

Функциональное моделирование. Моделирование бизнес- процессов.

Методология SADT, IDEF0

- Методология SADT разработана Дугласом Россом. Исходная работа над SADT началась в 1969 г. В начале 70-х годов методология SADT была реализована в виде четкой формальной процедуры.
- На ее основе разработана, известная методология IDEF0 (Icam DEFINition), которая является основной частью программы ICAM (Интеграция компьютерных и промышленных технологий), проводимой по инициативе ВВС США.

- Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области.
- Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Основные элементы этой методологии основываются на
следующих концепциях:

1. 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. Взаимодействие блоков друг с другом описываются посредством интерфейсных дуг, выражающих "ограничения", которые в свою очередь определяют, когда и каким образом функции выполняются и управляются;

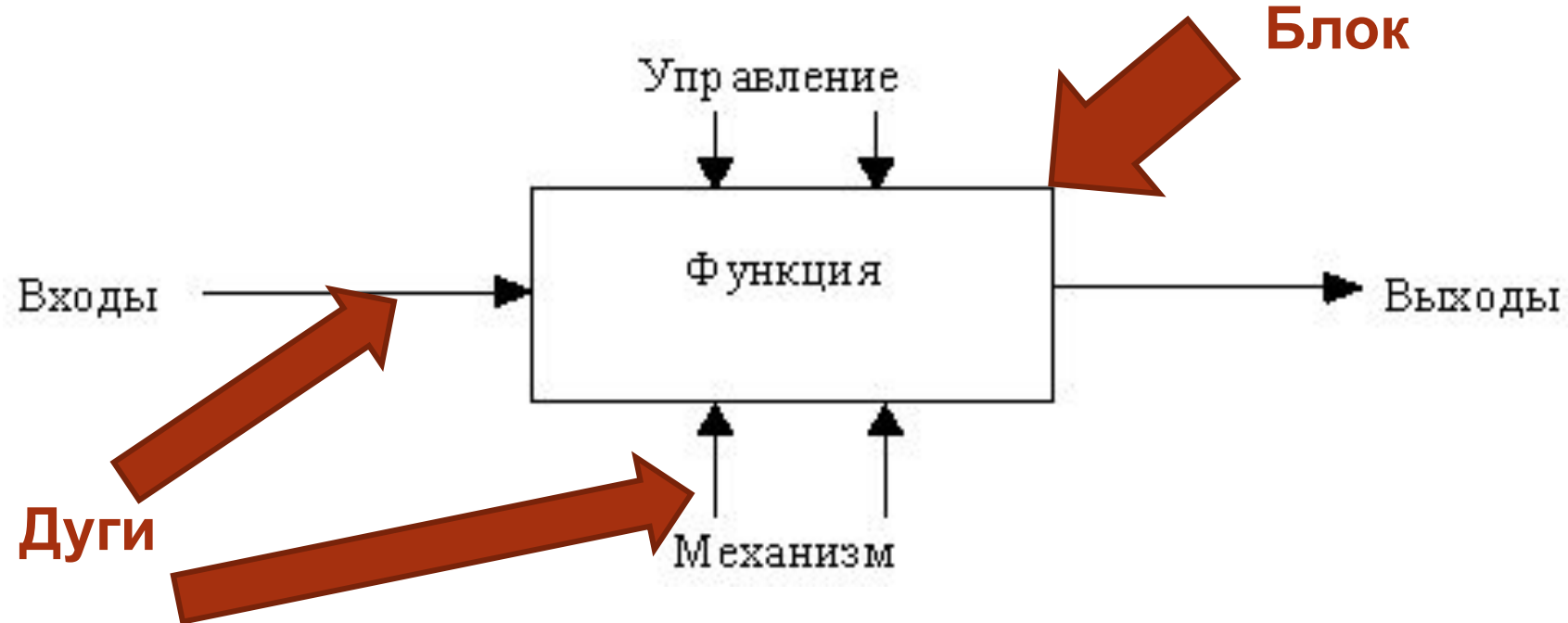
2. Строгость и точность.

Выполнение правил SADT требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика.

Правила SADT:

- 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.

Результатом применения методологии SADT является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга.

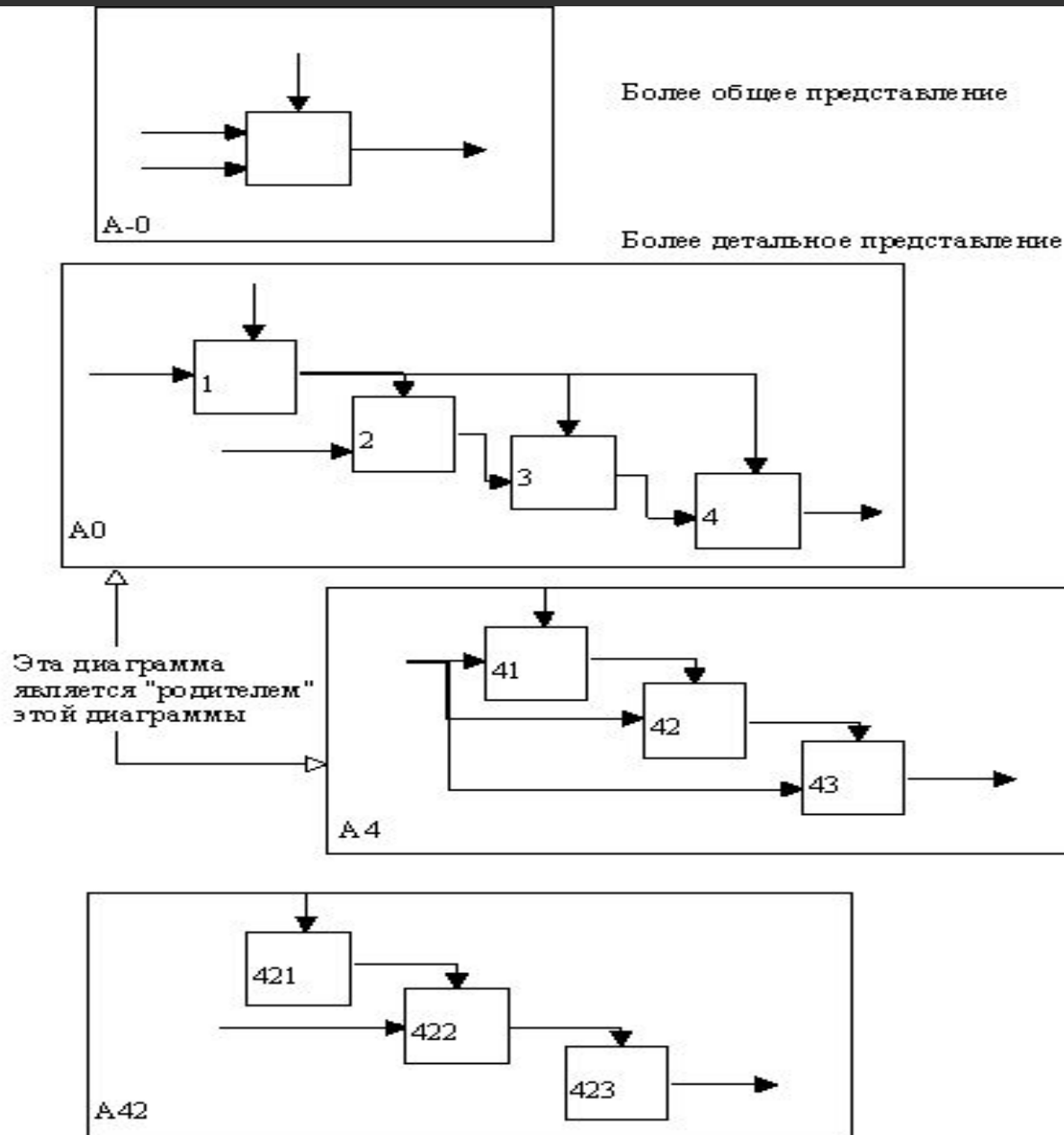


Диаграммы - главные компоненты модели, все функции ИС и интерфейсы на них представлены как блоки и дуги.

