

Методологии IDEF0, DFD

Основа методологии IDEF0

В основе методологии лежат четыре основных понятия:

1.функциональный блок,

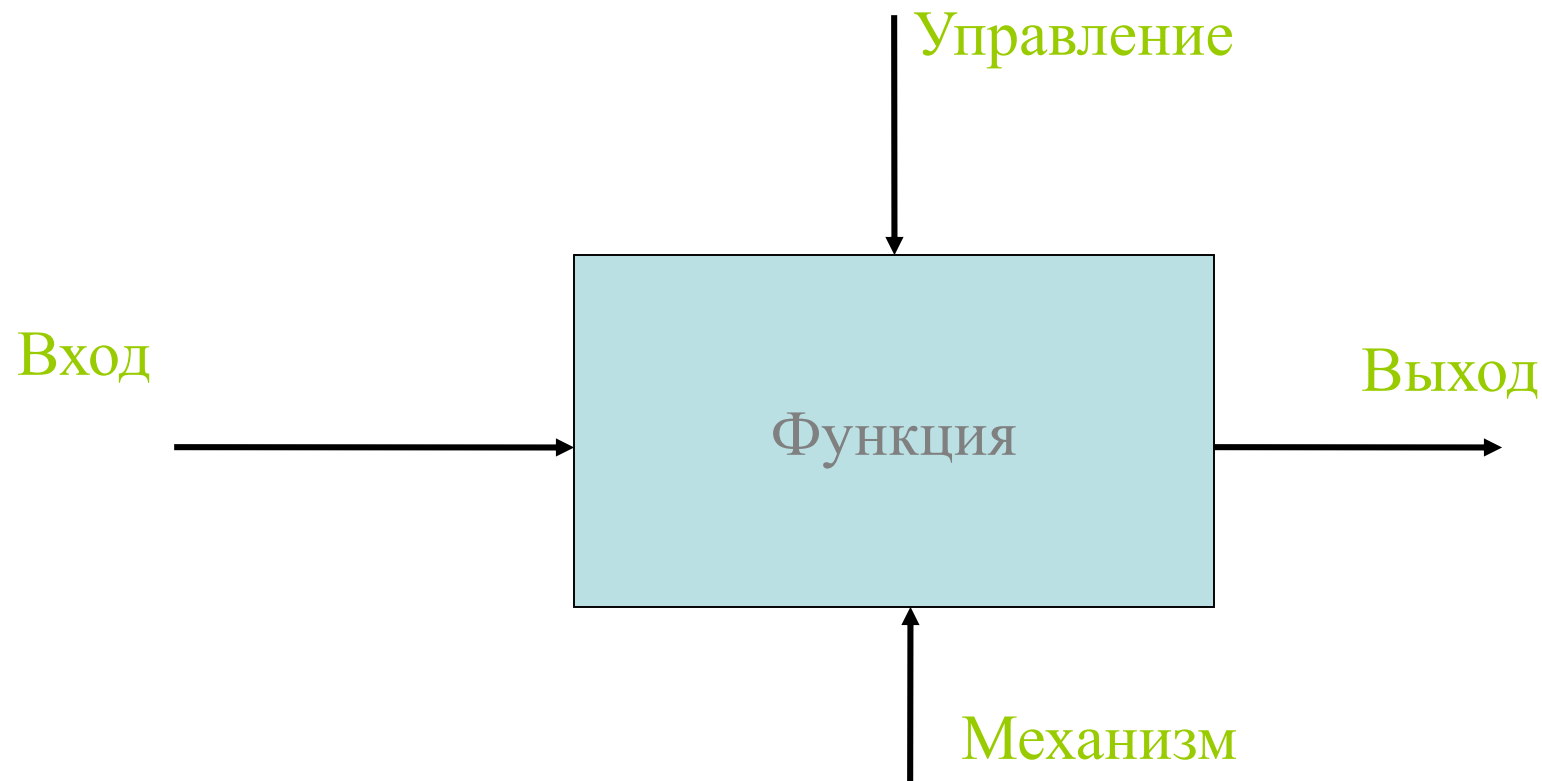
2.интерфейсная дуга,

3.декомпозиция,

4.гlossарий.

Функциональная модель SADT(Structured Analysis and Design Teqnique) (IDEF0)

отображает действия объекта и связи между этими действиями



Модель обеспечивает отделение функций от организационной структуры.

Требование стандарта IDEF0

Любой функциональный блок должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую.

