

Методология создания ИС

Моделирование потоков данных (процессов)

Методология функционального моделирования
SADT

Моделирование данных. Основные понятия

Моделирование потоков данных (процессов)

Модель системы определяется как иерархия диаграмм потоков данных (ДПД): описывают асинхронный процесс преобразования информации от ввода в систему до выдачи.

Диаграммы верхних уровней иерархии (контекстные) определяют основные **процессы** или **подсистемы** ИС с внешними входами и выходами.

Они детализируются при помощи диаграмм нижнего уровня. Декомпозиция продолжается до тех пор, пока не будет достигнут уровень декомпозиции, на кот. процессы становятся элементарными и детализировать их далее невозможно.

Источники информации (**внешние сущности**) порождают информационные потоки, переносящие информацию к системам или процессам.

Процессы преобразуют информацию и порождают новые **потоки данных**, кот. переносят информацию к другим процессам или подсистемам, **накопителям данных** или внешним сущностям — потребителям информации.

Основными компонентами ДПД являются:

внешние сущности; системы/подсистемы; процессы; накопители данных; потоки данных.

Внешняя сущность (ВС).

Материальный предмет или физич. лицо - источник или приемник информации (заказчики, персонал, поставщики, клиенты, склад).

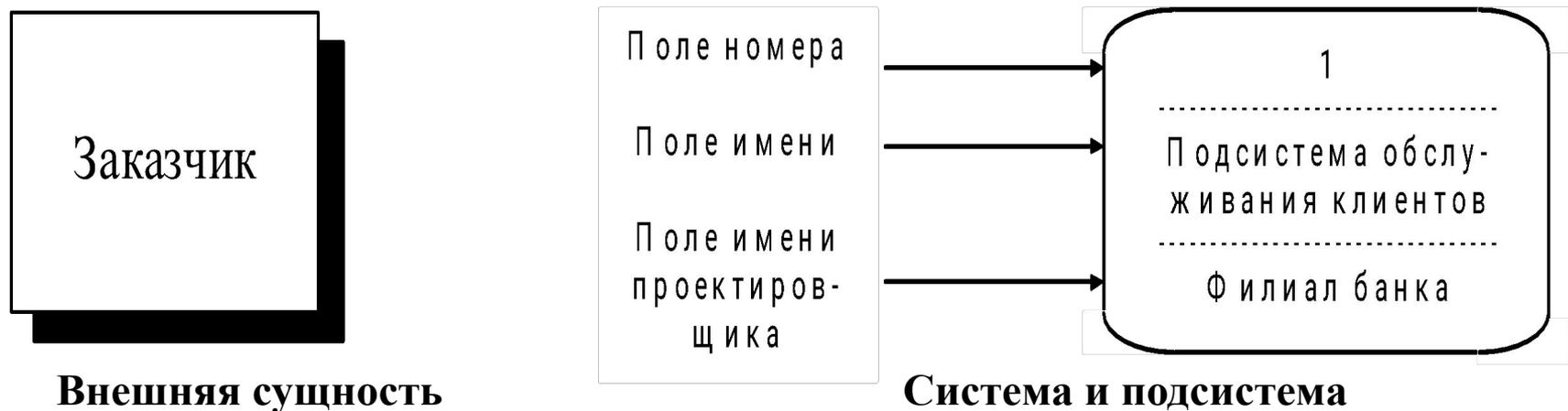
Определение объекта или системы в качестве ВС указывает на то, что она находится за пределами границ анализируемой ИС.

В процессе анализа некот. ВС м. б. перенесены внутрь диаграммы анализируемой ИС, или, наоборот, часть процессов ИС м. б. вынесена за пределы диаграммы и представлена как ВС.

ВС обозначается квадратом, расположенным как бы над диаграммой и бросающим на нее тень для того, чтобы м. б. выделить этот символ среди других обозначений.

Система и подсистема.

При построении модели сложной ИС она м. б. представлена в самом общем виде на так назыв. контекстной диаграмме в виде одной системы как единого целого либо м. б. декомпозирована на ряд подсистем.

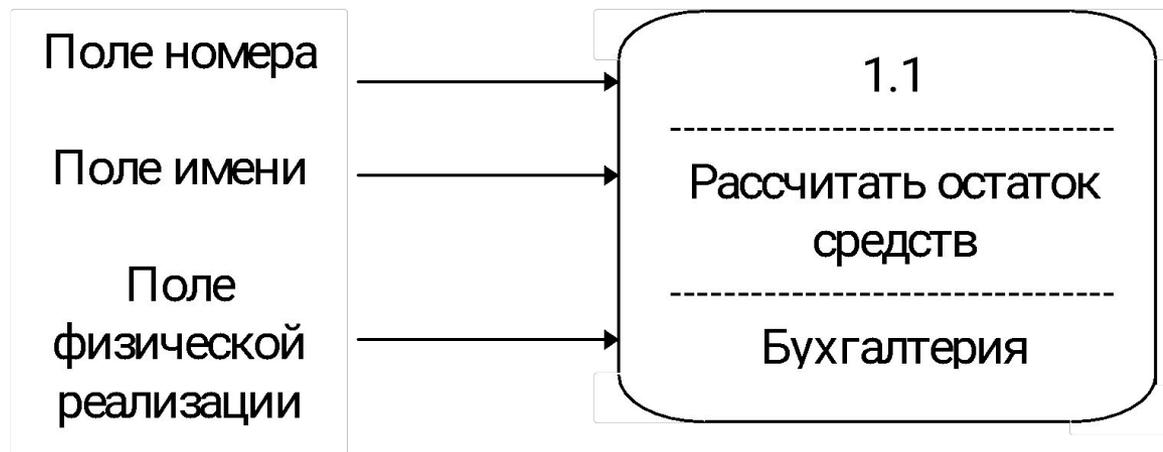


Процесс. Преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Физически процесс м. б. реализован различными способами: это м. б. подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т.д.

Номер процесса служит для его идентификации. *В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за кот. следуют существительные в винительном падеже, напр., “Ввести сведения о клиентах”, “выдать информацию о текущих расходах”, “Проверить кредитоспособность клиента”.*

Использование таких глаголов, как “обработать”, “модернизировать” или “отредактировать”, означает, к. п., недостаточно глубокое понимание данного процесса и требует дальнейшего анализа.

Информация в поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

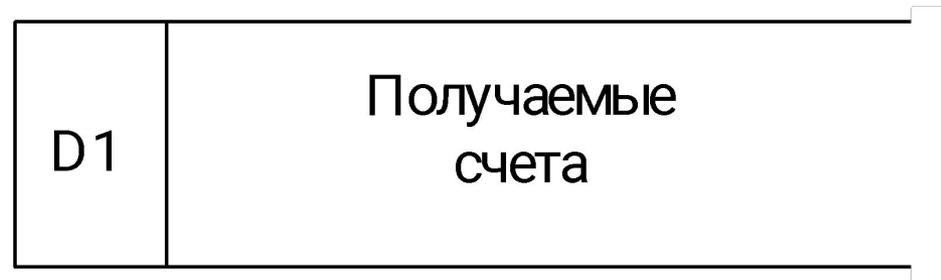


Накопитель данных. Это абстрактное устройство для хранения информации, кот. м. в любой момент поместить в накопитель и через некот. время извлечь, причем способы помещения и извлечения м. б. любыми.

Накопитель данных м. б. реализован физически в виде ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т.д.

Накопитель данных идентифицируется буквой “D” и произвольным числом. Имя накопителя выбирается из соображения наибольшей информативности для проектировщика.

Накопитель данных в общем случае является прообразом будущей БД, и описание хранящихся в нем данных д. б. увязано с информационной моделью.



Поток данных. Он определяет информацию, передаваемую через некот. соединение от источника к приемнику. Реальный поток данных м. б. информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т.д.

Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, кот. показывает направление потока (рис.). Каждый поток данных имеет имя, отражающее его содержание.



Построение иерархии диаграмм потоков данных (ДПД)

Проектирование простых ИС: строится единственная контекстная диаграмма со звездообразной топологией - в центре находится главный процесс, соединенный с приемниками и источниками информации.

Проектирование сложных ИС: единственная контекстная диаграмма б. содержать слишком большое количество источников и приемников информации, их трудно расположить на листе бумаги нормального формата, главный процесс не раскрывает структуры распределенной системы. Выход - детализация на верхнем уровне.

Признаки сложности (в смысле контекста):

- наличие большого количества внешних сущностей (десять и более),
- распределенная природа системы,
- многофункциональность системы с уже сложившейся или выявленной группировкой функций в отдельные подсистемы.

Иерархия контекстных диаграмм (КД) для сложных ИС.

КД верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. КД следующего уровня детализируют контекст и структуру подсистем.

Иерархия КД определяет взаимодействие основных функциональных подсистем проектируемой ИС между собой с внешними входными и выходными потоками данных и внешними объектами (источниками и приемниками информации).

Разработка КД решает проблему строгого определения функциональной структуры ИС. Это важно для сложных ИС, в разработке кот. участвуют разные организации и коллективы разработчиков.

После построения контекстных диаграмм полученную модель следует проверить на полноту исходных данных об объектах системы и изолированность объектов (отсутствие информационных связей с др. объектами).

Детализация подсистем при помощи ДПД

Каждый процесс на ДПД м. б. детализирован при помощи ДПД или мини-спецификации.

Правила детализации:

— *правило балансировки* — детализирующая диаграмма в качестве внешних источников/приемников данных м. иметь только те компоненты (подсистемы, процессы, внешние сущности, накопителем данных), с кот. имеют информационную связь детализируемые подсистема или процесс на родительской диаграмме;

— *правило нумерации* — при детализации процессов д. поддерживаться их иерархическая нумерация. Напр., процессы, детализирующие процесс с номером 12, получают номера 12.1, 12.2, 12.3 и т.д.

Мини-спецификация

- конечная вершина иерархии ДПД.

Решение о **завершении детализации** процесса принимается аналитиком исходя из следующих **критериев**:

- наличия процесса относительно небольшого количества **ВХОДНЫХ** и **ВЫХОДНЫХ** потоков данных (2-3 потока);
- возможности описания преобразования данных процессом в виде последовательного алгоритма;
- выполнения процессом единственной логической функции преобразования входной информации в выходную;
- возможности описания логики процесса при помощи мини-спецификации небольшого объема (не более 20-30 строк).

Переход к детализации процессов осуществляется после определения содержания всех потоков и накопителей данных, кот. описывается при помощи *структур данных*.

Структуры данных конструируются из элементов данных и м. содержать альтернативы, условные вхождения и итерации.

Условное вхождение означает, что данный компонент м. отсутствовать в структуре.

Альтернатива означает, что в структуру м. входить один из перечисленных элементов.

Итерация означает вхождение любого числа элементов в указанном диапазоне.

Для каждого элемента данных м. указываться его тип (непрерывные или дискретные данные).

Для непрерывных данных м. указываться единица измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования.

Для дискретных данных м. указываться таблица допустимых значений.

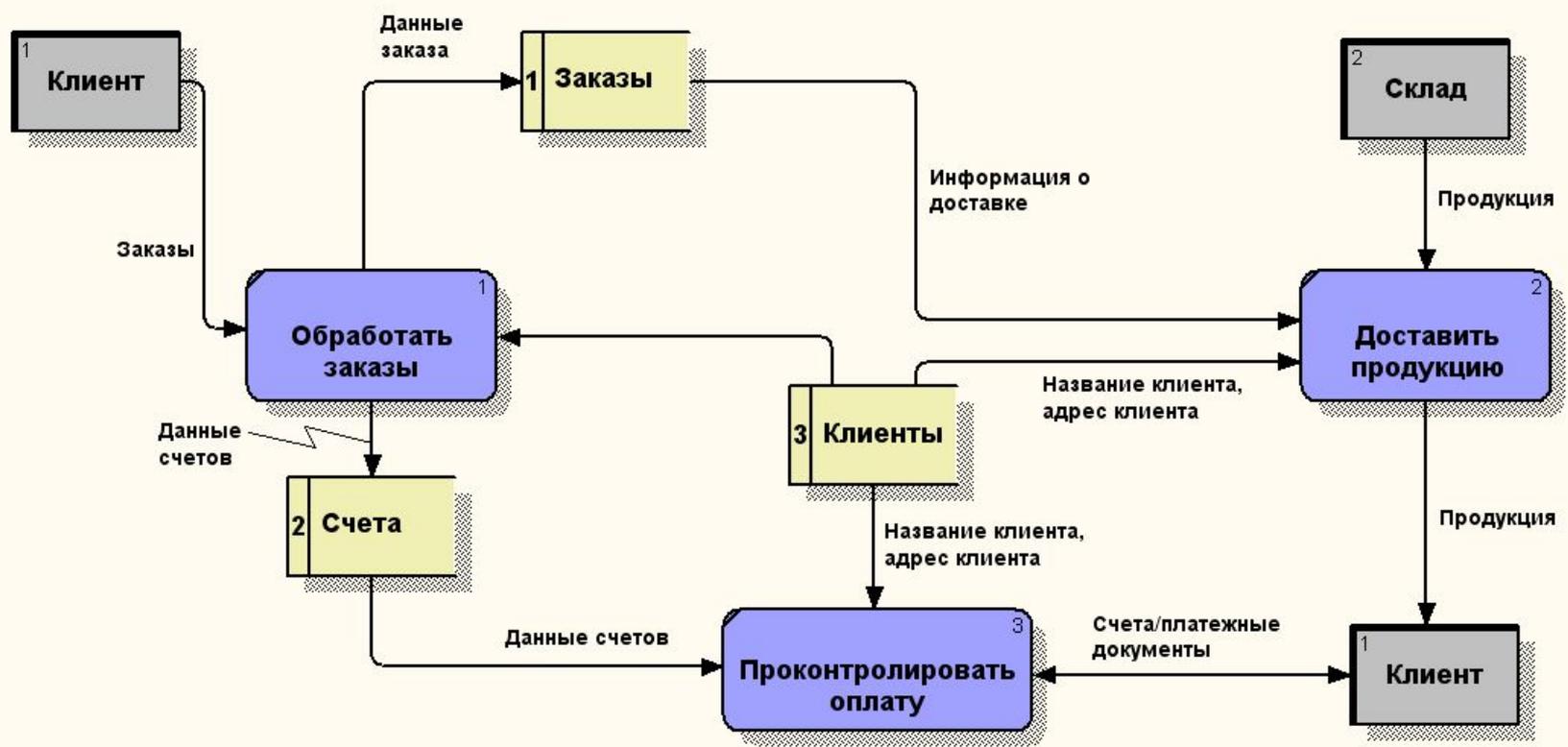
Верификация (*проверка на полноту и согласованность*)