Место соединения дуги с блоком определяет тип интерфейса.

Управляющая информация входит в блок сверху  в то время как информация, которая подвергается обработке, показана с левой стороны блока, а результаты (выходы) показаны с правой стороны. 

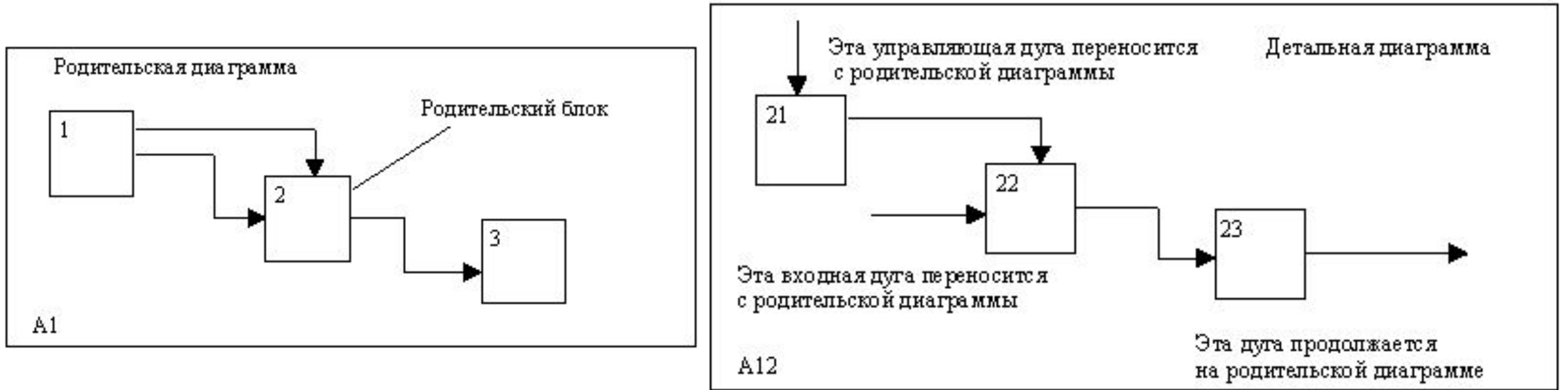
Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу. 



Построение SADT-модели начинается с одного блока и дуг, изображающих интерфейсы с функциями вне системы.

Затем данный блок детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки представляют основные подфункции исходной функции. Каждая из этих подфункций может быть декомпозирована подобным образом.

Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию.



Типы связей между функциями

(0) Тип случайной связности: наименее желательный.

Случайная связность возникает, когда конкретная связь между функциями мала или полностью отсутствует. Это относится к ситуации, когда имена данных на SADT-дугах в одной диаграмме имеют малую связь друг с другом.

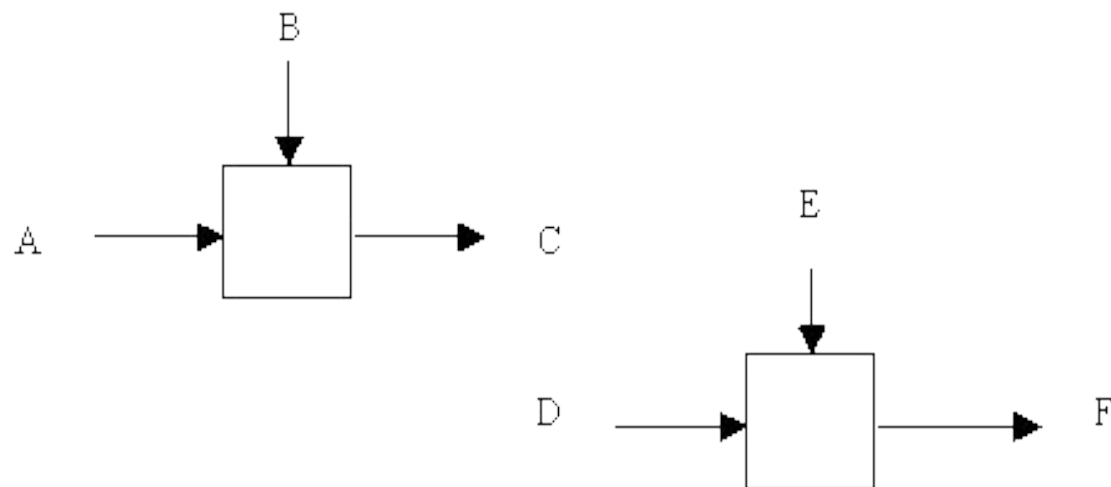


Рис. 1. Случайная связность

(1) Тип логической связности

Логическое увязывание происходит тогда, когда данные и функции собираются вместе благодаря тому, что они попадают в общий класс или набор элементов, но необходимых функциональных отношений между ними не обнаруживается.

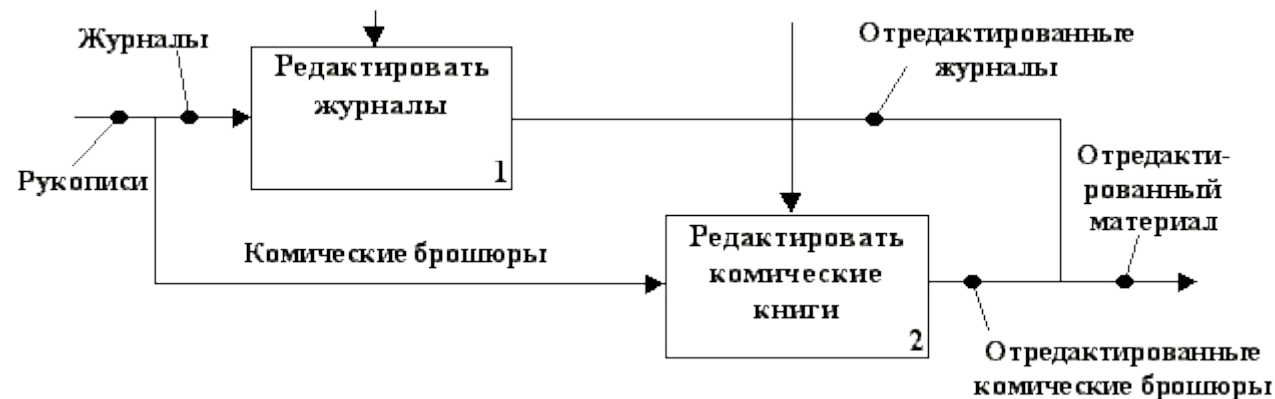
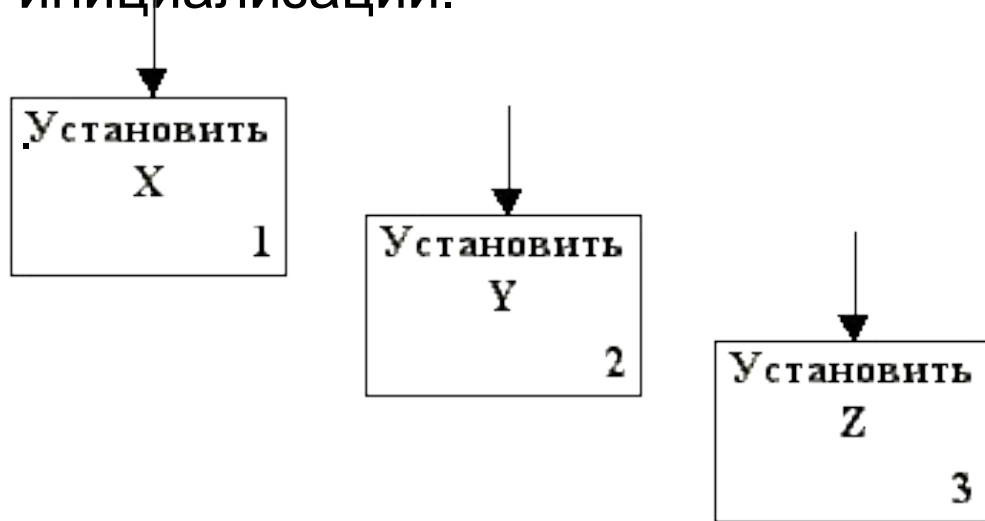


