

## Практическое занятие № 2.

### Метод моделирования IDEF0. Построение диаграмм и моделей в IDEF0.

**Цель практического занятия:** изучить правила построения диаграмм и моделей в IDEF0.

#### План проведения занятия:

1. Ознакомиться с лекционным материалом и материалом практического занятия № 2 по методу моделирования IDEF0.
2. Изучить правила построения диаграмм.
3. Изучить правила построения моделей.

#### 1. Правила построения диаграмм.

Диаграмма строится из функциональных блоков. Количество функциональных блоков на диаграмме может быть не менее 3 и не более 6. Связи (стрелки) связывают функциональные блоки на диаграмме и отображают взаимосвязи и взаимодействия между ними. Функциональные блоки никогда не размещаются на диаграмме случайным образом – размещение происходит по степени важности, как ее понимает автор (применяя принцип доминирования, то есть относительного порядка).

Доминирование (для диаграмм) – размещение блоков по степени важности (относительный порядок), как ее понимает автор. Доминирование понимается как влияние, которое один блок оказывает на другие блоки диаграммы. Наиболее доминирующий блок размещается над наименее доминирующим.

Обычно диаграмма представлена на листе вместе со служебной информацией. Служебная информация состоит из выделенных верхнего и нижнего колонтитулов (заголовка и «подвала»). Элементы заголовка используются для отслеживания процесса создания модели. Элементы «подвала» отображают наименование модели, к которой относится диаграмма, и показывают ее расположение относительно других диаграмм модели.

#### Элементы заголовка диаграммы IDEF0

Таблица 1

Поле	Назначение
Используется в:	Используется для отражения внешних ссылок на данную диаграмму (заполняется на печатном документе вручную)
Автор, дата, проект, пересмотр	Содержит ФИО автора диаграммы, дату создания, дату последнего внесения изменений, наименование проекта, в рамках которого она создавалась.

Замечания: 1.....10	При ручном редактировании диаграмм пользователи могут зачеркивать цифру каждый раз, когда они вносят очередное исправление
Статус:	Статус отражает состояние разработки или утверждения данной диаграммы. Это поле используется для реализации формального процесса публикации в шагами пересмотра и утверждения.
Рабочая версия	Новая диаграмма, глобальные изменения или новый автор для существующей диаграммы.
Эскиз	Диаграмма достигла некоторого приемлемого для читателей уровня и готова для представления на утверждение.
Рекомендовано	Диаграмма одобрена и утверждена. Какие-либо изменения не предвидятся.
Публикация	Диаграмма готова для окончательной печати и публикации.
Читатель	ФИО читателя.
Дата	Дата знакомства читателя с диаграммой.
Контекст	Набросок расположения функциональных блоков на родительской диаграмме, на которой выделяется функциональный блок, декомпозируемый данной диаграммой. Для диаграммы самого верхнего уровня (контекстной) в поле приводится контекст TOP.

#### Элементы «подвала» диаграммы IDEF0

Таблица 2

Поле	Назначение
Узел:	Номер диаграммы, совпадающий с номером родительского функционального блока.
Название:	Имя родительского функционального блока.
Номер:	Уникальный идентификатор данной версии данной диаграммы. Как правило, номер состоит из инициалов автора (которые должны быть уникальными среди всех аналитиков проекта) и последовательного уникального идентификатора – например, СКЛ005. При публикации эти номера могут быть заменены стандартными номерами страниц. Если диаграмма замещает другую диаграмму, то номер замещаемой диаграммы заключается в скобки - СКЛ005 (СКЛ004). Это обеспечивает сохранение всех версий диаграммы и хранение истории изменений всех диаграмм модели.

Для обеспечения точности представления предметной области диаграммы пересматриваются и изменяются. Автор готовит свой набор диаграмм для каждого автора, рецензенты свои предложения по изменениям возвращают автору, чтобы он внес их в модель. Если возникают разногласия между рецензентом и автором, то диаграмма обычно рассылается всем рецензентам для достижения общей договоренности по спорному вопросу.

Формально механизм рецензирования и модификации диаграмм поддерживается полями «Статус» и нумерацией диаграмм, контроль истории изменений – полем «Дата» и «Автор, дата, проект, *пересмотр*».

## **2. Правила построения моделей.**

Для описания системы требуется, как правило, несколько диаграмм. Диаграммы, собранные и связанные вместе, становятся моделью. В IDEF0 дополнительно к правилам синтаксиса диаграмм существуют правила синтаксиса моделей. Синтаксис IDEF0-моделей позволяет определить границу модели, связать диаграмму в одно целое и обеспечить точное согласование между диаграммами.

При построении моделей используют следующие приемы детализации процессов:

- декомпозиция (для моделей) – разбиение процесса на детализированную последовательность связанных подпроцессов. Родительский блок (процесс) декомпозируется на несколько «дочерних» подпроцессов. Декомпозируемый блок и касающиеся его дуги определяют точные границы диаграммы, представляющей декомпозицию данного блока;
- построение (для моделей) – группировка процессов по уровням декомпозиции.