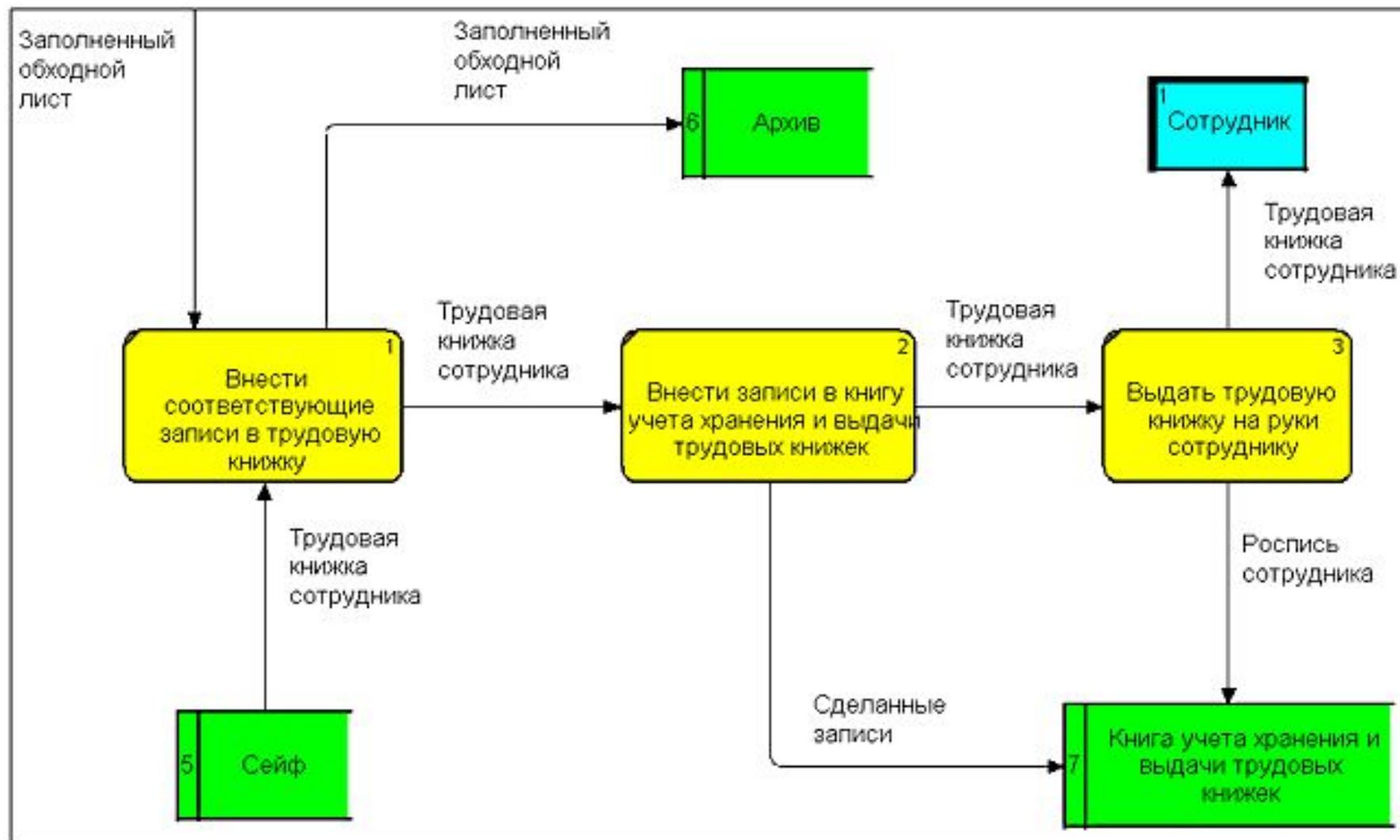
После построения законченной модели системы ее необходимо **верифицировать**.

В полной модели все ее объекты (подсистемы, процессы, потоки данных) д. б. подробно описаны и детализированы.

Выявленные недетализированные объекты следует детализировать, вернувшись на предыдущие шаги разработки.

В согласованной модели для всех потоков данных и накопителей данных д. выполняться правило сохранения информации: *все поступающие куда-либо данные д. б. считаны, а все считываемые данные д. б. записаны.*

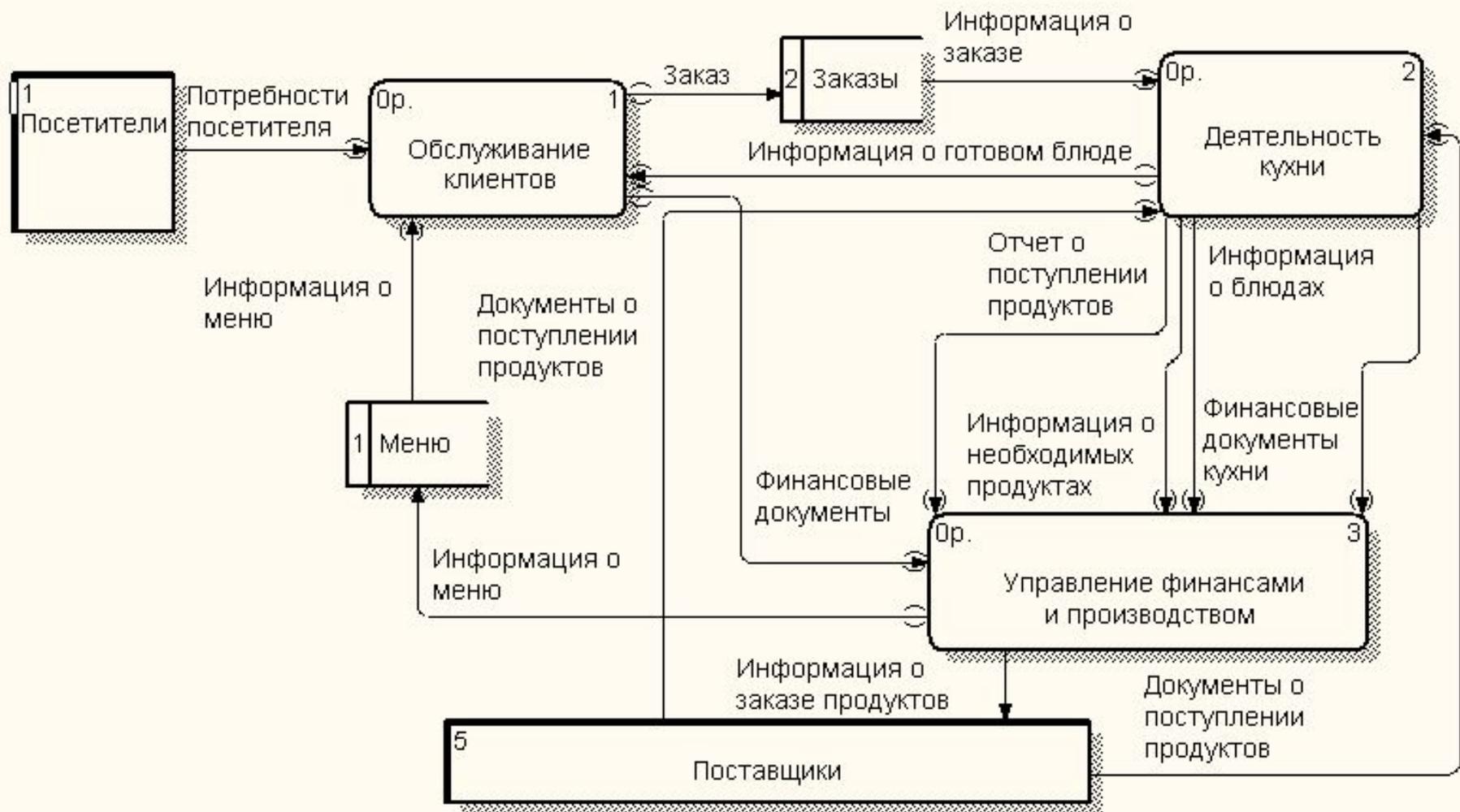




Пример DFD-диаграммы бизнес-процесса (нотация Гейна-Сарсона)

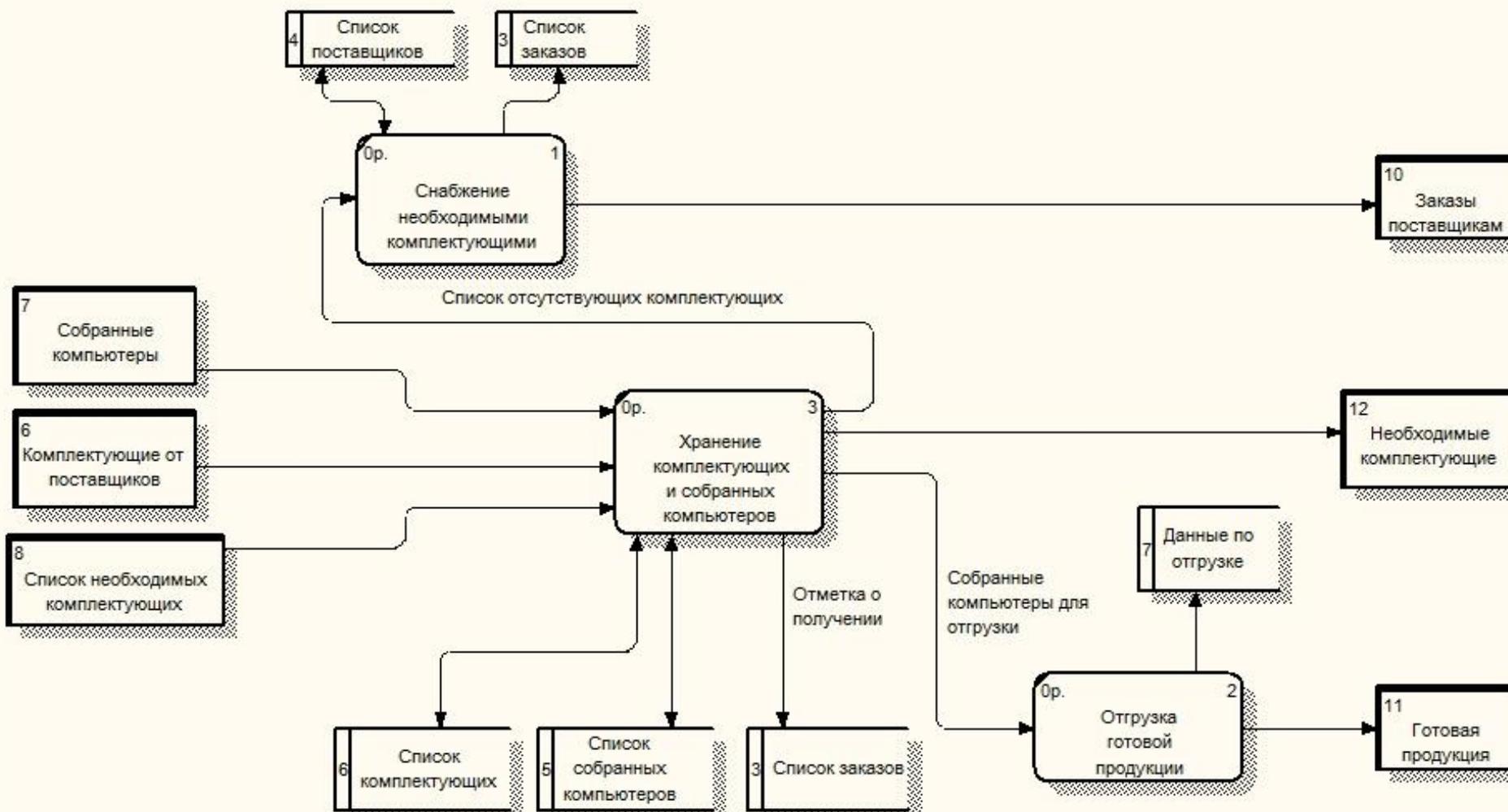


| | | | | | | |
|-----------------------------|-------------------|------------------|-------------|--------|------|----------|
| USED AT: | AUTHOR: | DATE: 04.05.2010 | WORKING | READER | DATE | CONTEXT: |
| | PROJECT: resDFD_2 | REV: 29.05.2010 | DRAFT | | | |
| | | | RECOMMENDED | | | |
| NOTES: 1 2 3 4 5 6 7 8 9 10 | | | PUBLICATION | | | A-0 |