Рис. 2. Логическая связность

Типы связей между функциями

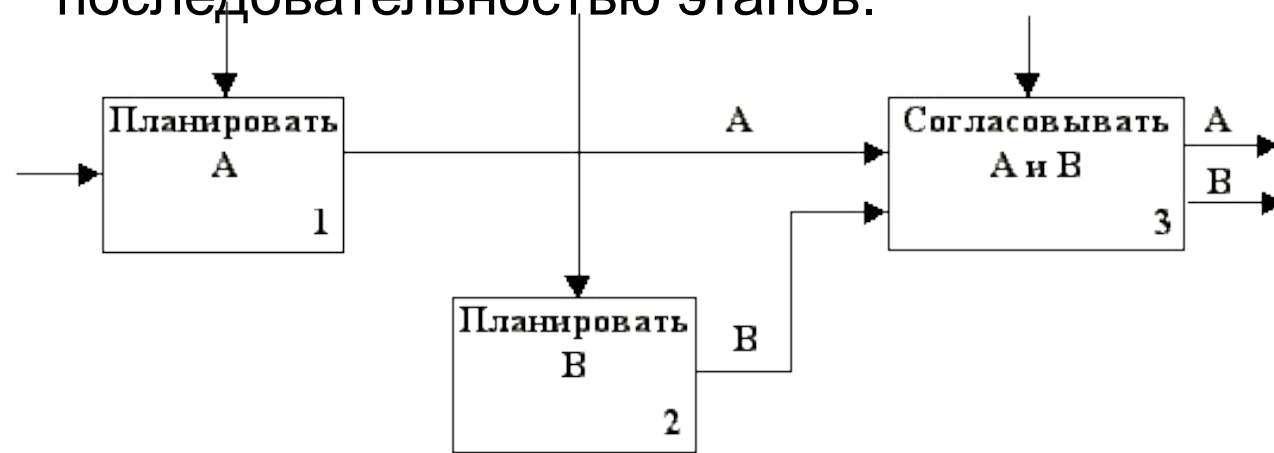
(2) Тип временной связности. Связанные по времени элементы возникают вследствие того, что они представляют функции, связанные во времени, когда данные используются одновременно или функции включаются параллельно, а не последовательно. Часто такой пример можно найти на диаграмме, показывающей функции, касающиеся операций инициализации.



Временная связность

(3) Тип процедурной связности

Процедурно-связанные элементы появляются сгруппированными вместе благодаря тому, что они выполняются в течение одной и той же части цикла или процесса. Общий элемент процесса может быть итерационным или ветвью, зависящей от принятого решения, или линейной последовательностью этапов.

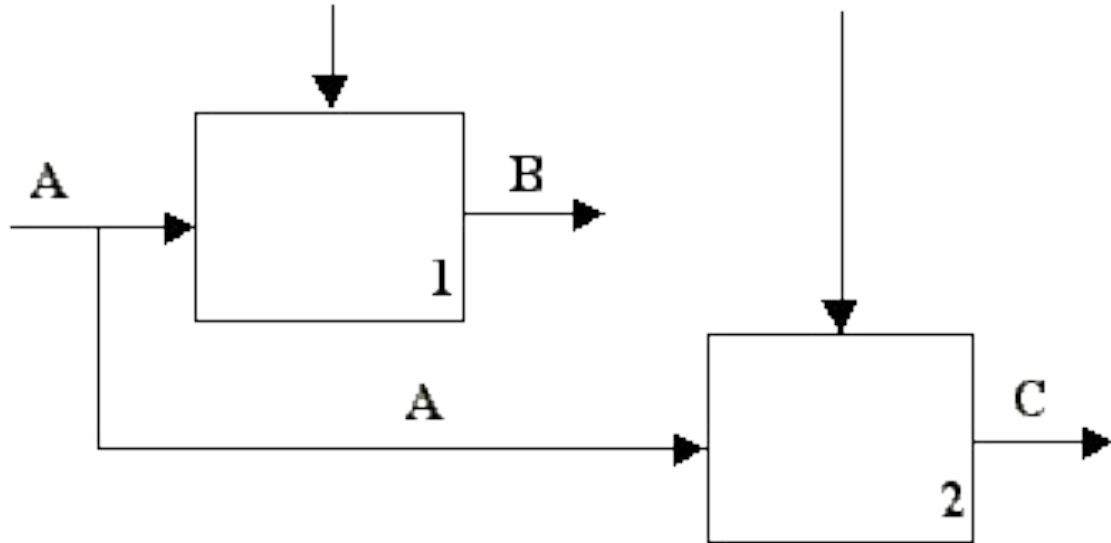


Процедурная связность

Типы связей между функциями

(4) Тип коммуникационной связности.

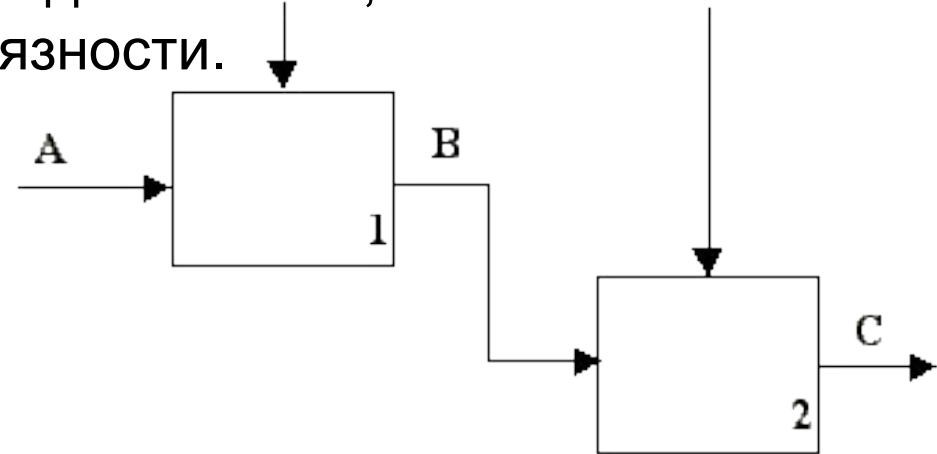
Диаграммы демонстрируют коммуникационные связи, когда блоки группируются вследствие того, что они используют одни и те же входные данные и/или производят одни и те же выходные данные.



