ПМ3
МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ ПО

Лекция 4-3. ООАП и структурный системный анализ
Методология объектно-ориентированного анализа и проектирования (ООАП)
Нотации структурного системного анализа: IDEF0 и ДПД (DFD)

Структура ЖЦ ПО в RUP

- Полный ЖЦ ПС разбивается на циклы развития, в ходе каждого из которых разрабатывается новое поколение или версия продукта.
- Каждый цикл развития состоит из четырех последовательных фаз
  - Первый цикл выполнения этих фаз – начальный цикл
  - Последующие циклы развития – эволюционные циклы
- Каждая фаза содержит все аспекты разработки и повторяет все основные потоки работ
- Акценты при выполнении деятельностей потоков работ зависят от фазы процесса
- Фазы завершаются главными вехами

Методология объектно-ориентированного анализа и проектирования (ООАП)

Методология объектно-ориентированного анализа и проектирования (ООАП)

- Необходимость анализа предметной области до начала написания программы была осознана при разработке масштабных проектов.
- Так, процесс создания баз данных существенно отличается от написания программного кода для решения вычислительной задачи. При проектировании базы данных возникает необходимость в предварительной разработке концептуальной схемы или модели, которая отражала бы общие взаимосвязи предметной области и особенности организации соответствующей информации.
- Предметная область (domain) - часть реального мира, которая имеет существенное значение или непосредственное отношение к процессу функционирования программы. Другими словами, предметная область включает в себя только те объекты и взаимосвязи между ними, которые необходимы для описания требований и условий решения конкретной задачи.

Методология ООАП

- Выделение исходных или базовых компонентов предметной области, требуемых для решения той или иной задачи, представляет нетривиальную проблему. Сложность данной проблемы проявляется в неформальном характере процедур или правил, которые можно применять для этой цели. Более того, эта работа должна выполняться совместно со специалистами или экспертами, хорошо знающими предметную область. Например, если разрабатывается база данных для обслуживания пассажиров крупного аэропорта, то в проектировании концептуальной схемы базы данных должны принимать участие штатные сотрудники аэропорта. Эти сотрудники хорошо знают весь процесс обслуживания пассажиров или данную предметную область. Сложность моделирования предметной области и разработки корпоративных информационных систем привело к появлению новой методологии объектно-ориентированный анализ и проектирование.
- Объектно-ориентированный анализ и проектирование (ООАП, Object-Oriented Analysis/Design) - технология разработки программных систем, в основу которых положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов.

Методология ООАП

- Методология ООАП тесно связана с концепцией автоматизированной разработки программного обеспечения (Computer Aided Software Engineering, CASE). К первым CASE-средствам отнеслись с определенной настороженностью. Со временем появились как восторженные отзывы об их применении, так и критические оценки их возможностей. Причин для столь противоречивых мнений было несколько.
- 1. Ранние CASE-средства были простой надстройкой над системой управления базами данных (СУБД). Визуализация процесса разработки концептуальной схемы БД имеет немаловажное значение, тем не менее, она не решает проблем создания приложений других типов.
- 2. В CASE-средствах реализована графическая нотация. Если языки программирования имеют строгий синтаксис, то попытки предложить подходящий синтаксис для визуального представления концептуальных схем БД, были восприняты далеко не однозначно.
- На этом фоне разработка и стандартизация унифицированного языка моделирования UML вызвала воодушевление у всего сообщества корпоративных программистов.

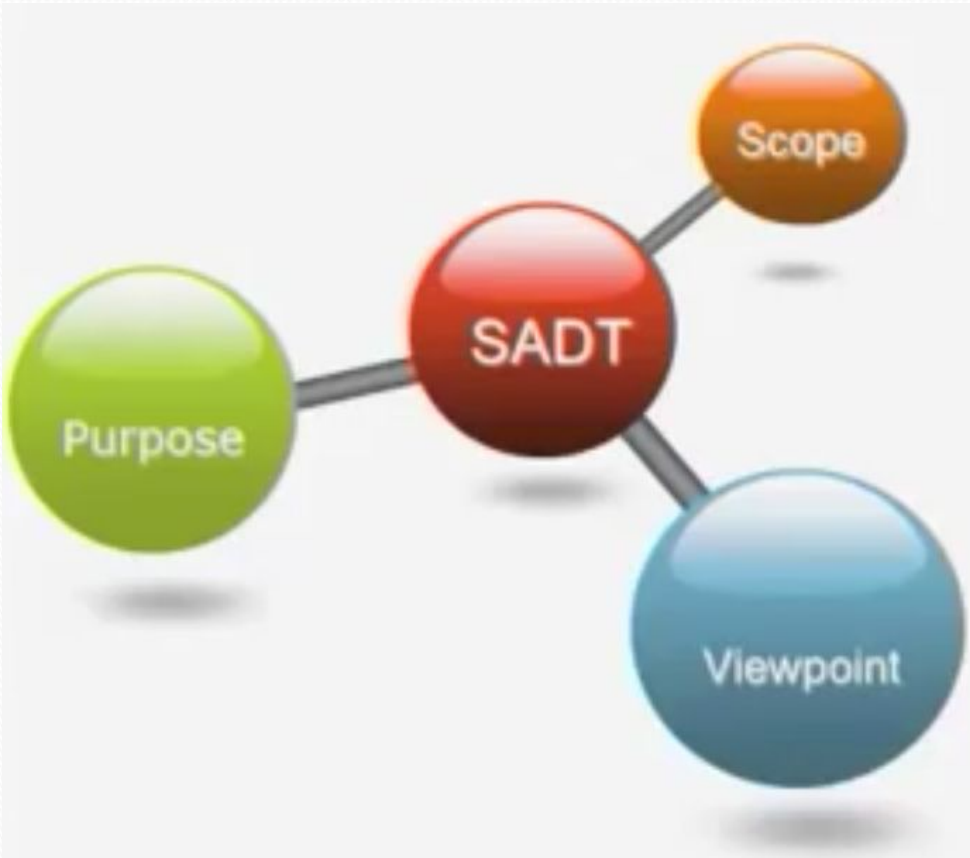
В рамках ООАП исторически рассматривались три графических нотации структурного системного анализа

1. диаграммы "сущность-связь" (Entity-Relationship Diagrams, ERD),
2. диаграммы функционального моделирования (Structured Analysis and Design Technique, SADT),
3. диаграммы потоков данных (Data Flow Diagrams, DFD).

Методология структурного анализа и проектирования

- В 70-е гг. XX века предложена *Дугласом Россом* (Douglas Ross) – методология **SADT** *(Structured Analysis and Design Technique)*. Основная идея данной методологии – построение древовидной иерархической модели предприятия.
- В начале *1990-х* на основе SADT принят стандарт моделирования бизнес-процессов **IDEF0**, являющийся одним из 14 стандартов линейки *IDEF – Integration Definition for Functional Modeling*.
- Положения методологии зафиксированы в разработанном в США стандарте IDEF0 (В СНГ – *РД IDEF0 – 2000*)

SADT.Основные понятия и правила



Viewpoint (Точка зрения)

указание на должностное лицо, с позиции которого разрабатывается модель

Purpose (Цель)

выражает причину создания модели

Scope (Область моделирования)

описывает круг функций системы

SADT - Structured Analysis and Design Technique, методология Росса

Техники структурных диаграмм

Средства анализа

Средства проектирования

Средства анализа

- Диаграммы потоков данных
- Диаграммы потоков управления
- Таблицы, деревья решений
- Матрицы
- Диаграммы зависимости
- Диаграммы декомпозиции
- SADT-диаграммы