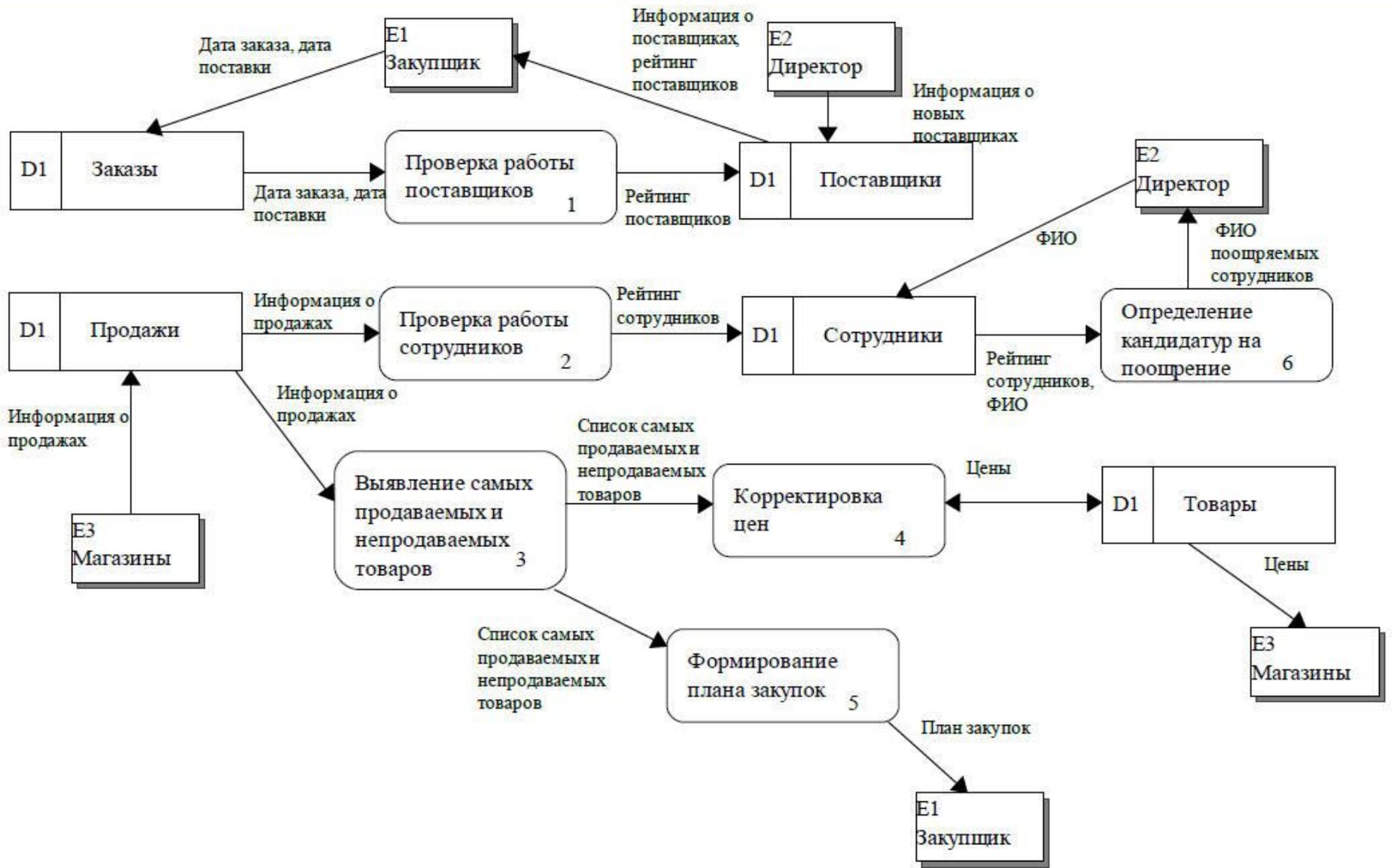


| | | |
|--------------------|---|---------|
| NODE: A0 | TITLE: Деятельность ресторана | NUMBER: |
|--------------------|---|---------|

| | | | | | | |
|----------|-----------------------------|------------------|-------------|--------|------|----------|
| USED AT: | AUTHOR: Fastowsky G. Eduard | DATE: 12.04.2010 | WORKING | READER | DATE | CONTEXT: |
| | PROJECT: Computer Firm | REV: 13.04.2010 | DRAFT | | | |
| | | | RECOMMENDED | | | |
| | NOTES: 1 2 3 4 5 6 7 8 9 10 | | PUBLICATION | | | A0 |



| | | |
|--------------------|---------------------------------------|---------|
| NODE: A4 | TITLE: Отгрузка и снабжение | NUMBER: |
|--------------------|---------------------------------------|---------|



А4 Анализировать работу

Методология функционального моделирования SADT

Методология SADT разработана Дугласом Россом.

На ее основе разработана известная методология IDEF0 (Icam DEFinition).

Методология функционального моделирования — совокупность:

- методов,
- правил и
- процедур,

предназначенных для построения функциональной модели (ФМ) объекта какой-либо предметной области.

ФМ отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Концепции методологии:

- *графическое представление блочного моделирования*. Графика блоков и дуг SADT-диаграммы отображает **функцию** в виде **блока**, а **интерфейсы входа/выхода** представляются **дугами**, соответственно входящими в блок и выходящими из него;
- *строгость и точность*. Правила SADT включают:
 - ограничение количества блоков на каждом **уровне декомпозиции** (правило **3-6 блоков**),
 - **связность** диаграмм (**номера блоков**),
 - **уникальность меток и наименований** (отсутствие повторяющихся имен),
 - синтаксические правила для графики (блоков и дуг),
 - **разделение входов и управлений** (правило определения роли данных);
- *отделение организации от функции*, т.е. исключение влияния организационной структуры на функциональную модель.

Методология SADT используется для моделирования широкого круга систем.

В уже существующих системах SADT используется для анализа функций, выполняемых системой, и указания механизмов, посредством кот. они осуществляются.

Состав ФМ

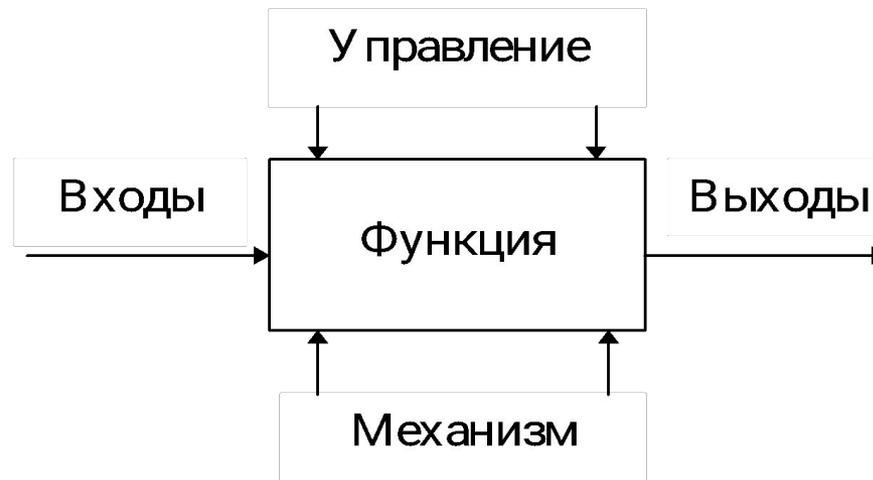
Модель состоит из

- *диаграмм,*
- *фрагментов текстов и*
- *гlossария,* имеющих ссылки друг на друга.

Диаграммы — главные компоненты модели, все **функции** ИС и *интерфейсы* на них представлены как **блоки** и *дуги*.

Место соединения дуги с блоком определяет *тип интерфейса*.

Управляющая информация входит в блок сверху, в то время как информация, кот. *подвергается обработке,* показана с левой стороны. *Механизм* (человек или автоматизированная система), кот. осуществляет операцию, представляется дугой, входящей в блок снизу.



Особенность методологии SADT

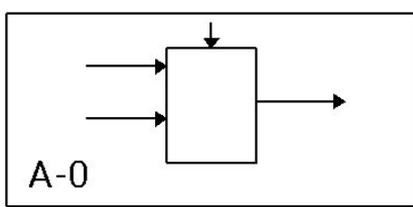
- постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель.

Каждый компонент модели м. б. декомпозирован на другой диаграмме.

Каждая диаграмма иллюстрирует “внутреннее строение” блока на родительской диаграмме.

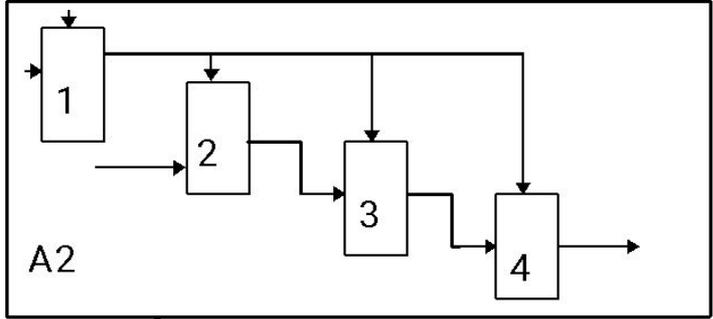
Построение SADT-модели *начинается* с представления всей системы в виде простейшего компонента — *одного блока и дуг*, изображающих интерфейсы с функциями вне системы.

Единственный блок отражает систему как единое целое - имя в блоке является общим.

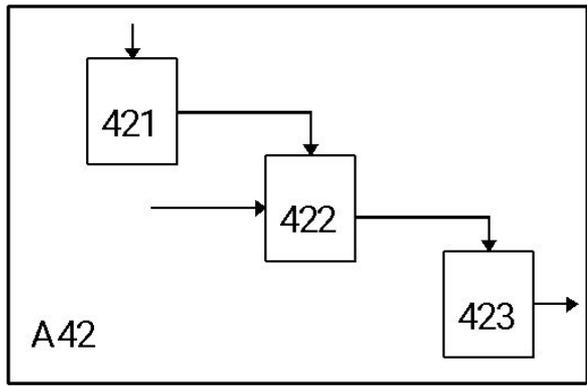
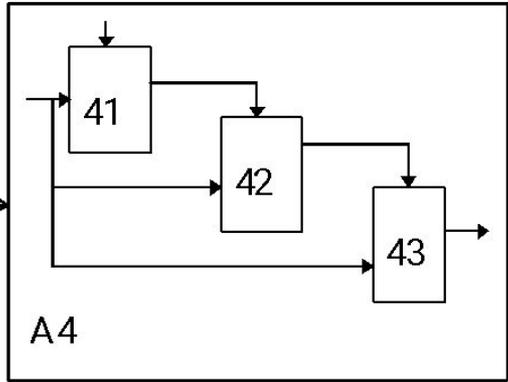


Общее представление

Более детальное представление



Верхняя диаграмма является "родителем" нижней диаграммы



Блок, представляющий систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами.

Эти блоки определяют основные подфункции исходной функции. Данная декомпозиция выявляет полный подбор подфункций, каждая из кот. показана как блок, границы кот. определены интерфейсными дугами.

Каждая из подфункций м. б. декомпозирована подобным образом в целях большей детализации.

Во всех случаях каждая подфункция м. содержать только те элементы, кот. входят в исходную функцию.

Кроме того, ФМ не м. опустить какие-либо элементы, т.е., как уже отмечалось, родительский блок и его интерфейсы обеспечивают контекст.

К нему нельзя ничего добавить, и из него не м.б. ничего удалено.

Модель SADT - серия диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, кот. изображены в виде блоков.

Детали каждого из основных блоков показаны в виде блоков на других диаграммах.