IDEF0-модель является иерархически организованной совокупностью диаграмм. Сначала формируется контекстная диаграмма модели, а потом она декомпозируется в соответствии с правилами IDEF0. Контекстная диаграмма содержит один функциональный блок, который описывает общую функцию, выполняемую системой. Дуги, касающиеся этого блока, описывают главные управления, входы, выходы и механизмы всей системы. При этом, блок изображает границу системы: все, лежащее внутри него, является частью описываемой системы, а все, лежащее вне его, образует среду системы.

IDEF0-модели развиваются в процессе структурной декомпозиции «сверху-вниз». Сначала декомпозируется один блок контекстной диаграммы на диаграмму, содержащую от 3 до 6 блоков, затем декомпозируется один или несколько из этих блоков и так далее. Название диаграммы совпадает с названием декомпозируемого блока. Декомпозиция формирует границы новой диаграммы. Декомпозируемый блок и касающиеся его дуги определяют точную границу диаграммы, представляющей декомпозицию этого блока. Декомпозируемый блок называется «родительским» блоком, а содержащая его диаграмма – «родительской». Диаграмма, представляющая декомпозицию данного блока называется диаграммой-потомком или «дочерней». Она описывает все, что связано с декомпозируемым блоком и его дугами и не описывает ничего вне этой границы. Таким образом, IDEF0-диаграмма является декомпозицией некоторого ограниченного объекта. Принцип ограничения объекта встречается на каждом уровне.

Граничные дуги на обыкновенной (не контекстной) диаграмме представляют входы, выходы, управления и механизмы соответствующего этой диаграмме родительского

функционального блока. Источник или точка применения граничных дуг может быть установлена путем изучения родительской диаграммы. Все граничные дуги на дочерней диаграмме (за исключением тоннельных дуг) должны соответствовать дугам, подсоединенным к родительскому функциональному блоку.

Результатом этого процесса является модель, диаграмма верхнего уровня которой описывает систему в общих терминах «черного ящика», а диаграммы нижнего уровня описывают детализированные аспекты и операции системы.

### **Построение моделей.**

Ни одна модель не должна строиться без ясного осознания объекта и целей моделирования. Выбранное определение цели моделирования должно отвечать на следующие вопросы:

- Почему моделируется данный процесс?
- Что выявит данная модель?
- Как смогут ее применить те, кто с ней ознакомиться?

Модели строятся для того, чтобы ответить на набор поставленных вопросов. Такие вопросы формулируются на ранних стадиях моделирования и впоследствии служат основой для четкого и краткого определения цели моделирования.

Примеры вопросов:

- Кто контролирует работу?
- Какая технология нужна для выполнения работы?
- Как продвигаются по цеху материалы?
- На каких этапах требуется инструмент?
- Каковы обязанности рабочего?
- И т.д.

Рекомендуемый порядок разработки моделей «с нуля»: формулирование цели моделирования, выбор точки зрения, определение границ моделирования.

**Цель моделирования** – это краткое описание причины существования модели, то есть получения ответов на некоторую совокупность вопросов.

### **Точка зрения.**

С методической точки зрения при моделировании полезно использовать мнение экспертов, у которых разные взгляды на предметную область. Но при этом каждая модель должна разрабатываться только с одной определенной точки зрения. При этом другие точки

зрения можно задокументировать в дополнительных диаграммах ГЕО для наглядности изложения и лучшего понимания предметной области.

Точка зрения выбирается исходя из цели моделирования. Наименованием точки зрения может быть наименование должности, роли, подразделения и т.п. (например, мастер цеха, директор по производству).

**Точка зрения** – это краткое определение перспективы модели; это позиция, с которой наблюдается система и создается ее модель.

Как и в случае с определением цели моделирования, четкое определение точки зрения необходимо для обеспечения внутренней целостности модели и предотвращения постоянного изменения ее структуры. Может оказаться необходимым построение моделей с разных точек зрения для детального отражения всех особенностей выделенных в системе функциональных блоков.

### ***Границы моделирования.***

**Границы моделирования (или область моделирования)** – некие рамки, дающие понять, какие объекты и процессы должны быть включены в модель и рассмотрены как ее составные части, а какие – являются внешними для модели и не будут рассматриваться подробно.

Границы моделирования имеют два компонента: ширину охвата и глубину детализации. Ширина охвата обозначает внешние границы моделируемой системы. Глубина детализации определяет степень подробности, с которой нужно проводить декомпозицию функциональных блоков.

### ***Контекстная диаграмма.***

Чтобы правильно определить границы моделирования, необходимо корректно разработать контекстную диаграмму модели. Иногда прибегают к построению дополнительной диаграммы для отображения уровня, более высокого, чем контекстный для данной системы, что позволяет обозначить систему, внутри которой располагается объект для моделирования.

Наименование контекстного блока обобщает определение границ моделирования. Правила подбора имени для контекстного блока в целом не отличаются от общих правил наименования функциональных блоков.

Стрелки диаграмм IDEF0 обычно проще проектировать в следующем порядке: *выход, вход, механизм исполнения, управление*. Каждый функциональный блок обозначает отдельную функцию, которая имеет ясно и кратко описываемые результаты работы. Наличие неясностей при анализе выходов какого-либо функционального блока является сигналом для проведения реинжиниринга рассматриваемого процесса.