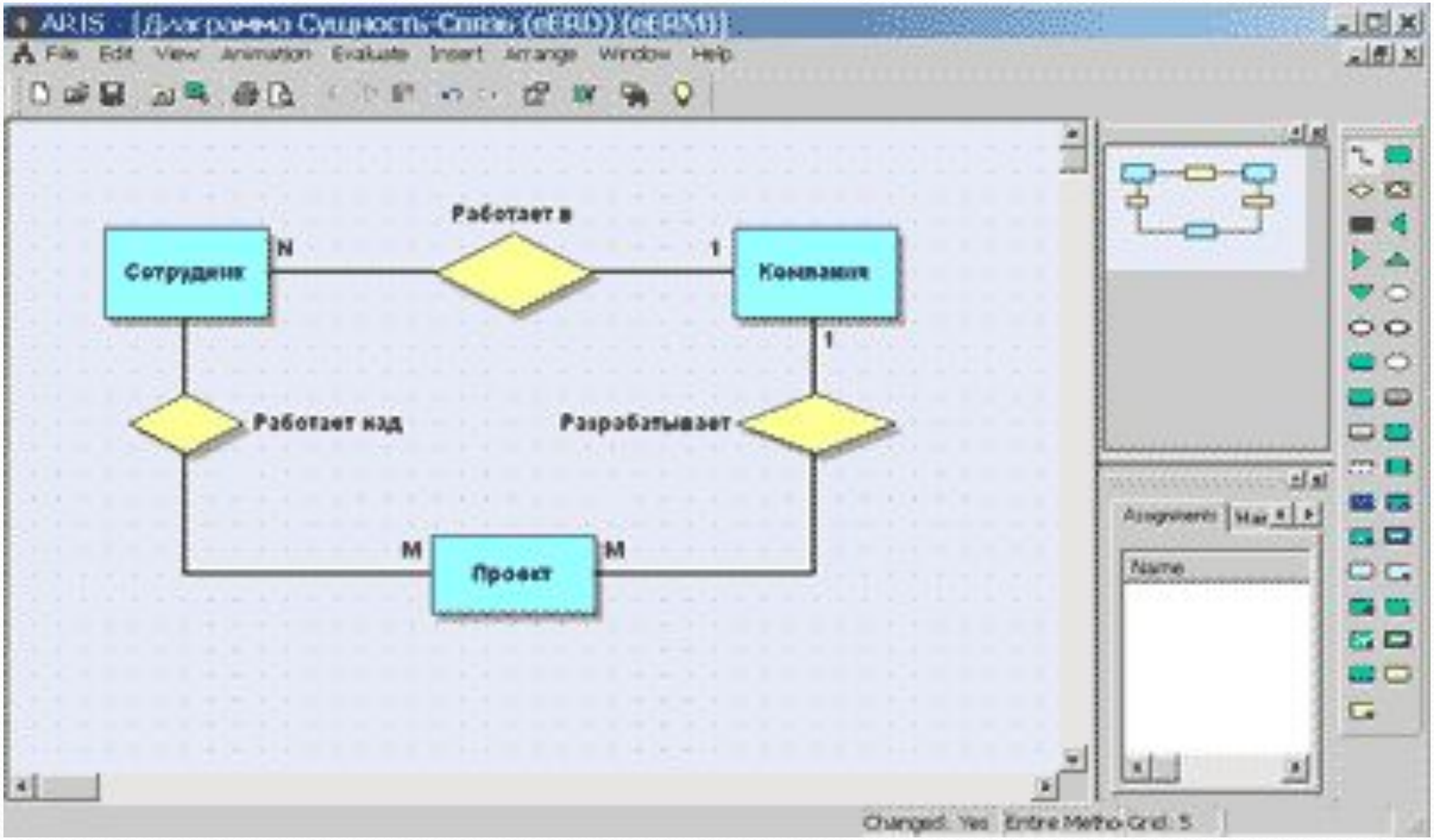
Средства проектирования

- Структурные карты
- Диаграммы деятельности
- Диаграммы Варнье-Орра
- Диаграммы переходов состояний
- Языки проектирования спецификаций
- Блок-схемы
- Схемы-экранов
- Диаграммы сущность-связь

Диаграммы "сущность-связь"

- Диаграммы "сущность-связь" (ERD) предназначены для графического представления моделей данных разрабатываемой программной системы и предлагают набор стандартных обозначений для определения данных и отношений между ними. С помощью этого вида диаграмм можно описать отдельные компоненты концептуальной модели данных и совокупность взаимосвязей между ними.
- Основными понятиями данной нотации являются понятия сущности и связи. При этом под сущностью (entity) понимается произвольное множество реальных или абстрактных объектов, каждый из которых обладает одинаковыми свойствами и характеристиками. В этом случае любой рассматриваемый объект может быть экземпляром одной и только одной сущности, должен иметь уникальное имя или идентификатор, а также отличаться от других экземпляров данной сущности.
- Связь (relationship) определяется как отношение или ассоциация между отдельными сущностями. Примерами связей могут являться родственные отношения, в частности "отец-сын" или производственные - "начальник-подчиненный". Другой тип связей задается отношениями "иметь в собственности" или "обладать свойством". Различные типы связей графически изображаются в форме ромба с соответствующим именем данной связи.
- Графическая модель данных строится таким образом, чтобы связи между отдельными сущностями отражали не только семантический характер соответствующего отношения, но и дополнительные аспекты обязательности связей, а также кратность участвующих в данных отношениях экземпляров сущностей. Нотация диаграмм (ERD) реализована в различных программных средствах.

Диаграмма "сущность-связь" для примера сотрудников компании, работающих над различными проектами, в ARIS



Диаграммы "сущность-связь"

- Ограниченность диаграмм ERD проявляется при конкретизации концептуальной модели в более детальное представление моделируемой программной системы, которое кроме статических связей должно содержать информацию о поведении или функционировании отдельных ее компонентов.

2. Диаграммы функционального моделирования

Существует несколько графических языков моделирования:

- Нотация IDEF0 - для документирования процессов производства и отображения информации об использовании ресурсов на каждом из этапов проектирования систем
- Нотация IDEF1 - для документирования информации о производственном окружении систем (в 1985 году была расширена и переименована в IDEF1X)
- Нотация IDEF2 - для документирования поведения системы во времени (никогда не была полностью реализована)

Методология IDEF нашла применение в правительственных и коммерческих организациях после появления в 1993 г. стандарта FIPS правительства США для двух технологий IDEF0 и IDEF1X. Позже этот стандарт продолжал активно развиваться и послужил основой для реализации в некоторых CASE-средствах, наиболее известным из которых является BPwin.

Диаграммы функционального моделирования

- Процесс моделирования IDEF представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели системы какой-либо предметной области. Функциональная модель IDEF отображает структуру процессов функционирования системы и ее отдельных подсистем, т.е. выполняемые ими действия и связи между этими действиями. Для этой цели строятся специальные модели, которые позволяют в наглядной форме представить последовательность определенных действий. Исходными строительными блоками любой модели нотации IDEF0 процесса являются деятельность (activity) и стрелки (arrows).
- Одна из наиболее важных особенностей нотации IDEF0 - постепенное введение все более детальных представлений модели системы по мере разработки отдельных диаграмм. Построение модели IDEF0 начинается с представления всей системы в виде простейшей диаграммы, состоящей из одного блока процесса и стрелок ICOM, служащих для изображения основных видов взаимодействия с объектами вне системы. Поскольку исходный процесс представляет всю систему как единое целое, данное представление является наиболее общим и подлежит дальнейшей декомпозиции.

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 **4 основных понятия**:
- 1) функциональный блок;
- 2) интерфейсная дуга (стрелка);
- 3) декомпозиция;
- 4) глоссарий.

Стандарт IDEF0

Вход — материал или информация, которые используются или преобразуются работой для получения результата

Выход — материал или информация, которые производятся работо

Управление — правила, стратегии, процедуры или стандарты, которыми руководствуется работа

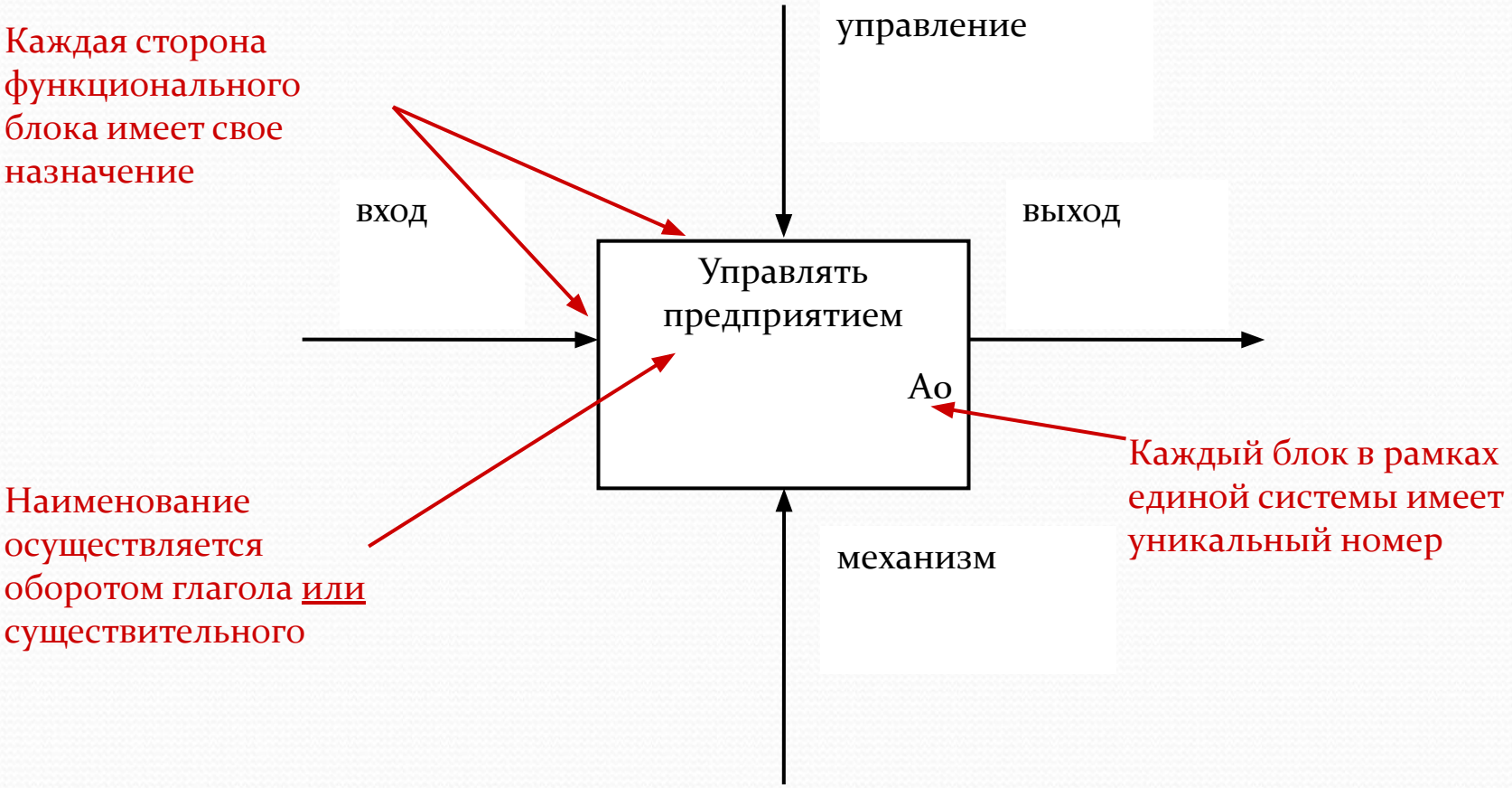
Механизм — ресурсы, которые выполняют работу