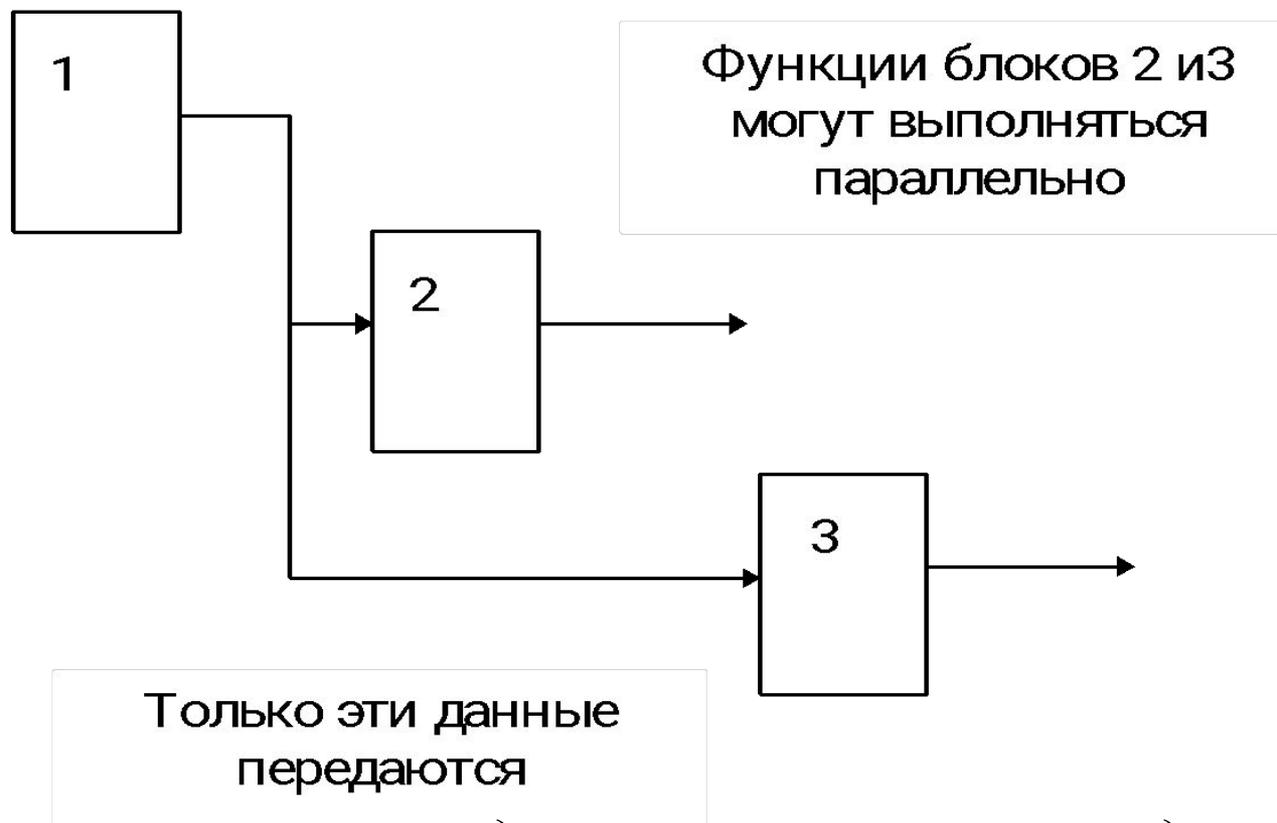
Каждая детальная диаграмма является декомпозицией блока из более общей диаграммы.

На каждом шаге декомпозиции более общая диаграмма наз. *родительской* для более детальной диаграммы.

Дуги, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются *точно теми же самыми*, что и дуги, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграмма изображают одну и ту же часть системы.

На рис. приведены различные варианты выполнения функций и соединения дуг с блоками.

Некоторые дуги присоединены к блокам диаграммы обоими концами, у других же один конец остается неприсоединенным.



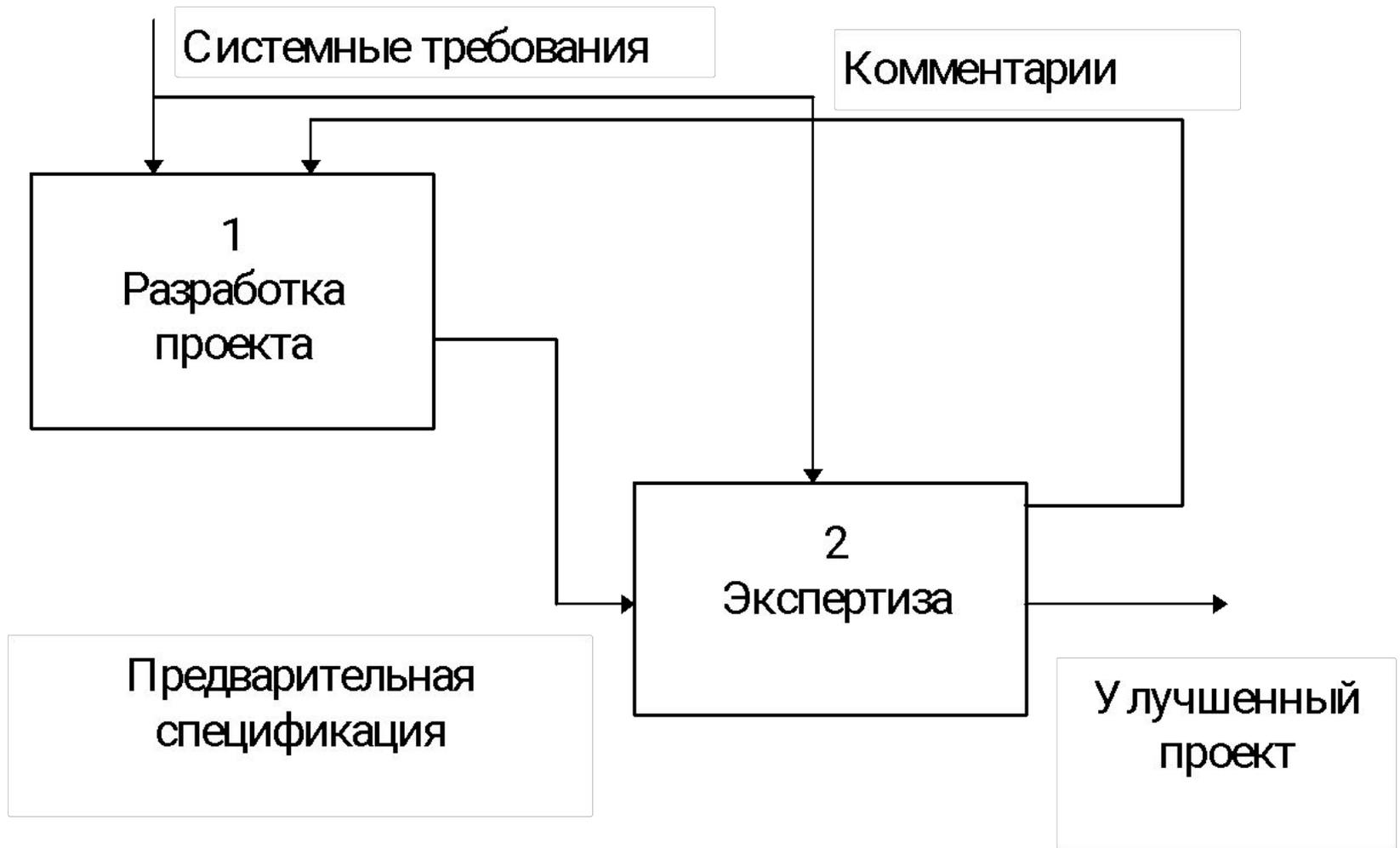
Неприсоединенные дуги соответствуют входам, управлениям и выходам родительского блока.

Источник или получатель этих пограничных дуг м. б. обнаружен только на родительской диаграмме.

Неприсоединенные концы д. соответствовать дугам на исходной диаграмме.

Все граничные дуги должны продолжаться на родительской диаграмме, чтобы она была полной и непротиворечивой.

На SADT-диаграммах не указаны явно ни последовательность, ни время.

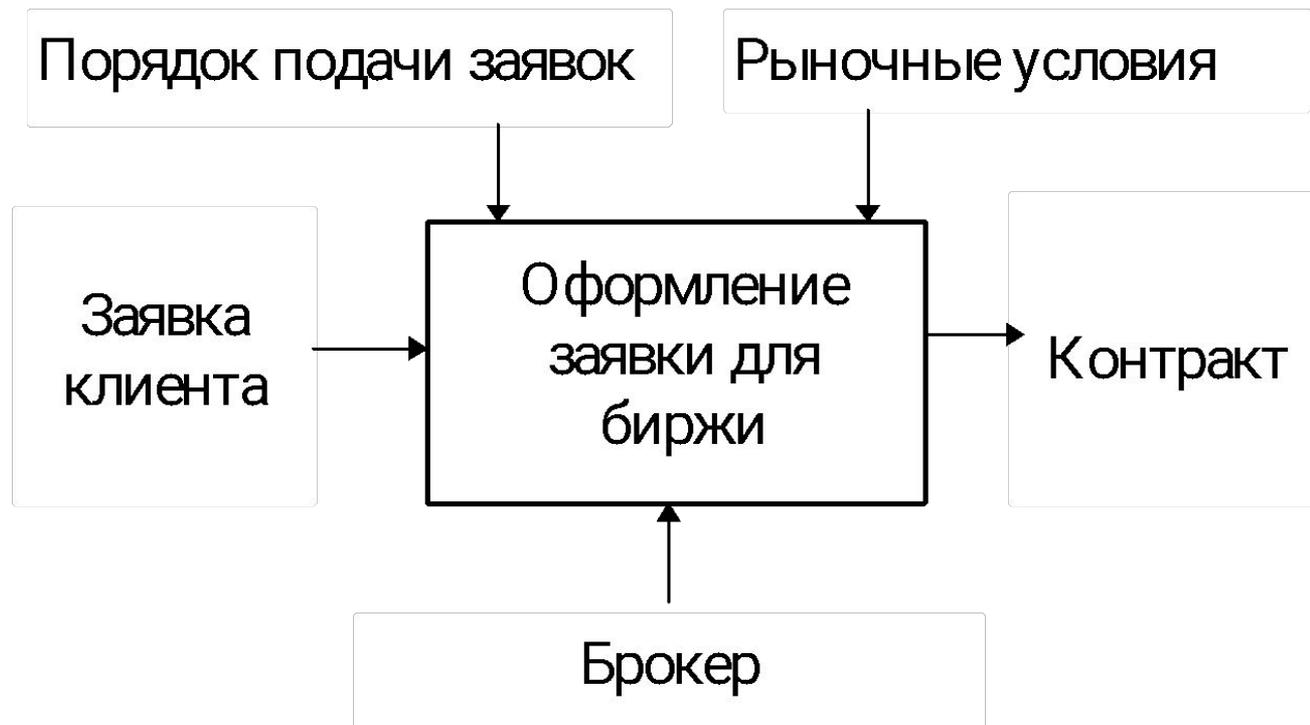


Обратные связи, итерации, продолжающиеся процессы и перекрывающиеся (по времени) функции м. б. изображены с помощью дуг.

Обратные связи м. выступать в виде комментариев, замечаний, исправлений и т.д.

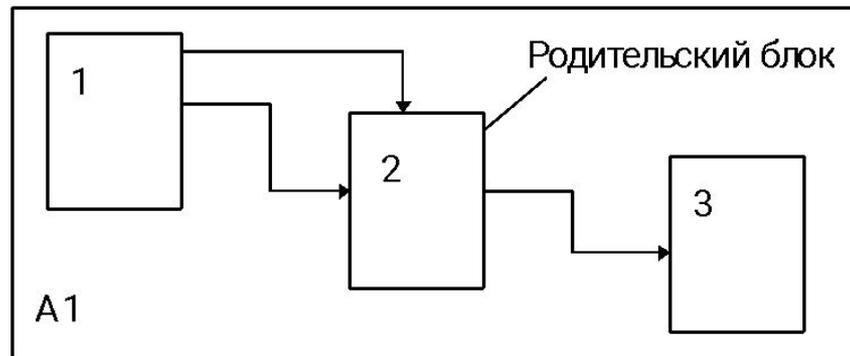
Механизмы (дуги с нижней стороны) показывают средства, с помощью кот. осуществляется выполнение функций.

Механизм м. б. человеком, компьютером или любым др. устройством, кот. помогает выполнять данную функцию.