Причина: каждый процесс должен происходить по каким-то правилам (отображаемым управляющей дугой) и должен выдавать некоторый результат (выходящая дуга)

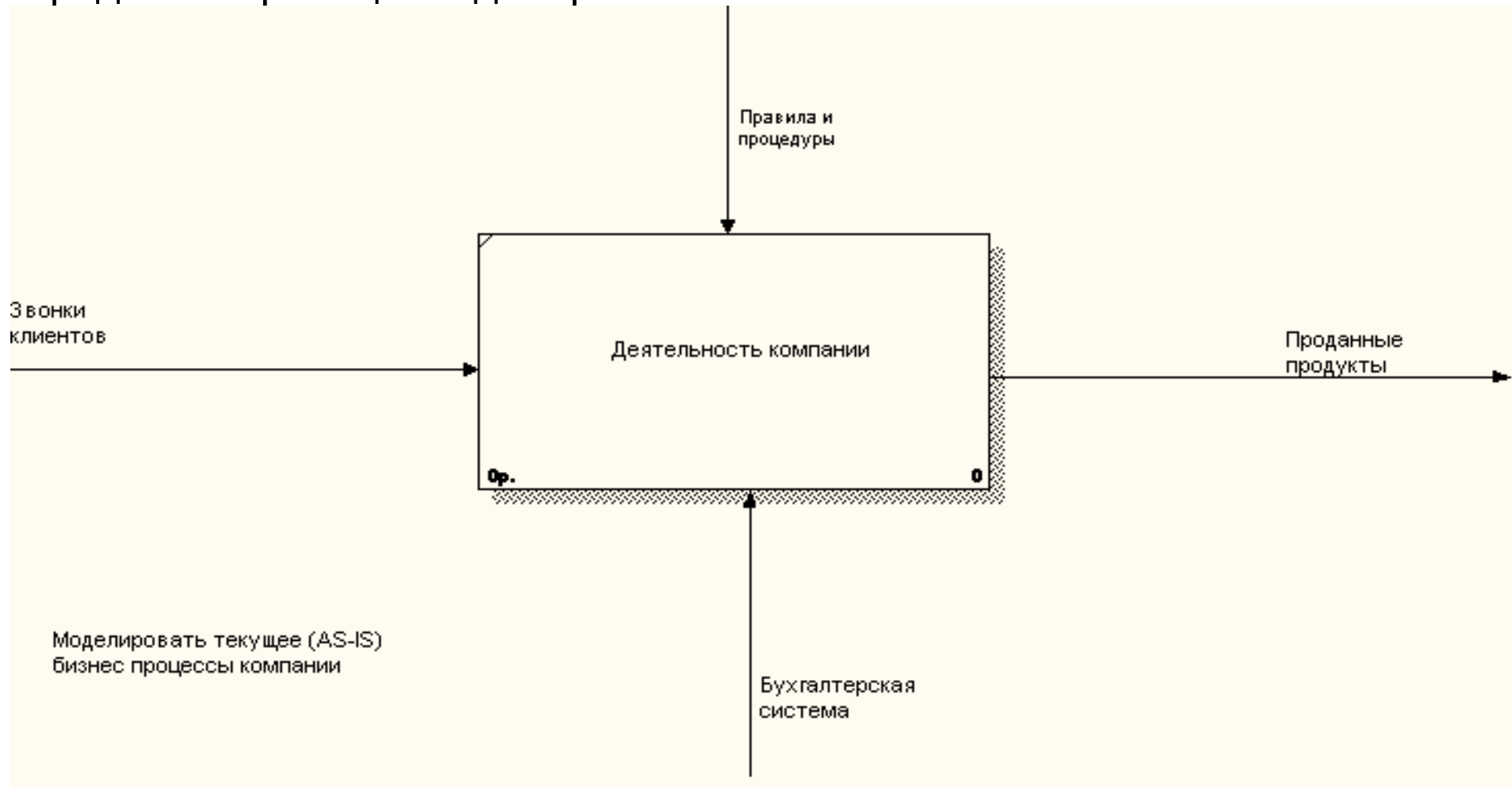
Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий классов DFD (Data Flow Diagram) и WFD (Work Flow Diagram).

Модель IDEF0 всегда начинается с представления системы как единого целого – одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области.

Контекстная диаграмма - диаграмма самого высокого уровня.

Определяет

- общее представление о деятельности организации
- задает единую точку зрения на описание деятельности исходя из цели моделирования
- определяет границы моделирования системы и ее компонентов



Цель и точка зрения

В пояснительном тексте к контекстной диаграмме должна быть указана **цель** (Purpose) построения диаграммы в виде краткого описания и зафиксирована **точка зрения** (Viewpoint).

Цель определяет соответствующие области в исследуемой системе, на которых необходимо фокусироваться в первую очередь.

Точка зрения определяет основное направление развития модели и уровень необходимой детализации.