### *Определение выходов.*

После идентификации возможных выходов полезно провести анализ модели на предмет покрытия ею всех возможных сценариев поведения процесса. Это означает, что если существует вероятность возникновения той или иной ситуации в ходе процесса, модель должна отражать возможность возникновения такой ситуации. Многие начинающие аналитики забывают отразить негативные результаты работы функциональных блоков. Например, блок «Провести экзамен», произведет поток студентов, которые сдали экзамен и получили оценки, но вполне правомерно ожидать и таких студентов, которые не сдали экзамен. Негативные результаты часто используются в качестве обратных связей, анализ на их наличие должен проводиться для каждого блока. Также важно включать в модель спорные стрелки, принятие решения о наличии которых в модели вполне можно переложить на экспертов.

### *Определение входов.*

Входы можно рассматривать как особым образом преобразуемые функциональными блоками сырье или информацию для производства выхода. В производственных отраслях определить, как входное сырье преобразуется в готовую продукцию, достаточно просто. Однако при моделировании информационных потоков входной поток данных может представляться не потребляемым и не обрабатываемым вообще. Случаи, когда входящие и исходящие стрелки называются в точности одинаково, очень редки и в основном указывают на бесполезность данного блока для системы в целом или на некорректный выбор имени для исходящей стрелки. Решением может служить применение более подробного описания для входящих и исходящих потоков данных. Например, вход может иметь название «Предварительные данные о количестве деталей», а выход – «Уточненные данные о количестве деталей».

### *Определение механизмов исполнения.*

После создания входов и выходов, приступают к рассмотрению механизмов исполнения или ресурсов. В понятие механизма исполнения входят персонал, информационные системы, оборудование и т.п. Например, блок «Собрать деталь» может потребовать использования какого-либо оборудования, например гаечного ключа.

### *Определение управления.*

Для каждого функционального блока обязательно должно быть определено хотя бы одно управление, контролирующее ход работы функционального блока. В случае, когда неясно, отнести стрелку к входу или управлению, следует ее рисовать как управление. Управление можно рассматривать как особую форму входа функционального блока.

Когда создание контекстной диаграммы завершено, можно задать следующие вопросы:

- Обобщает ли диаграмма моделируемый процесс?
- Согласуется ли диаграмма с границами моделирования, точкой зрения и целью моделирования?

- Подходит ли выбранный уровень детализации стрелок для контекстного блока?

(Обычно на контекстной диаграмме рекомендуется рисовать не более шести стрелок каждого типа.)

### ***Нумерация блоков и диаграмм.***

Все функциональные блоки IDEF0 нумеруются. В номерах допускается использование префиксов произвольной длины, но в основном используют префикс А. Номер блока проставляется за префиксом. Контекстный блок всегда имеет номер А0.

Префикс повторяется для каждого блока модели. Номера используются для отражения уровня декомпозиции, на котором находится блок. Блок А0 декомпозируется в блоки А1, А2, А3 и т.д. Блок А1 декомпозируется в А11, А12, А13 и т.д. Блок А11 декомпозируется в А111, А112, А113 и т.д. Для каждого уровня декомпозиции в конец номера добавляется одна цифра.

### ***Связь между диаграммой и ее родительским функциональным блоком.***

Функциональный блок декомпозируется, если необходимо детально описать его работу. При декомпозиции полезно рассмотреть его жизненный цикл, что помогает определить функциональные блоки получающейся «дочерней» диаграммы. При моделировании IDEF0 важно иметь в виду, что граница «дочерней» диаграммы есть граница родительского функционального блока. Это означает, что вся работа выполняется блоками самого нижнего уровня.

На концах граничных стрелок (начинающихся или заканчивающихся за пределами диаграммы) «дочерних» диаграмм помещают коды ICOM, чтобы показать, где находится соответствующая стрелка на родительской диаграмме (см. рис. 3.1).

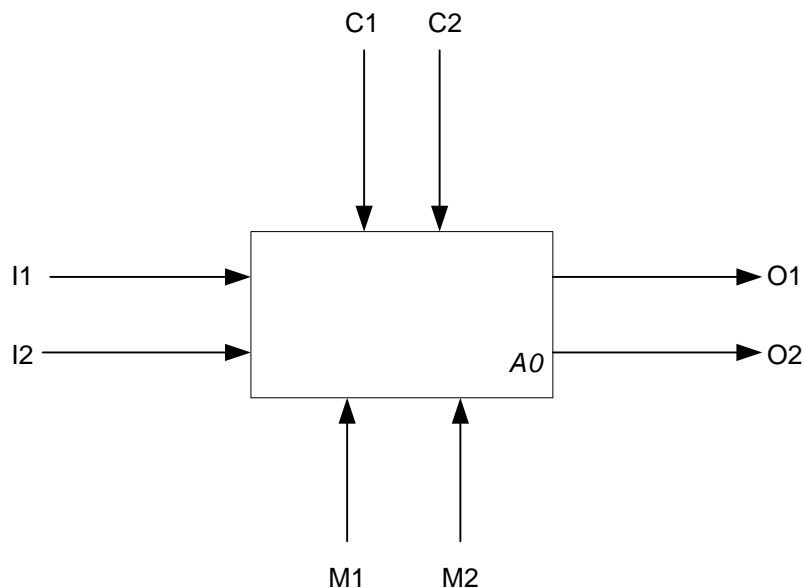


Рис. 3. 1. ICOM-коды на граничных стрелках.

