

Проектирование информационных систем

Функциональное проектирование

Основные вопросы

- IDEF
- IDEF0

История

Составные части

IDEF

Определения

- IDEF – сокращение от:
 - Изначально: *ICAM* (Integrated Computer-Aided Manufacturing) *Definition*
 - С 1999 года: *Integration Definition*
- IDEF – набор методологий моделирования сложных систем. Разные методологии семейства позволяют отображать и анализировать деятельность систем с различных точек зрения. Ключевые составляющие:
 - Использование визуальных моделей
 - Иерархичность – позволяет:
 - Описывать сложные системы (правило 7 ± 2)
 - Останавливаться на разном уровне глубины обследования при анализе
 - «Общеприменимость»

История

- Методологии IDEF создавались в рамках предложенной ВВС США (как обычно ^_^) программы компьютеризации промышленности — ICAM в 1970х – 1980х годах
- В 1990х годах основные методологии стали широко использоваться в ВВС США → МО США → гражданских приложениях
- 1991 – National Institute of Standards and Technology (NIST) начал формировать стандарт Federal Information Processing Standard (FIPS)
- 2008 – стандарт был отозван -_-

Состав группы – 1/2

- IDEF0 – описание функций системы
- IDEF1 – построение информационной модели
- IDEF1X – построение модели данных
- IDEF2 – построение имитационных моделей систем
- IDEF3 – моделирование бизнес-процессов
- IDEF4 – объектно-ориентированное представление системы
- IDEF5 – создание онтологического описания
- IDEF6 – обоснование принятых проектных решений

Состав группы – 2/2

- IDEF7 – аудит ИС (information system auditing)
- IDEF8 – описание пользовательских интерфейсов
- IDEF9 – описание ограничений предметной области (бизнес-ограничений)
- IDEF10 – описание реализации архитектуры
- IDEF11 – проектирование документов (information artifact modeling)
- IDEF12 – описание организационных структур
- IDEF13 – Three schema mapping design
- IDEF14 – проектирование сети

Статус

- На 1995 год только IDEF0, IDEF1X, IDEF2, IDEF3 и IDEF4 были полностью разработаны
- Часть методологий описана на уровне черновика. Например, описаны методы:
 - Обнаружения ограничений предметной области IDEF9
 - Формулирования обоснований архитектурных решений IDEF6
 - Описаний взаимодействия с человеком IDEF8
 - Описания сетей IDEF14
- Методы IDEF7, IDEF10, IDEF11, IDEF 12 и IDEF13 не определены дальше начальных формулировок

Статус в РФ

- IDEF0 задается документом РД IDEF0 – 2000 «Методология функционального моделирования. Руководящий документ»

История

Составные части

IDEFO

Основные понятия

- Блок или Функциональный блок
- Стрелка или Интерфейсная дуга
- Декомпозиция
- Глоссарий

Функциональный блок – 1/2

- Представляет модель функции (работы) описываемой системы
- Определение из РД IDEF0 – 2000: прямоугольник, содержащий имя и номер и используемый для описания функции