Вызов — специальная стрелка, указывающая на другую модель

Типы связей между работами:

Функциональный блок

- Олицетворяет некоторую конкретную функцию или работу в рамках рассматриваемой системы
- *РД IDEFo – 2000*: прямоугольник, содержащий имя и номер и используемый для описания функции



Интерфейсная дуга

- Интерфейсная дуга отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, отображаемую функциональным блоком.
- *Графически* изображается в виде однонаправленной стрелки.
- Каждая дуга должна иметь свое уникальное *название*, сформулированное оборотом существительного (должно отвечать на вопросы кто?, что?). Примеры: информация, разработчик, документ, обработанная заявка.
- В зависимости от того, к какой стороне блока она подходит, интерфейсная дуга будет являться *входящей, выходящей, управления, механизма*.

Интерфейсная дуга

Стрелки входа может не быть. Остальные интерфейсные дуги обязательны.

Декомпозиция

- Принцип декомпозиции применяется при разбиении сложных процессов на составляющие его функции. При этом уровень детализации определяется непосредственно разработчиком модели.
- Модель IDEF0 всегда начинается с рассмотрения системы как единого целого, т.е. одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области. Такая диаграмма называется *контекстной*, она обозначается идентификатором A-0.
- Для определения границ системы на контекстной диаграмме обязательно должны быть цель и точка зрения.

Цель моделирования

Цель моделирования должна отвечать на следующие вопросы:

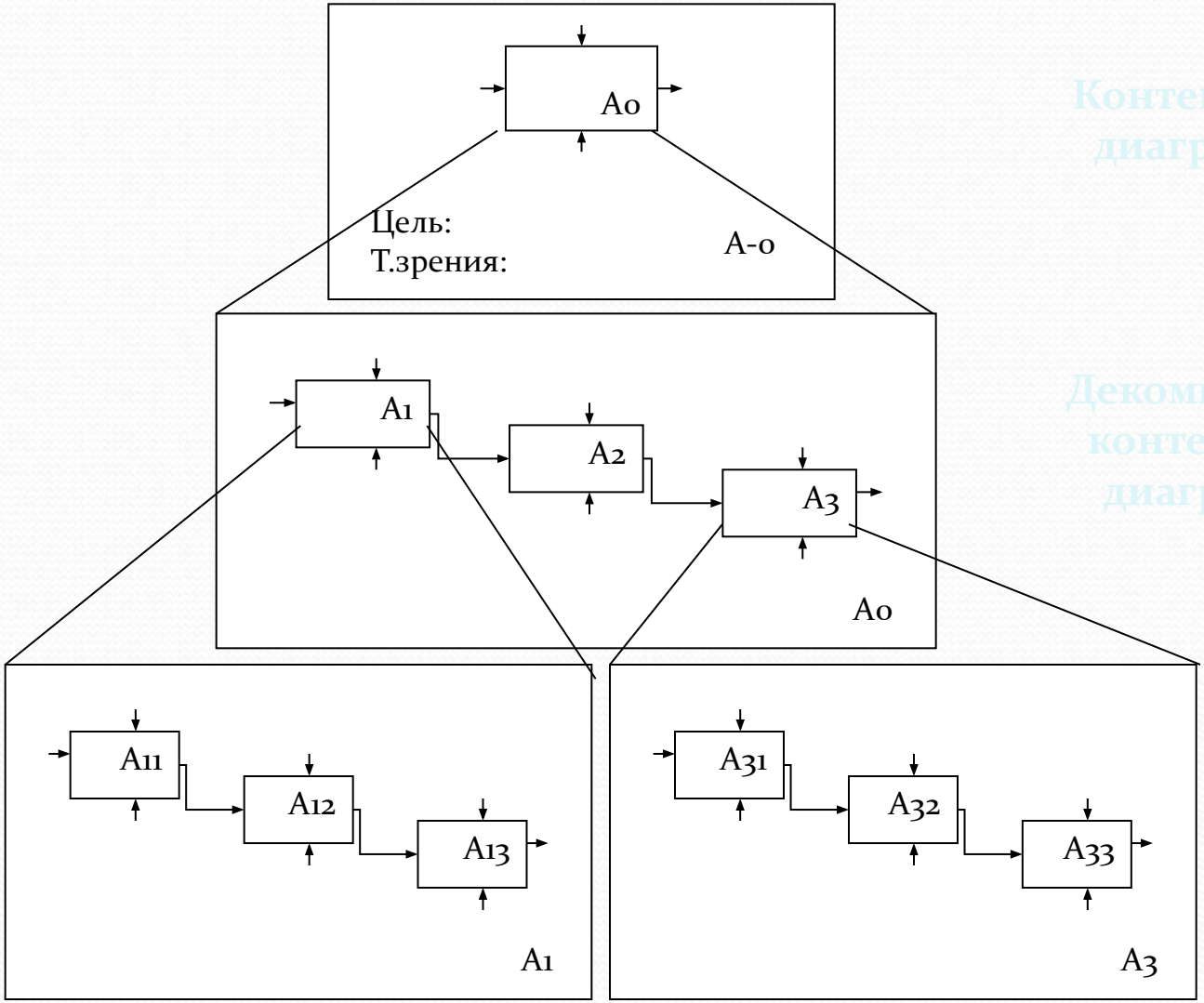
- Почему процесс должен быть замоделирован?
- Что должна показывать модель?
- Что может получить читатель?

Примеры целей: «Идентифицировать слабые стороны процесса сбора данных», «Определить ответственность сотрудников для написания должностных инструкций» и т.п.

Точка зрения

- Точка зрения – позиция, с которой будет строиться модель. В качестве точки зрения берется взгляд человека, который видит систему в нужном для моделирования аспекте.
- Как правило, выбирается точка зрения человека, ответственного за выполнение моделируемой работы.
- Между целью и точкой зрения должно быть жесткое соответствие.