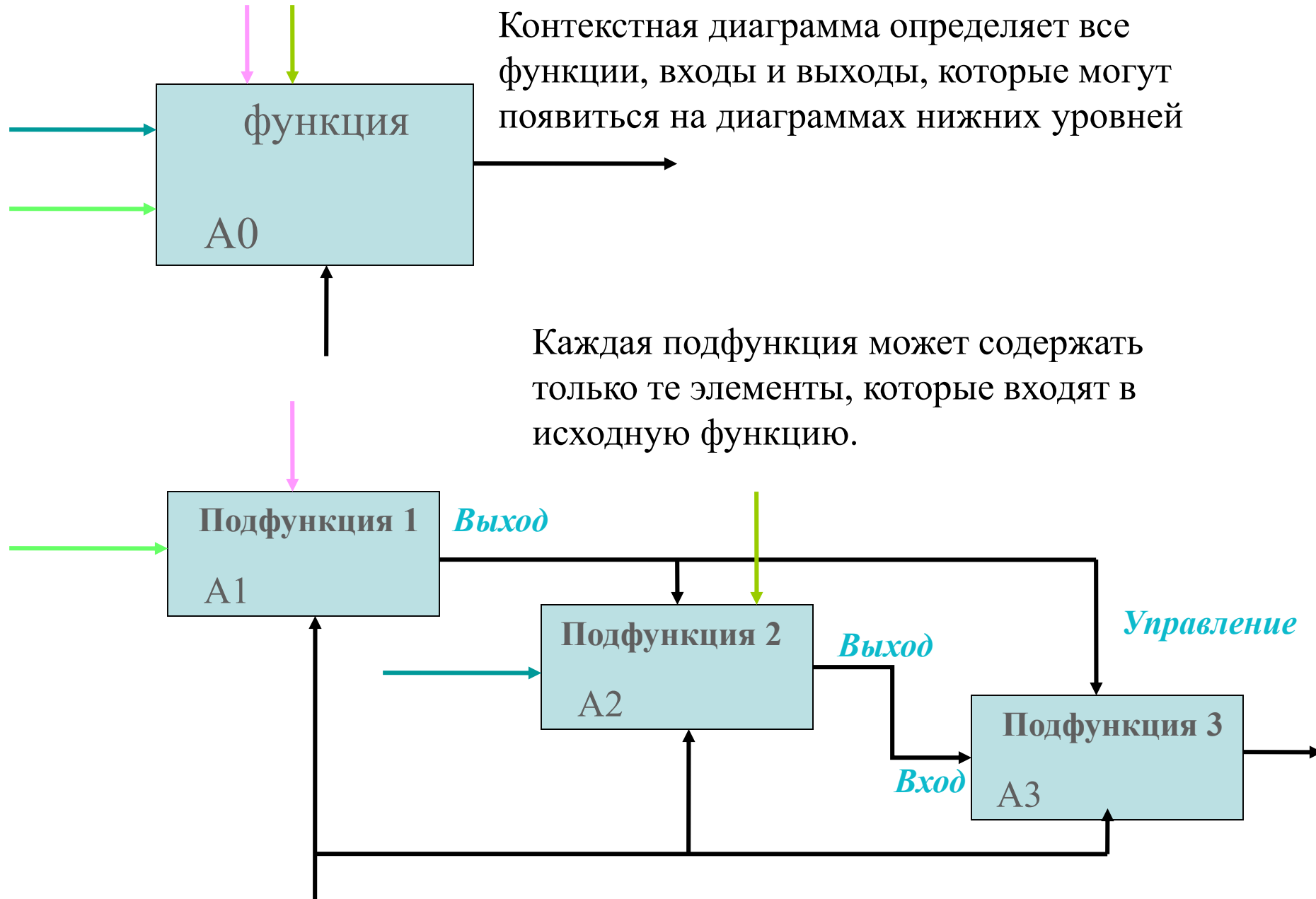
Декомпозиция IDEF0

Декомпозиция (Decomposition) является основным понятием стандарта IDEF0.

Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции. При этом уровень детализации процесса определяется непосредственно разработчиком модели.

Позволяет: постепенно и структурированно представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