Они нужны для проверки целостности модели и могут быть полезны, когда порядок расположения стрелок на «дочерней» диаграмме отличается от порядка их расположения на родительской диаграмме. Код ICOM состоит из латинской буквы I, C, O и M и числа, показывающего расположение стрелки на родительской диаграмме в порядке сверху вниз или слева направо.

Сформулированная цель моделирования содержит вопросы, на которые должна отвечать модель. Когда становится возможным получение ответов на них с помощью модели, модель считается удовлетворяющей поставленным требованиям и рассматривается как завершенная. При построении декомпозиции первого уровня необходимо следить за тем, чтобы все блоки на диаграмме лежали внутри определенных ранее границ моделирования. Перед декомпозированием блока надо удостовериться, не приведет ли это к превышению установленной ранее глубины детализации для данной модели. Еще одно правило состоит в том, что моделирование IDEF0 должно продолжаться до тех пор, пока стрелки предшествования (вход и выход) преобладают на диаграммах.

При необходимости дальнейшей детализации отдельных процессов могут быть использованы диаграммы IDEF3.

### **Другие диаграммы IDEF0.**

В дополнение к контекстным диаграммам и диаграммам декомпозиции при разработке и представлении моделей могут применяться другие виды IDEF0-диаграмм.

#### ***Дерево модели.***



Это обзорная диаграмма, показывающая структуру всей модели (см. рис. 3.2).

Обычно вершина дерева соответствует контекстному блоку, под вершиной выстраивается вся иерархия блоков модели. Однако можно назначать вершиной произвольный блок, помещая под ним все его «дочерние» блоки. Из-за высокой итеративности функционального моделирования можно ожидать, что дерево модели будет неоднократно изменяться существенным образом до тех пор, пока не будет получена его стабильная версия. Обзор модели с использованием дерева помогает сконцентрироваться на функциональной декомпозиции модели.

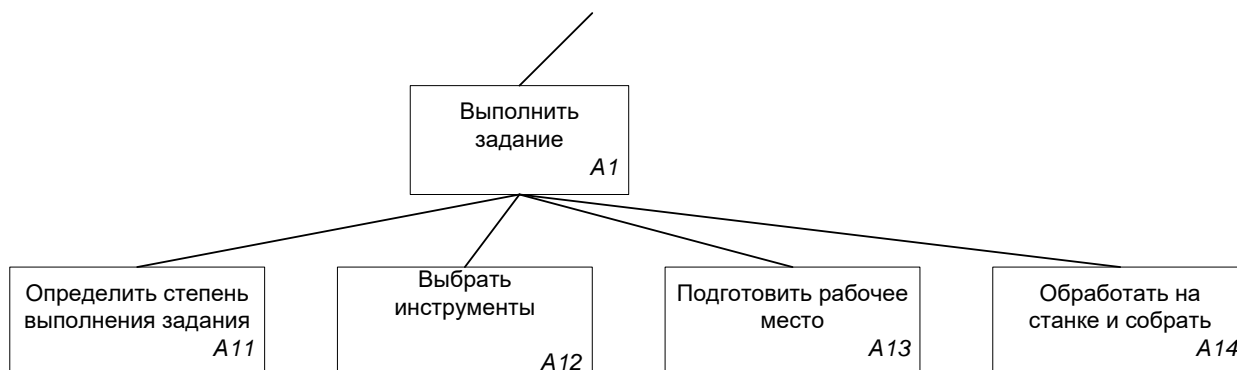


Рис. 3. 2. Фрагмент дерева модели.

**Презентационные диаграммы** (For Exposition Only diagrams – FEO diagrams) часто включают в модели, чтобы проиллюстрировать другие точки зрения или детали, выходящие за рамки традиционного синтаксиса IDEF0. Диаграммы FEO допускают нарушение любых правил построения диаграмм IDEF0 в целях выделения важных с точки зрения аналитика частей модели. Естественно, если диаграмма FEO включена в модель исключительно для отображения другой точки зрения на систему, она скорее всего будет выглядеть как обыкновенная диаграмма IDEF0, удовлетворяя всем ограничениям IDEF0.

Один из способов использования FEO-диаграмм состоит в отделении функционального блока от его окружения посредством создания диаграммы с единственным блоком и всеми относящимися к нему стрелками наподобие контекстной диаграммы. Это бывает полезно в ситуациях, когда необходимо быстро получить информацию об интерфейсе (стрелках) функционального блока, а соответствующая диаграмма декомпозиции содержит слишком много объектов.