Функциональный блок – 2/2

Стороны блока имеют стандартное значение с точки зрения связи блок/стрелки

управление

Управлять
предприятием

A0

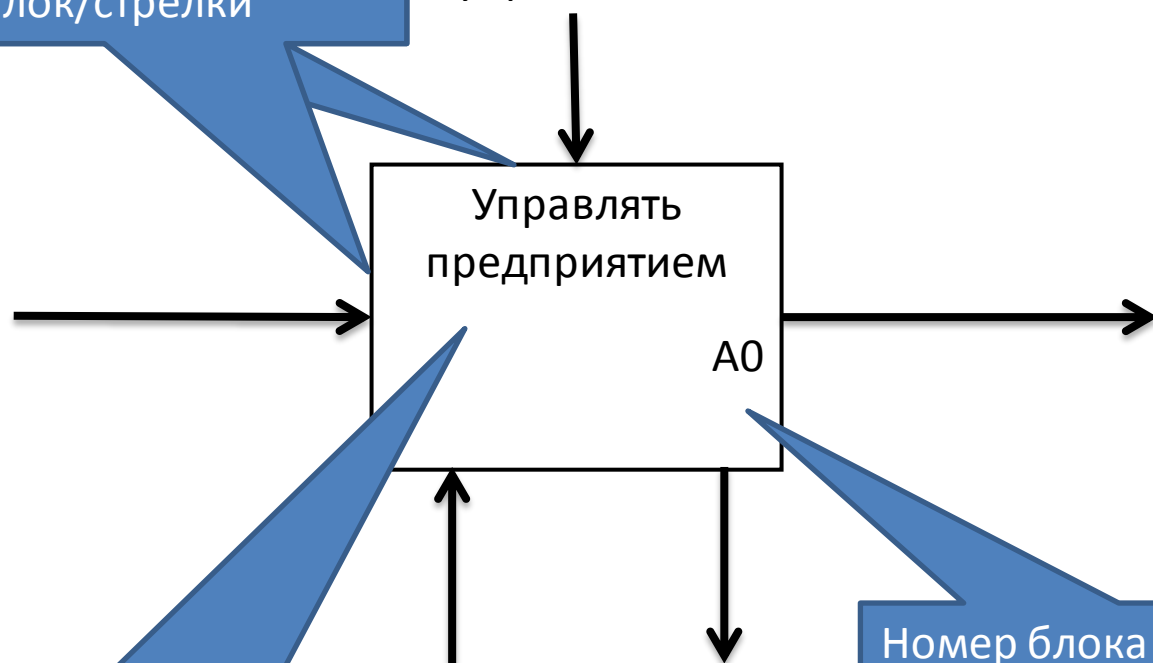
ВХОД

ВЫХОД

механизм вызов

Номер блока
Уникален в рамках системы

Имя
Активный глагол или глагольный оборот, описывающий функцию



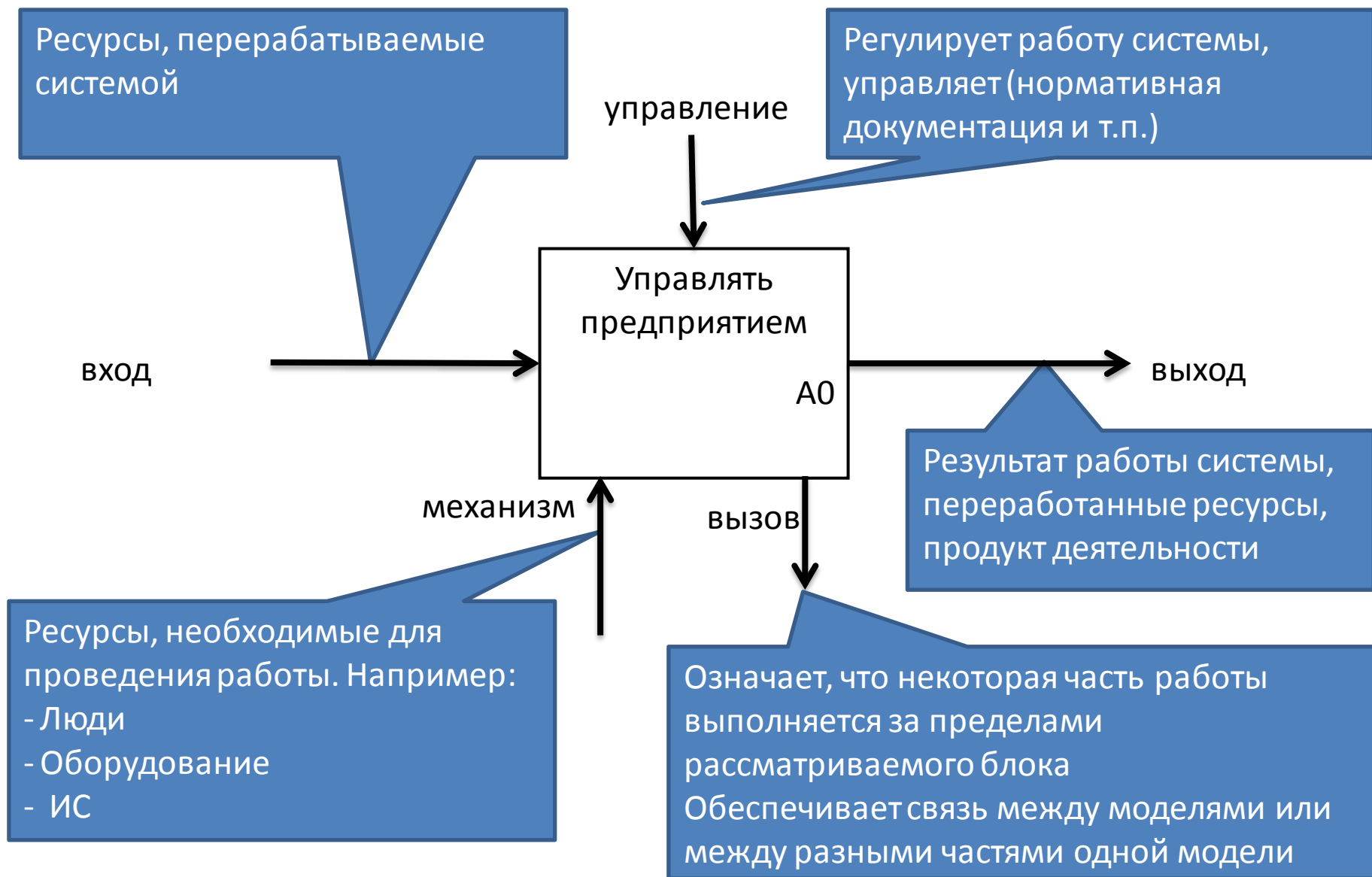
Именованние элементов

- Имена функций – глаголы или глагольные обороты. Примеры:
 - Производить детали
 - Наблюдать
 - Изготовить компонент
- Стрелки обозначают данные или материальные объекты (используемые, производимые, обрабатываемые, etc.). Имена стрелок – существительные или обороты существительных:
 - Отчет об испытаниях
 - Бюджет
 - Инженер-конструктор
 - Деталь

