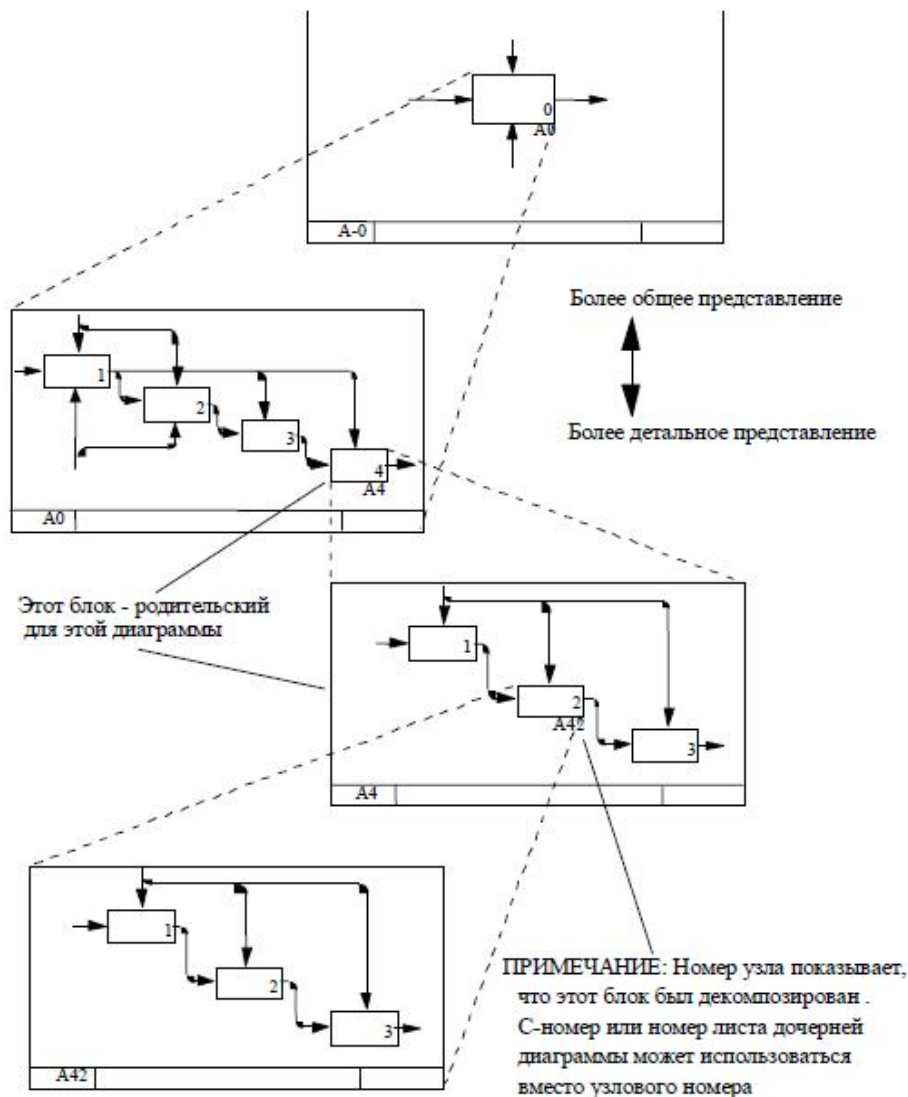


Рис. 25

Родительская диаграмма

Родительская диаграмма – это диаграмма, которая содержит один или более родительских блоков. Каждая обычная (не-контекстная) диаграмма является также дочерней диаграммой, поскольку, по определению, она подробно описывает некоторый родительский блок. Таким образом, любая диаграмма может быть как родительской диаграммой (содержать родительские блоки), так и дочерней (подробно описывать собственный родительский блок). Аналогично, блок может быть как родительским (подробно описываться дочерней диаграммой) так и дочерним (появляющимся на дочерней

диаграмме). Основное иерархическое отношение существует между родительским блоком и дочерней диаграммой.

То, что блок является дочерним и раскрывает содержание родительского блока на диаграмме предшествующего уровня, указывается специальным ссылочным кодом, написанным ниже правого нижнего угла блока. Этот ссылочный код может формироваться несколькими способами, из которых самый простой заключается в том, что код, начинающийся с буквы А (по имени диаграммы А-0), содержит цифры, определяемые номерами родительских блоков.

На диаграмме все функциональные блоки должны быть пронумерованы. Номер состоит из префикса и одной или нескольких цифр. Обычно используется префикс «А», но допустимо использовать префикс любой длины. Контекстная функция всегда именуется А0. Функция А0 декомпозируется в функции А1, А2, А3 и т.д.. Функция А2 декомпозируется в функции А21, А22, А23 и т.д. Каждый уровень декомпозиции добавляет один разряд в номер функционального блока.

Например, показанные на рисунке Рис. 26 коды означают, что диаграмма является декомпозицией 1-го блока диаграммы, которая, в свою очередь является декомпозицией 6-го блока диаграммы А0, а сами коды образуются присоединением номера блока (Рис. 27).

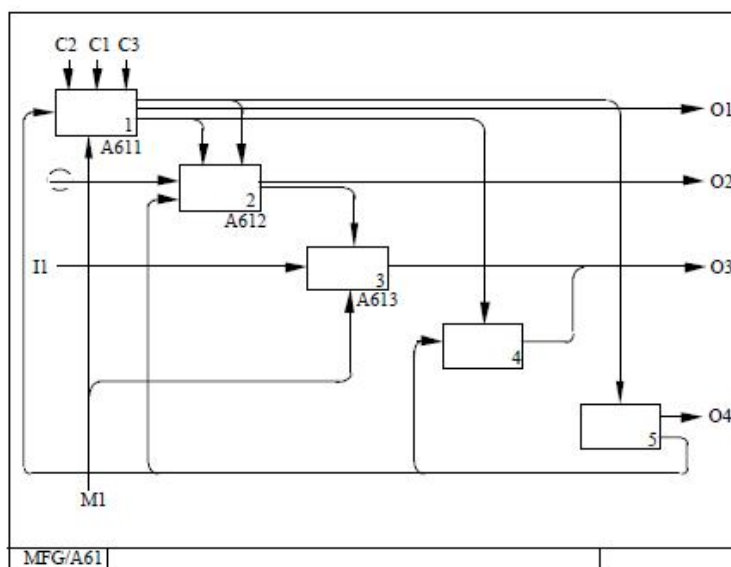


Рис. 26

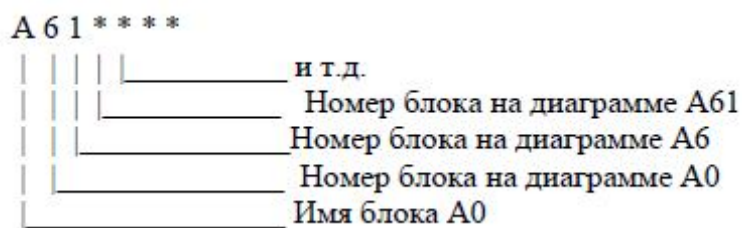


Рис. 27

Текст и глоссарий

Диаграмме может быть поставлен в соответствие структурированный текст, представляющий собой краткий комментарий к содержанию диаграммы. Текст используется для объяснений и уточнений характеристик, потоков, внутриблочных соединений и т.д. Текст не должен использоваться для описания и без того понятных блоков и стрелок на диаграммах.

Глоссарий предназначен для определения аббревиатур (акронимов), ключевых слов и фраз, используемых в качестве имен и меток на диаграммах. Глоссарий определяет понятия и термины, которые должны быть одинаково понимаемы всеми участниками разработки и пользователями модели, чтобы правильно интерпретировать ее содержание.

5. СВОЙСТВА ДИАГРАММ

Стрелки как ограничения

Стрелки на диаграмме IDEF0, представляя данные или материальные объекты, одновременно задают своего рода ограничения (условия). Входные и управляющие стрелки блока, соединяющие его с другими блоками или с внешней средой, по сути описывают условия, которые должны быть выполнены для того, чтобы реализовалась функция, записанная в качестве имени блока.

Рисунок (Рис. 28) иллюстрирует случай, при котором "функция 3" может быть выполнена только после получения данных, выработанных "функцией 1" и "функцией 2".

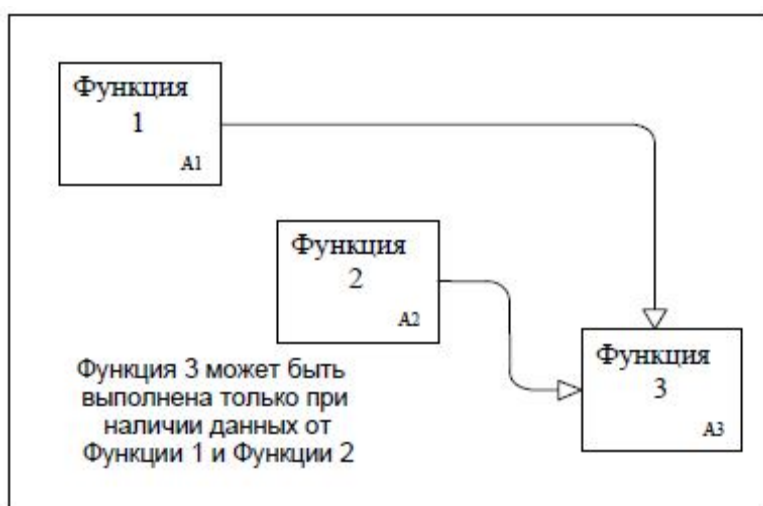


Рис. 28

Параллельное функционирование

Различные функции в модели могут быть выполнены параллельно, если удовлетворяются необходимые ограничения (условия). Один блок может управлять выполнением другой функции (Рис. 29 а) или создать данные, материальные объекты, необходимые для параллельной работы нескольких блоков (Рис. 29 б).