Дополнительно встречаются следующие виды презентационных диаграмм:

- копия диаграммы IDEF0, которая содержит все функциональные блоки и стрелки, относящиеся только к одному из функциональных блоков, - это позволяет отразить взаимодействие между этим блоком и другими объектами диаграммы;

- копия диаграммы IDEF0, которая содержит все функциональные блоки и стрелки, непосредственно относящиеся только к входу и (или) выходу родительского блока;
- различные точки зрения, как правило, на глубину одного уровня декомпозиции.

### Пример.

ИСПОЛЬЗУЕТСЯ В:	АВТОР: Марса ПРОЕКТ: ЭМЦ ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10	ДАТА: 02/20/93 ПЕРЕСМОТР:	<input checked="" type="checkbox"/> РАБОЧАЯ ВЕРСИЯ <input type="checkbox"/> ЭСКИЗ <input type="checkbox"/> РЕКОМЕНДОВАНО <input type="checkbox"/> ПУБЛИКАЦИЯ	ЧИТАТЕЛЬ _____ _____ _____	ДАТА _____ _____ _____	КОНТЕКСТ: Тор
-----------------	--	------------------------------	---	-------------------------------------	---------------------------------	------------------

**Вопросы:**

Каковы обязанности мастера ?  
 Каковы обязанности механика ?  
 Кто контролирует задания ?  
 Как продвигаются по цеху материалы ?  
 На каких этапах требуется чертеж ?  
 В какой момент на процесс влияют стандарты качества ?  
 На каких этапах требуются инструменты ?  
 Что происходит с забракованными деталями ?

**Цель:**

Определить обязанности каждого работника экспериментального механического цеха и понять, как эти обязанности взаимосвязаны между собой с тем, чтобы написать учебное пособие.

**Претенденты :**

Мастер  
 Механик  
 Контролер  
 Начальник

**Точка зрения:** Начальника цеха

**2** Процесс обучения для различных типов работников требует декомпозиции в зависимости от обязанностей, которые выполняют эти работники в цехе (см. замечание N5 на диаграмме DAM001).

**1** Только с этой точки зрения можно показать взаимосвязи между отдельными работами и обязанностями персонала.

УЗЕЛ: ЭМЦ/ А-0	НАЗВАНИЕ: Цель и точка зрения модели ЭМЦ	НОМЕР: DAM002
----------------	--	---------------

Рис.3.3. Цель и точка зрения модели.

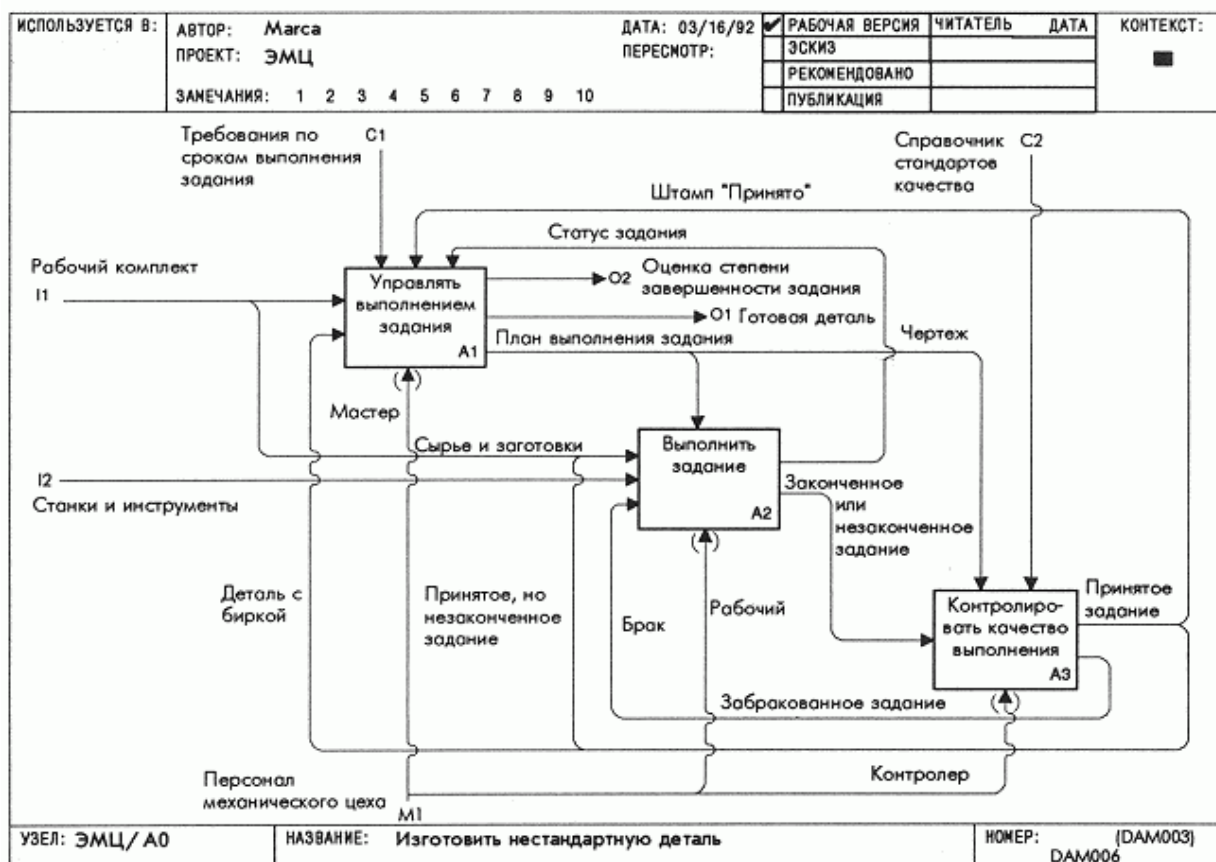


Рис. 3.4. Представление функций и их детализация в IDEF0-диаграммах.