Декомпозиция функциональных диаграмм

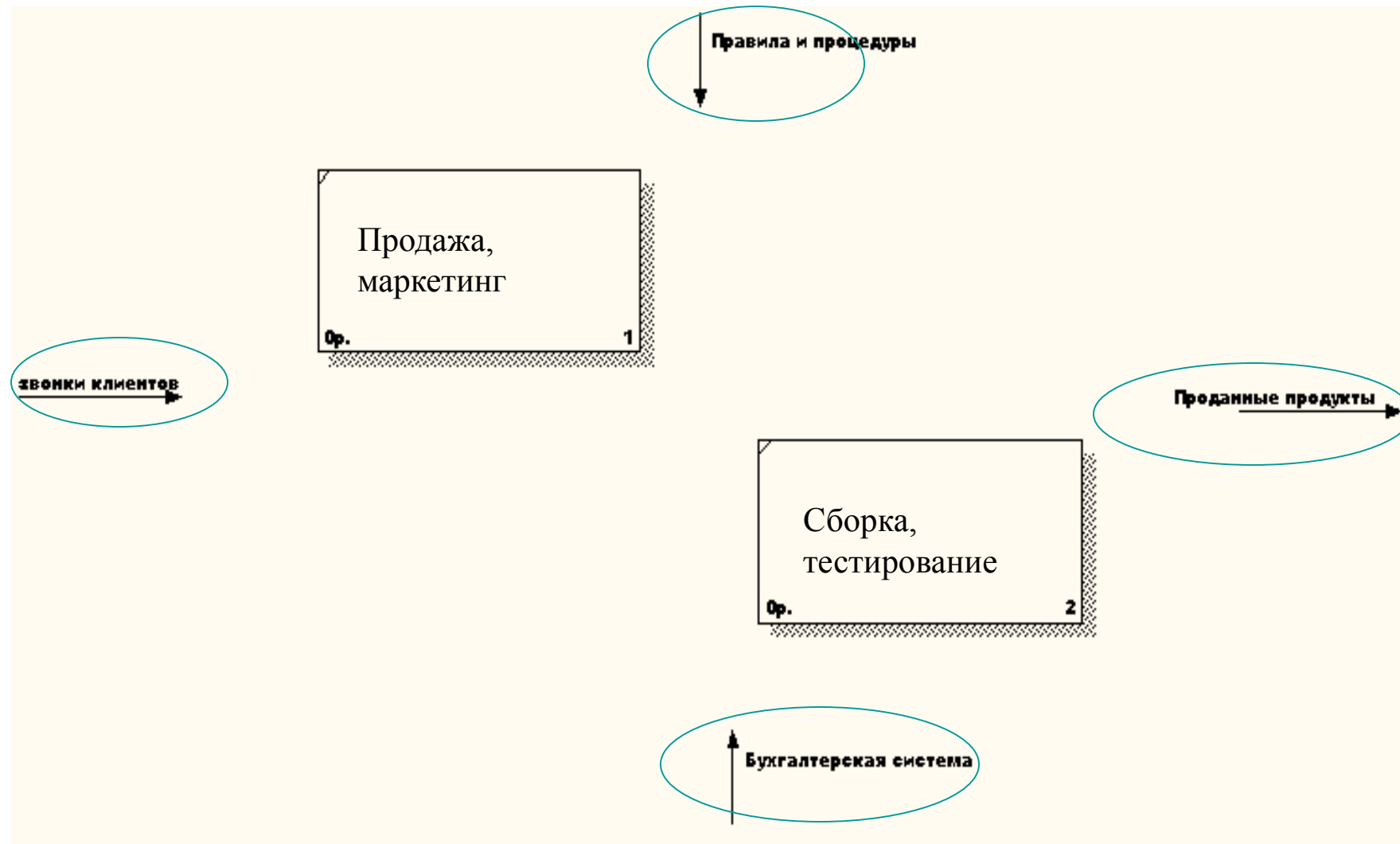


Структурная целостность модели IDEF0

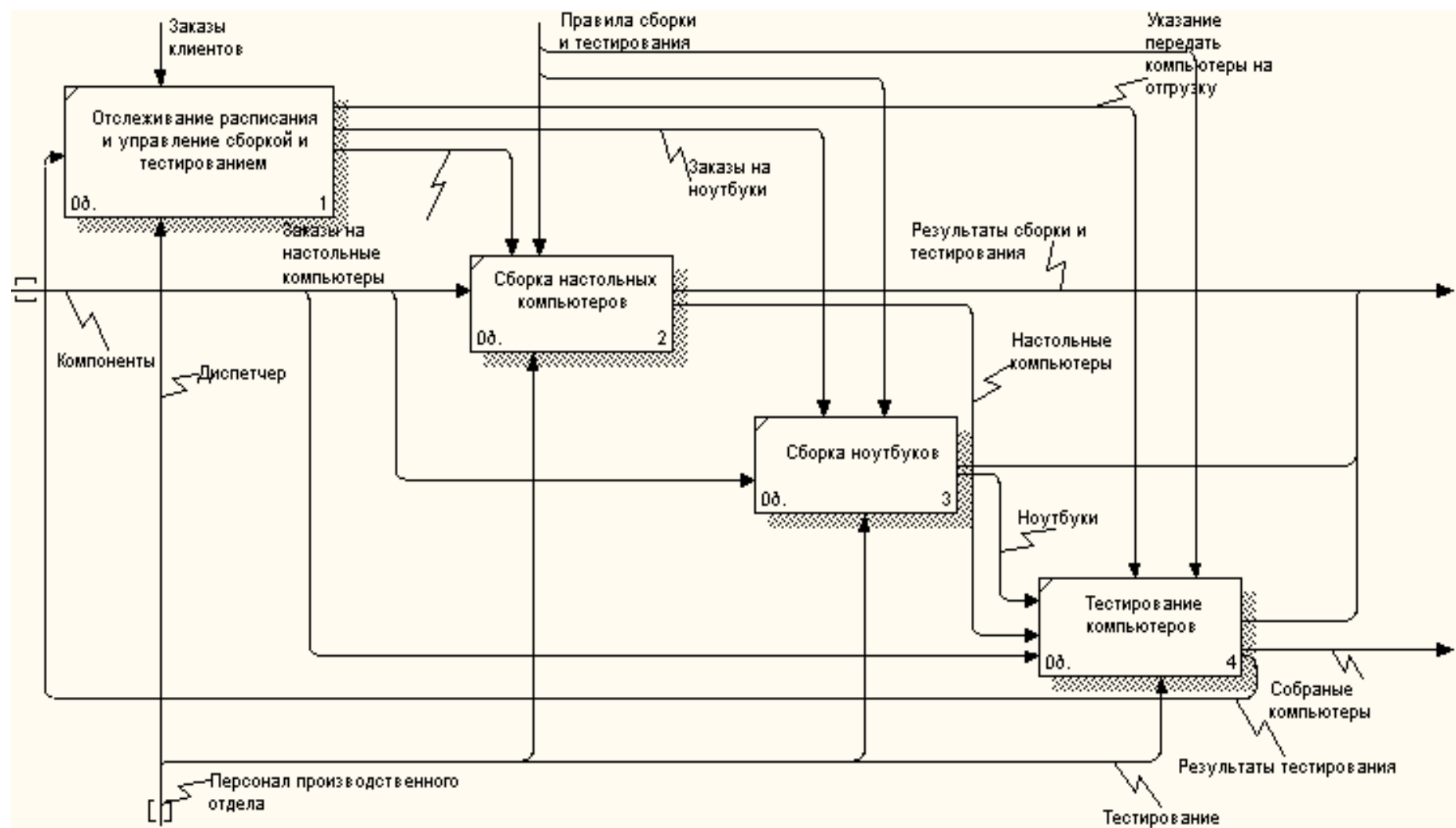
В каждом случае декомпозиции функционального блока все интерфейсные дуги, входящие в данный блок или исходящие из него, фиксируются на дочерней диаграмме.