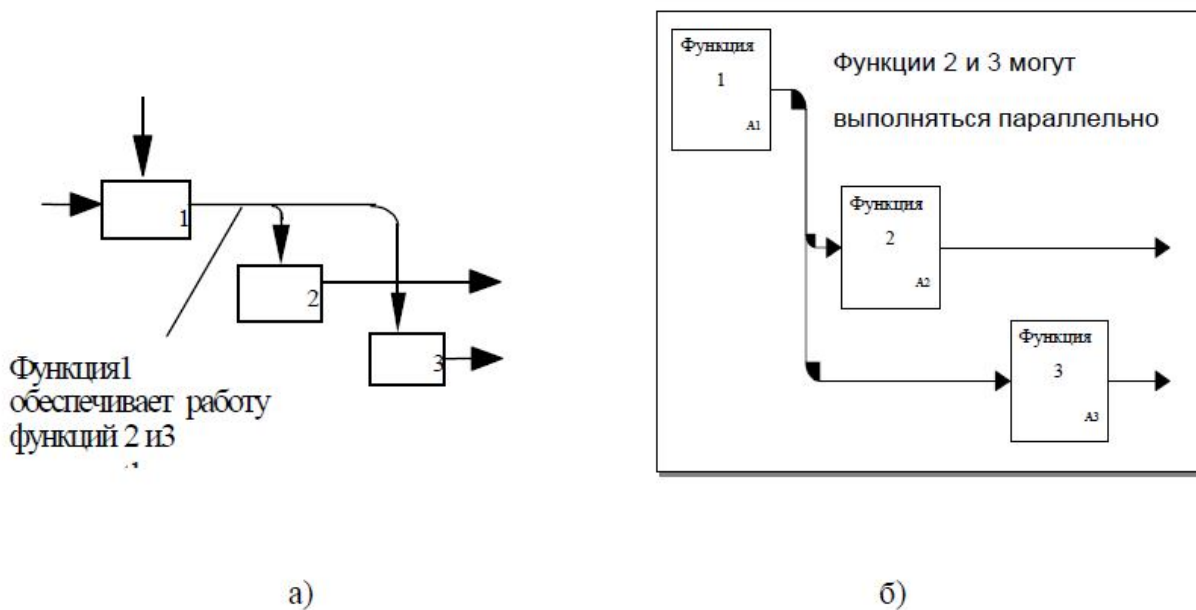


Рис. 29

Ветвление и слияние сегментов стрелок

Ветвление и слияние стрелок призвано уменьшить загруженность диаграмм графическими элементами (линиями). Чтобы стрелки и их сегменты правильно описывали связи между блоками - источниками и блоками - потребителями, используется аппарат меток. Метки связываются с сегментами посредством тильд. При этом между сегментами возникают определенные отношения, описанные ниже:

- непомеченные сегменты (Рис. 30) содержат все объекты, указанные в метке стрелки перед ветвлением (т.е. все объекты принадлежат каждому из сегментов);

- сегменты, помеченные после точки ветвления (Рис. 31), содержат все объекты, указанные в метке стрелки перед ветвлением, или их часть, описываемую меткой каждого конкретного сегмента;

- при слиянии непомеченных сегментов объединенный сегмент стрелки содержит все объекты, принадлежащие сливаемым сегментам и указанные в общей метке стрелки после слияния (Рис. 32);

- при слиянии помеченных сегментов (Рис. 33) объединенный сегмент содержит все или некоторые объекты, принадлежащие сливаемым сегментам и

перечисленные в общей метке после слияния; если общая метка после слияния отсутствует, это означает, что общий сегмент передает все объекты, принадлежащие сливаемым сегментам.

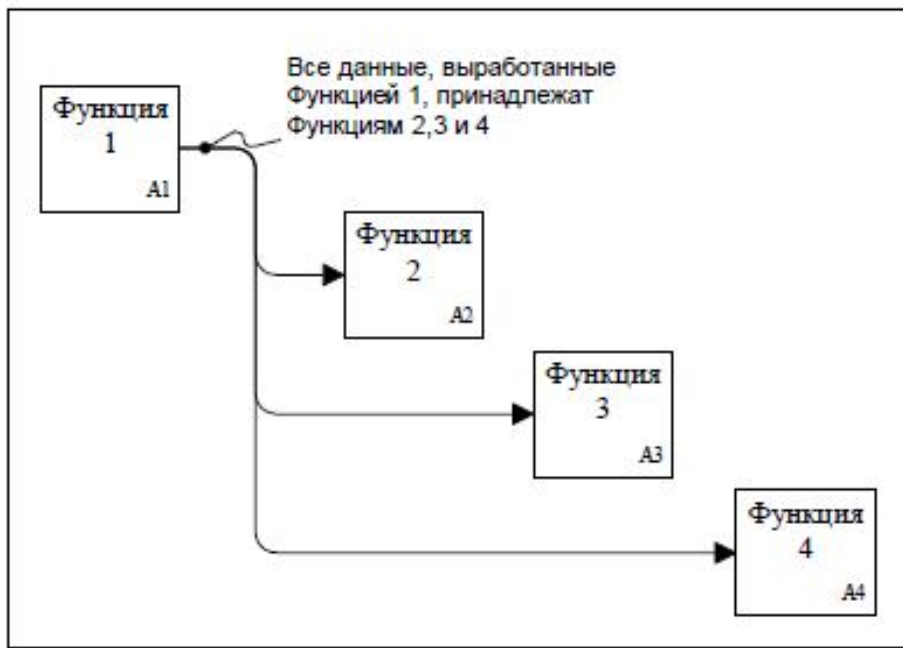


Рис. 30

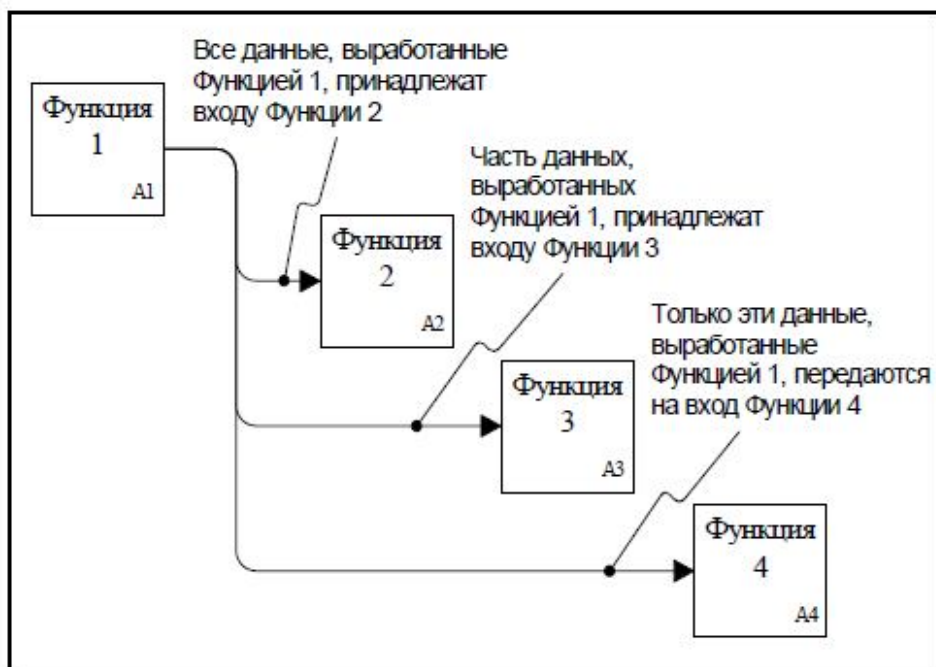


Рис. 31



Рис. 32

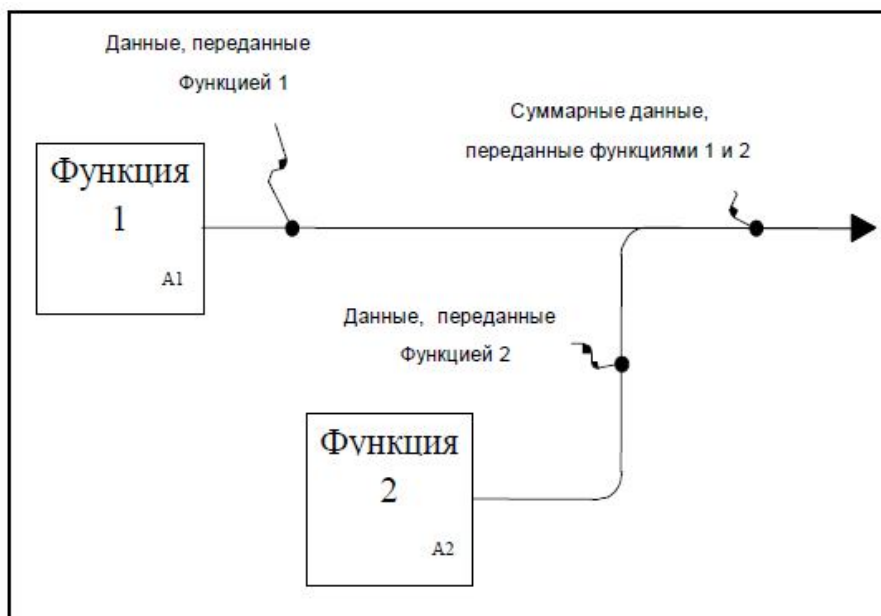


Рис. 33

Пример слияния стрелок

Функция производит объекты, которые используются в нескольких других функциях. Объекты, полученные в результате работы нескольких функций, объединяются в один общий поток (Рис. 34)