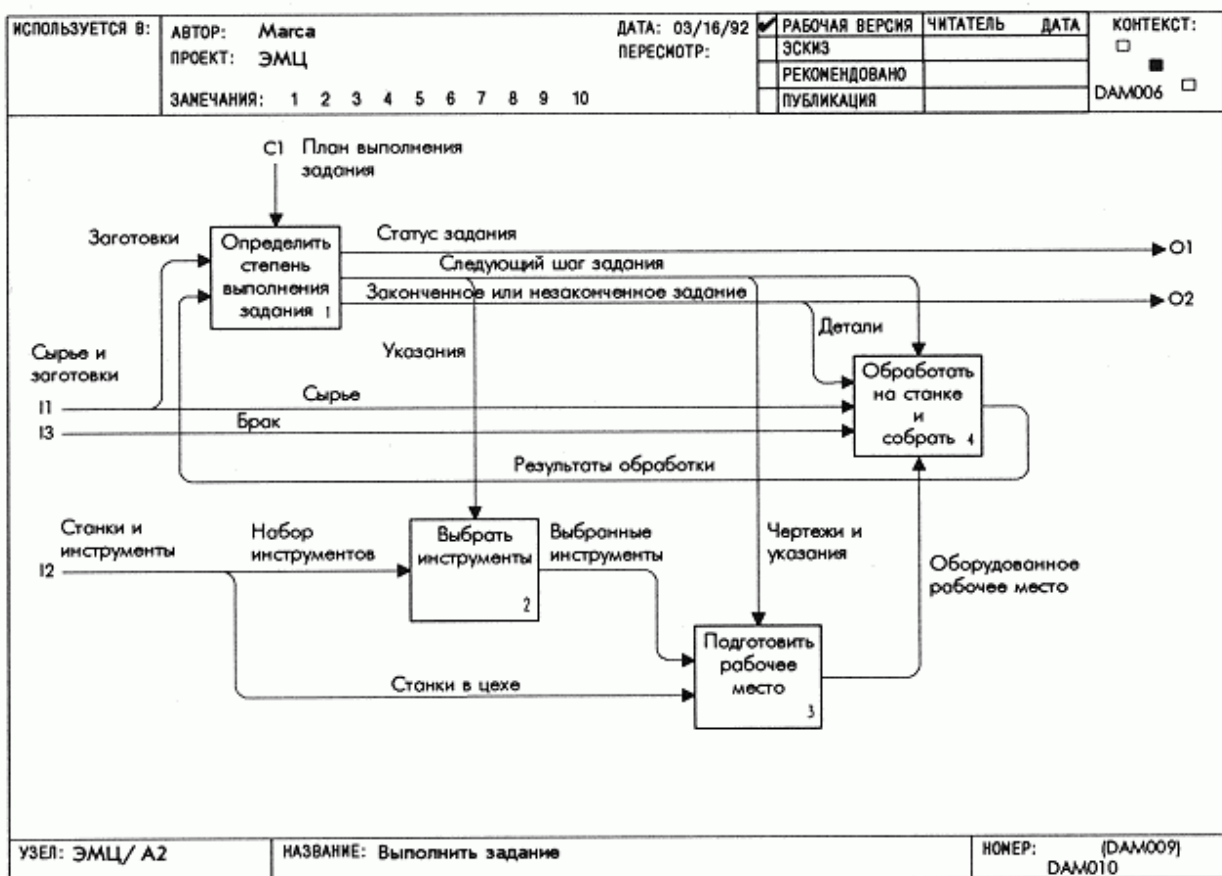


Рис. 3.5. Типичная IDEF0-диаграмма.