Этим достигается структурная целостность IDEF0–модели.

Перенос контекста на декомпозицию



декомпозиция



Туннелирование в IDEF0

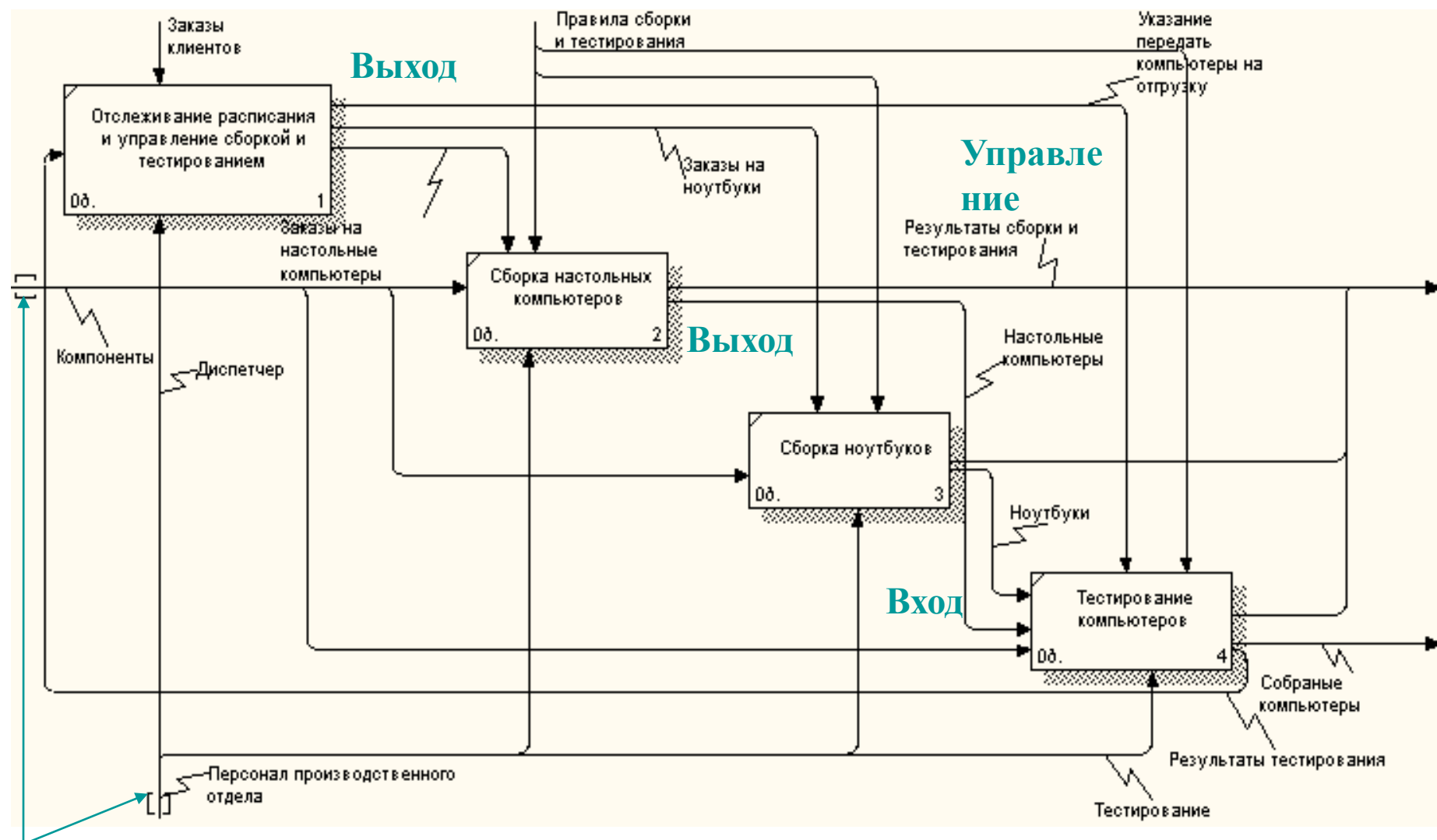
Иногда отдельные интерфейсные дуги высшего уровня не имеет смысла продолжать рассматривать на диаграммах нижнего уровня, или наоборот — отдельные дуги нижнего отражать на диаграммах более высоких уровней — это будет только перегружать диаграммы и делать их сложными для восприятия.