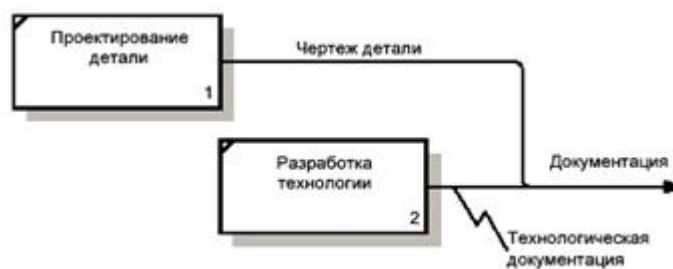


Рис. 34

Случай, когда какой либо однотипный результат, получается в итоге взаимодействия двух различных функций. Достаточно отметить только общую часть стрелки (Рис. 35).

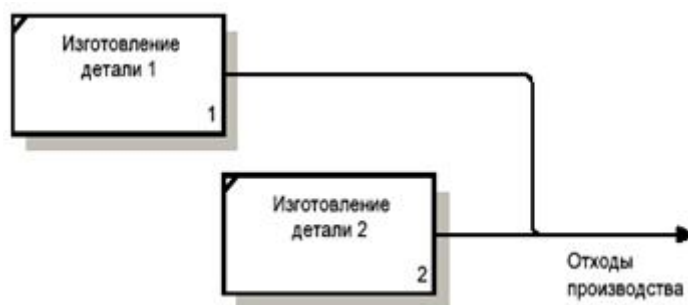


Рис. 35

Пример разветвления стрелок

Если общий поток, разветвляясь, сохраняет первоначальное содержание, то подпись необходима только для общей части стрелки (Рис. 36).

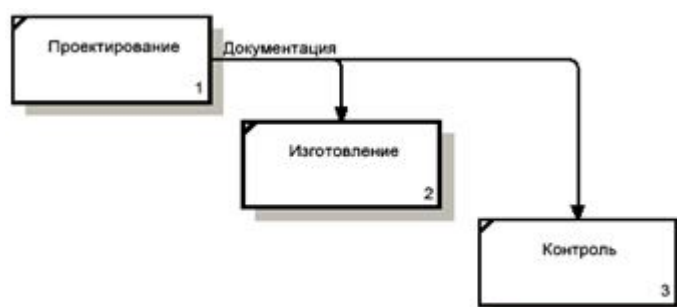


Рис. 36

Поток ответвляется от общего потока, неся в себе часть объектов (чертежи). Подписываются общая стрелка и ответвления. Если ответвление не подписано, то оно несет в себе общий поток объектов (Рис. 37).

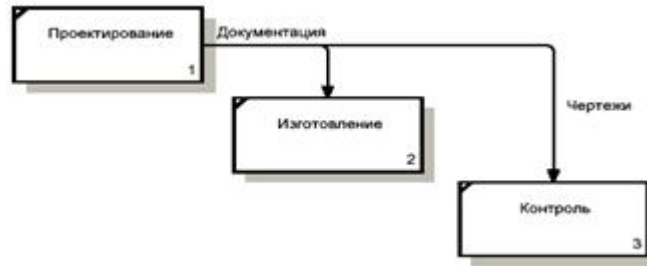


Рис. 37

Если общий поток разделяется на несколько независимых потоков, то необходимо обозначать как общую часть стрелки, так и каждое ответвление (Рис. 38).



Рис. 38

При разветвлении общая часть стрелки обязательно должна быть именованной. Именование стрелок-ответвлений зависит от конкретных условий, описанных выше. На рисунке показана ошибочная ситуация – неименованная общая стрелка.

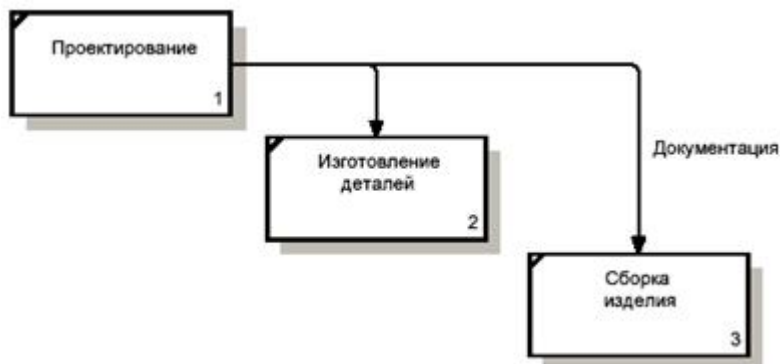


Рис. 39

Отношения блоков на диаграммах

В методологии IDEF0 существует шесть типов отношений между блоками в пределах одной диаграммы:

- доминирование;
- управление;
- выход - вход;
- обратная связь по управлению;
- обратная связь по входу;
- выход – механизм.

Первое из перечисленных отношений определяется взаимным расположением блоков на диаграмме. Предполагается, что блоки, расположенные на диаграмме выше и левее, «доминируют» над блоками, расположенными ниже и правее. «Доминирование» понимается как влияние, которое один блок оказывает на другие блоки диаграммы.

Остальные пять отношений описывают связи между блоками и изображаются соответствующими стрелками.

Отношения управления и выход – вход являются простейшими, поскольку отражают прямые взаимодействия, которые понятны и очевидны.

Отношение управления (Рис. 40) возникает тогда, когда выход одного блока служит управляющим воздействием на блок с меньшим доминированием.

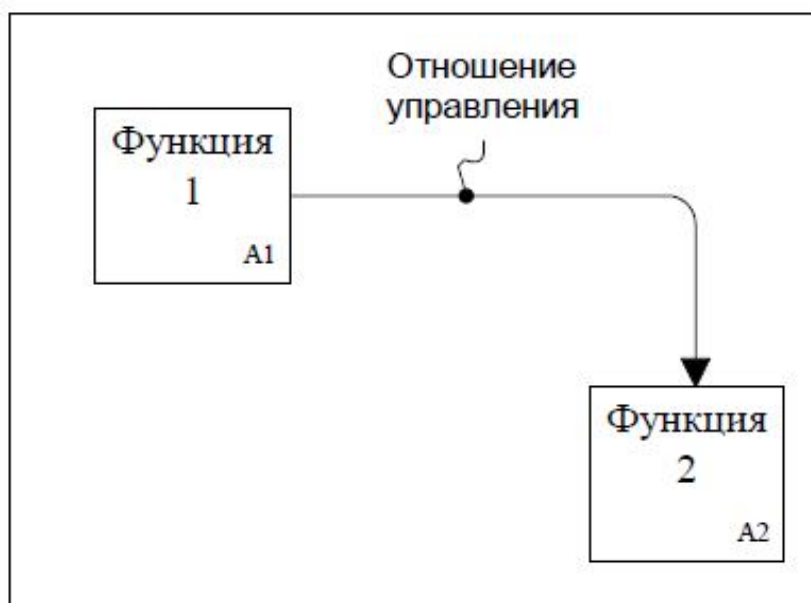


Рис. 40

Отношение выход – вход (Рис. 41 а) возникает при соединении выхода одного блока с входом другого блока с меньшим доминированием.

Обратная связь по управлению и обратная связь по входу являются более сложными типами отношений, поскольку они представляют итерацию (выход функции влияет на будущее выполнение других функций с большим доминированием, что впоследствии влияет на исходную функцию).

Обратная связь по управлению (Рис. 41 б) возникает тогда, когда выход некоторого блока создает управляющее воздействие на блок с большим доминированием.

Отношение обратной связи по входу (Рис. 41 в) имеет место тогда, когда выход блока становится входом другого блока с большим доминированием.

Связи «выход – механизм» (Рис. 41 г) отражают ситуацию, при которой выход одной функции становится средством достижения цели для другой.

Связи «выход – механизм» возникают при отображении в модели процедур пополнения и распределения ресурсов, создания или подготовки средств для выполнения функций системы (например, приобретение или изготовление требуемых инструментов и оборудования, обучение персонала, организация физического пространства, финансирование, закупка материалов и т.д.