Декомпозиция



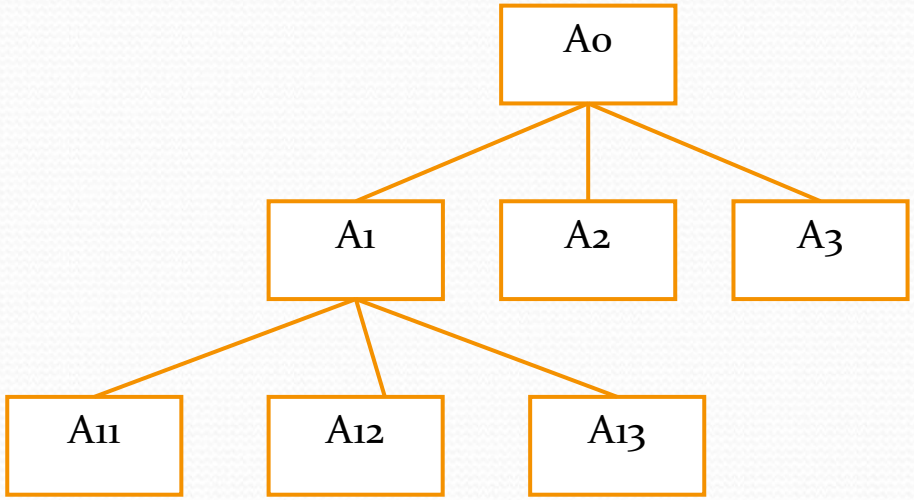
Контекстная диаграмма

Декомпозиция контекстной диаграммы

Декомпозиция блока A1

Декомпозиция блока A3

Декомпозиция



Дерево узлов

A0 _____
A1_____
A11_____
A12_____
A13_____
A2_____
A3_____

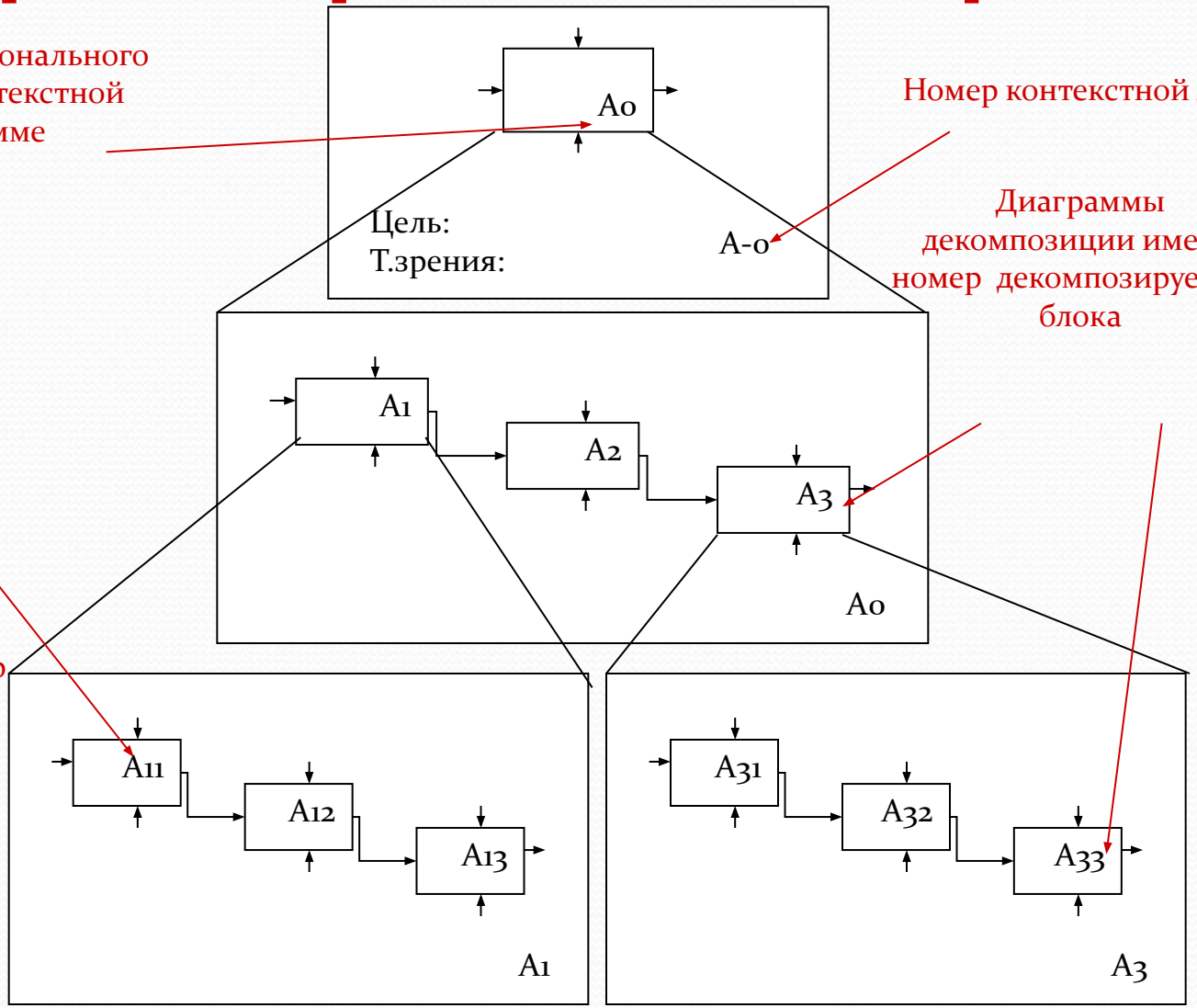
Индекс узлов

Нумерация работ и диаграмм

Номер функционального блока на контекстной диаграмме

Номер контекстной диаграммы

Диаграммы декомпозиции имеют номер декомпозируемого блока

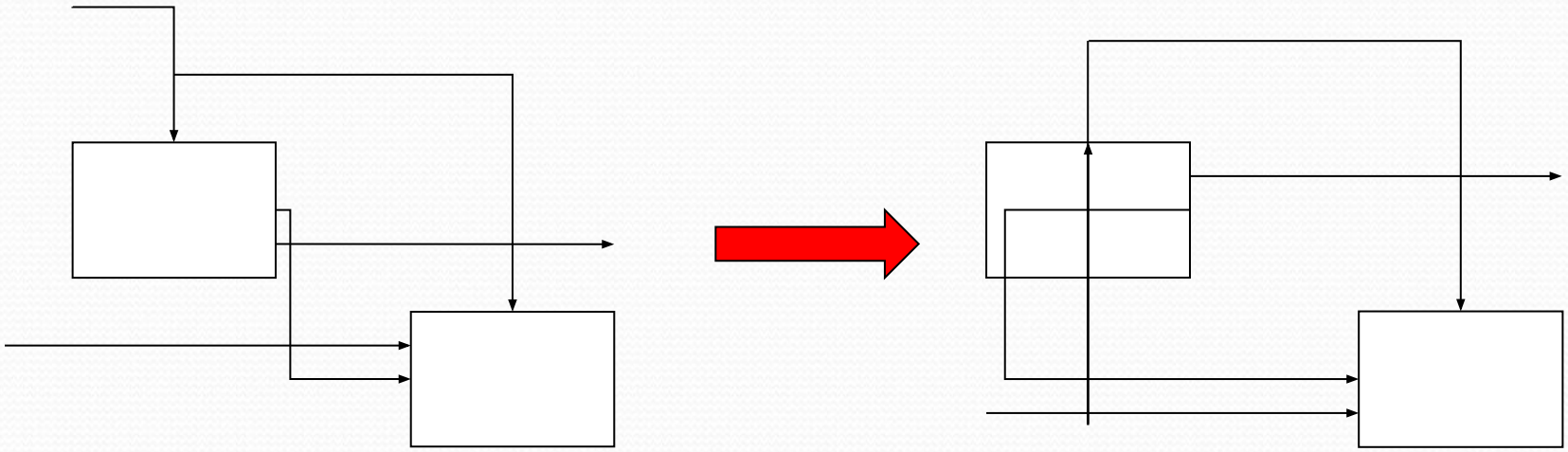


Формат номера блока:

1. Префикс
2. Номер родительской работы
3. Собственный порядковый номер

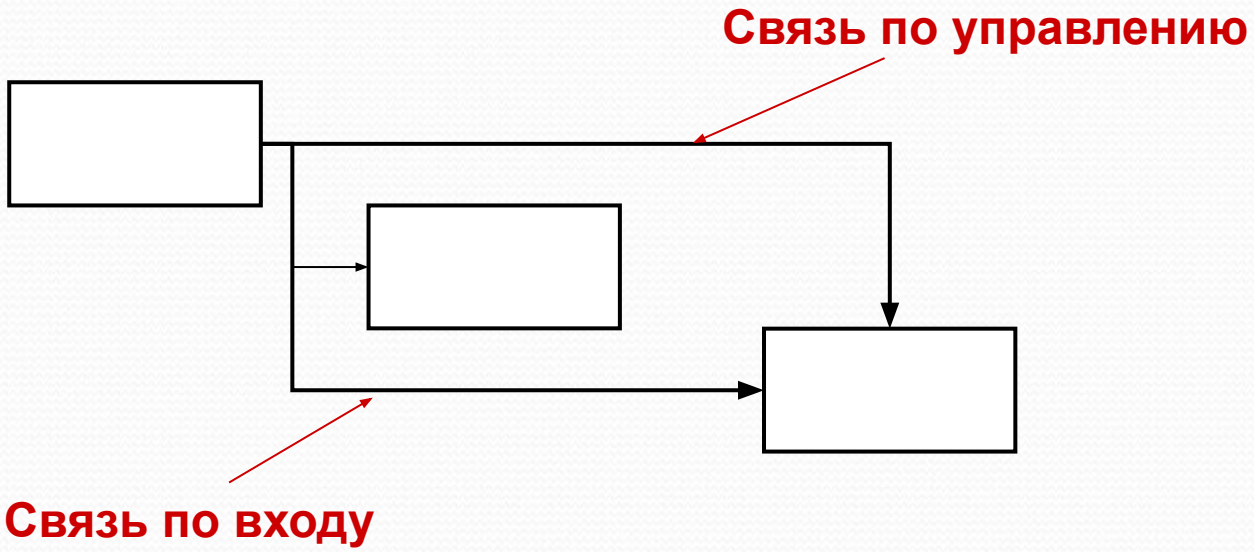
Основные правила построения диаграмм

1. На одной диаграмме рекомендуется рисовать от 3 до 6 блоков. Иначе диаграмма будет плохо читаемой.
2. Функциональные блоки должны располагаться слева направо сверху вниз в порядке доминирования.
3. Следует избегать излишнего пересечения стрелок.