Коммуникационная связность

(5) Тип последовательной связности

На диаграммах, имеющих последовательные связи, выход из одной функции служит входными данными для следующей функции. Связь между элементами на диаграмме является более тесной, чем на рассмотренных выше уровнях связей, поскольку моделируются причинно-следственные зависимости. Следовательно, это высокая степень связности.

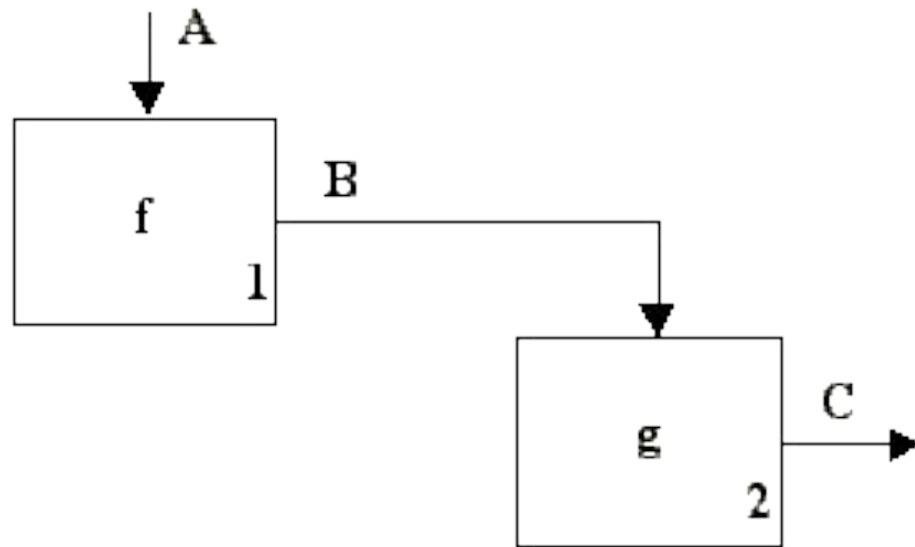


Последовательная связность

Типы связей между функциями

(6) Тип функциональной связности

Диаграмма отражает полную функциональную связность, когда все элементы функции влияют на выполнение одной единственной функции или результата. Диаграмма, которая является чисто функциональной, не содержит чужеродных элементов, относящихся к последовательному или более слабому типу связности. Одним из способов определения функционально-связанных диаграмм является рассмотрение двух блоков, связанных через управляющие дуги, как показано на рис.



Функциональная связность

Правила определения типов связей между функциями

Если существует единственно приемлемый путь описания диаграмм с использованием сложного предложения, предложения, содержащего запятую, или предложения, содержащего более одного глагола, то менее вероятно, что диаграмма является функциональной. Она, видимо, последовательная, коммуникационная или логическая в терминах связности.

Если описывающее предложение содержит такие временные ориентиры, как "первый", "следующий", "после", "затем", "начать", "когда", "до тех пор, пока" или "для всех", то диаграмма имеет временную или процедурную связность; реже такие слова обозначают последовательную связность.

Если утверждение описывающего предложения не содержит одного конкретного объекта, следующего за глаголом, диаграмма является, вероятно, логически связанной.

Моделирование бизнес- процессов. Методология IDEF0

Бизнес-процесс – это последовательность взаимосвязанных активностей или задач, которые приводят к созданию определённого продукта или услуг для потребителей.

Выделяют три вида бизнес-процессов:

Управляющие – это бизнес-процессы, которые управляют функционированием системы. Примером управляющих бизнес-процесса может служить корпоративное управление и стратегический менеджмент.

Операционные – это бизнес-процессы, которые составляют основной бизнес компании и создают основной поток доходов. Примером операционных бизнес-процессов является: снабжение, производство, маркетинг и продажа.

Поддерживающие – это бизнес-процессы, которые обслуживают основной бизнес. Примером поддерживающих бизнес-процессов являются: бухгалтерский учёт, подбор персонала, техническая поддержка.

Бизнес-процесс может быть декомпозирован на несколько подпроцессов, которые имеют собственные атрибуты, однако также направлены на достижение цели основного бизнес-процесса.

Бизнес-процессы могут подвергаться моделированию с помощью различных методов. Одним из способов является составление модели бизнес-процесса «как есть» (англ. as is). После этого модель бизнес-процесса подвергается критическому анализу или обрабатывается специальным программным обеспечением. В результате строится модель бизнес-процесса «как должно быть» (англ. to be). Некоторые консультанты опускают фазу «как есть» и сразу предлагают модель «как должно быть».

Важнейшим этапом в процессе разработки сложных систем является этап системного анализа и моделирование соответствующей проектной области, его цель заключается в разработке спецификации проекта технического задания (ТЗ) на разработку проекта.

Цель построения модели некоторого процесса предметной области - специфицирование операций и действий, выполняемых в процессе предметной области и взаимодействий между ним. Существует множество методов и техник применяемых для моделирования бизнес-процессов. Одной из самых используемых является методология функционального моделирования IDEF0.

Методология IDEF0 следующий этап развития хорошо известного графического языка описания функциональных систем *SADT*. Исторически *IDEF0* как стандарт был разработан в 1981 году.

Основополагающими понятиями IDEF0 – методологии является цель моделирования, точка зрения и субъект моделирования.

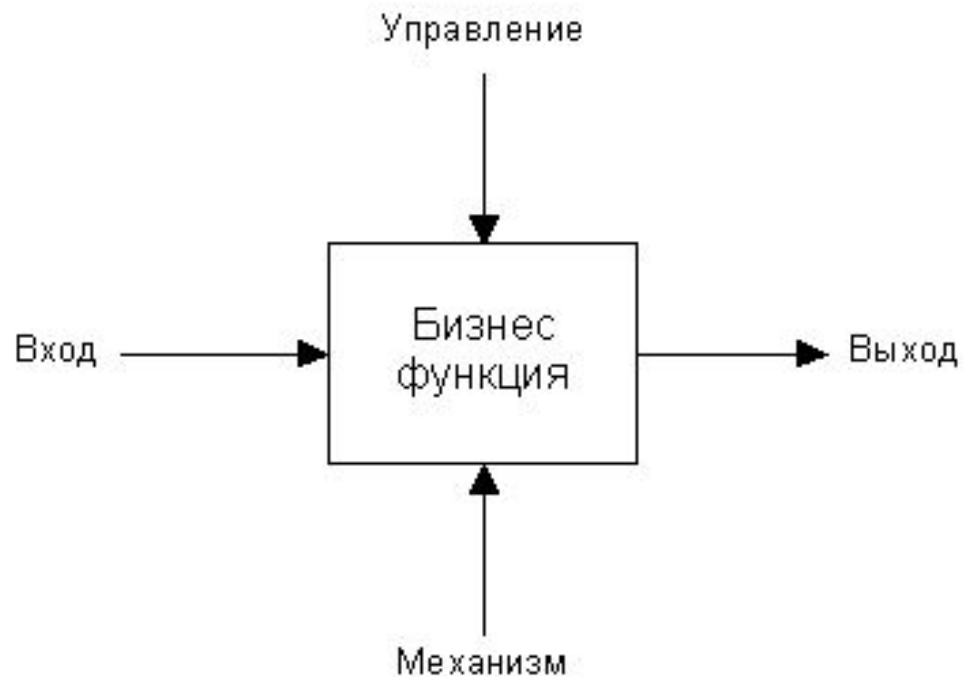
Точка зрения – это представление о системе с позиции некоторого участника процесса. Например, при моделировании некоторого производственного процесса, может использоваться точка зрения руководителя производства, технолога, рабочего, экономиста, юриста, специалиста по качеству и др. Точка зрения должна соответствовать цели моделирования.