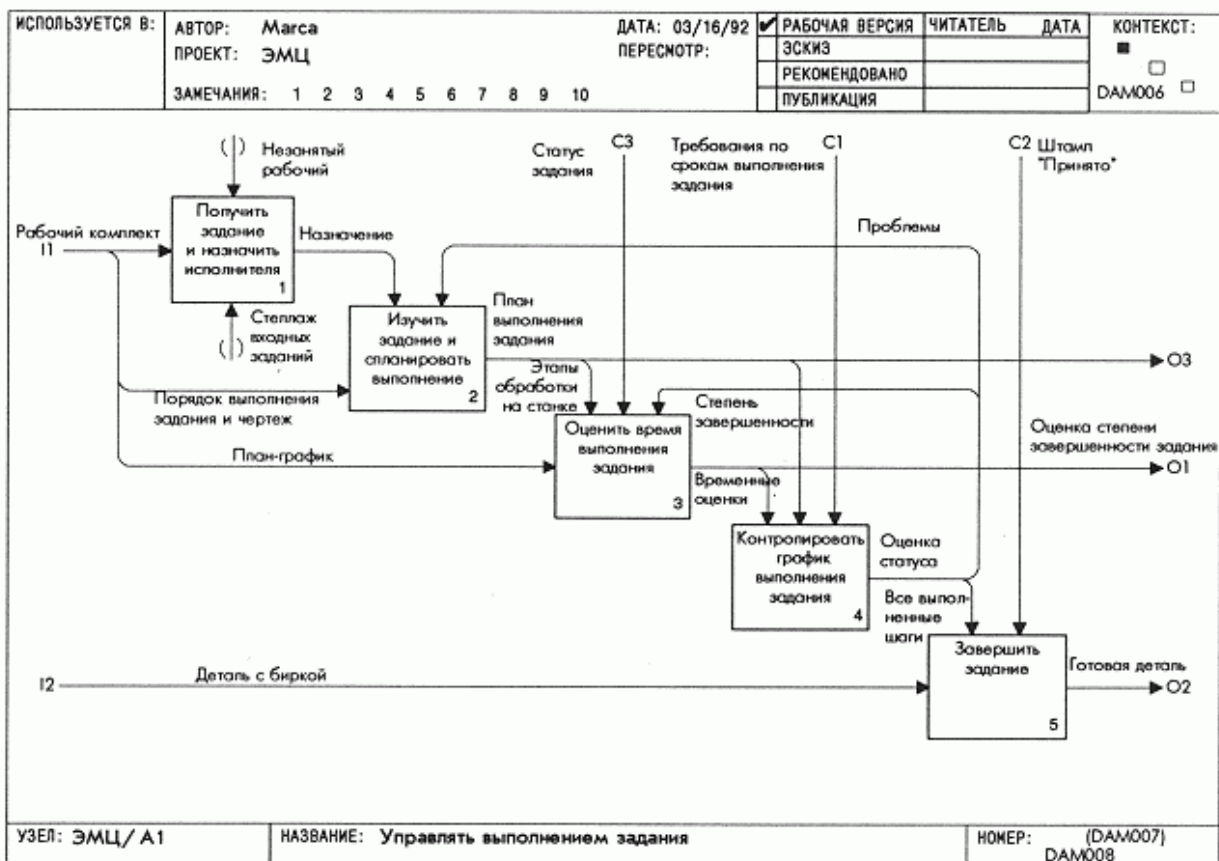
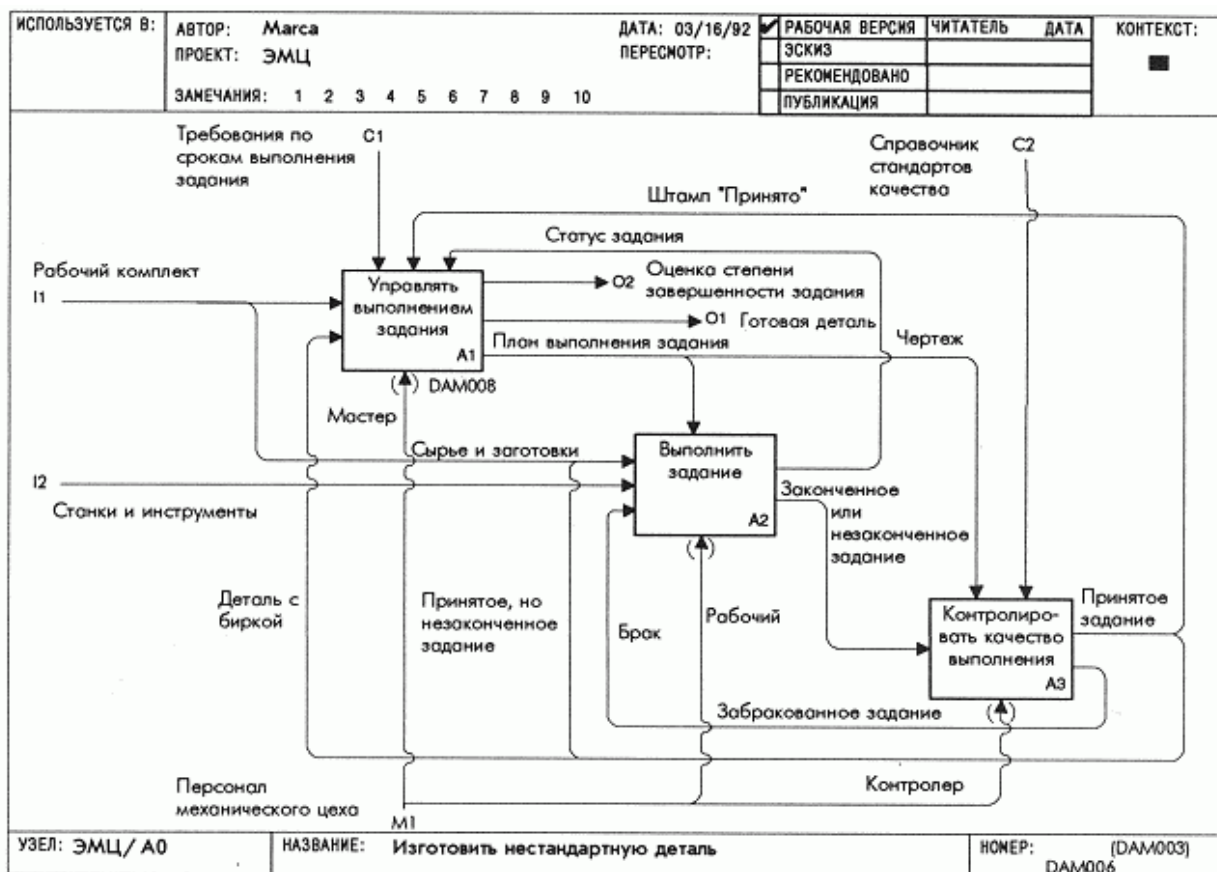