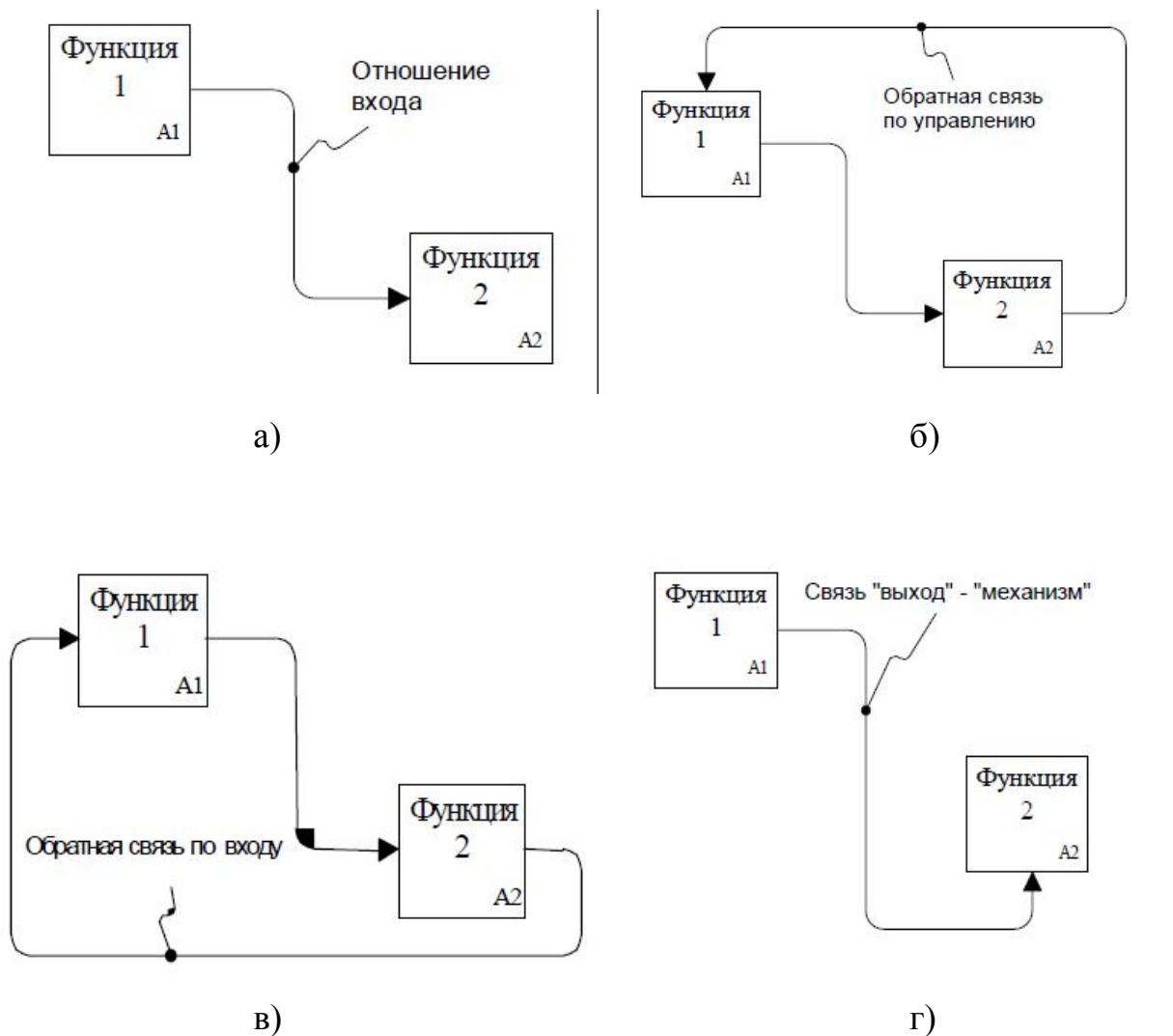


Рис. 41

6. ОТНОШЕНИЯ МЕЖДУ БЛОКАМИ ДИАГРАММЫ И ДРУГИМИ ДИАГРАММАМИ

Все описанные выше отношения отображаются внутренними стрелками, т.е. такими, у которых оба конца связаны с блоками диаграммы. Отношения между блоками диаграммы и другими диаграммами, являющимися по отношению к рассматриваемой диаграмме окружающей средой (окружением), описываются граничными стрелками. Обе ситуации отражены на Рис. 42.

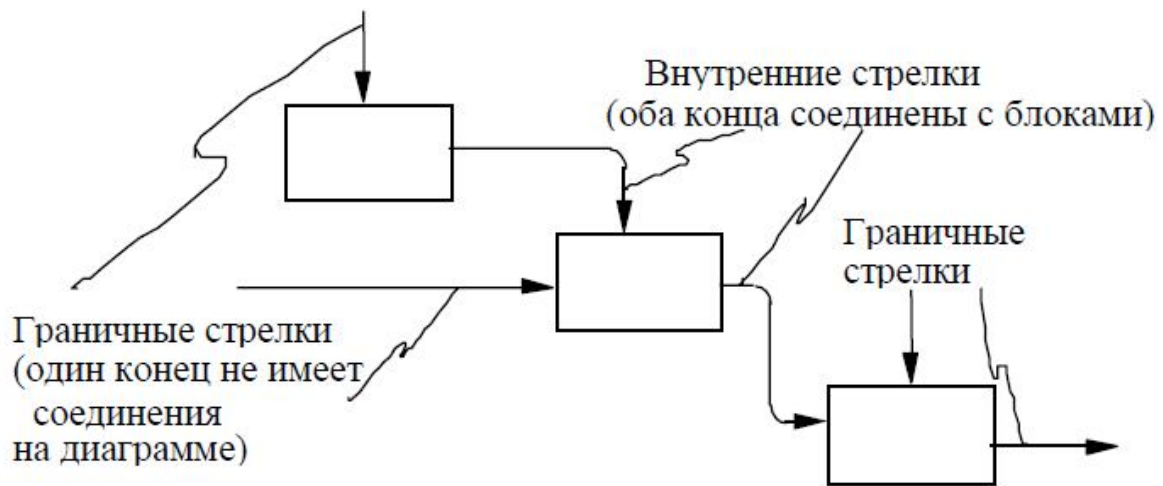


Рис. 42

Граничные стрелки

На обычной (не контекстной) диаграмме граничные стрелки представляют входы, управления, выходы или механизмы родительского блока диаграммы. Источник или потребитель граничных стрелок можно обнаружить, только изучая родительскую диаграмму. Все граничные стрелки на дочерней диаграмме (за исключением стрелок, помещенных в тоннель должны соответствовать стрелкам родительского блока, как показано на Рис. 43.

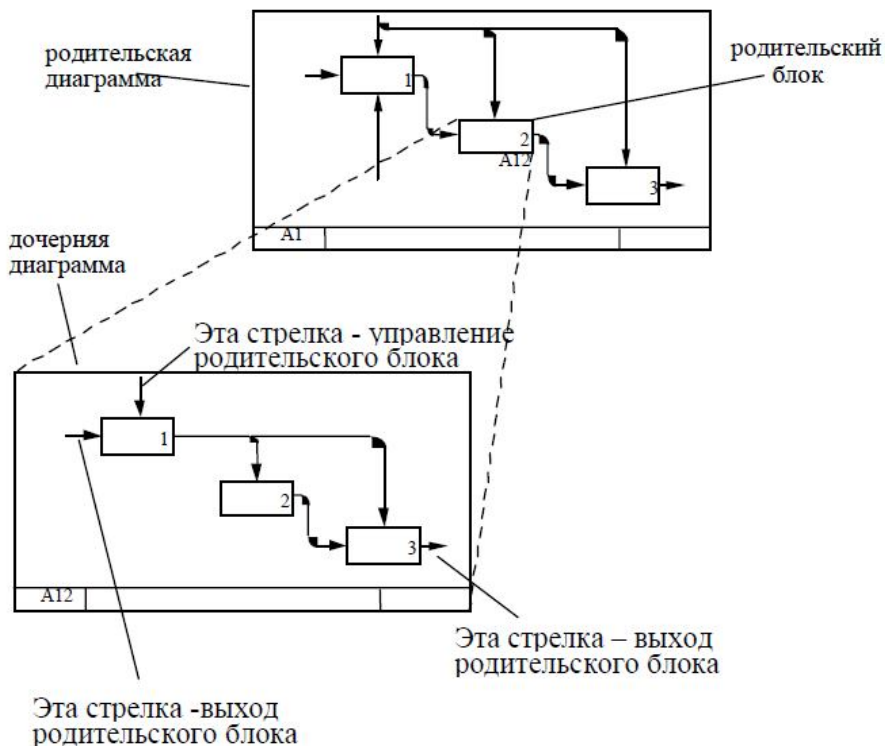


Рис. 43

ICOM - кодирование граничных стрелок

ICOM - коды связывают граничные стрелки на дочерней диаграмме со стрелками родительского блока. Нотация, названная ICOM - кодом, определяет значения соединений. Буквы I, C, O или M, написанные около несвязанного конца граничной стрелки на дочерней диаграмме идентифицируют стрелку как Вход (Input), Управление (Control), Выход (Output) или Механизм (Mechanism) в родительском блоке. Буква следует за числом, определяющим относительное положение точки подключения стрелки к родительскому блоку; это положение определяется слева направо или сверху вниз. Например, код "С3", написанный возле граничной стрелки на дочерней диаграмме, указывает, что эта стрелка соответствует третьей (считая слева) управляющей стрелке родительского блока.