Цель моделирования определяет степень детализации разрабатываемой модели.

Субъект моделирования определяет границы моделируемой системы и окружающей ее внешней среды.

IDEF0 – модель предполагает наличие четко сформулированной цели, единственного субъекта моделирования и одной точки зрения.

Основной структурной единицей IDEF0 модели является диаграмма представляющая собой графическое окно модели предметной области или её части. Каждая **IDEF0-диаграмма** содержит блоки и дуги. Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними. Блоки именуются **глаголами или глагольными оборотами. Дуги – существительными.**



Вход – материал или информация, которые используются или преобразуются блоком для получения результата (выхода). Блок может не иметь ни одной входной дуги. Данный вид дуги поступает на левую сторону блока.

Управление – условия, правила, стратегии, стандарты, которые влияют на выполнение функции. Каждый блок должен иметь хотя бы одну дугу управления. Данный вид дуг поступает на верхнюю сторону блока.

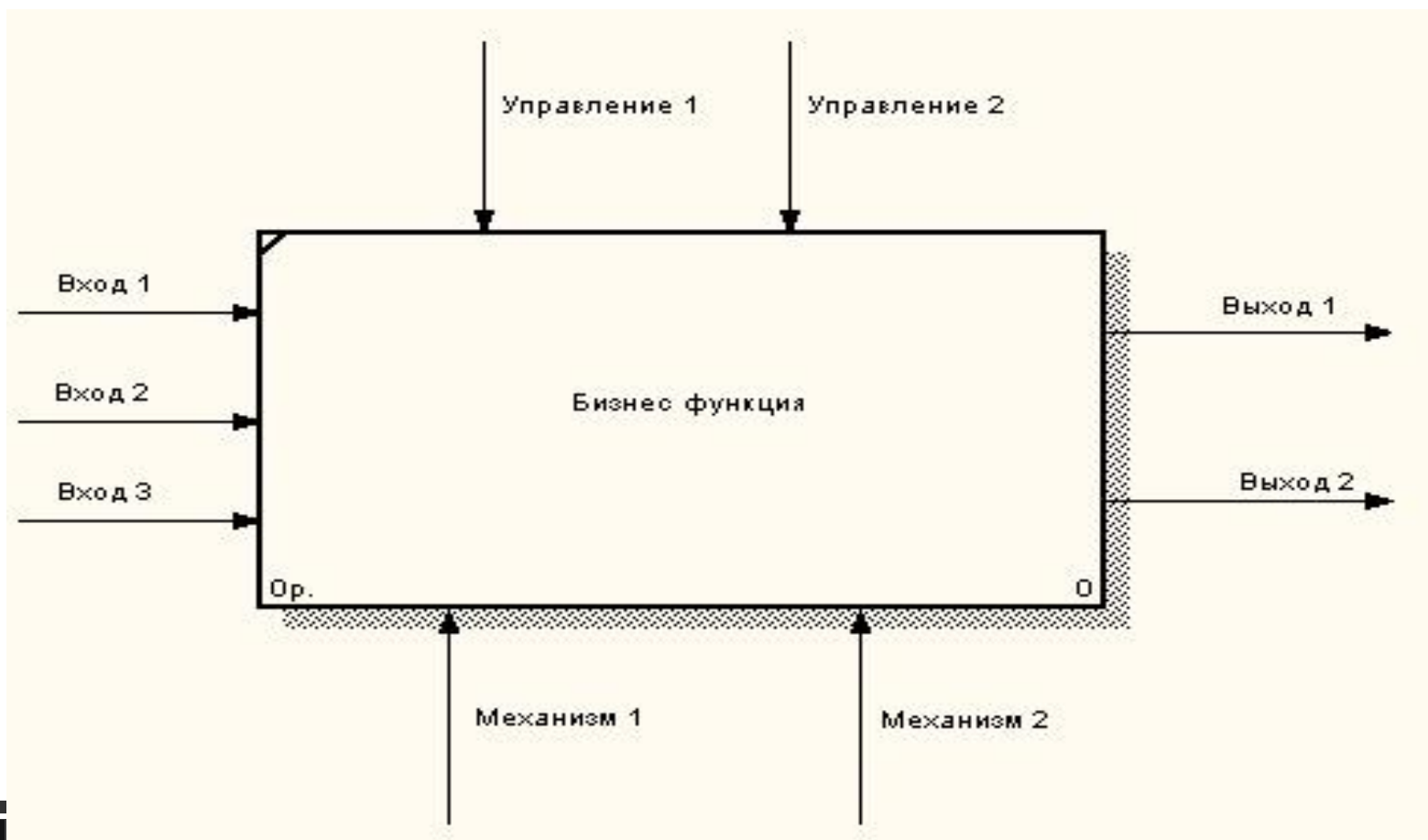
Выход – результат выполнения функции (материал или информация). Каждая функция должна иметь хотя бы одну выходную дугу. Данный вид дуг выходит из правой стороны блока.

Механизм – ресурсы, с помощью которых выполняется работа. Это могут быть, например, денежные средства, персонал предприятия, станки и другое оборудование. Данный вид дуг поступает на нижнюю сторону блока.

Базовые принципы моделирования IDEF0

1. Принцип контекстной диаграммы.