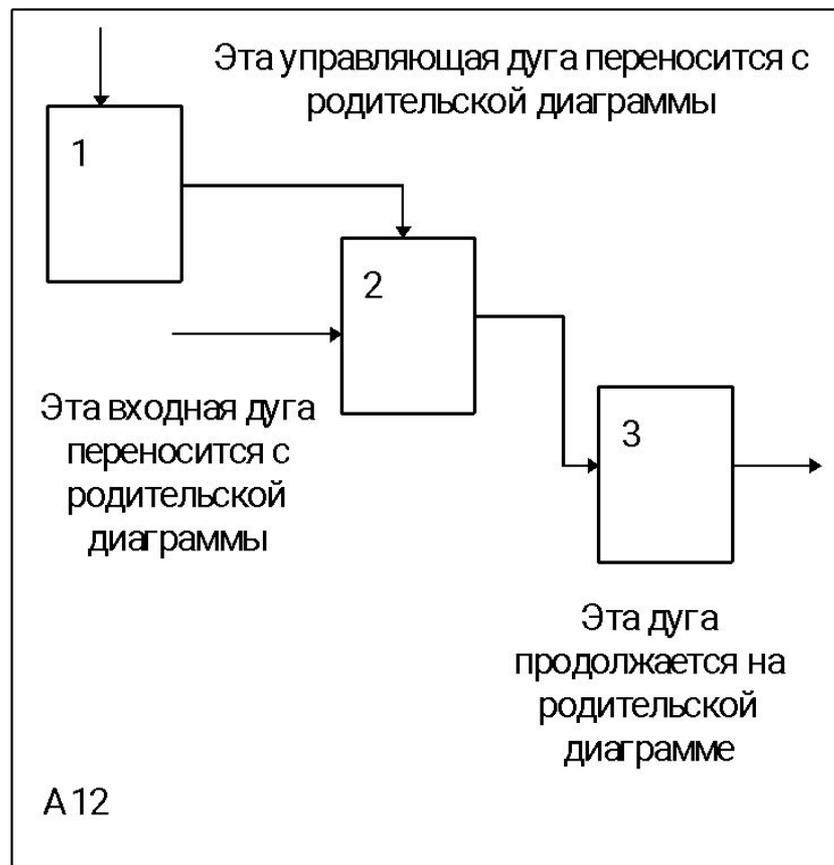


Каждый блок на диаграмме имеет свой номер.

Блок любой диаграммы м. б. далее описан диаграммой нижнего уровня, кот., в свою очередь, м. б. далее детализирована с помощью необходимого числа диаграмм. Т. о., формируется иерархия диаграмм.

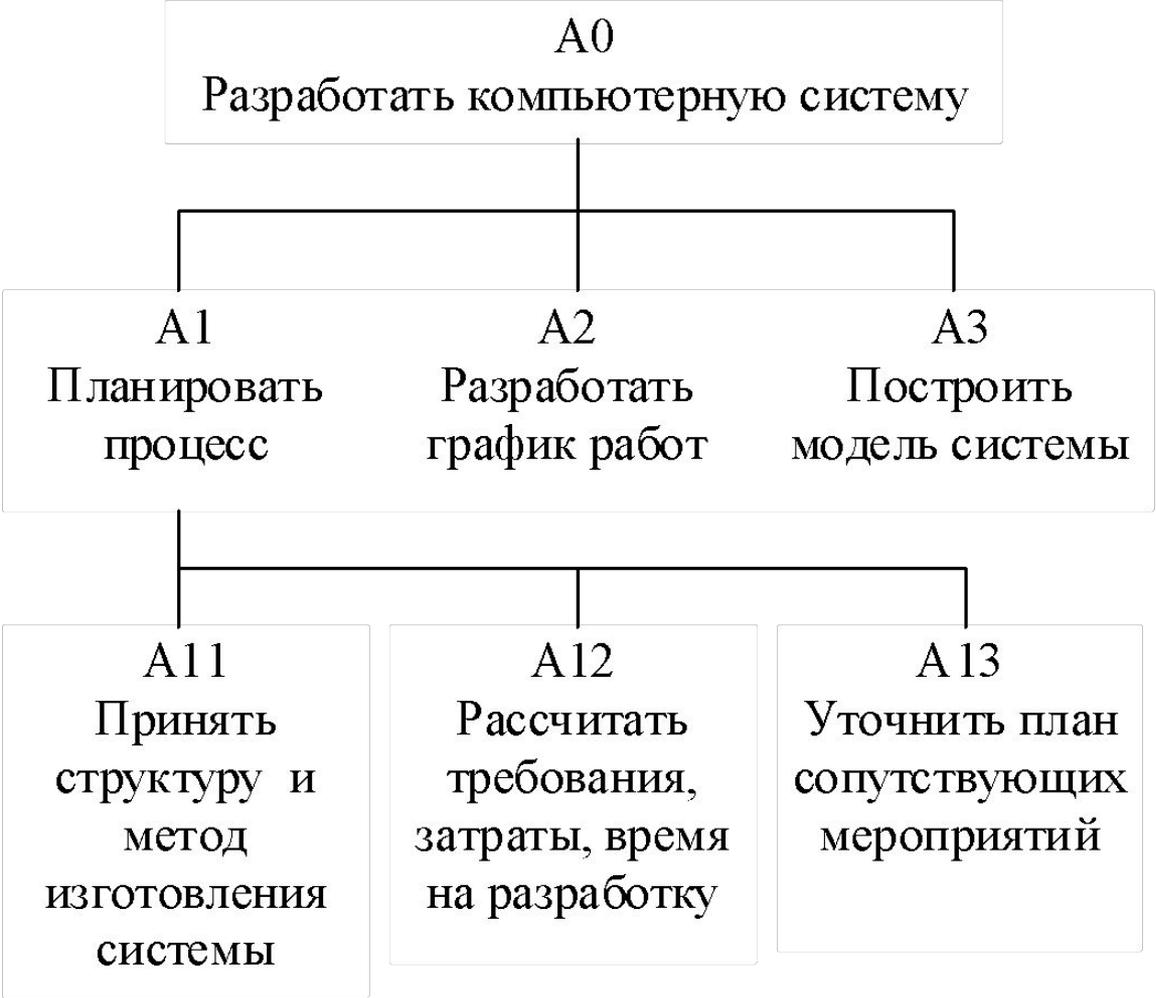


а



б

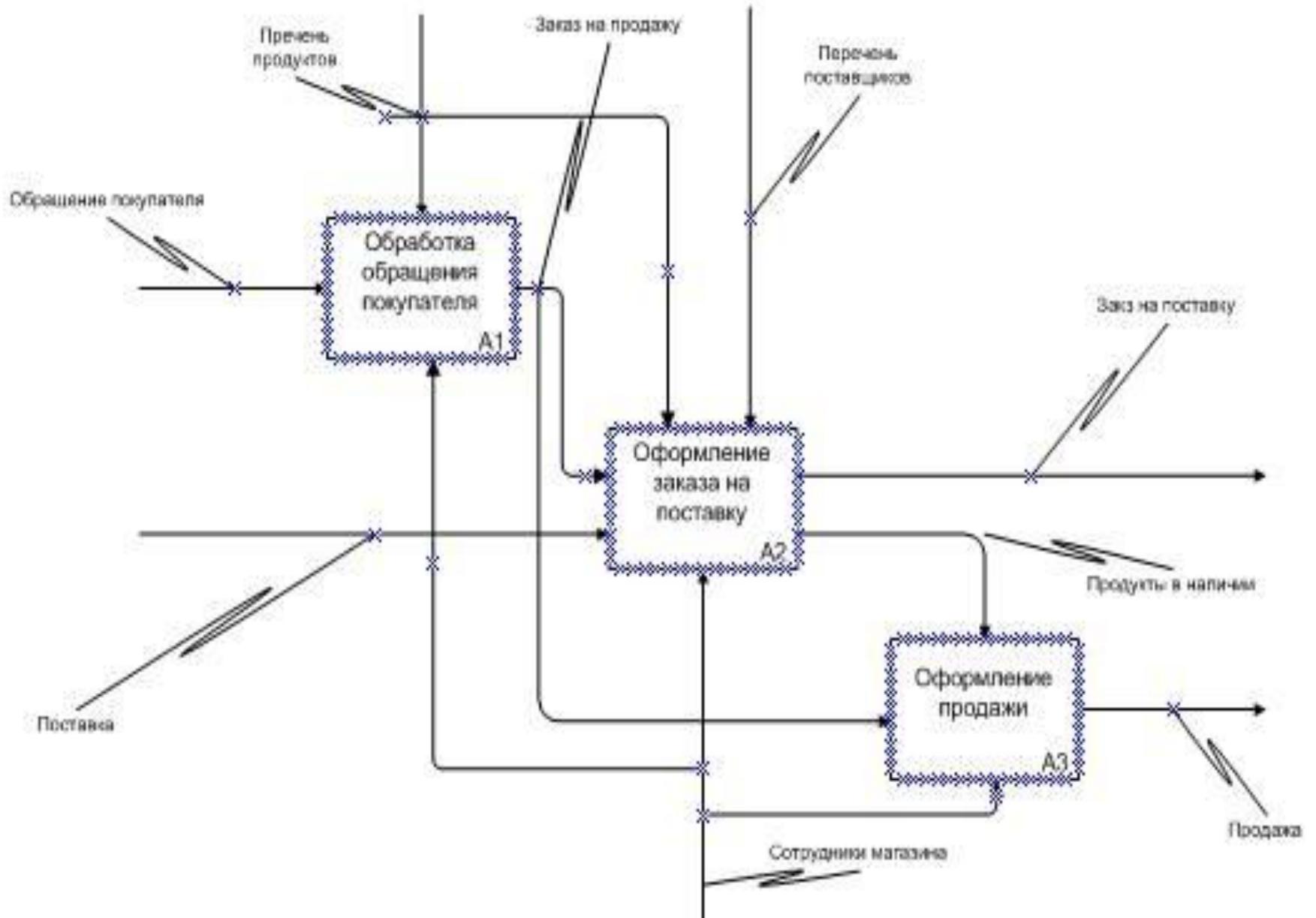
Для того чтобы указать положение любой диаграммы или блока в иерархии, используются номера диаграмм. Например, A21 является диаграммой, кот. детализирует блок 1 на диаграмме A2. Аналогично A2 детализирует блок 2 на диаграмме A0, кот. является самой верхней диаграммой модели. На рис. показано типичное дерево диаграмм.



Функциональная модель Магазина с точки зрения покупателя



Диаграмма второго уровня функциональной модели магазина



| Объект | Название | Определение | Описание |
|------------|--------------------------------|---|--|
| Функция А1 | Обработка обращения покупателя | Приведение желаний покупателя в соответствие с возможностями магазина. | Сопоставление списка продуктов, подготовленного покупателем с перечнем продуктов, представленных в магазине. |
| Функция А2 | Оформление заказа на поставку | Анализ наличия нужных покупателю продуктов и обеспечение поставки отсутствующих продуктов | Поиск поставщика, оформление заказа и поставки отсутствующих продуктов. |
| Функция А3 | Оформление продажи | Оформление отчета об обработке магазином обращении покупателя | Подготовка отчетного документа (накладной), подтверждающей передачу продуктов покупателю |

| | | | |
|--------------------------------|----------------------|--|---|
| Вход А1 | Обращение покупателя | Обращение покупателя, представленное в удобной для него форме. | Обращение может быть представлено в устной или письменной форме. |
| Вход А2 | Поставка | Документы, подтверждающие поставку продуктов, которых нет в наличии. | Документы, подтверждающие поставку продуктов, которых нет в наличии, но были заказаны покупателем |
| Вход А2 Вход А3 Выход А1 | Заказ на продажу | Заказ, содержащий продукты из перечня, предоставленного магазином | Обращение покупателя, приведенное в соответствие с перечнем продаваемых продуктов |
| Выход А3 | Продажа | Результат обработки обращения покупателя. | Документом, подтверждающим результат обработки, является накладная, содержащая перечень и характеристики приобретенных продуктов. |

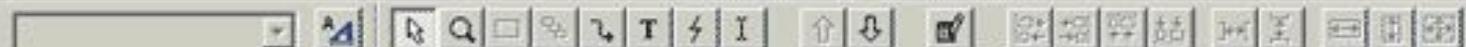
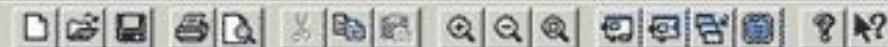
| | | | |
|-------------------------|----------------------|---|--|
| Выход А2 | Заказ на поставку | Обращение к поставщику о поставке отсутствующих на складе продуктов | Заказ поставщику на продукты, запрошенные покупателем, но отсутствующие на складе |
| Выход А2 Контроль А3 | Продукты в наличии | Перечень продуктов, подготовленных для покупателя | Перечень продуктов, как со склада, так и заказанных у поставщиков |
| Контроль А1, А2 | Перечень продуктов | Перечень продуктов, предоставленных на выбор покупателю. | Перечень может включать продукты, отсутствующие в наличии (на складе). Такие продукты могут быть получены от поставщиков, с которыми работает магазин. |
| Контроль А2 | Перечень поставщиков | Данные о поставщиках, с которыми работает магазин | Поставщики, к которым магазин обращается при исчерпании продуктов на складе. |