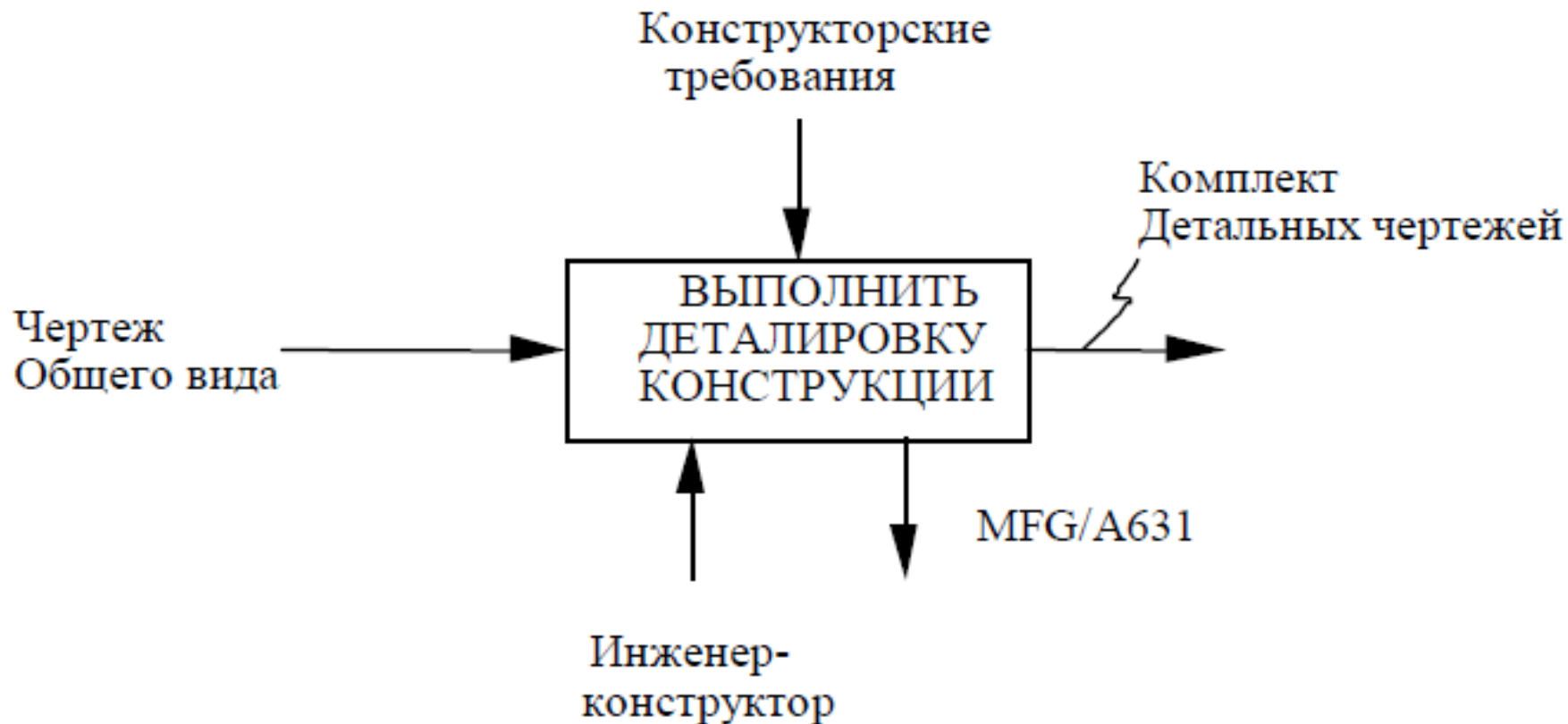
Интерфейсная дуга – 1/7

- Интерфейсная дуга отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, отображаемую функциональным блоком
- Графически изображается в виде стрелки
- Каждая дуга имеет уникальное название
- В зависимости от того, к какой стороне блока она подходит, интерфейсная дуга будет являться входящей, выходящей, управления, механизма, вызова
- Может не быть стрелки входа. Остальные интерфейсные дуги обязательны

Интерфейсная дуга – 2/7



Интерфейсная дуга – 3/7



Интерфейсная дуга – 4/7

- Вход – материал или информация, которые используются и преобразуются работой для получения результата (выхода)
 - Отвечает на вопрос «Что подлежит обработке?»
 - Входом может быть:
 - Материальный объект (сырье, деталь, экзаменационный билет)
 - Нематериальный Не имеющий физического воплощения (запрос к БД, вопрос преподавателя)
 - Работа может не иметь ни одной стрелки входа. Тогда предполагается, что она активируется как-то иначе
 - Рисуеться входящей в левую грань работы

Интерфейсная дуга – 5/7

- Управление – управляющие, регламентирующие и нормативные данные, которыми руководствуется работа
 - Отвечает на вопрос «В соответствии с чем выполняется работа?»
 - Управление влияет на работу, но не преобразуется ей
 - выступает в качестве ограничения. Примеры:
 - Правила
 - Стандарты
 - Нормативы
 - Расценки
 - Устные указания
 - Рисуеться входящей в верхнюю грань работы
 - Когда непонятно, «управление или вход?», то рисуют вход

Интерфейсная дуга – 6/7

- Выход (англ. output) – материал или информация, которые представляют результат выполнения работы
 - Отвечает на вопрос «Что является результатом работы?»
 - Выходом может быть:
 - Материальный объект (деталь, автомобиль, платежные документы, ведомость)
 - ~~Нематериальный~~ Не имеющий физического воплощения (выборка данных из БД, ответ на вопрос, устное указание)