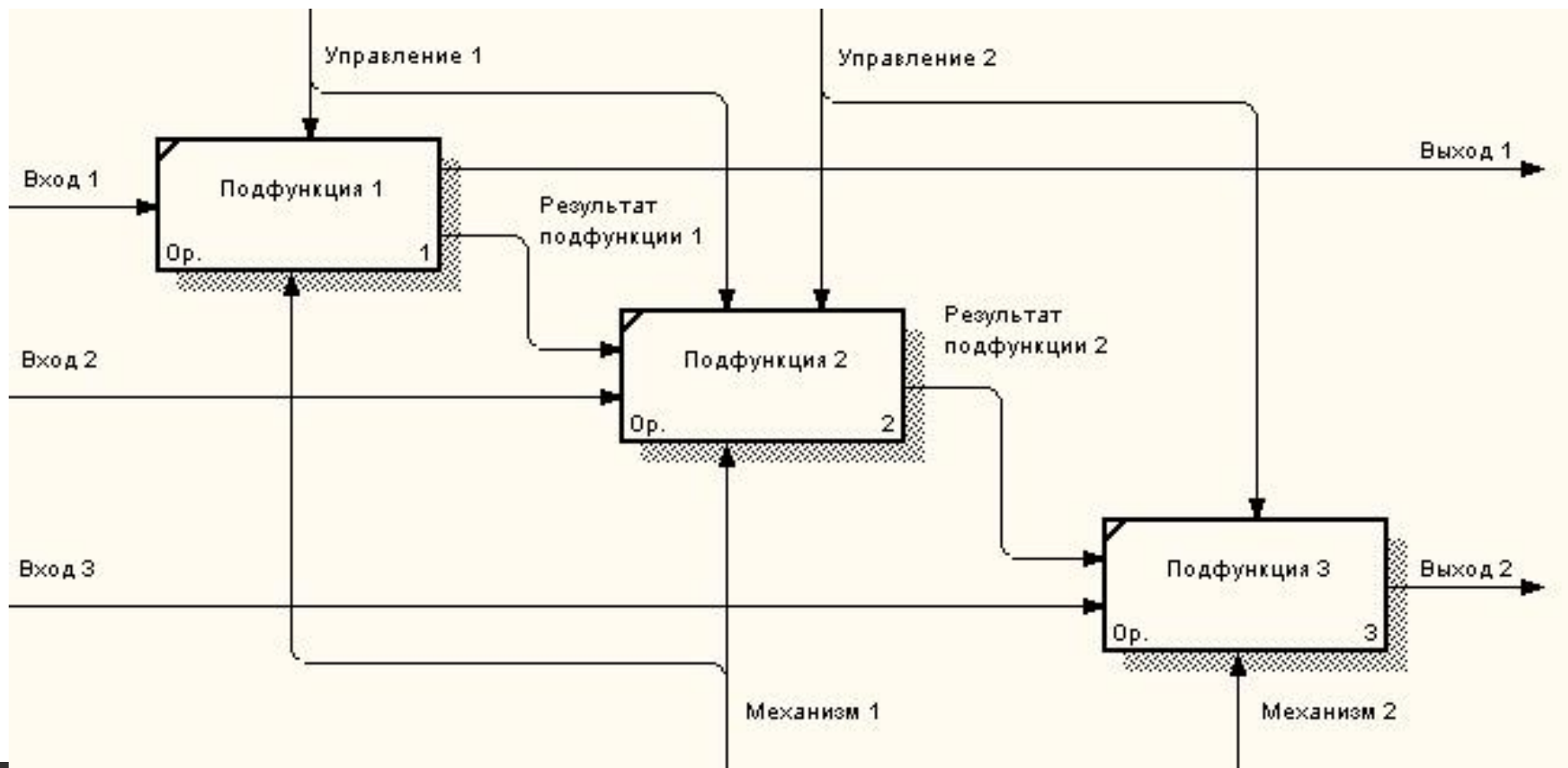
Заключается в том, что моделирование предметной области начинается с построения контекстной диаграммы. На этой диаграмме изображается один блок, представляющий собой главную функцию моделируемой системы и определяющий границы системы.



Базовые принципы моделирования IDEF0

2. Принцип функциональной декомпозиции.

Представляет собой разбиение действий, операций, функций предметной области на более простые действия, операции, функции. В результате сложная бизнес – функция представляется совокупностью более простых функций, которые в свою очередь также могут быть декомпозированы на более простые функции.



Базовые принципы моделирования IDEF0

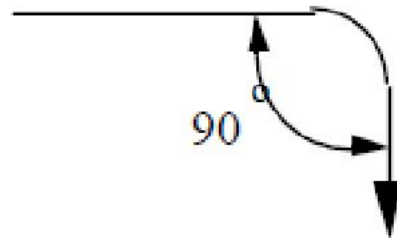
3. Принцип ограничения сложности.

Обеспечивает понятность и удобочитаемость IDEF0 – диаграмм. Количество блоков на диаграмме должно быть не менее трех и не больше шести.

Синтаксические правила IDEF0



Прямолинейный отрезок стрелки



- Ломаный сегмент стрелки.
Дуга сопряжения -90 град.