Меха-
низм А1,
А2, А3

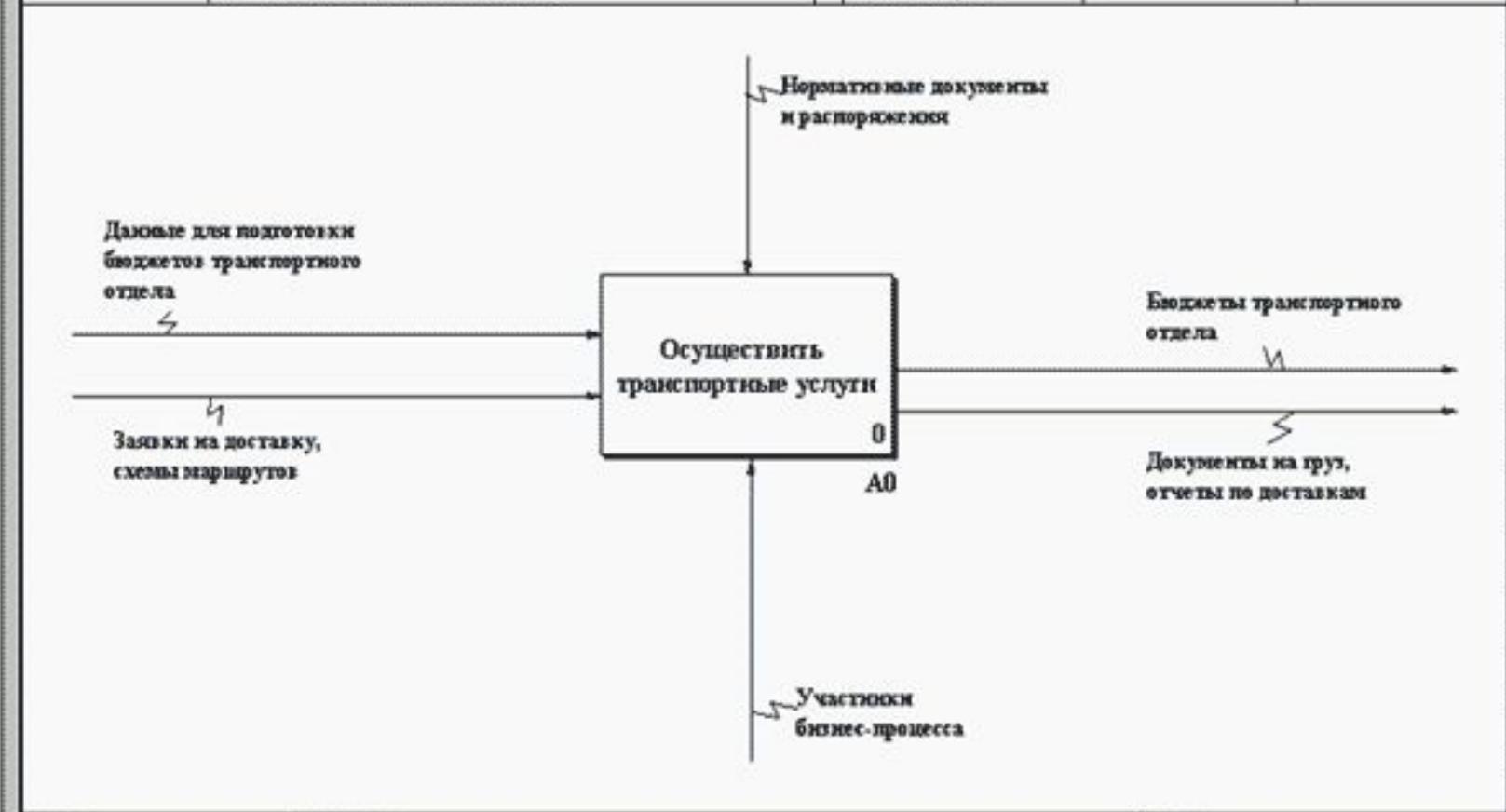
Сотрудники
магазина

Сотрудники, прямо или
косвенно участвующие в
обработке обращения
покупателя.

Количество и состав
сотрудников зависит от
количества покупателей,
ассортимента продуктов и от
количества поставщиков.

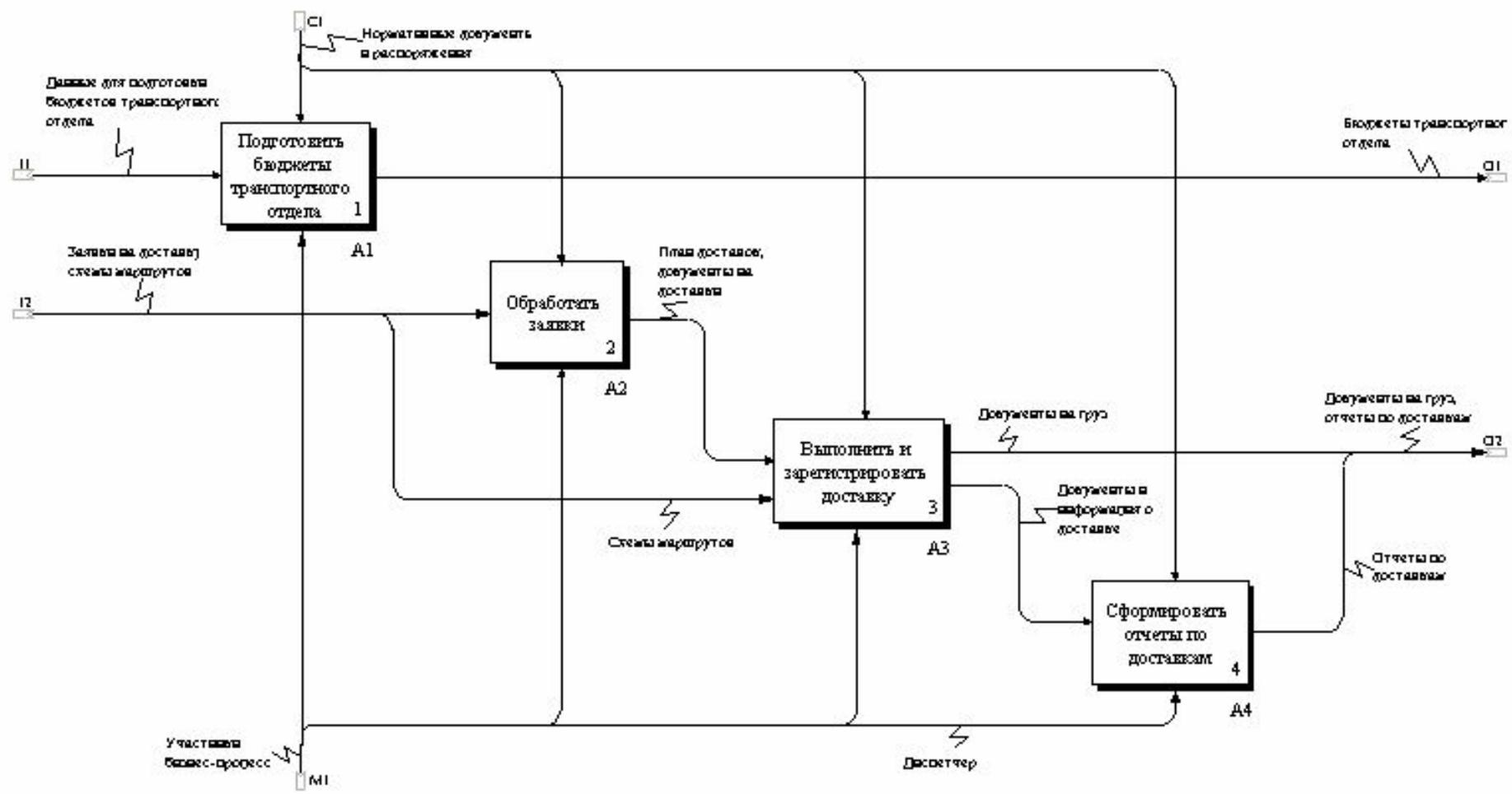


| | | | | | | |
|---------------|----------------------------------|-----------------------|----------------|----------|------|----------------------|
| ИСПОЛЬЗУЮТСЯ: | АВТОР: <i>Аналитик</i> | ДАТА: 15 ян 2002 | РАБОЧАЯ ВЕРСИЯ | ЧИТАТЕЛЬ | ДАТА | КОНТЕКСТ: TOP |
| | ПРОЕКТ: <i>Торговая компания</i> | ПЕРЕСМОТР: 17 ян 2002 | ЭСКИЗ | | | |
| | ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10 | | РЕКОМЕНДОВАНО | | | |
| | | | ПУБЛИКАЦИЯ | | | |

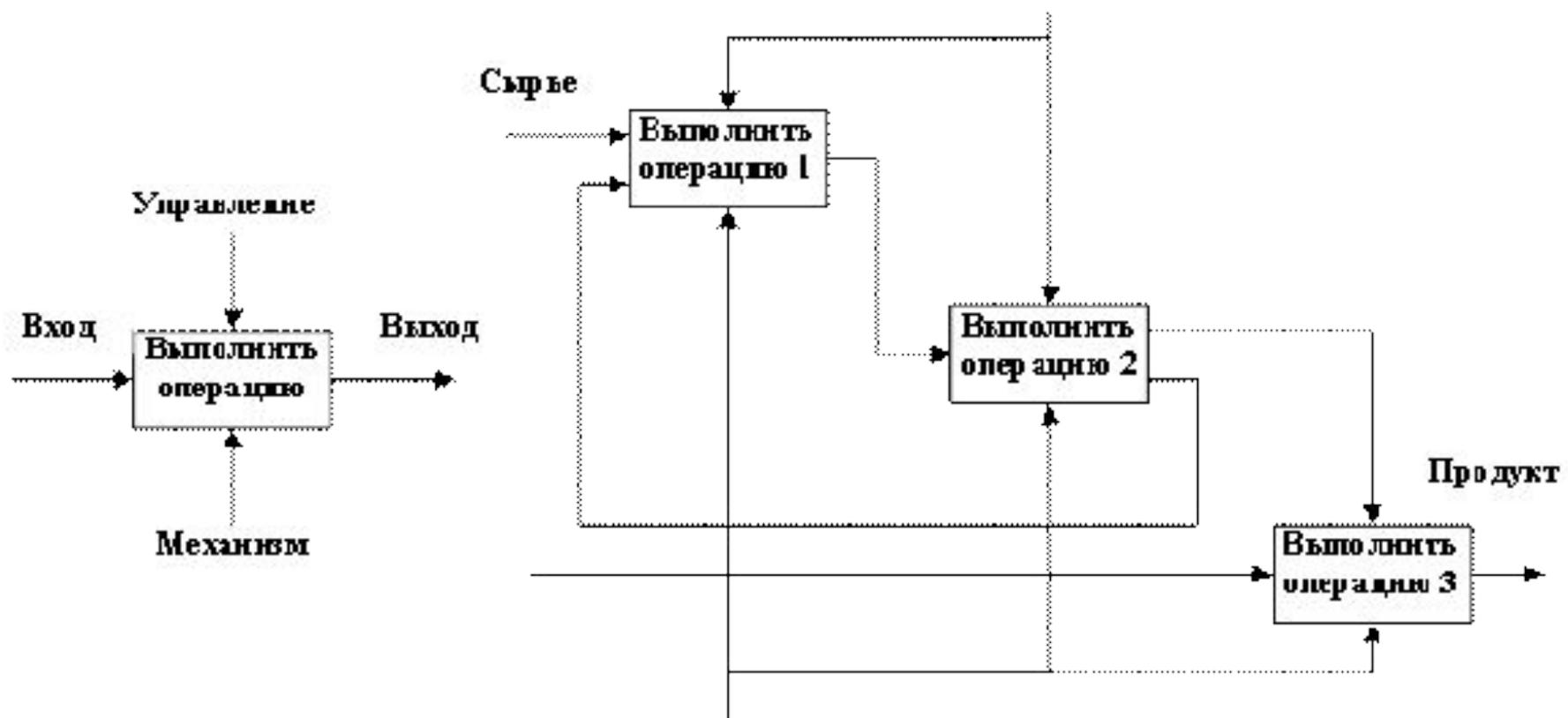


| | | | |
|--------------|--|----------------|---------|
| УЗЕЛ: A-0 | НАЗВАНИЕ: Осуществить транспортные услуги | НОМЕР: | Стр.: 1 |
|--------------|--|----------------|---------|

| | | | | | | | |
|---------------|---------------------------------|------------------------|----------------|----------|------|-----------|--|
| ИСПОЛЬЗУЮТСЯ: | АВТОР: Аналитик | ДАТА: 15 янв 2002 | РАБОЧАЯ ВЕРСИЯ | ЧИТАТЕЛЬ | ДАТА | КОНТЕКСТ: | |
| | ПРОЕКТ: Торговая компания | ПЕРЕСМОТР: 17 янв 2002 | ЭСКИЗ | | | | |
| | ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9 10 | | РЕКОМЕНДОВАНО | | | | |
| | | | ПУБЛИКАЦИЯ | | | | |