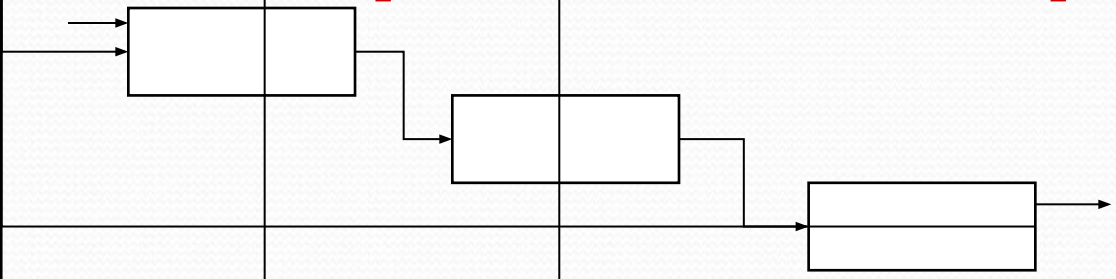


Основные правила построения диаграмм

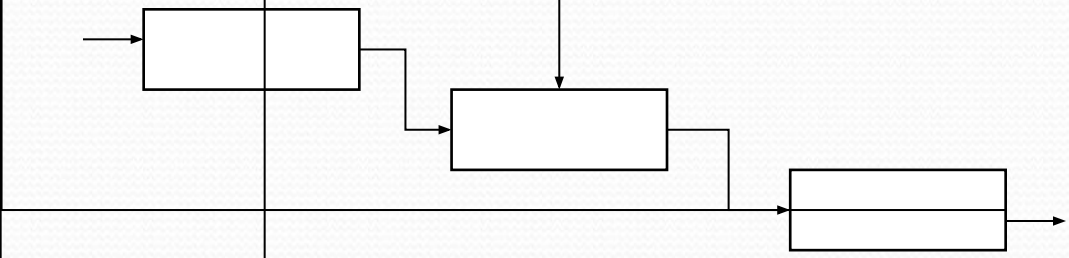
4. Выход одного блока может являться входом (управлением) для другого. Могут быть и обратные связи по входу и управлению.



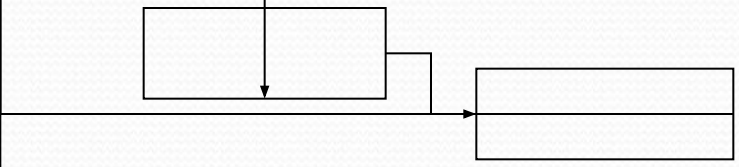
Основные правила построения диаграмм



а) обратная связь по входу



б) обратная связь по управлению



в) обратная связь по механизму

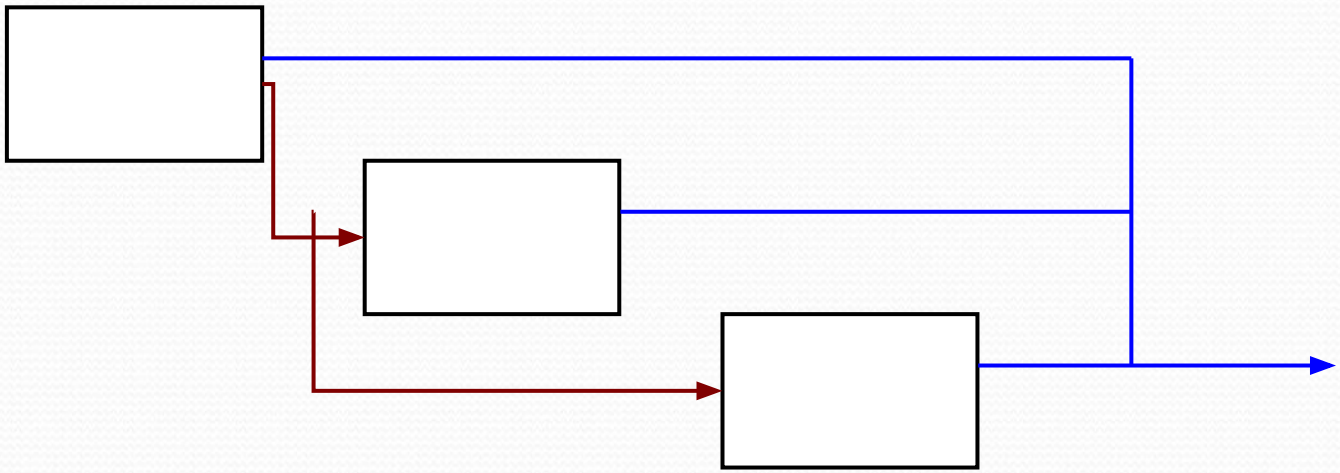
Обратная связь по входу, как правило, используется для описания циклов.

Обратная связь по управлению – выход нижестоящей работы передается на управление вышестоящей

Обратная связь по механизму – выход нижестоящей работы создает ресурсы, выполняющие вышестоящую работу

Основные правила построения диаграмм

5. Стрелки могут быть сливающимися и разветвляющимися

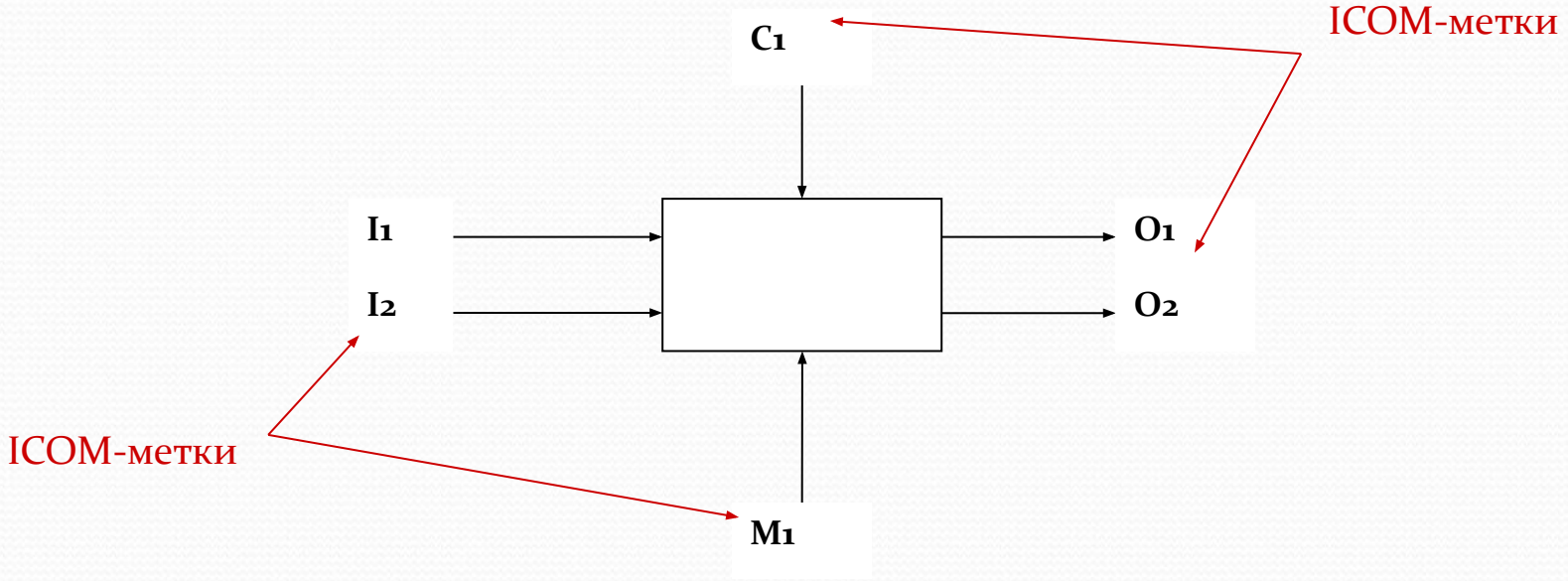


Слияние стрелок

Разветвление стрелок

Граничные стрелки

Стрелки на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром. Они могут начинаться у границы диаграммы и заканчиваться у функционального блока и наоборот. Такие стрелки называются граничными. Граничные стрелки помечаются с помощью **ICOM-меток** (Input, Control, Output, Mechanism)



Тоннельные стрелки

Иногда необходимо отобразить граничные стрелки, которые значимы на данном уровне и не значимы на родительской диаграмме. Например, некоторые данные используются только на данном уровне и не используются на других. Без использования механизма тоннелирования малозначимая стрелка появится на всех уровнях модели, что затруднит чтение диаграмм.

Глоссарий и FEO-страница

- Для каждого из элементов в IDEF0 существует стандарт, подразумевающий создание и поддержку набора соответствующих определений, ключевых слов, повествований, изложений и т.д, которые характеризуют объект, отраженный данным элементом. Этот набор – **глоссарий**, являющийся описанием сущности данного элемента.
- **FEO-диаграмма** (*For Exposition Only*) – это диаграмма, которая поясняет особо интересные и тонкие аспекты диаграмм. Эти диаграммы не ограничены синтаксисом IDEF0. В них может быть текстовая, графическая информация, схемы, альтернативная точка зрения на процесс и т.п.