Для решения подобных задач в стандарте IDEF0 предусмотрено понятие туннелирования.

Обозначение "туннеля" (Arrow Tunnel) в виде двух круглых скобок вокруг начала интерфейсной дуги обозначает, что эта дуга не была унаследована от функционального родительского блока и появилась (из "туннеля") только на этой диаграмме.

Такое же обозначение вокруг конца (стрелки) интерфейсной дуги в непосредственной близи от блока–приемника означает тот факт, что в дочерней по отношению к этому блоку диаграмме эта дуга отображаться и рассматриваться не будет.

Преобразование типов стрелок



Появление новых стрелок, отсутствующих на родительской диаграмме, отображается «туннелем»

Ограничения сложности IDEF0-диаграмм

- Ограничение количества функциональных блоков на диаграмме: три-семь. Верхний предел (семь) обусловлен физиологическими возможностями восприятия информации человеком и заставляет разработчика использовать иерархии при описании сложных предметов. Нижний предел (три) гарантирует, что на соответствующей диаграмме достаточно деталей, чтобы оправдать ее создание.
- Ограничение количества подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг четырьмя.

Для моделирования бизнес-функции обычно достаточно 2-3 уровней детализации.

Общее число уровней в модели обычно не превышает 6-7.

Групповой процесс разработки моделей IDEF0

Первый этап. Поиск ответов на вопросы:

- 1) Что поступает в подразделение "на входе"?
- 2) Какие функции и в какой последовательности выполняются в рамках подразделения?
- 3) Кто является ответственным за выполнение каждой из функций?
- 4) Чем руководствуется исполнитель при выполнении каждой из функций?
- 5) Что является результатом работы подразделения (на выходе)?

На основе имеющихся положений, документов и результатов опросов создается черновик (Model Draft) модели.

Второй этап.

Распространение черновика для рассмотрения, согласований и комментариев. На этой стадии происходит обсуждение черновика модели с широким кругом компетентных лиц (в терминах IDEF0 — читателей) на предприятии.

Каждая из диаграмм черновой модели письменно критикуется и комментируется.

Автор также письменно соглашается с критикой или отвергает ее с изложением логики принятия решения.

Групповой процесс разработки моделей IDEF0

Третий этап. Официальное утверждение модели.

Утверждение согласованной модели происходит руководителем рабочей группы в том случае, если у авторов модели и читателей отсутствуют разногласия по поводу ее адекватности.

Окончательная модель представляет собой согласованное представление о предприятии (системе) с заданной точки зрения и для заданной цели.

Функциональная методика потоков данных

Цель методики - построение модели рассматриваемой системы в виде диаграммы потоков данных (Data Flow Diagram — DFD), обеспечивающей правильное описание выходов (отклика системы в виде данных) при заданном воздействии на вход системы (подаче сигналов через внешние интерфейсы).