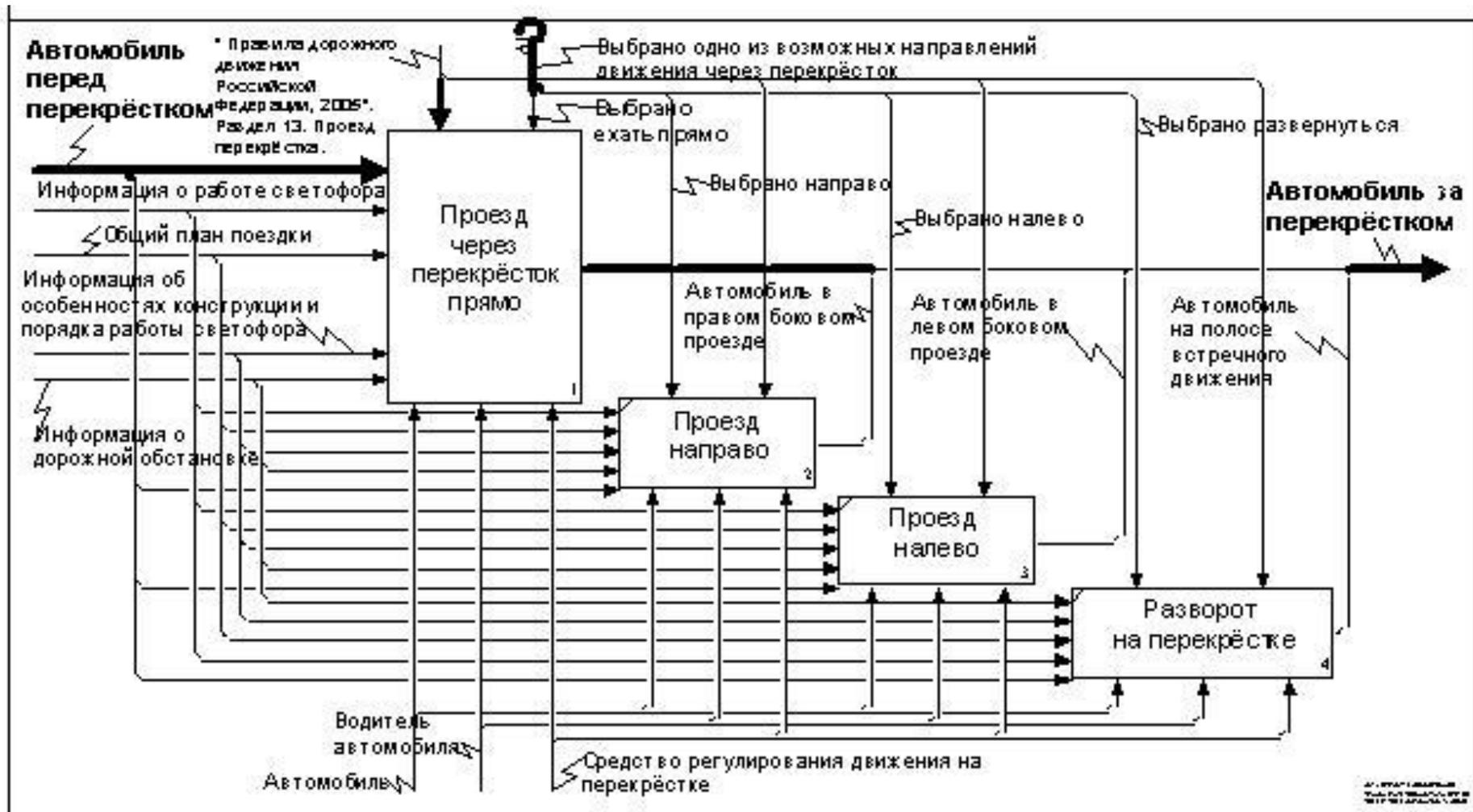
| | | |
|-------|---------------------------------|--------|
| УЗЕЛ: | НАЗВАНИЕ: | НОМЕР: |
| A0 | Осуществить транспортные услуги | |



| | | | | | | |
|--|--|------------------|-------------|--------|------|---------------------|
| USED AT: Или инструкции ФМ и DEF0, DFD и DEF3 | AUTHOR: © Дубининский В И, 2008 г | DATE: 17.01.2008 | WORKING | READER | DATE | CONTEXT: TOP |
| | PROJECT: Технологическая разработка для стандарта | REV: 10.02.2008 | DRAFT | | | |
| | | | RECOMMENDED | | | |
| | | | PUBLICATION | | | |



| | | |
|--------------|---|---|
| CODE: A-0 | TITLE: Проезд перекрёстка автодороги | NUMBER: "Содержание и структура документации стандарта DEF0- DEF3 Проект 10.02.08" |
|--------------|---|---|



| | | |
|----------|--|--|
| CODE: 40 | TITLE: Проезд через средства автоматизации | NUMBER: 1 С: Осиновский И.И. Союзгипротранспорт/Работы по проекту МДЕР - ДФЗ П.р.р.р.р.р.р. |
|----------|--|--|

Типы связей между функциями

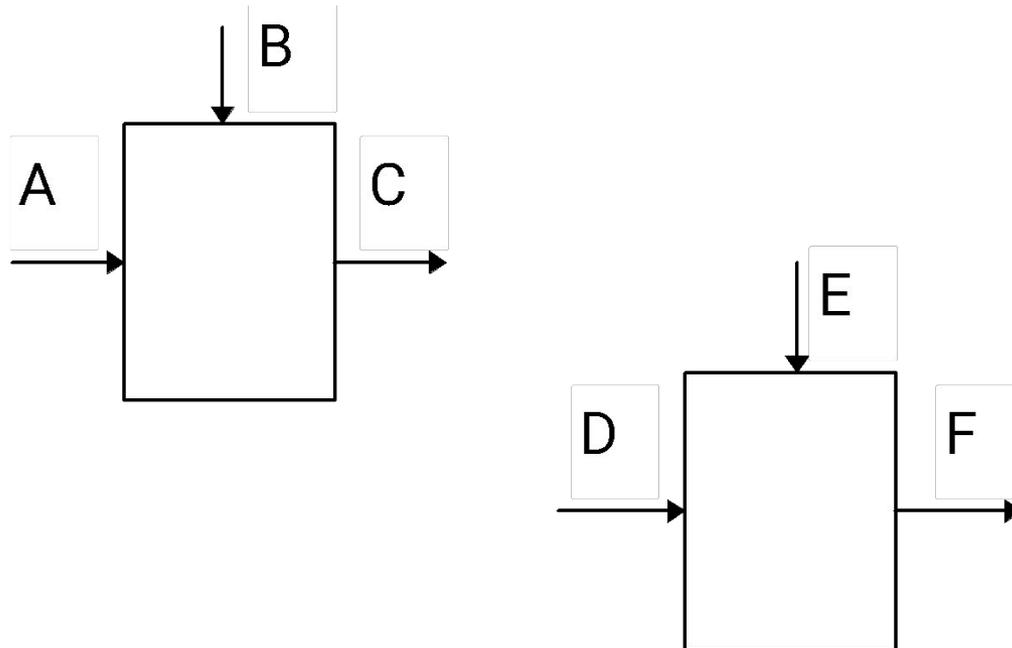
Одним из важных моментов при проектировании ИС с помощью методологии SADT является точная согласованность типов связей между функциями.

Типы связи (в порядке возрастания их относительной значимости):

- случайная;
- логическая;
- временная;
- процедурная;
- коммуникационная;
- последовательная;
- функциональная.

Случайная связь — конкретная связь между функциями мала или полностью отсутствует.

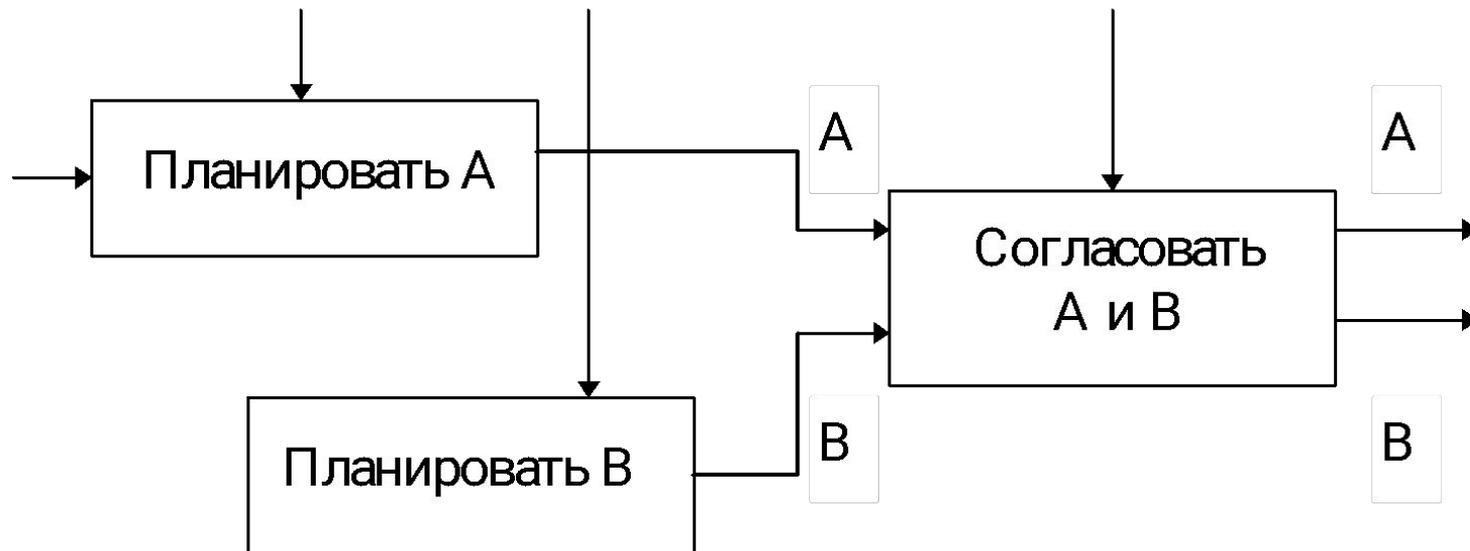
Пример: ситуация, когда имена данных на SADT-дугах в одной диаграмме имеют малую связь друг с другом.



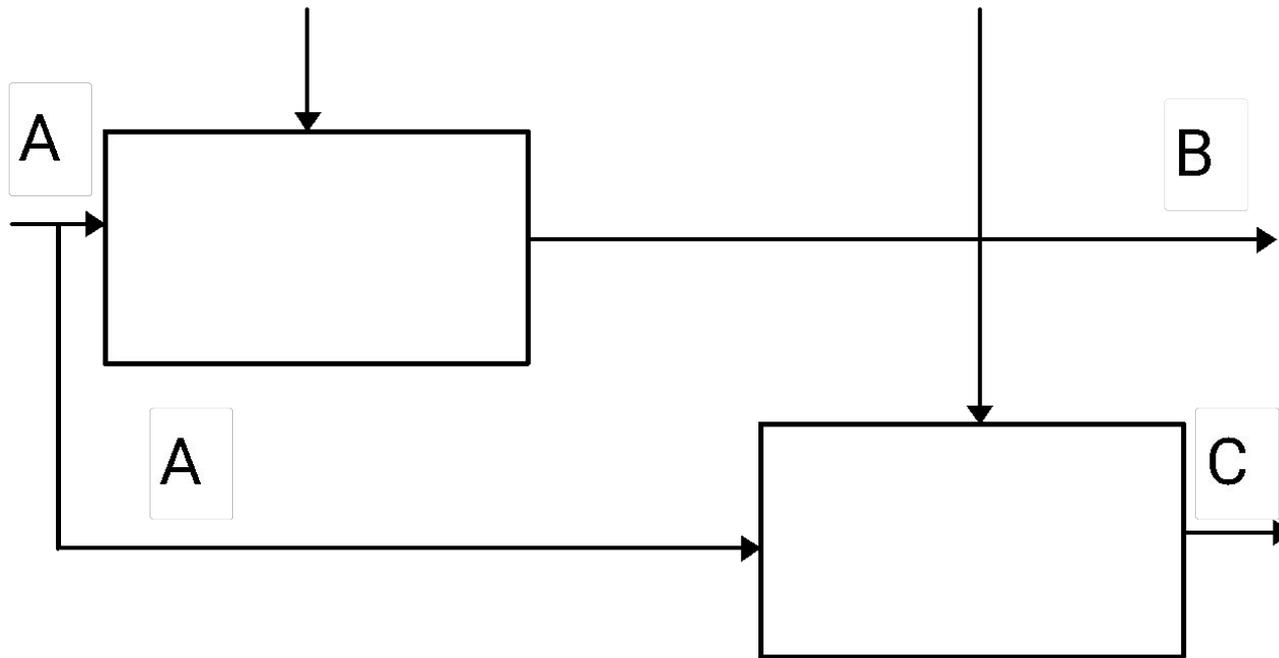
Логическая связь — данные и функции собираются вместе на диаграмме благодаря тому, что они попадают в общий класс или набор элементов, но необходимых функциональных отношений между ними не обнаруживается.

Временная связь — представляет функции, связанные во времени, когда данные используются одновременно или функции включаются параллельно, а не последовательно.

Процедурная связь (рис.) — функции сгруппированы вместе благодаря тому, что они выполняются в течение одной и той же части цикла или процесса.