 - Рисуетя исходящей из правой грани работы

Интерфейсная дуга – 7/7

- Механизм – ресурсы, выполняющие работу
 - Отвечает на вопрос «Кто выполняет работу или посредством чего?»
 - Механизмом могут быть персонал предприятия, студент, станок, оборудование, программа
 - Рисует входящей в нижнюю грань работы
- Вызов – стрелка указывает, что некоторая часть работы выполняется за пределами рассматриваемого блока
 - Отображает передачу управления в нематериальном виде (вызов процедуры, выдача указания)
 - Рисует исходящей из нижней грани работы

Декомпозиция – 1/6

- Принцип декомпозиции применяется при разбиении сложных процессов на составляющие его функции. Уровень детализации:
 - Определяется разработчиком модели
 - Зависит от проекта: предметной области, системы, потребностей заказчика
- Верхний уровень модели IDEF0 всегда представляет систему как единое целое:
 - Один (!) функциональный блок
 - Интерфейсные дуги представляют взаимодействие системы с пользователями и/или другими системами

Декомпозиция – 2/6

- Диаграмма верхнего уровня:
 - Называется контекстной
 - Обозначается идентификатором A-0
 - Для определения границ системы на контекстной диаграмме обязательно приводят краткие утверждения, определяющие:
 - **Точку зрения** должностного лица или подразделения, с позиций которого создается модель
 - **Цель**, для достижения которой разрабатывается система

Цель – 1/2

- Формулировка цели должна отвечать на следующие вопросы:
 - Зачем создается модель процесса?
 - Что должна показывать модель?
 - Какая информация доводится до читателя?
- Формулировка цели выражает причину создания модели, то есть содержит перечень вопросов, на которые должна отвечать модель, что в значительной мере определяет ее структуру

Цель – 2/2

- Примеры целей:
 - Идентифицировать слабые стороны процесса сбора данных
 - Определить ответственность сотрудников для написания должностных инструкций