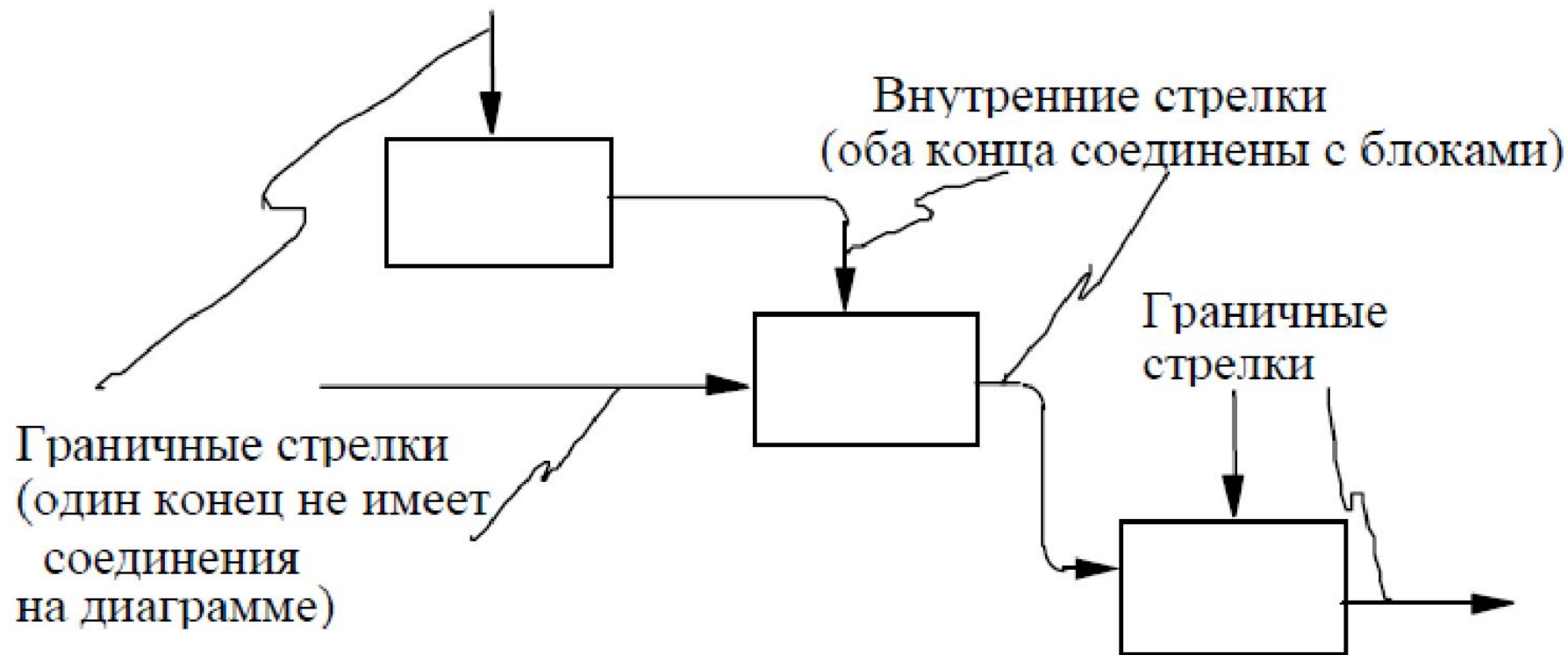
Ветвление стрелок



- Слияние стрелок

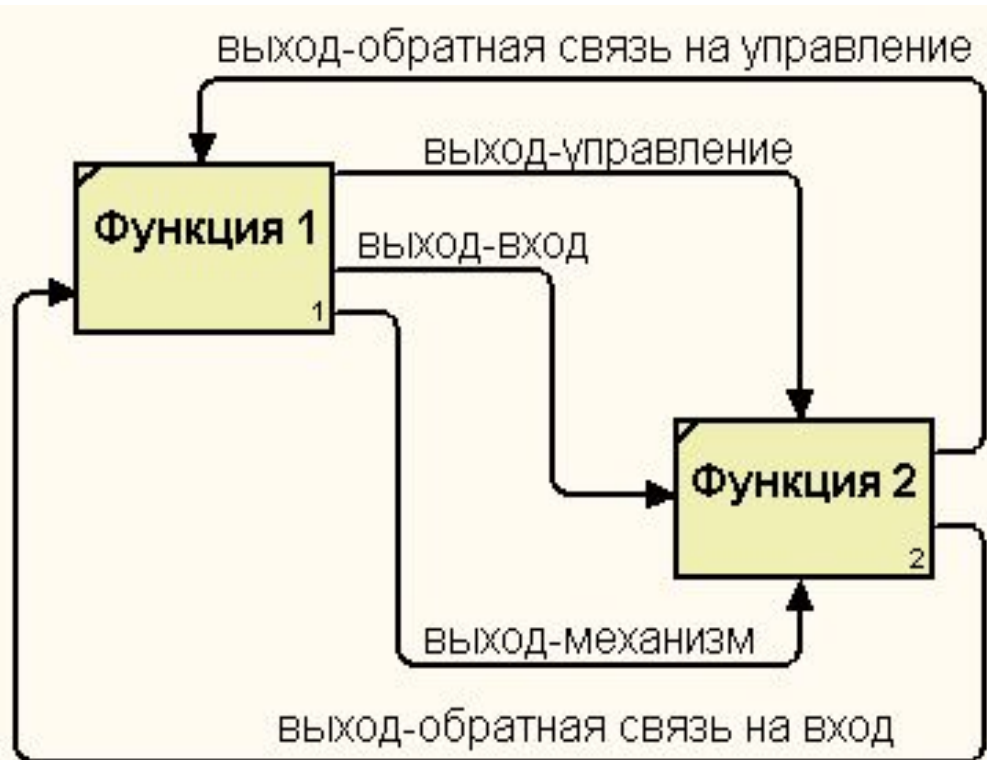
Граничные и внутренние дуги

На обычной (не контекстной) диаграмме граничные стрелки представляют входы, управления, выходы или механизмы родительского блока диаграммы. Источник или потребитель граничных стрелок можно обнаружить, только изучая родительскую диаграмму.



Основные виды связей между функциональными блоками.

В IDEF0 различают пять типов связей работ, три из них прямые и две обратные (внутренние дуги).



1. Связь по входу (output-input). Данная связь возникает, когда стрелка выхода вышестоящей работы (далее - просто выход) направляется на вход нижестоящей.

2. Связь по управлению (output-control). Данная связь возникает, когда выход вышестоящей работы направляется на управление нижестоящей. Связь по управлению показывает доминирование вышестоящей работы

3. Связь выход-механизм (output-mechanism). Данная связь возникает, когда выход одной работы направляется на механизм другой. Эта взаимосвязь показывает, что одна работа подготавливает ресурсы, необходимые для проведения другой работы

5. Обратная связь по входу (output-input feedback). Данная связь возникает, когда выход нижестоящей работы направляется на вход вышестоящей. Такая связь, как правило, используется для описания циклов (если выявлен брак, например).

4. Обратная связь по управлению (output-control feedback). Данная связь возникает, когда выход нижестоящей работы направляется на управление вышестоящей. Обратная связь по управлению часто свидетельствует об эффективности бизнес-процесса.

Основные виды связей между функциональными



1

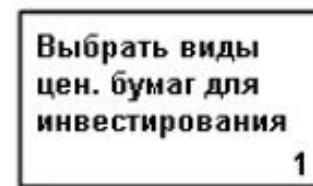
3



Зажим

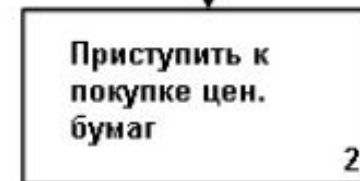
5

Некачественная
деталь

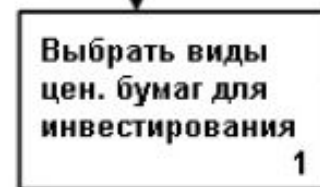


Принципы формирования
инвестиционного портфеля

2

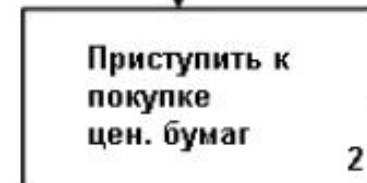


Информация о текущих курсах



Принципы формирования
инвестиционного портфеля

4



Приступить к
покупке
цен. бумаг

Слияние стрелок

- § Функция производит объекты, которые используются в нескольких других функциях.
- § Объекты, полученные в результате работы нескольких функций, объединяются в один общий поток.



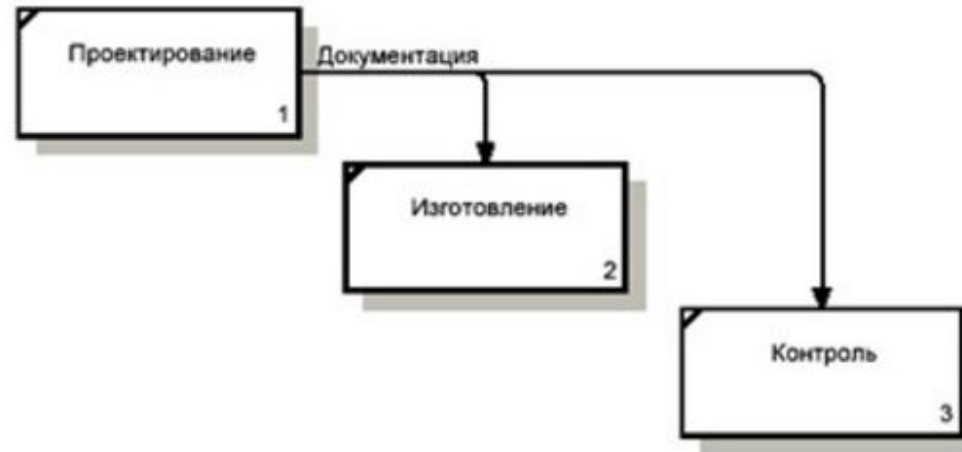
- § Случай когда какой либо **однотипный** результат получается от двух различных функций. Достаточно отметить только **общую** часть стрелки.



- § Два **различных** выхода сливаются в один общий. Должны быть отмечены **каждая ветвь** и **общий** участок связи.

Разветвление

- § Поток разветвляясь сохраняет первоначальное содержание.
- § Подпись необходима только для общей части стрелки.



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



Применение туннелей

§ Применяются когда:

- хотят чтобы стрелка используемая только начиная с какого либо уровня не присутствовала на всех промежуточных уровнях декомпозиции. Это помогает освободить промежуточные диаграммы от неиспользуемых стрелок.
- необходимо скрыть граничную стрелку на диаграмме декомпозиции.