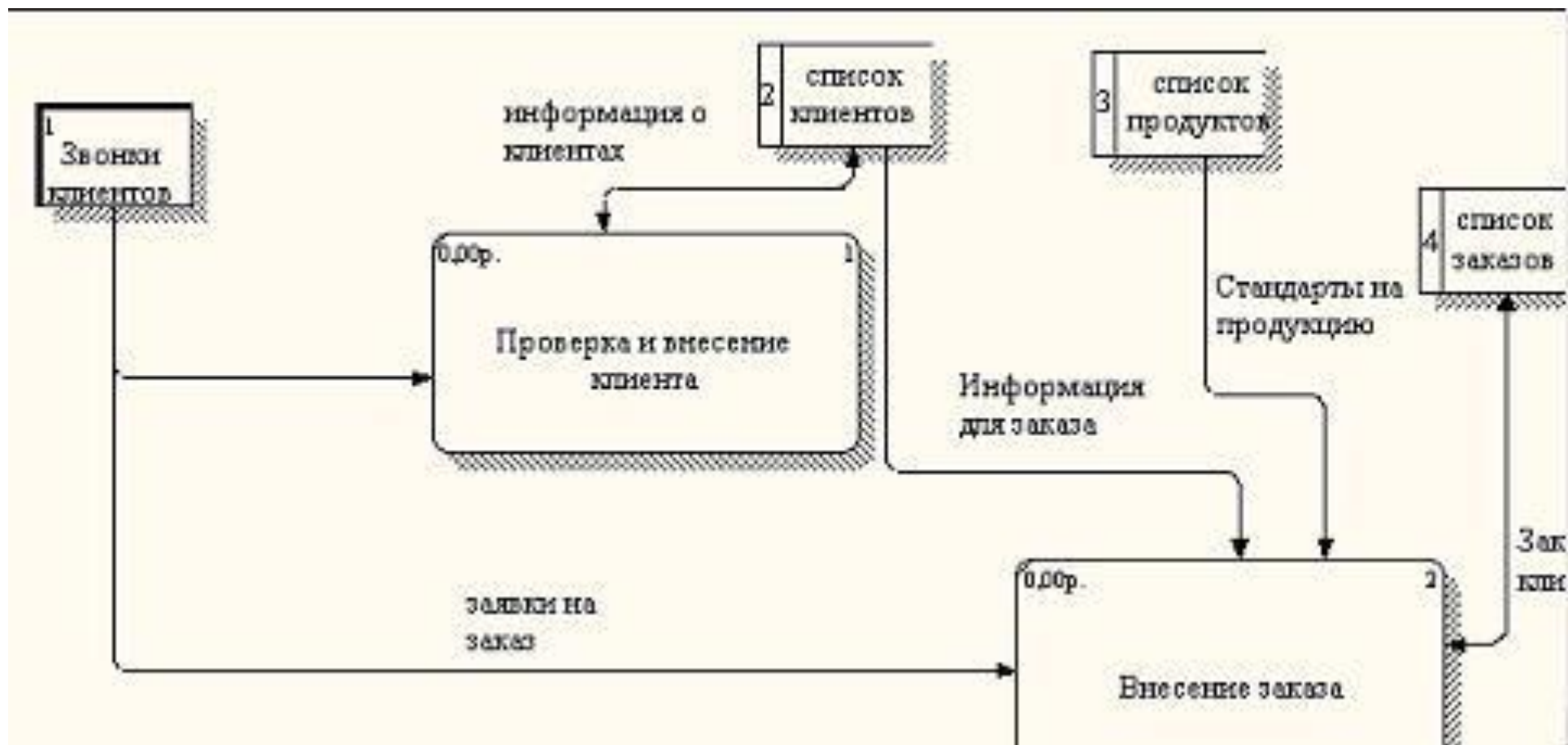
Диаграммы потоков данных являются основным средством моделирования функциональных требований к проектируемой системе.

Основные понятия модели потоков данных

При создании диаграммы потоков данных используются четыре основных понятия:

- **потоки данных,**
- **процессы (работы) преобразования входных потоков данных в выходные,**
- **внешние сущности,**
- **накопители данных (хранилища).**

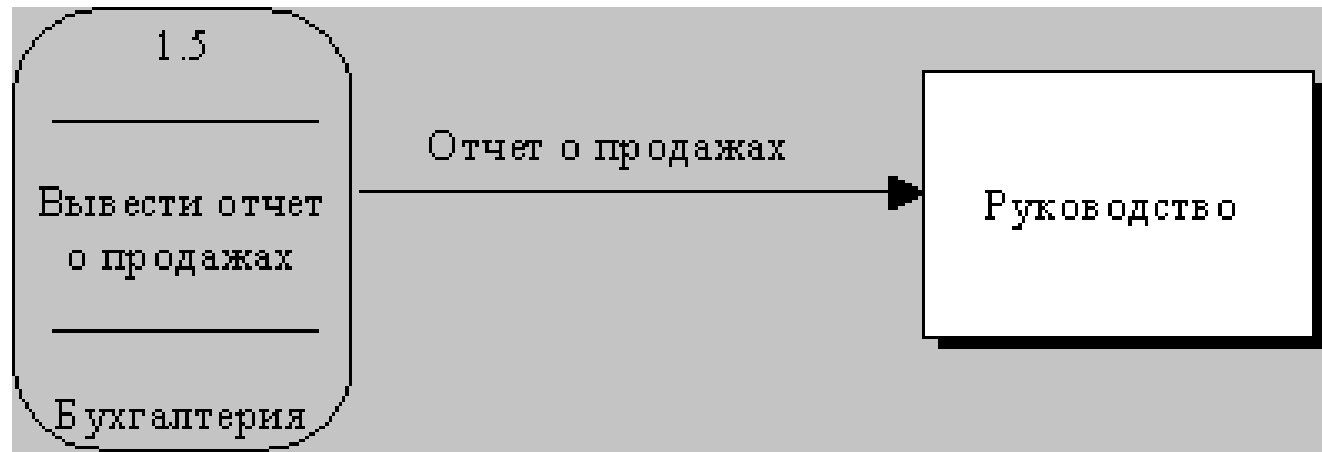
Пример модели потоков данных



Потоки данных в DFD

Потоки данных являются абстракциями, использующимися для моделирования передачи информации (или физических компонент) из одной части системы в другую.