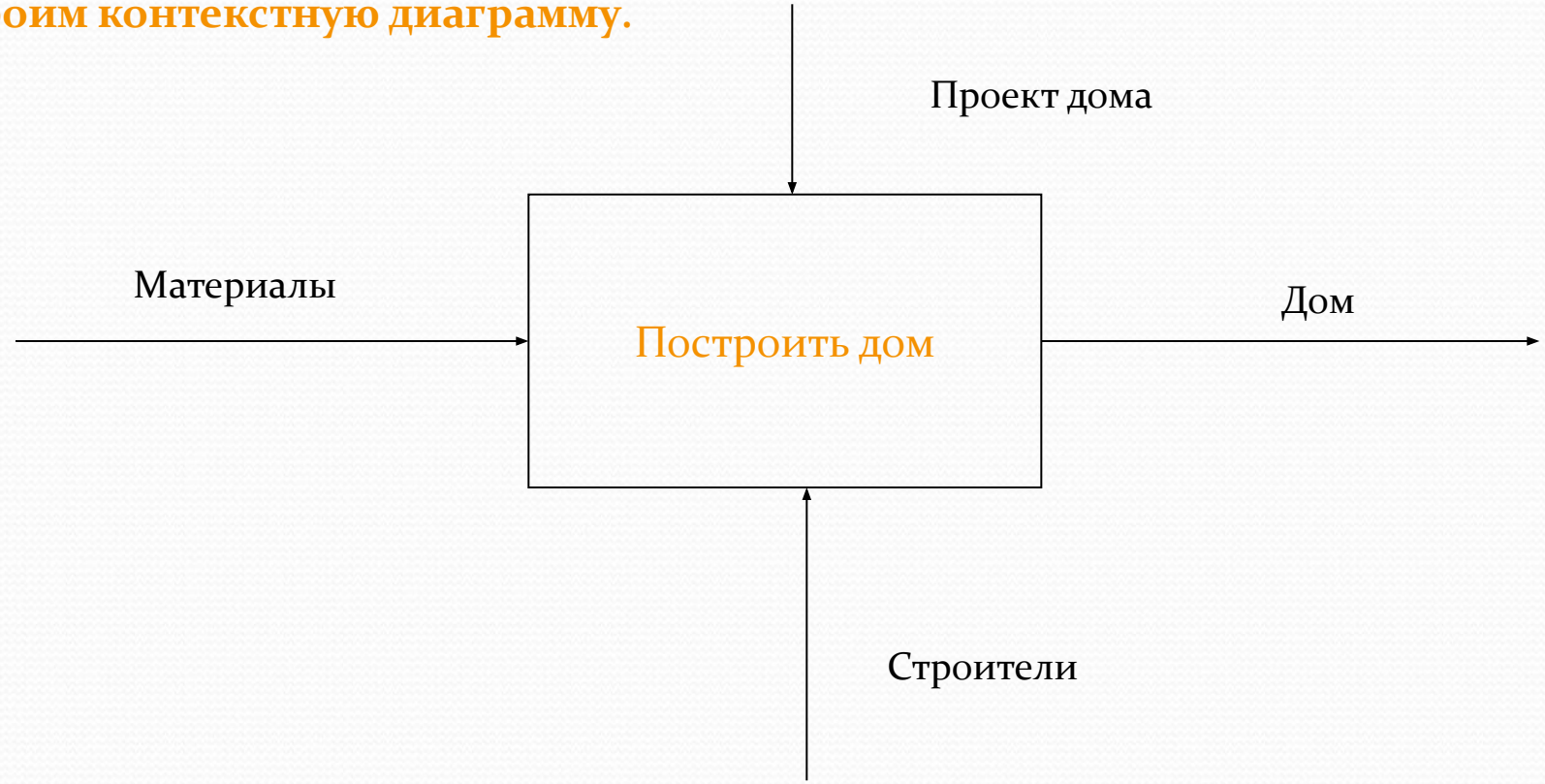
Мастерская страница (каркас диаграммы)

- Стандартный бланк для диаграмм (облегчает подшивку и копирование)
- Разделен на 3 основные части:

1) **поле рабочей информации** (для отслеживания диаграммы в процессе моделирования)
2) **поле сообщений** (область рисования диаграммы)
3) **поле идентификации** (идентификация диаграммы и ее позиционирование в иерархии)

Пример модели процесса постройки садового домика

1. Строим контекстную диаграмму.

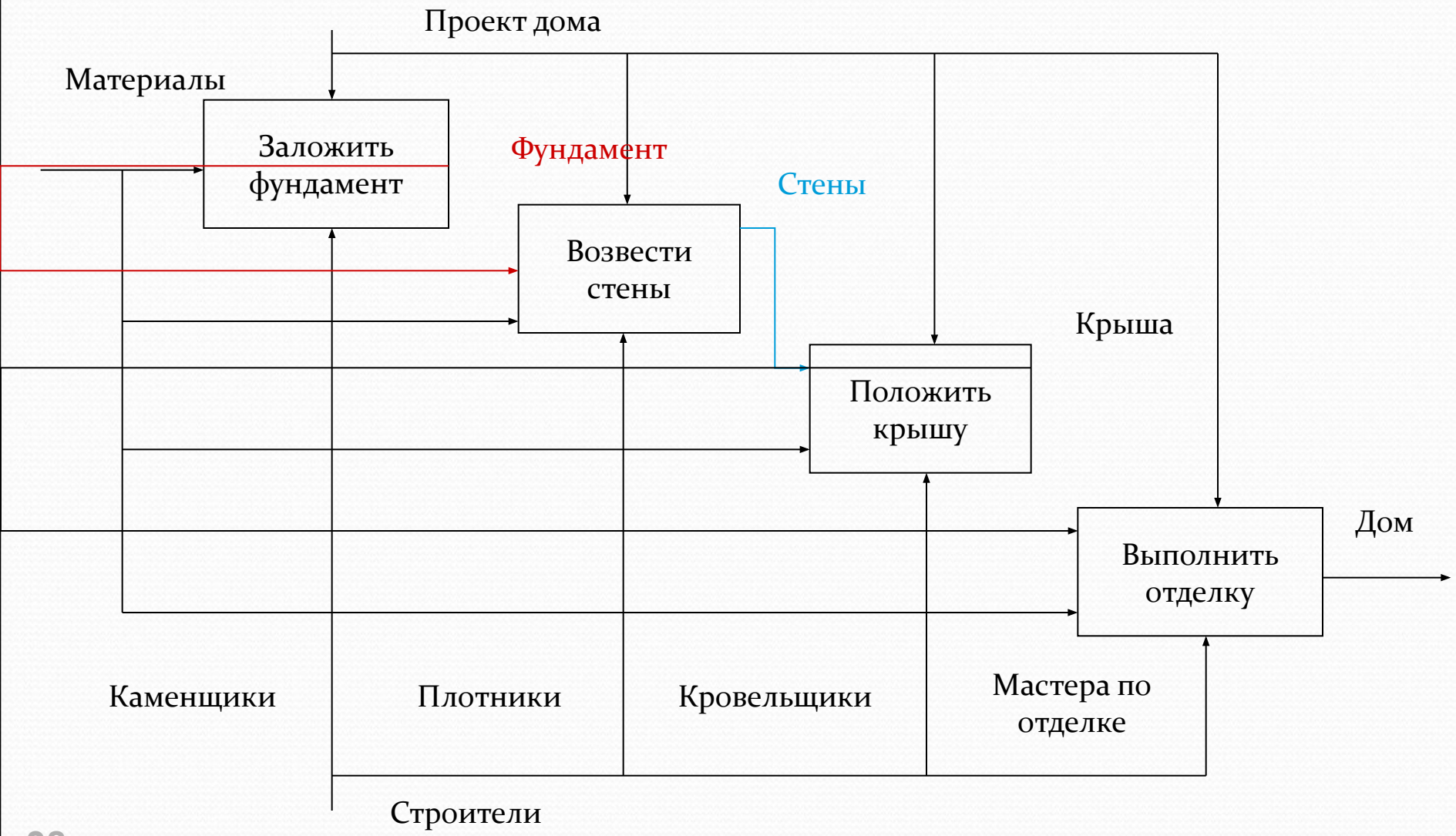


Цель: Определить действия, необходимые для постройки дачного домика

Точка зрения: владельца дачного участка

Пример модели процесса постройки садового домика

2. Декомпозируем контекстную диаграмму



Пример модели, построенной с использованием CASE-средства BPWin

Пример модели, построенной с использованием CASE-средства BPWin

Дерево узлов

FEO-страница

- В конечном итоге **модель IDEF0** представляет собой набор иерархически взаимосвязанных диаграмм с сопроводительной документацией, которая разбивает исходное представление сложной системы на отдельные составные части. Детали каждого основного процесса представляются в виде более подробных процессов на других диаграммах. В этом случае каждая диаграмма нижнего уровня является декомпозицией процесса из более общей диаграммы. Поэтому на каждом шаге декомпозиции более общая диаграмма конкретизируется на ряд детальных диаграмм.
- Основной недостаток данной методологии связан с отсутствием явных средств для объектно-ориентированного представления моделей сложных систем. Некоторые аналитики отмечают важность знания и применения нотации IDEF0, однако отсутствие возможности реализации соответствующих графических моделей в объектно-ориентированном программном коде существенно сужают диапазон решаемых с ее помощью задач.

Диаграммы потоков данных (ДПД) Data Flow Diagrams (DFD)

- В основе графического моделирования информационных систем с помощью диаграмм потоков данных лежит специальная технология построения диаграмм потоков данных DFD.
- В разработке методологии DFD приняли участие многие аналитики, среди которых следует отметить Э. Йордона. Он автор одной из первых графических нотаций DFD.
- Для изображения ДПД традиционно использовали две различные нотации: Йордона и Гейна-Сарсона.