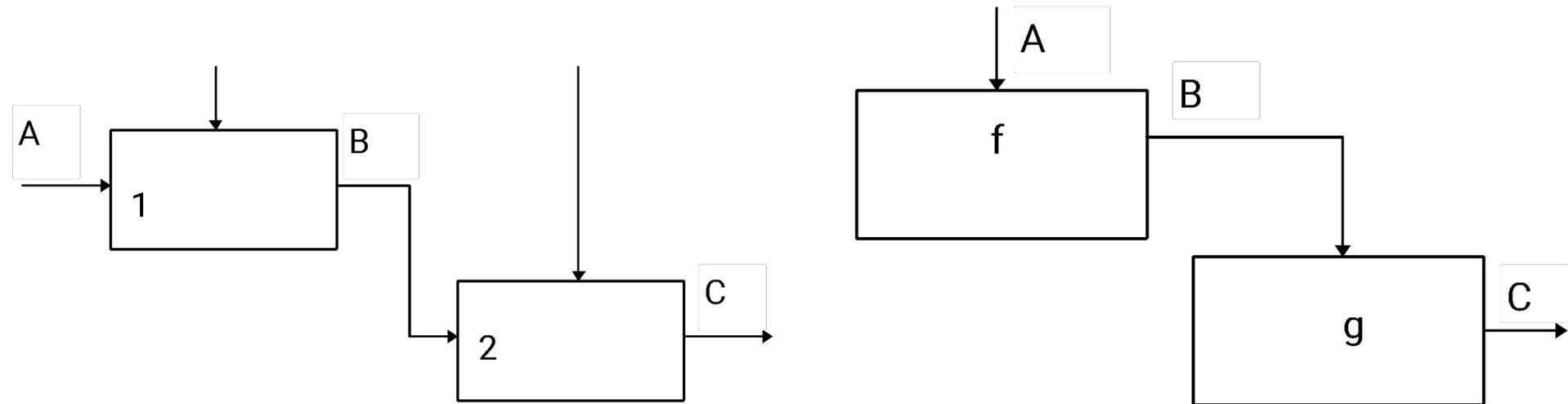


Коммуникационная связь — функции группируются благодаря тому, что они используют одни и те же входные данные и/или производят одни и те же выходные данные (рис.).



Последовательная связь — выход одной функции служит входными данными для следующей функции. Связь между элементами на диаграмме является более тесной, чем в рассмотренных выше случаях, поскольку моделируются причинно-следственные зависимости (рис.).

Функциональная связь — все элементы функции влияют на выполнение одной и только одной функции. Диаграмма, являющаяся чисто функциональной, не содержит чужеродных элементов, относящихся к последовательному или более слабому типу связи. Одним из способов определения функционально-связанных диаграмм является рассмотрение 2-х блоков, связанных через управляющие дуги (рис.).



Моделирование данных. Основные понятия.

Цель - обеспечение разработчика ИС концептуальной схемой БД в форме одной модели или нескольких локальных моделей, кот. относительно легко м. б. отображены в любую систему БД.

Наиболее распространенным средством моделирования данных являются диаграммы “сущность-связь” (ERD).

Базовые понятия ERD:

Сущность (Entity) — реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области.

Каждая сущность д. обладать уникальным идентификатором.

Каждый экземпляр сущности д. однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности.

Каждая сущность д. обладать некоторыми свойствами:

- иметь уникальное имя;
- обладать одним или несколькими **атрибутами**, кот. либо принадлежат сущности, либо наследуются через связь;
- обладать одним или несколькими атрибутами, кот. однозначно идентифицируют каждый экземпляр сущности.

Каждая сущность м. обладать любым количеством **связей** с другими сущностями модели.

СВЯЗЬ - поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области.

Связь - ассоциация между сущностями, при кот. каждый экземпляр одной сущности ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, и наоборот.

Атрибут - любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности.

Атрибут - тип характеристик или свойств, ассоциированных со множеством реальных или абстрактных объектов (людей, мест, событий, состояний, идей, предметов и т.д.).

Экземпляр атрибута - определенная характеристика отдельного элемента множества. Экземпляр атрибута определяется типом характеристики и ее значением, наз. значением атрибута.

В ER-модели атрибуты ассоциируются с конкретными сущностями. Т. о., экземпляр сущности д. обладать единственным определенным значением для ассоциированного атрибута.

Пример: Предметная область - компания по торговле автомобилями. Ниже приведены выдержки из интервью, проведенного с персоналом компании.

Главный менеджер: одна из основных обязанностей - содержание автомобильного имущества. Он д. знать, сколько заплачено за машины и каковы накладные расходы. Обладая этой информацией, он м. установить нижнюю цену, за кот. мог бы продать данный экземпляр. Кроме того, он несет ответственность за продавцов и ему нужно знать, кто что продает и сколько машин продал каждый из них.

Продавец: ему нужно знать, какую цену запрашивать и какова нижняя цена, за кот. м. совершить сделку. Кроме того, ему нужна основная информация о машинах: год выпуска, марка, модель и т. п.

Администратор: его задача сводится к составлению контрактов, для чего нужна информация о покупателе, автомашине и продавце, поскольку именно контракты приносят продавцам вознаграждения за продажи.

Первый шаг моделирования - извлечение информации из интервью и выделение сущностей.

Сущности м. б. идентифицированы с главным менеджером - автомашины и продавцы.

Продавцу важны автомашины и связанные с их продажей данные.

Для администратора важны покупатели, автомашины, продавцы и контракты. Исходя из этого **выделяются четыре сущности** (автомашина, продавец, покупатель, контракт), кот. изображаются на диаграмме так, как они представлены на рис.



Второй шаг моделирования - идентификация связей.

Связь - ассоциация между сущностями, при кот., к. п., каждый экземпляр одной сущности, наз. *родительской* сущностью, ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, наз. *сущностью-потомком*, а каждый экземпляр сущности-потомка ассоциирован в точности с одним экземпляром сущности-родителя. Т. о., экземпляр сущности-потомка м. существовать только при существовании сущности-родителя.

Связи м. даваться имя, выражаемое грамматическим оборотом глагола и помещаемое возле линии связи. Имя каждой связи между двумя данными сущностями д.б. уникальным, но имена связей в модели не обязаны б. уникальными.

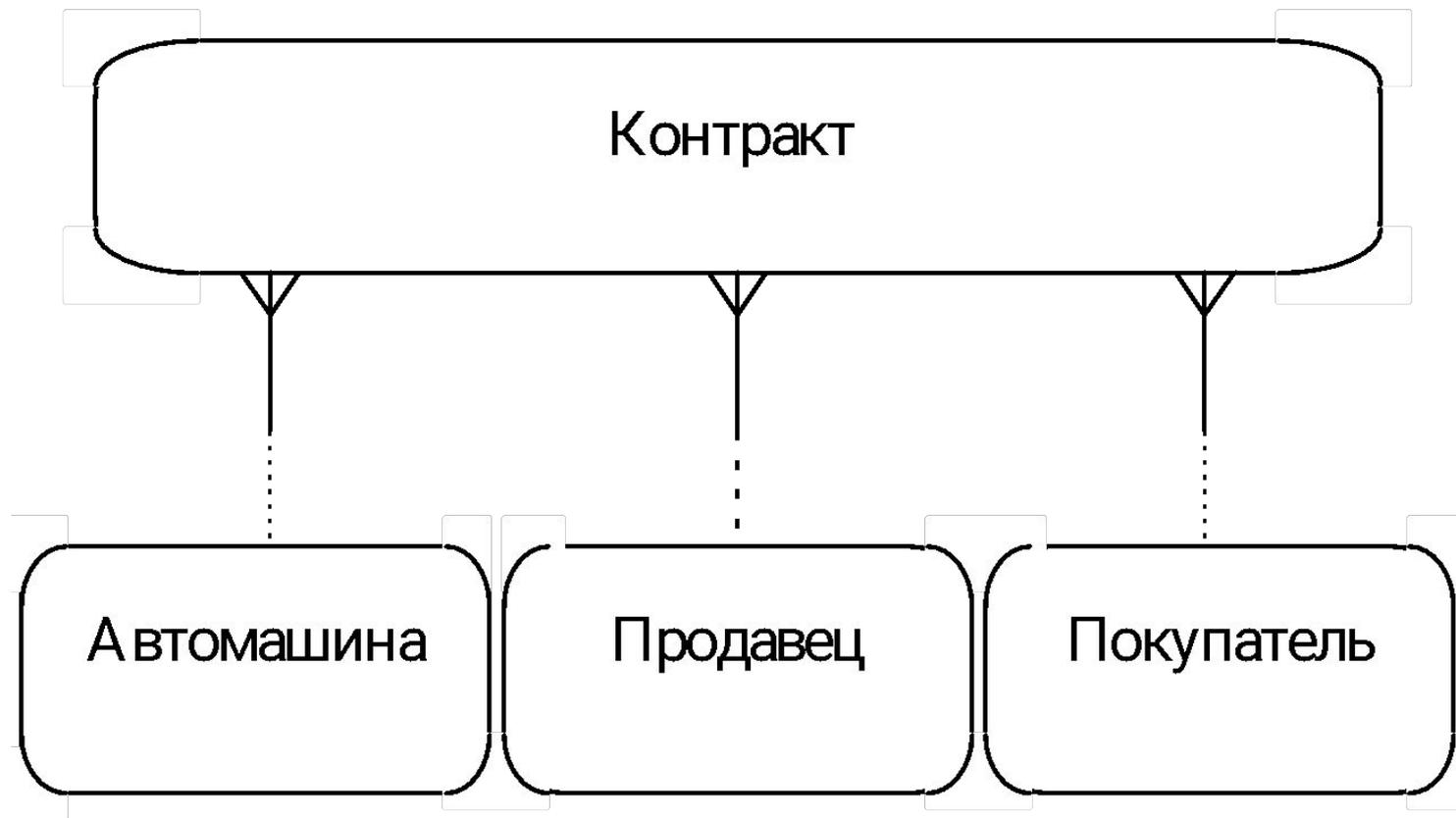
Имя связи всегда формируется с точки зрения родителя, так что м. б. образовано предложение соединением имени сущности-родителя, имени связи, выражения степени и имени сущности-потомка.



Например, связь продавца с контрактом м. б. выражена с. о.:

- продавец м. получить вознаграждение за один или более контрактов;
- контракт д. б. инициирован ровно одним продавцом.

Описав также связи остальных сущностей, получим следующую схему:



Третий шаг моделирования - идентификация атрибутов.

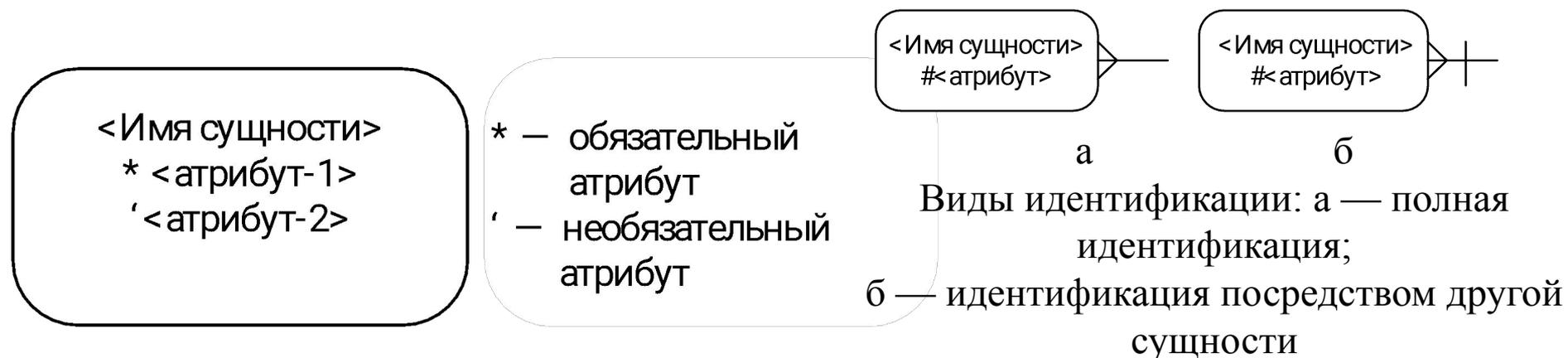
Атрибут м. б. либо обязательным, либо необязательным (рис.).

Обязательность означает, что атрибут не м. принимать неопределенных значений.

Атрибут м. б. либо описательным (обычным дескриптором сущности), либо входить в состав уникального идентификатора (первичного ключа).

Уникальный идентификатор - атрибут или совокупность атрибутов и/или связей, предназначенные для уникальной идентификации каждого экземпляра данного типа сущности.

В случае полной идентификации каждый экземпляр данного типа сущности полностью идентифицируется своими собственными ключевыми атрибутами, в противном случае в его идентификации участвуют также атрибуты другой сущности-родителя (рис.).



Каждый атрибут идентифицируется уникальным именем, выражаемым грамматическим оборотом существительного, описывающим представляемую атрибутом характеристику. Атрибуты изображаются в виде списка имен внутри блока ассоциированной сущности, причем каждый атрибут занимает отдельную строку. Атрибуты, определяющие первичный ключ, размещаются наверху списка и выделяются знаком “#”.

Каждая сущность д. обладать хотя бы одним возможным ключом.

Возможный ключ сущности - один или несколько атрибутов, чьи значения однозначно определяют каждый экземпляр сущности.

При существовании нескольких возможных ключей один из них обозначается в качестве первичного ключа, а остальные - как альтернативные ключи.