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

Иногда буквенные ICOM - коды, определяющие роли граничных стрелок (вход, управление, механизм), могут меняться при переходе от родительского блока к дочерней диаграмме. Например, управляющая стрелка в родительском блоке может быть входом на дочерней диаграмме. Аналогично, вход родительского блока может быть управлением для одного или более дочерних блоков. Примеры изменения ролей стрелок можно видеть на рисунке (Рис. 44).

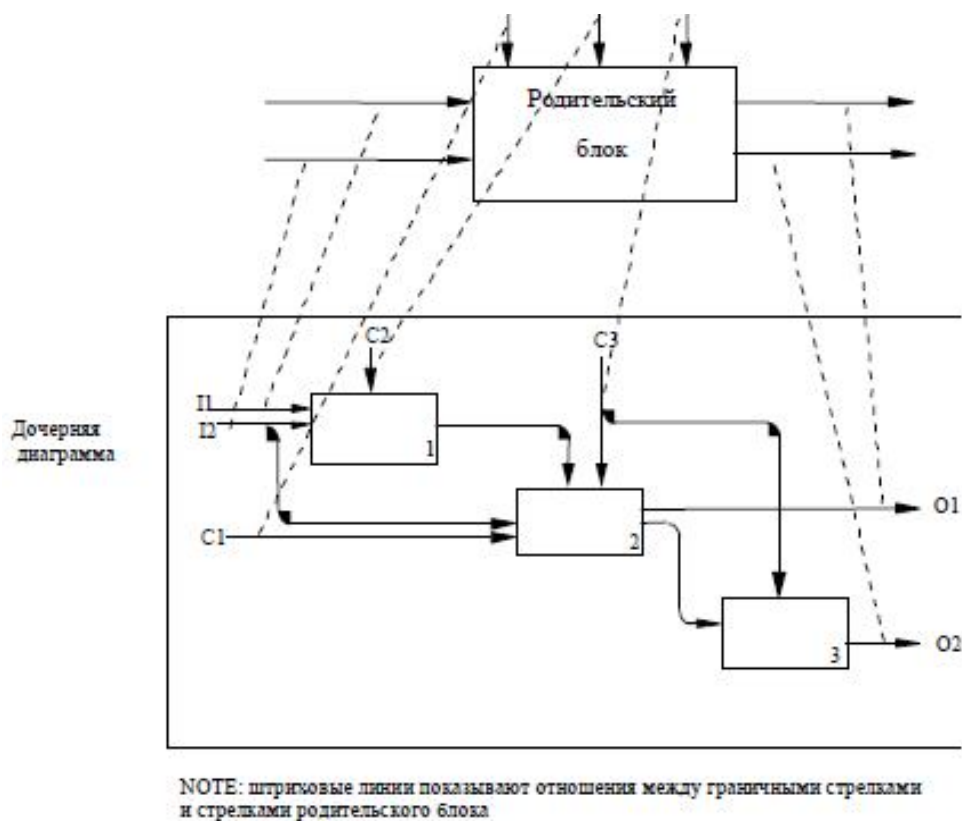


Рис. 44

Стрелки, помещенные в «туннель»

Туннели являются важной частью диаграмм. Применяются когда хотят чтобы стрелка используемая только начиная с какого либо уровня не присутствовала на всех промежуточных уровнях декомпозиции. Это помогает освободить промежуточные диаграммы от неиспользуемых стрелок или необходимо скрыть граничную стрелку на диаграмме декомпозиции.

Туннель - круглые скобки в начале и/или окончании стрелки. Туннельные стрелки означают, что данные, выраженные этими стрелками, не рассматриваются на родительской диаграмме и/или на дочерней диаграмме.

Стрелка, помещенная в туннель там, где она присоединяется к блоку (Рис. 45 а), означает, что данные, выраженные этой стрелкой, не обязательны на следующем уровне декомпозиции.

Стрелка, помещаемая в туннель на свободном конце (Рис. 45 б) означает, что выраженные ею данные отсутствуют на родительской диаграмме.

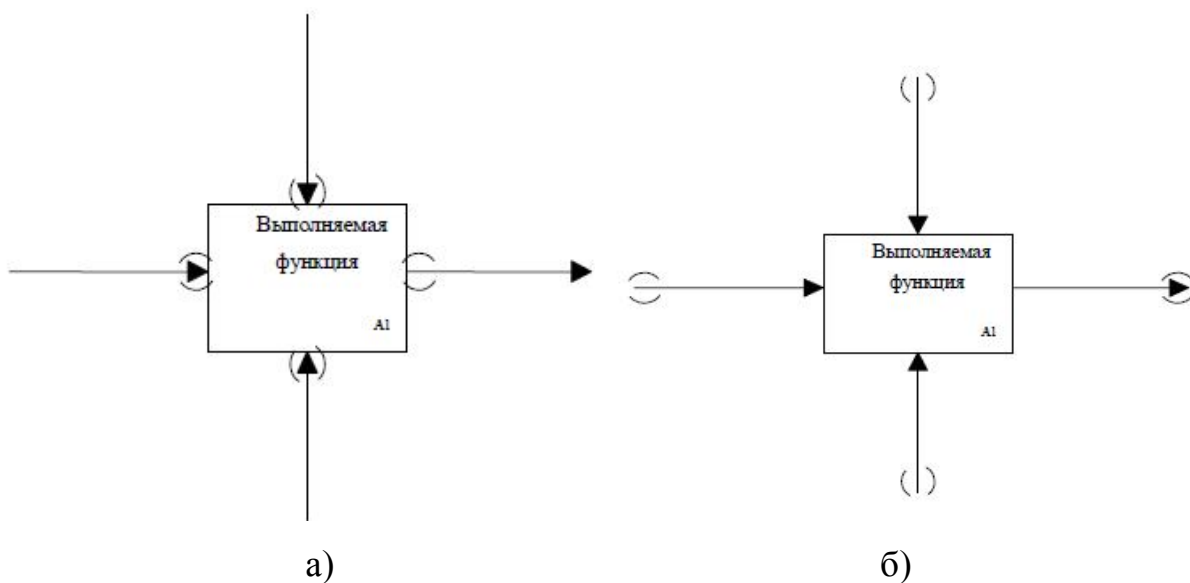


Рис. 45

7. ССЫЛОЧНЫЕ ВЫРАЖЕНИЯ (КОДЫ)

Ссылочные выражения (коды) присваиваются всем элементам модели: диаграммам, блокам, стрелкам и примечаниям. Ссылочные выражения затем могут использоваться в различных контекстах для точного указания на нужный элемент модели.

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

Номера блоков

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

На контекстной диаграмме А-0 единственному блоку присваивается номер 0 (ноль). На всех других диаграммах блоки нумеруются цифрами от 1 до 6,

начиная с верхнего левого блока (при их диагональном размещении) и кончая нижним правым блоком. Если некоторые блоки на диаграмме размещены не по диагонали, то сначала нумеруются «диагональные» блоки (также начиная с левого верхнего блока), а затем – «недиагональные» блоки, начиная с нижнего правого против часовой стрелки.

Узловые номера

Узловой номер базируется на положении блока в иерархии модели. Обычно узловой номер формируется добавлением номера блока к номеру диаграммы, на которой он появляется. Например, узловой номер блока 2 на диаграмме A25 - A252. Все узловые номера IDEF0 начинаются с заглавной буквы, например, "А". Когда родительский блок подробно описывается дочерней диаграммой, узловые номера родительского блока и дочерней диаграммы совпадают.

Контекстные диаграммы и дочерняя диаграмма верхнего уровня - исключения в вышеуказанной схеме узловой нумерации. Каждая модель IDEF0 имеет контекстную диаграмму верхнего уровня - диаграмму А-0. Эта диаграмма содержит единственный "высший блок", который является уникальным родителем всей модели и несет уникальный номер 0 (нуль) и узловой номер А0. Каждая модель IDEF0 должна также иметь по крайней мере одну дочернюю диаграмму, содержащую декомпозицию блока А0 на 3 ... 6 дочерних блоков. Этим блокам присваиваются уникальные узловые номера А1, А2, А3, ... А6. Таким образом, последовательность [А0, А1, ..., А2, ..., А3, ...] начинает нумерацию узлов для любой модели.

Например, модель может иметь следующие узловые номера:

А-1	Дополнительная контекстная диаграмма
А-0	Обязательная контекстная диаграмма верхнего уровня (содержащая высший блок А0)
А0	Верхняя дочерняя диаграмма

A1, A2, ..., A5	Дочерние диаграммы
A11, A12, ..., A15, ..., A51, ... , A55	Дочерние диаграммы
A111, A112, ..., A151, ..., A511, ..., A555	Дочерние диаграммы
...	Дочерние диаграммы нижнего уровня

Узловой номер используется также для обозначения того, что блок подвергнут декомпозиции. В этом случае узловой номер, совпадающий с номером дочерней диаграммы, помещается под правым нижним углом блока на родительской диаграмме.