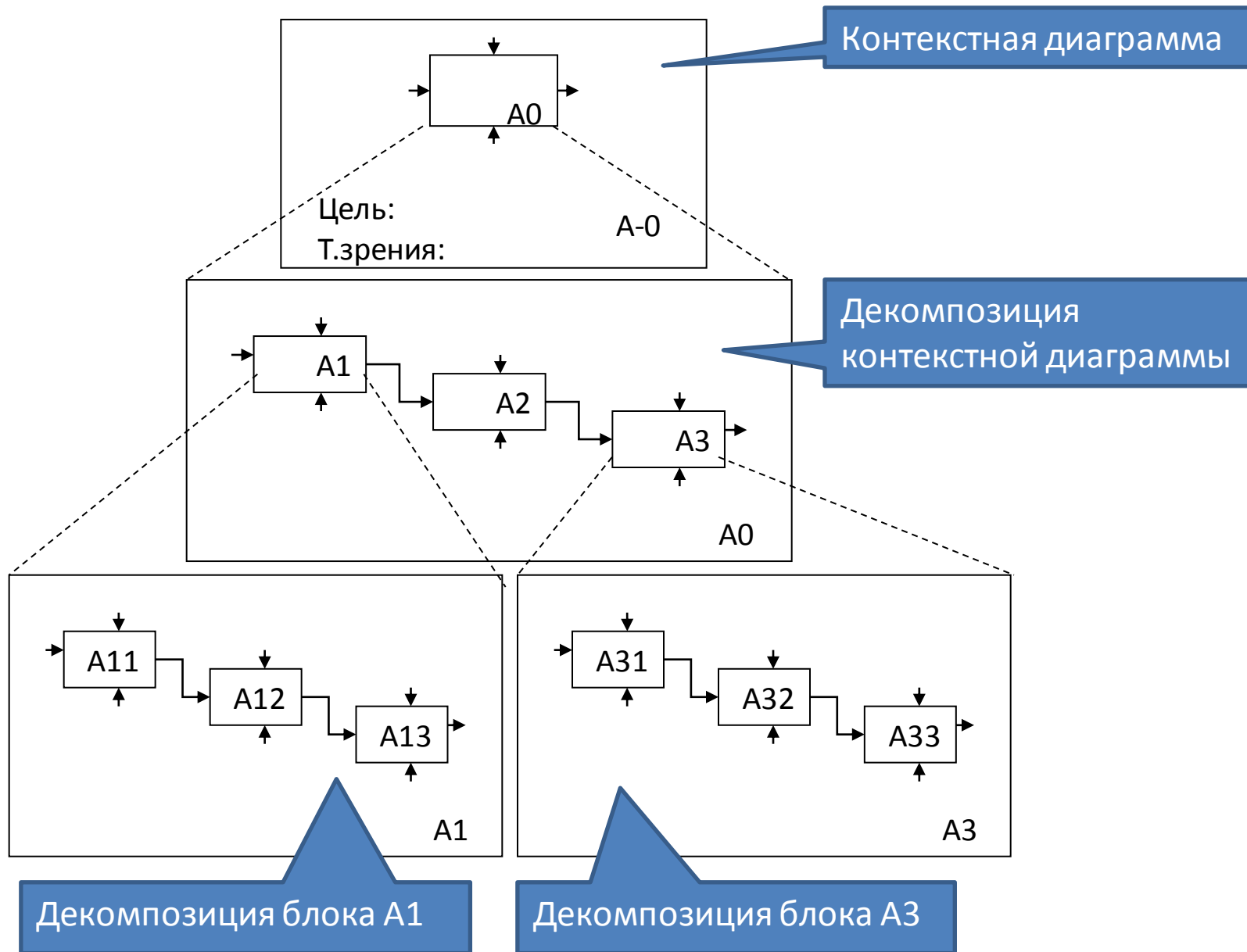
Точка зрения – 1/2

- Точка зрения определяет, что и с какой стороны будет показано в проекте (на модели). Аспекты, важные с одной точки зрения, могут не появиться в модели, разрабатываемой с другой точки зрения на тот же самый объект
- Изменение точки зрения → Рассмотрение других аспектов системы → Другая модель