ДПД

- Основным средством моделирования функциональных требований АИС являются диаграммы потоков данных. С их помощью эти требования разбиваются на функциональные компоненты (процессы) и представляются в виде сети, связанной потоками данных. Главная цель таких средств – продемонстрировать как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами

ДПД

- В соответствии с методологией модель системы определяется как иерархия диаграмм потоков данных (ДПД или DFD), описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю. Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы ИС с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут такой уровень декомпозиции, на котором процессы становятся элементарными и детализировать их далее невозможно.

ДПД

- Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те в свою очередь преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям - потребителям информации. Таким образом, основными компонентами диаграмм потоков данных являются:
  - внешние сущности;
  - системы/подсистемы;
  - процессы;
  - накопители данных;
  - потоки данных.

ДПД.Внешняя сущность

- Внешняя сущность представляет собой материальный предмет или физическое лицо, представляющее собой источник или приемник информации, например, заказчики, персонал, поставщики, клиенты, склад. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой ИС. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой ИС, если это необходимо, или, наоборот, часть процессов ИС может быть вынесена за пределы диаграммы и представлена как внешняя сущность.

ДПД.Внешняя сущность

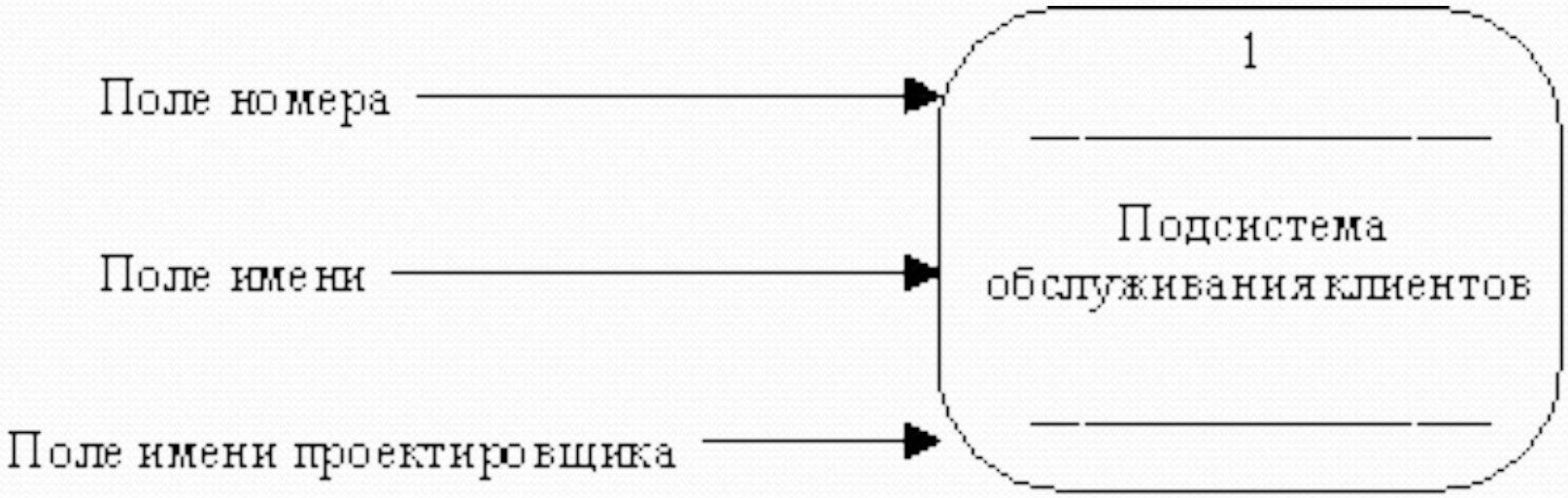
- Внешняя сущность обозначается квадратом, расположенным как бы "над" диаграммой и бросающим на нее тень, для того, чтобы можно было выделить этот символ среди других обозначений:

ДПД. Системы и подсистемы

- При построении модели сложной ИС она может быть представлена в самом общем виде на так называемой контекстной диаграмме в виде одной системы как единого целого, либо может быть декомпозирована на ряд подсистем.

ДПД. Системы и подсистемы

- Подсистема (или система) на контекстной диаграмме изображается следующим образом



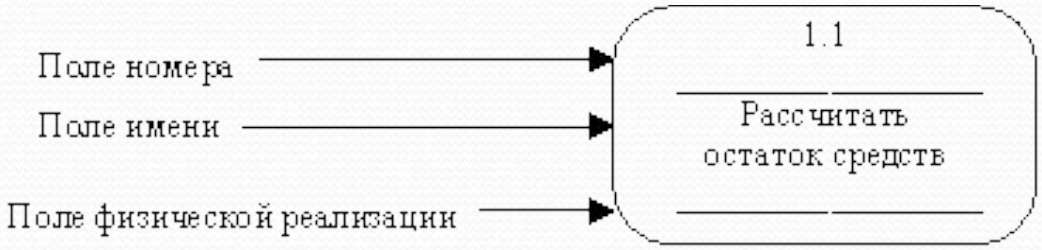
- Номер подсистемы служит для ее идентификации. В поле имени вводится наименование подсистемы в виде предложения с подлежащим и соответствующими определениями и дополнениями.

ДПД. Процесс

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

ДПД. Процесс

- Процесс на диаграмме потоков данных изображается так:



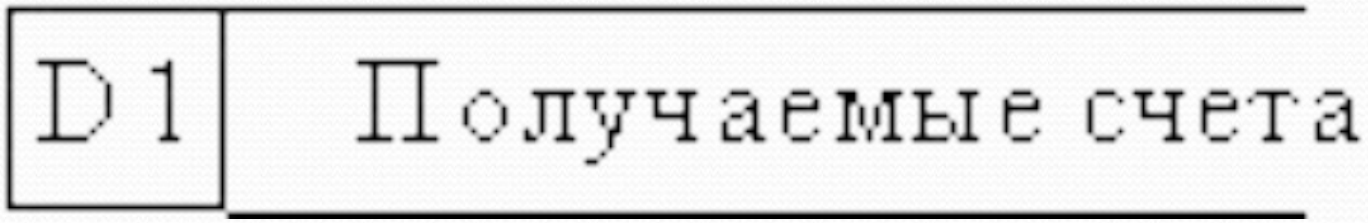
- Номер процесса служит для его идентификации. В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже, например: "Ввести сведения о клиентах"; "Выдать информацию о текущих расходах";"Проверить кредитоспособность клиента".
- Использование таких глаголов, как "обработать", "модернизировать" или "отредактировать" означает, как правило, 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.

ДПД. Накопитель данных

- Накопитель данных представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.
- Накопитель данных может быть реализован физически в виде микрофиши, ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т.д.

ДПД. Накопитель данных

- Накопитель данных на диаграмме потоков данных изображается так:



- 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 базы данных и описание хранящихся в нем данных должно быть увязано с информационной моделью.

ДПД. Потоки данных

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

ДПД. Потоки данных

- Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока.

- Каждый поток данных имеет имя, отражающее его содержание.

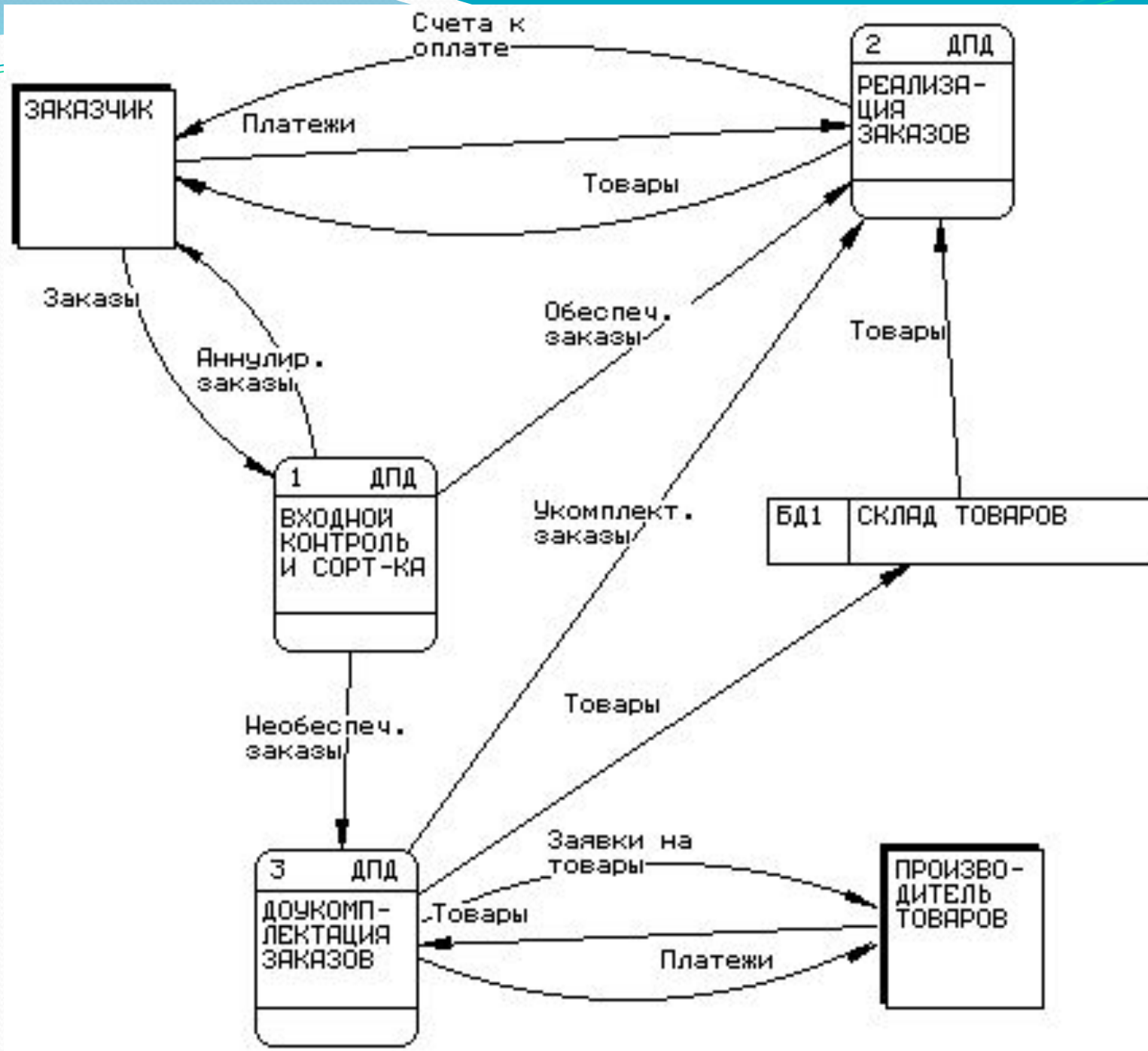
ДПД. Построение иерархии диаграмм потоков данных