Перечень узлов

Перечень узлов представляет информацию о входящих в модель узлах в форме списка, напоминающего обычное оглавление и отражающего иерархическую структуру модели.

A0 Производить продукт

A1 Планировать производство

A11 Выбрать технологию производства

A12 Оценить требуемое время и затраты на производство

A13 Разработать производственные планы

A14 Разработать план вспомогательных действий

A2 Разрабатывать и управлять графиком выпуска и ресурсами

A21 Разработать основной график

A22 Разработать график координации работ

A23 Оценивать затраты и приобретать ресурсы

A24 Следить за выполнением графика и расходом ресурсов

A3 Планировать выпуск продукции

Дерево узлов

Разработанная модель IDEF0 со всеми уровнями структурной декомпозицией может быть представлена на единственной диаграмме в виде дерева узлов, дополняющего перечень узлов. Для изображения этого дерева нет стандартного формата. Единственное требование состоит в том, что вся иерархия узлов модели должна быть представлена наглядно и понятно. Пример дерева узлов показан на рисунке Рис. 46.

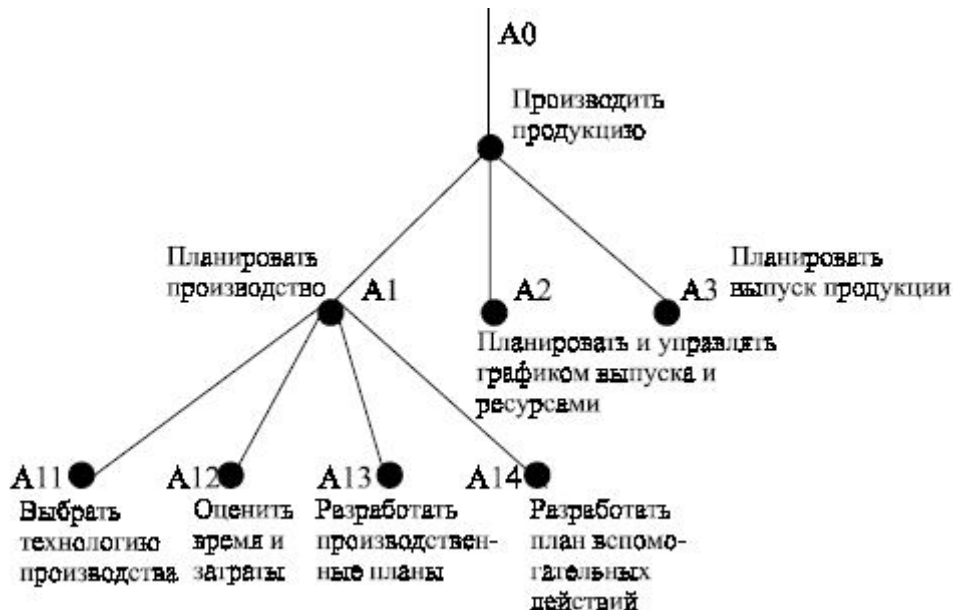


Рис. 46

8. МЕТОДИКА РАЗРАБОТКИ ФУНКЦИОНАЛЬНЫХ МОДЕЛЕЙ IDEF 0

В предыдущих разделах описаны инструментальные возможности методологии IDEF0 как средства функционального моделирования производственно-технических и организационно-экономических систем. В настоящем разделе кратко излагаются некоторые методические приемы построения моделей, облегчающие практическое применение этой методологии.

Общие положения

Как уже отмечалось во Введении, объектами функционального моделирования и структурного анализа по методологии IDEF0 являются организационно-экономические и производственно-технические системы. Согласно основным положениям системного анализа и системотехники

системой называется совокупность взаимодействующих объектов любой, в том числе различной, физической природы, обладающая выраженным системным свойством (свойствами), т.е. свойством, которого не имеет ни одна из частей системы при любом способе членения, и не выводимым из свойств частей. Части системы, обладающие собственными системными свойствами, называются подсистемами. Объединение нескольких систем, обладающее системным свойством, называют надсистемой или системой более высокого (2-го, 3-ьего и т.д.) порядка. Элементом системы является объект с однозначно определенными известными свойствами, вытекающими из физических или экономических законов.

Система (подсистема, элемент) имеют входы и выходы. Входом называется дискретное или непрерывное множество «контактов», через которое воздействие среды передается системе. Выход – множество «контактов», через которое система воздействует на среду. Любой элемент системы имеет по крайней мере один вход и один выход. Воздействие может состоять в передаче вещества, энергии, информации или комбинации этих сущностей.

Приведенные определения корреспондируются с определением функционального блока IDEF0 с той лишь разницей, что в методологии входные контакты подразделяются на собственно входы и управления.

Функциональный блок, как отображающий моделируемую систему в целом (блок A0), так и блок на любом уровне декомпозиции являются преобразующими блоками. Преобразующий блок – блок IDEF0 – диаграммы, преобразующий входы в выходы под действием управлений при помощи «механизмов». Преобразование – цель и результат работы любого блока на диаграмме любого уровня декомпозиции.

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

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

Информационный поток – множество информационных объектов, распределенное во времени.

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

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

Ограничительная информация - сведения о том, чего нельзя делать:

а) никогда, ни при каких обстоятельствах (кроме, быть может, форсмажорных) в любой фазе и на любом этапе функционирования системы в целом;

б) в рамках функционирования конкретного блока.

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

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

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

Схематическое изображение связей преобразующего блока в соответствии с соглашениями системы IDEF0 показано на рисунке Рис. 47. Ограничительная и предписывающая информация изображается стрелками, присоединяемыми к

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

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



Рис. 47

Классификация функций, моделируемых блоками IDEF0

Единообразное представление явлений и событий реального мира, происходящих в моделируемых системах, в виде функциональных блоков является большим преимуществом графического языка IDEF0. Вместе с тем, практика построения моделей требует введения классификации явлений и событий с целью облегчения построения и интерпретации (понимания) функциональных моделей. Такая классификация облегчает выбор глубины декомпозиции моделируемых систем и способствует выработке единообразных подходов и приемов моделирования в конкретных предметных областях.

Классификация делит все функции таких систем на четыре основных и два дополнительных вида. Каждая рубрика в классификации представляет собой класс преобразующих блоков, экземпляры которого возникают и используются при моделировании конкретной системы

Основные виды функций.

1. **Деятельность** (синонимы: дело, бизнес) – совокупность процессов, выполняемых (протекающих) последовательно или/и параллельно,

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

В модели IDEF0 деятельность описывается блоком A0 на основной контекстной диаграмме A-0.

При моделировании крупных, многопрофильных структур (фирм, организаций, предприятий), которые по своему статусу занимаются различными видами деятельности, последние представляют собой различные экземпляры класса «деятельность» и могут найти отражение в дополнительной контекстной диаграмме A-1. В этом случае общая модель такой сложной структуры будет состоять из ряда частных моделей, каждая из которых относится к конкретному виду деятельности. Связь между этими частными моделями представляет отдельную методическую проблему, которая в рамках настоящего РД не рассматривается.

2. Процесс (синоним: бизнес-процесс) – совокупность последовательно или/и параллельно выполняемых операций, преобразующая материальный или/и информационный потоки в соответствующие потоки с другими свойствами.

Процесс протекает в соответствии с управляющими директивами, вырабатываемыми на основе целей деятельности. В ходе процесса потребляются финансовые, энергетические, трудовые и материальные ресурсы и выполняются ограничения со стороны других процессов и внешней среды.

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

Операция выполняется :

а) в соответствии с директивами, вырабатываемыми на основе директив, определяющих протекание процесса, в состав которого входит операция;

б) с потреблением всех видов потребных ресурсов;

в) с соблюдением ограничений со стороны других операций и внешней среды.

4. **Действие** – преобразование какого-либо свойства материального или информационного объекта в другое свойство. Действие выполняется в соответствии с командой, являющейся частью директивы на выполнение операции, с потреблением необходимых ресурсов и с соблюдением ограничений, налагаемых на осуществление операции.

Дополнительные виды функций.

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

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

Введенные выше понятия группы основных функций образуют естественную иерархию блоков на IDEF0-диаграммах при декомпозиции, предусматривая четыре уровня последней. Однако при анализе сложных видов деятельности могут потребоваться промежуточные уровни декомпозиции, основанные на применении функций группы Б.

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

Организационно-технические структуры и механизмы IDEF0-моделей.

Все функции, входящие в приведенную выше классификацию, находятся между собой в отношениях иерархической подчиненности по принципу «сверху вниз»: деятельность – субдеятельность – процесс – подпроцесс – операция – действие. Согласно методологии IDEF0 каждая функция выполняется посредством механизма. В большинстве систем, анализируемых

при помощи функциональных моделей такими механизмами служат организационно-технические структуры. Одним из концептуальных принципов функционального моделирования является «отделение «организации» от функций». Вместе с тем анализ показывает, что между иерархией функций (преобразований) и иерархией механизмов существует соответствие, иллюстрируемое на рисунке Рис. 48.