Рис. 3.6. Связь между родительской диаграммой и диаграммой-потомком.

## **Проверка достоверности модели.**

Моделирование в IDEF0 - инженерная дисциплина. Это означает, что модели создаются и проходят серию последовательных улучшений до тех пор, пока они в точности не будут представлять объект моделирования. Одной из основных компонентов методологии IDEF0 является итерационное рецензирование, в процессе которого автор и эксперт неоднократно проверяют достоверность создаваемой модели. Итерационное рецензирование называется циклом *автор-читатель*.

Цикл автор-читатель начинается в тот момент, когда автор принимает решение распространить информацию о какой-либо части своей работы с целью получения отзыва о ней. Организация своевременной обратной связи имеет важнейшее значение для эффективного моделирования, потому что даже малая доля неверной или неверно интерпретированной аналитиком информации, заложенной в IDEF0-диаграмму, может привести к недостоверному описанию моделируемой системы.

Процесс чтения диаграммы можно разбить на четыре последовательных этапа. На первом этапе необходимо понять детали данной диаграммы. Второй этап - концентрация внимания на ближайшем контексте диаграммы. На третьем этапе следует уточнить место диаграммы в модели. Четвертый этап заключается в конструктивной критике авторского изложения.

Изучение деталей конкретной диаграммы необходимо для понимания ее роли в ближайшем контексте. Поняв ближайший контекст диаграммы, можно приступить к изучению ее связей с другими диаграммами модели.

Для понимания деталей отдельной диаграммы необходимо:

- прочитать название и номер узла;
- изучить каждый блок;
- изучить внутренние дуги;
- прочитать все замечания автора;
- просмотреть весь связанный с диаграммой дополнительный материал.

Чтение осуществляется наиболее эффективно, если все эти элементы диаграммы читаются последовательно.

Изучив все внутренние детали диаграммы, сосредоточьтесь на их контексте, определив связи между диаграммой и ее родителем. Вы получите более глубокое понимание диаграммы, потому что граница объекта определяет, как диаграмма входит в остальную часть модели. Понять контекст диаграммы позволяет чтение:

- родительского блока и дуг, появляющихся на родительской диаграмме и представляющих ограничения для изучаемой диаграммы;
- связей изучаемой диаграммы с другими блоками родительской диаграммы;
- дополнительного материала к родительской диаграмме.

Поняв непосредственный контекст диаграммы, посмотрите, как ее данные стыкуются с остальной частью модели. Для этого проследите потоки данных и управления от вершины модели вниз, вплоть до рассматриваемой диаграммы. Это обеспечит полное понимание диаграммы, потому что чтение сверху вниз дает возможность проследить последовательность функциональных деталей, которая привела к созданию диаграммы и указывает пути декомпозиции.

Критическая оценка означает постановку вопросов к содержанию диаграммы:

- верен ли синтаксис диаграммы;
- понимаю ли я, что хотел сказать автор;
- согласен ли я с тем, что выразил автор.

Эти вопросы задают в указанном порядке с тем, чтобы вначале разрешить мелкие, а потом перейти к более глобальным. Все эти шаги предпринимаются для выработки мнения об обоснованности и правильности диаграммы.

Таким образом, по IDEF0 только модель, прошедшая проверку на корректность синтаксиса, связей между составляющими её диаграммами и адекватность описания объекта моделирования может быть допущена к использованию в дальнейшей работе.

### **Список контрольных вопросов:**

1. В чем заключается метод IDEF0?
2. По каким правилам формируется диаграмма в IDEF0?
3. По каким правилам формируется модель в IDEF0?
4. Назовите принципы кодирования функциональных блоков и граничных стрелок в модели.
5. Дайте определение понятию «цель моделирования».
6. Дайте определение понятию «точка зрения».
7. Дайте определение понятию «граница моделирования».
8. Назовите дополнительные диаграммы, которые можно строить вместе с моделью IDEF0.
9. Опишите как можно проверить достоверность модели IDEF0.