- Первым шагом при построении иерархии ДПД является построение контекстных диаграмм. Обычно при проектировании относительно простых ИС строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.
- Если же для сложной системы ограничиться единственной контекстной диаграммой, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и кроме того, единственный главный процесс не раскрывает структуры распределенной системы. Признаками сложности (в смысле контекста) могут быть:
  - наличие большого количества внешних сущностей (десять и более);
  - распределенная природа системы;
  - многофункциональность системы с уже сложившейся или выявленной группировкой функций в отдельные подсистемы.
- Для сложных ИС строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.
- Иерархия контекстных диаграмм определяет взаимодействие основных функциональных подсистем проектируемой ИС как между собой, так и с внешними входными и выходными потоками данных и внешними объектами (источниками и приемниками информации), с которыми взаимодействует ИС.
- Разработка контекстных диаграмм решает проблему строгого определения функциональной структуры ИС на самой ранней стадии ее проектирования, что особенно важно для сложных многофункциональных систем, в разработке которых 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 другими объектами).
- Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи ДПД. Каждый процесс на ДПД, в свою очередь, может быть детализирован при помощи ДПД или миниспецификации. При детализации должны выполняться следующие правила:
- правило балансировки - означает, что при детализации подсистемы или процесса детализирующая диаграмма в качестве внешних источников/приемников данных может иметь только те компоненты (подсистемы, процессы, внешние сущности, накопители данных), с которыми имеет информационную связь детализируемая подсистема или процесс на родительской диаграмме;
- правило нумерации - означает, что при детализации процессов должна поддерживаться их иерархическая нумерация. Например, процессы, детализирующие процесс с номером 12, получают номера 12.1, 12.2, 12.3 и т.д.

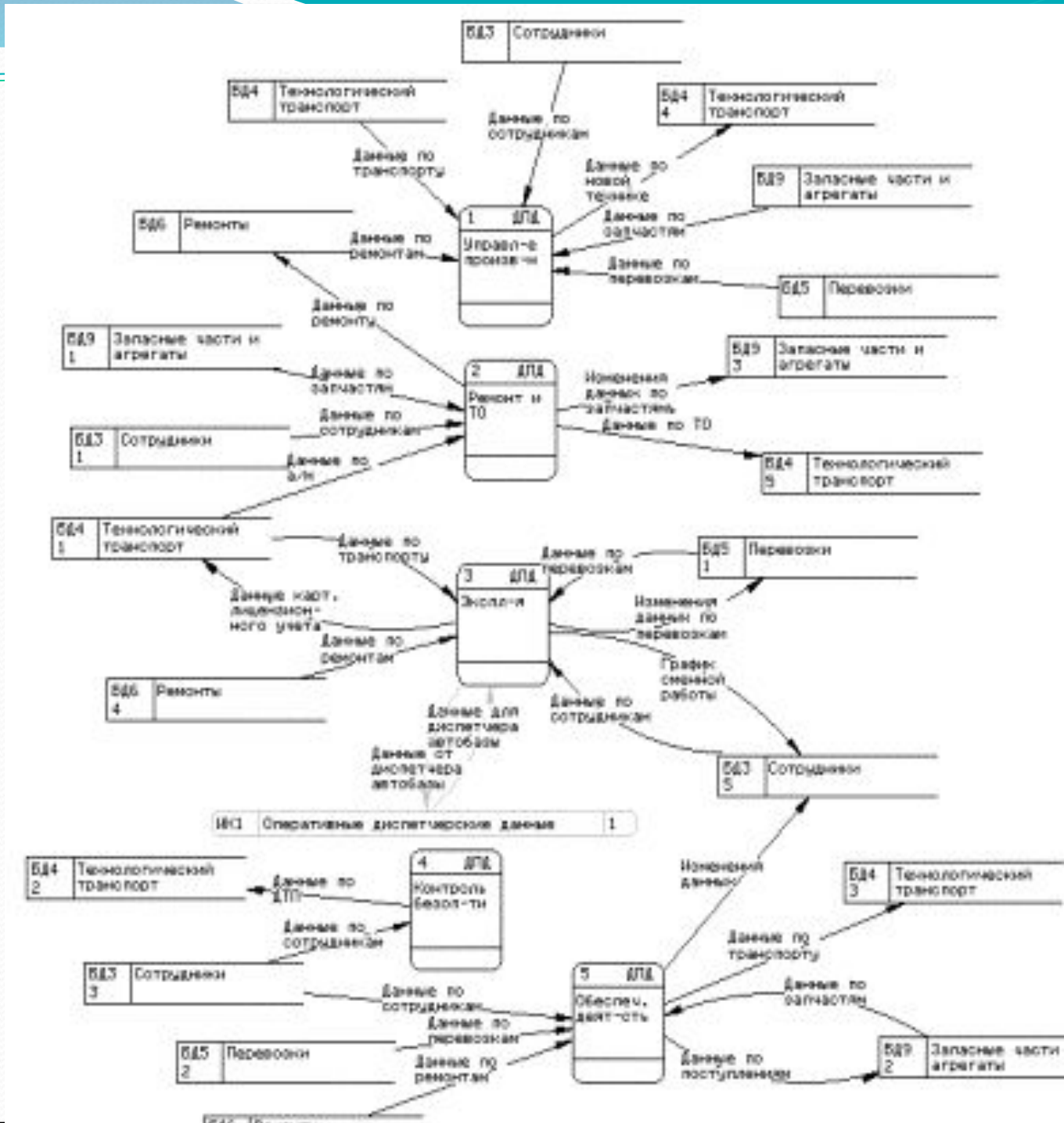
ДПД. Построение иерархии диаграмм потоков данных

- Миниспецификация (описание логики процесса) должна формулировать его основные функции таким образом, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу.
- Миниспецификация является конечной вершиной иерархии ДПД. Решение о завершении детализации процесса и использовании миниспецификации принимается аналитиком исходя из следующих критериев:
  - наличия у 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 миниспецификации небольшого объема (не более 20-30 строк).
- При построении иерархии ДПД переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается при помощи структур данных. Структуры данных конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что данный компонент может отсутствовать в структуре. Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает вхождение любого числа элементов в указанном диапазоне. Для каждого элемента данных может указываться его тип (непрерывные или дискретные данные). Для непрерывных данных может указываться единица измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования. Для дискретных данных может указываться таблица допустимых значений.
- После построения законченной модели системы ее необходимо верифицировать (проверить на полноту и согласованность). В полной модели все ее объекты (подсистемы, процессы, потоки данных) должны быть подробно описаны и детализированы. Выявленные недетализированные объекты следует детализировать, вернувшись на предыдущие шаги разработки. В согласованной модели для всех потоков данных и накопителей данных должно выполняться правило сохранения информации: все поступающие куда-либо данные должны быть считаны, а все считываемые данные должны быть записаны.

Пример ДПД Гейна-Сарсона



Фрагмент системного проекта в нотации ДПД Гейна-Сарсона



Недостаток рассмотренных нотаций структурного системного анализа

Связан с отсутствием

- явных средств для объектно-ориентированного представления моделей сложных систем,
- а также сложных алгоритмов обработки данных.

Поскольку на рассмотренных типах диаграмм не указываются характеристики времени выполнения отдельных процессов и передачи данных между процессами, то модели систем, реализующих синхронную обработку данных, не могут быть адекватно представлены в этих нотациях.

Все эти особенности методов структурного системного анализа ограничили возможности широкого применения соответствующих нотаций и послужили основой для разработки унифицированного языка моделирования UML.