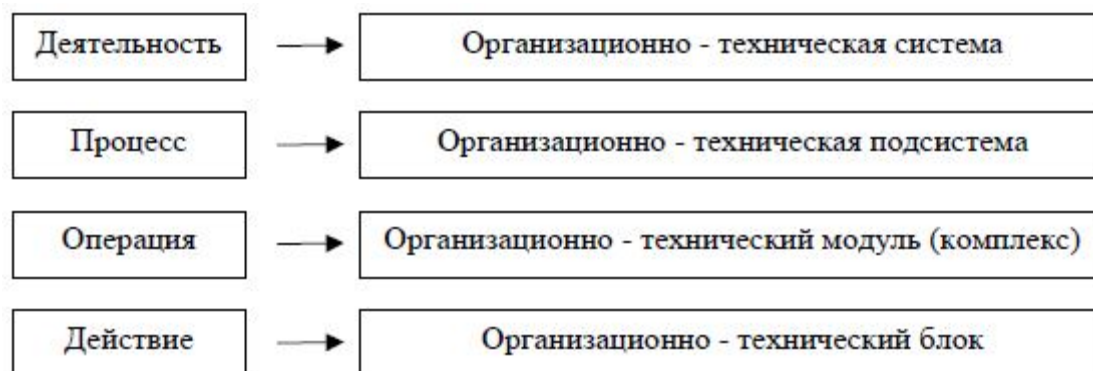


Рис. 48

Используя приведенные выше понятия системного анализа, определим элементы иерархии механизмов следующим образом.

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

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

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

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

Таким образом, при корректном построении модели (без априорной привязки к «организации») появляется возможность связать ее блоки на разных уровнях декомпозиции с объектами организационно-технической структуры, выступающими в качестве механизмов. В этом случае, и это методически

крайне важно, организационно-техническая структура становится результатом функционального моделирования.

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

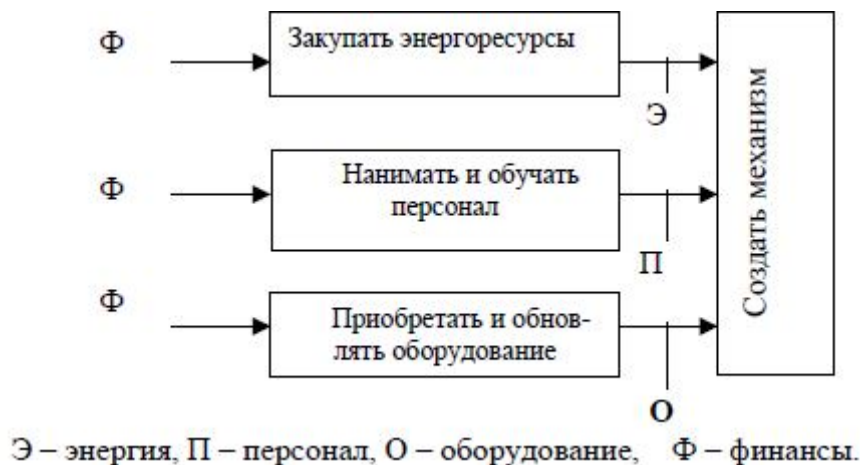


Рис. 49

Явление отражается в модели как субдеятельность, поскольку почти всегда состоит из нескольких процессов. Укрупненная схема этой субдеятельности приведена на Рис. 49. Согласно схеме входом и одновременно потребляемым ресурсом субдеятельности являются финансы, преобразуемые в другие виды ресурсов – энергетические, трудовые, материальные (оборудование, вспомогательные материалы и т.п.).

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

Существенный признак вспомогательного процесса: этот процесс не создает конечного продукта деятельности и, следовательно, прибыли.

Управление – особый вид процесса, операции, действия

Один из общих принципов методологии IDEF 0 требует, чтобы к каждому блоку на диаграмме должна быть присоединена хотя бы одна управляющая стрелка, отображающая условия правильного функционирования блока. Это требование есть следствие положения системотехники, согласно которому управление есть такое воздействие (преимущественно информационное) на систему, которое стимулирует ее функционирование в направлении достижения некоторой цели. В связи с этим можно сформулировать ряд определений и методических положений, которыми следует руководствоваться при отражении управлений на функциональных моделях.

Управление деятельностью – процесс, состоящий, как минимум, из следующих операций:

- формулирование целей деятельности;
- оценивание ресурсов, необходимых для осуществления деятельности и их сопоставление с имеющимися ресурсами;
- сбор информации об условиях протекания и фактическом состоянии деятельности («глобальная» обратная связь);
- выработка и принятие решений, направленных на достижение целей, в частности, решений о распределении ресурсов по процессам, входящим в состав деятельности; оформление решений в виде директив на управление процессами;
- реализация решений (исполнение директив) и оценка их результатов («локальная обратная связь»);
- корректировка (в случае необходимости, например, при нехватке ресурсов) ранее сформулированных целей (самонастройка, адаптация).

Именно решения и их реализация – суть те стимулирующие воздействия на систему, о которых говорилось выше.

Управление процессом – операция, состоящая, как минимум, из следующих действий:

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

- сбор (прием по каналам связи) информации о ходе выполнения операций, ее обобщение и формирование сведений о состоянии процесса; передача данных в подсистему управления деятельностью;

- сопоставление информации о ходе операций с данными директив и выработка локальных решений, направленных на устранение отклонений: корректировка (в случае необходимости) директив на выполнение операций.

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

Блоки управления должны присутствовать на каждой IDEF 0-диаграмме (кроме тех, которые являются декомпозициями самих таких блоков). Через них осуществляются управляющие воздействия на остальные блоки диаграммы. Именно эти блоки воспринимают ограничивающую и предписывающую информацию и преобразуют ее в соответствующие директивы и команды. Имена блоков управления, как правило, содержат глагол «Управлять...».

Стрелки, исходящие из блока с именем «Управлять ...», описывают централизованную схему управления (управленческую «вертикаль»). Возможны варианты структур, в которых выходная информация одного из блоков является управляющей для другого. Это отображает децентрализацию управления («горизонтальные» связи).

Типизация функциональных моделей и IDEF 0– диаграмм

Эффективность и производительность труда разработчиков функциональных моделей могут быть повышены за счет применения типовых

моделей и отдельных диаграмм, ориентированных на применение в конкретных предметных областях. Так, например, на основе представлений о жизненном цикле продукции (изделия) можно предложить типовую диаграмму уровня А0 для промышленного предприятия, которая может иметь вид, схематически показанный на рисунке (Рис. 50).

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



Рис. 50

Методика построения модели

1. Определить предмет моделирования.
2. Определить цели и точки зрения.
3. Создать контекстную функцию.
4. Определить основные граничные ISOM для контекстной функции:
 - определить выходы функций,
 - определить входы,
 - определить управления,
 - определить механизмы.
5. Декомпозиция – определить основные функции диаграммы верхнего уровня.

6. Построить основные связи - выходы на диаграмме верхнего уровня:

- определить все возможные варианты связанные с результатами работы функции,

- определить действия функций, которые могут заканчиваться неудачно, т.к. выходы должны отражать любое развитие событий,

- использовать неудачные действия для создания стрелок обратных связей,

- включить в модель сомнительные или неясные стрелки, обозначенные знаком вопроса, чтобы потом обсудить их с экспертом.

7. Построить основные связи - входы на диаграмме верхнего уровня.

Входами могут быть:

- объекты, из которых получаются объекты на выходе,

- материальные объекты, которые преобразуются в выходное изделие или уничтожаются в результате действия функции,

- информационные объекты, которые не меняют функцию.

8. Построить связи – управления на диаграмме верхнего уровня.

Управлениями могут быть: правила, стандарты, рекомендации, инструкции.

9. Построить связи – механизмы на диаграмме верхнего уровня.

Управлениями могут быть: люди, машины и механизмы, вычислительные системы, любые материальные ресурсы, силами или с помощью которых выполняются действия функции.

10. При необходимости продолжить декомпозицию дальше, повторить пункты 5-9.

Правила построения диаграмм

1. В составе модели должна присутствовать контекстная диаграмма А-0, которая содержит только один блок. Номер единственного блока на контекстной диаграмме А-0 должен быть 0.

2. Блоки на диаграмме должны располагаться по диагонали – от левого верхнего угла диаграммы до правого нижнего в порядке присвоенных номеров. Блоки на диаграмме, расположенные вверху слева «доминируют» над блоками,

расположенными внизу справа. «Доминирование» понимается как влияние, которое блок оказывает на другие блоки диаграммы. Расположение блоков на листе диаграммы отражает авторское понимание доминирования. Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные.

3. Неконтекстные диаграммы должны содержать не менее трех и не более шести блоков. Эти ограничения поддерживают сложность диаграмм на уровне, доступном для чтения, понимания и использования. Диаграммы с количеством блоков менее трех вызывают серьезные сомнения в необходимости декомпозиции родительской функции. Диаграммы с количеством блоков более шести сложны для восприятия читателями и вызывают у автора трудности при внесении в нее всех необходимых графических объектов и меток.

4. Каждый блок неконтекстной диаграммы получает номер, помещаемый в правом нижнем углу; порядок нумерации - от верхнего левого к нижнему правому блоку (номера от 1 до 6).