•Функциональная структура бизнес-процесса «Увольнение

сотрудника»



Диаграмма А-0 (контекстная диаграмма) бизнес-процесса «Увольнение сотрудника»

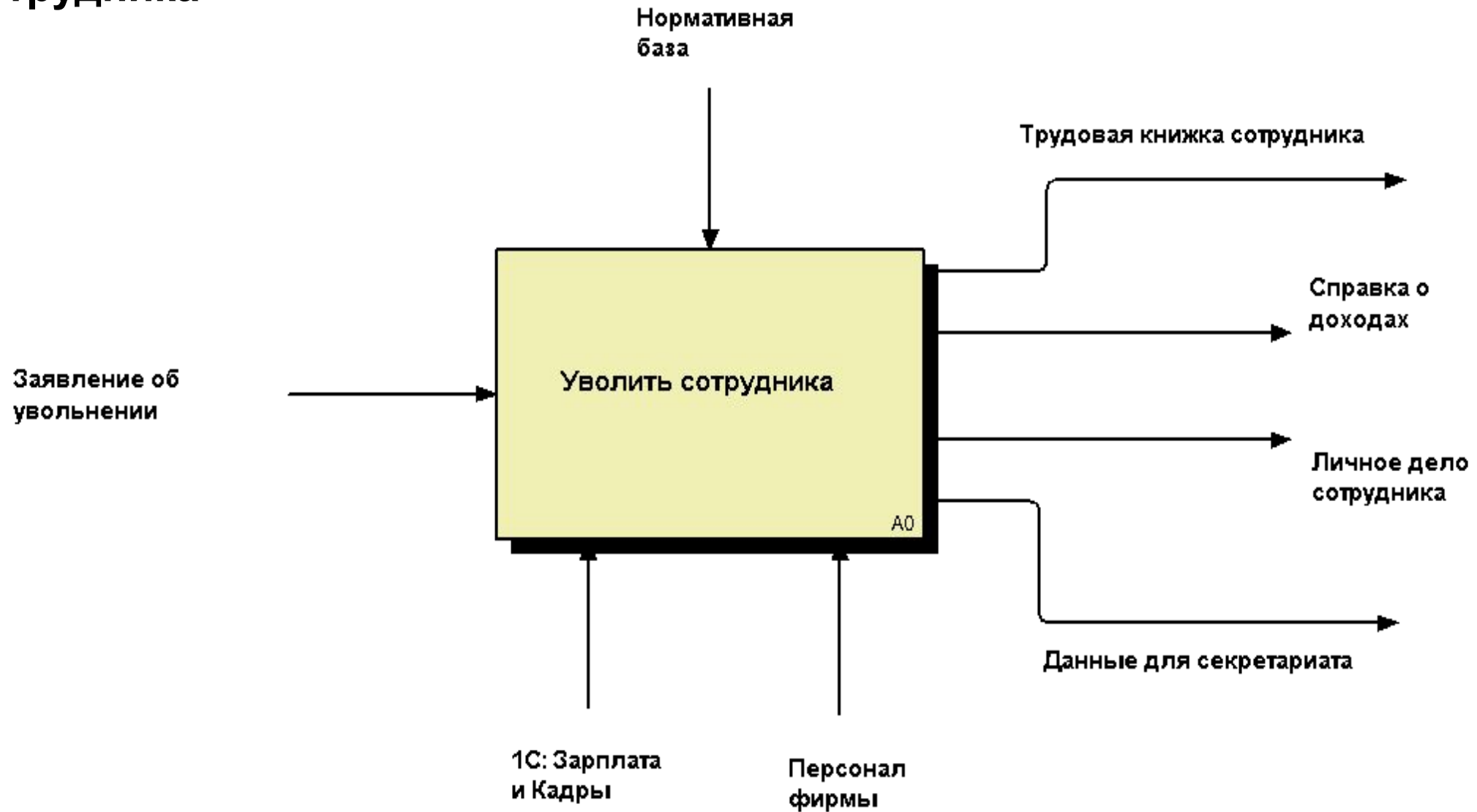


Диаграмма A0 (схема верхнего уровня) бизнес-процесса «Увольнение сотрудника»

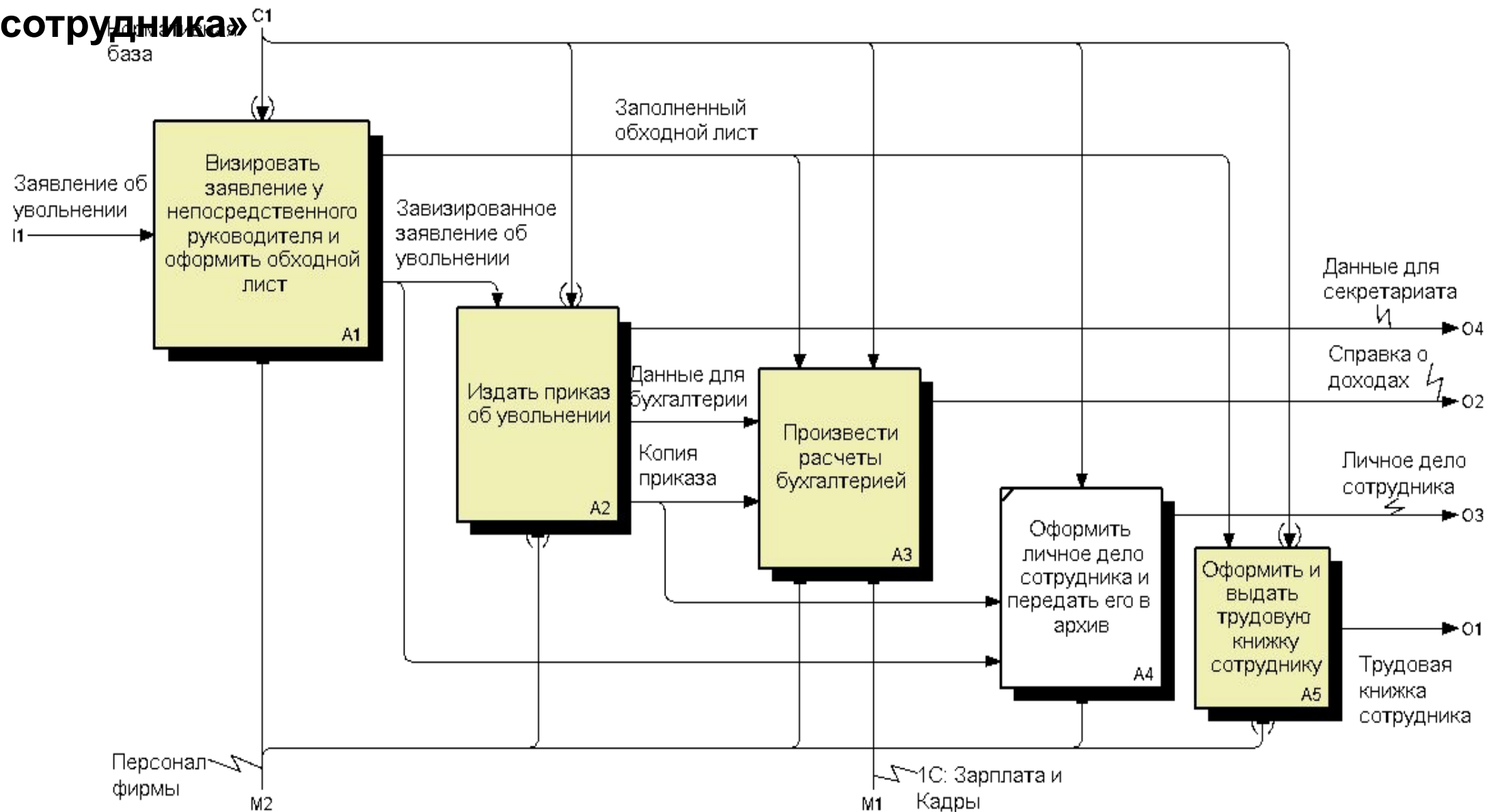


Диаграмма A1 бизнес-процесса «Визирование заявления у непосредственного руководителя и оформление обходного листа»