Потоки на диаграммах изображаются именованными стрелками, ориентация которых указывает направление движения информации.



Процессы (работы) в DFD

Назначение **процесса** (работы) состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса.

Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением (например, "получить документы по отгрузке продукции").

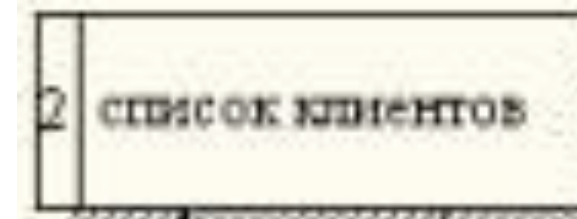
Каждый процесс имеет уникальный номер для ссылок на него внутри диаграммы, который может использоваться совместно с номером диаграммы.

Хранилища данных в DFD

Хранилище (накопитель) данных позволяет на указанных участках определять данные, которые будут сохраняться в памяти между процессами.

Информация, которую оно содержит, может использоваться в любое время после ее получения, при этом данные могут выбираться в любом порядке.

Имя хранилища должно определять его содержимое и быть существительным.

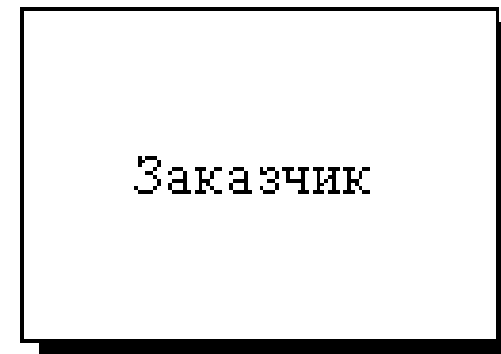


Внешние сущности в DFD

Внешняя сущность представляет собой материальный объект вне контекста системы, являющейся источником или приемником системных данных.

Имя внешней сущности должно содержать существительное, например, "склад товаров".

Предполагается, что объекты, представленные как внешние сущности, не должны участвовать ни в какой обработке.



Процесс построения DFD (шаг 1)

Шаг 1. Создание так называемой основной диаграммы типа "звезда", на которой представлен моделируемый процесс и все внешние сущности, с которыми он взаимодействует.

В случае сложного основного процесса он сразу представляется в виде декомпозиции на ряд взаимодействующих процессов.

Критериями сложности являются: наличие большого числа внешних сущностей, многофункциональность системы, ее распределенный характер.

Процесс построения DFD (внешние сущности)

Для определения внешних сущностей необходимо выделить поставщиков и потребителей основного процесса.

На этом этапе описание взаимодействия заключается в выборе глагола, дающего представление о том, как внешняя сущность использует основной процесс или используется им.

Например: основной процесс – "учет обращений граждан", внешняя сущность – "граждане", описание взаимодействия – "подает заявления и получает ответы".

Процесс построения DFD (таблица событий)

Для всех внешних сущностей строится таблица событий, описывающая их взаимодействие с основным потоком.

Таблица событий включает в себя:

- наименование внешней сущности,
- событие,
- тип события (типичный для системы или исключительный, реализующийся при определенных условиях),
- реакцию системы.

Процесс построения DFD (шаг 2)

Шаг 2. Происходит декомпозиция основного процесса на набор взаимосвязанных процессов, обменивающихся потоками данных.

Сами потоки не конкретизируются, определяется лишь характер взаимодействия.

Декомпозиция завершается, когда процесс становится простым: процесс имеет два-три входных и выходных потока; процесс может быть описан в виде преобразования входных данных в выходные; процесс может быть описан в виде последовательного алгоритма.

Процесс построения DFD (шаг 3)

Шаг 3. выделяются **потоки данных**, которыми обмениваются процессы и внешние сущности.

Происходит анализ таблиц событий. События преобразуются в потоки данных от инициатора события к запрашиваемому процессу, а реакции – в обратный поток событий.

После построения входных и выходных потоков аналогичным образом строятся внутренние потоки.

Процесс построения DFD (последний шаг)

Диаграмма проверяется на полноту и непротиворечивость.

Полнота диаграммы обеспечивается, если в системе нет "повисших" процессов, не используемых в процессе преобразования входных потоков в выходные.

Преимущества методики DFD

К преимуществам методики DFD относятся:

1. возможность однозначно определить внешние сущности, анализируя потоки информации внутри и вне системы;
2. возможность проектирования сверху вниз, что облегчает построение модели "как должно быть";
3. наличие спецификаций процессов нижнего уровня, что позволяет преодолеть логическую незавершенность функциональной модели и построить полную функциональную спецификацию разрабатываемой системы.

Недостатки методики DFD

К недостаткам методики DFD относят:

- необходимость искусственного ввода управляющих процессов, поскольку управляющие воздействия (потoki) и управляющие процессы с точки зрения DFD ничем не отличаются от обычных;
- отсутствие понятия времени, т.е. отсутствие анализа временных промежутков при преобразовании данных (все ограничения по времени должны быть введены в спецификациях процессов).