5. Каждый блок, подвергнутый декомпозиции, должен иметь ссылку на дочернюю диаграмму; ссылка (например, узловой номер, C-номер или номер страницы) помещается под правым нижним углом блока.

6. Имена блоков (выполняемых функций) и метки стрелок должны быть уникальными. Если метки стрелок совпадают, это значит, что стрелки отображают тождественные данные.

7. При наличии стрелок со сложной топологией целесообразно повторить метку для удобства ее идентификации.

8. Следует обеспечить максимальное расстояние между блоками и поворотами стрелок, а также между блоками и пересечениями стрелок для облегчения чтения диаграммы. Одновременно уменьшается вероятность перепутать две разные стрелки.

9. Блоки всегда должны иметь хотя бы одну управляющую и одну выходную стрелку, но могут не иметь входных стрелок.

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

11. Максимально увеличенное расстояние между параллельными стрелками облегчает размещения меток, их чтение и позволяет проследить пути стрелок, как показано на рисунке (Рис. 51, а, б).

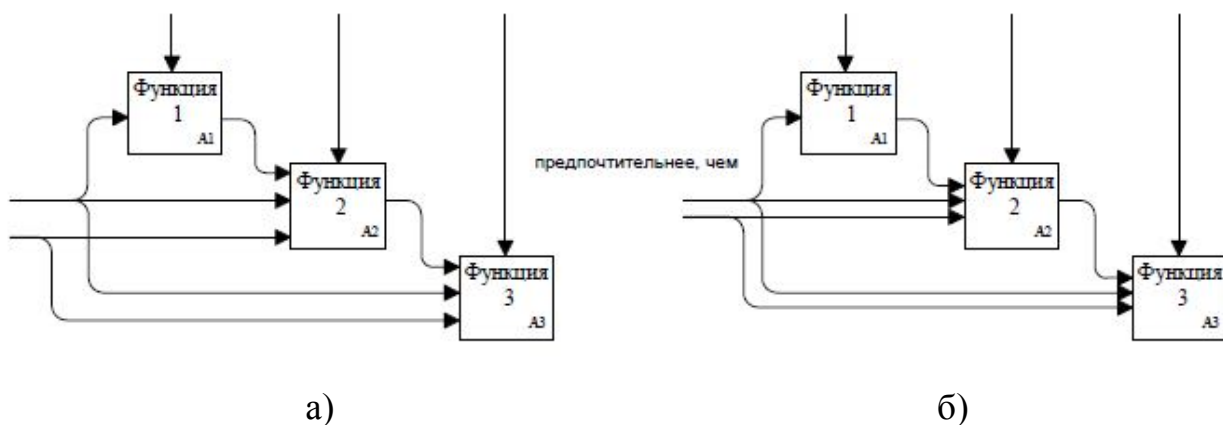


Рис. 51

12. Стрелки связываются (сливаются), если они представляют сходные данные и их источник не указан на диаграмме (Рис. 52).

13. Обратные связи по управлению должны быть показаны как "вверх и над" (Рис. 53 а). Обратные связи по входу должны быть показаны как "вниз и под" (Рис. 53 б). Так же показываются обратные связи посредством механизма. Таким образом обеспечивается показ обратной связи при минимальном числе линий и пересечений.

14. Циклические обратные связи для одного и того же блока изображаются только для того, чтобы их выделить. Обычно обратную связь изображают на диаграмме, декомпозирующей блок. Однако иногда требуется выделить повторно используемые объекты (Рис. 54).



Рис. 52

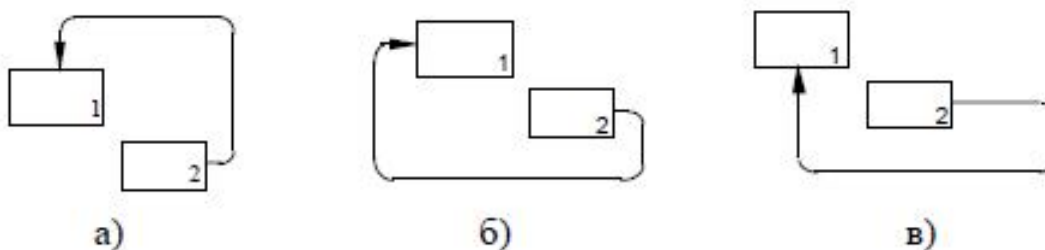


Рис. 53

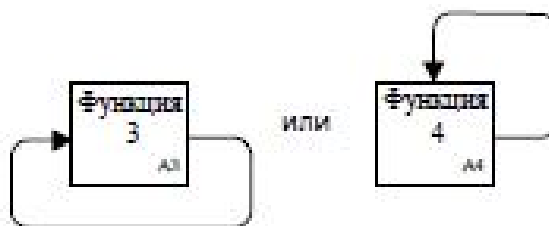


Рис. 54

15. Стрелки объединяются, если они имеют общий источник или приемник, или они представляют связанные данные. Общее название лучше описывает суть данных. Следует минимизировать число стрелок, касающихся каждой стороны блока, если, конечно, природа данных не слишком разнородна (Рис. 55).

16. Если возможно, стрелки присоединяются к блокам в одной и той же позиции. Тогда соединение стрелок конкретного типа с блоками будет согласованным и чтение диаграммы упростится.

17. При соединении большого числа блоков необходимо избегать необязательных пересечений стрелок. Следует минимизировать число петель и поворотов каждой стрелки.

18. Блоки (функции) являются сопряженными через среду, если они имеют связи с источником, генерирующим данные, без конкретного определения отношения отдельной части данных к какому-либо блоку.

19. Две или более функций являются сопряженными через запись, если они связаны с набором данных и не обязательно зависят от того, представлены ли все возможные интерфейсы как сопряжение через среду. Тип интерфейса, показанный на рисунке (Рис. 56), предпочтителен, поскольку определяют отношения конкретных элементов данных к каждому блоку.

20. Необходимо использовать (где это целесообразно) выразительные возможности ветвящихся стрелок.

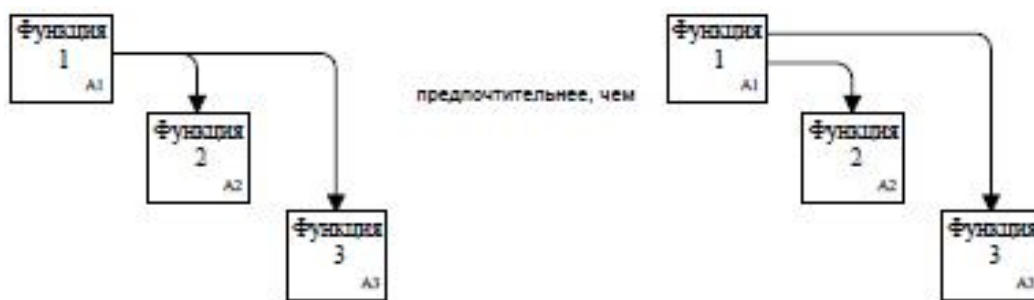


Рис. 55

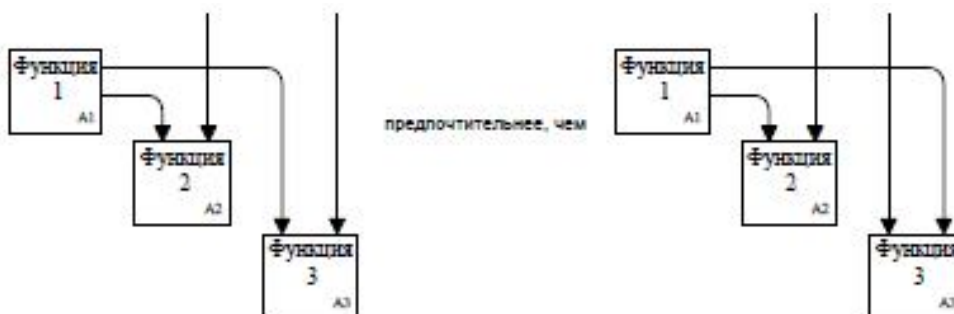


Рис. 56

ЛИТЕРАТУРА

1. Вендеров А.М. CASE_технологии. Современные методы и средства проектирования информационных систем М.: Финансы и статистика, 2005.
2. Новикова Т.Б. Методологии IDEF0: Типы связей, туннелирование стрелок // Международный журнал экспериментального образования. – 2016. – № 12-3. – С. 377-383.
3. Черемных С. В. и др. Структурный анализ систем: IDEF-технологии. — М.: Финансы и статистика, 2001. — 208 с.
4. Бахтизин В.В., Глухова Л.А. Методология функционального проектирования IDEF0. Учебное пособие по курсу «Технология разработки программного обеспечения» для студ. спец. 40 01 01 Программное обеспечение информационных технологий дневной формы обучения. – Минск: БГУИР, 2003. – 24 с.
5. Черемных С., Семенов И., Ручкин В. Моделирование и анализ систем. IDEF-технологии: практикум — М.: Финансы и статистика, 2006. — 188 с.
6. Методология функционального моделирования IDEF0. Руководящий документ. – М.: Госстандарт России, 2000. – 75 с.