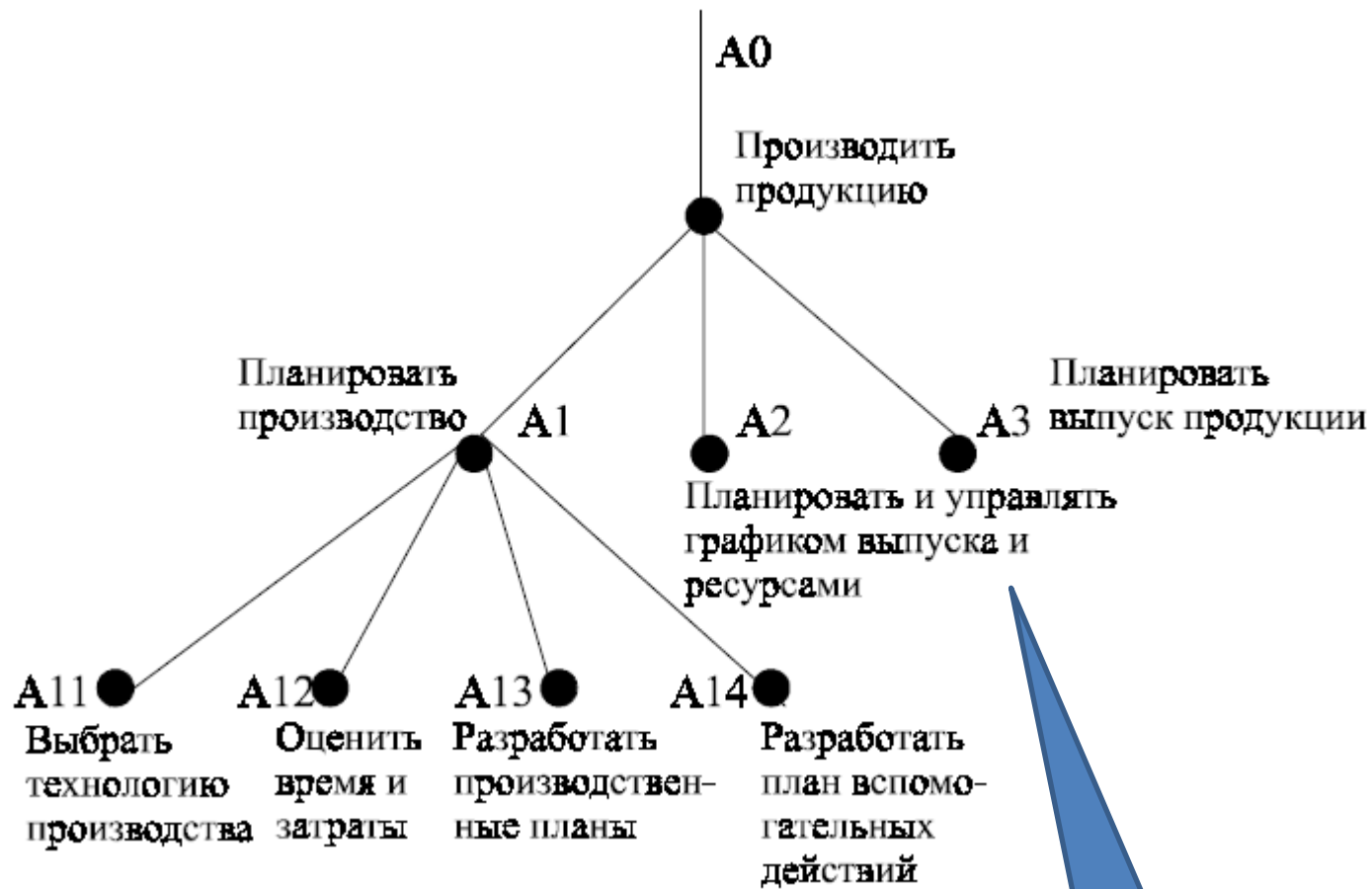
Точка зрения – 2/2

- Должна выбираться «правильная» точка зрения, то есть берется взгляд человека, который видит систему в нужном для моделирования аспекте. Каком?
 - [Стандарт] Как правило, выбирается точка зрения человека, ответственного за выполнение моделируемой работы
 - [Жизнь] Точка зрения должна быть согласована с такой у заказчика
 - [Логика] Между целью и точкой зрения должно быть жесткое соответствие

Декомпозиция – 3/6



Декомпозиция – 4/6



Дерево узлов

Декомпозиция – 5/6

A0 Производить продукт

A1 Планировать производство

A11 Выбрать технологию производства

A12 Оценить требуемое время и затраты на производство

A13 Разработать производственные планы

A14 Разработать план вспомогательных действий

A2 Разрабатывать и управлять графиком выпуска и ресурсами

A21 Разработать основной график

A22 Разработать график координации работ

A23 Оценивать затраты и приобретать ресурсы

A24 Следить за выполнением графика и расходом ресурсов

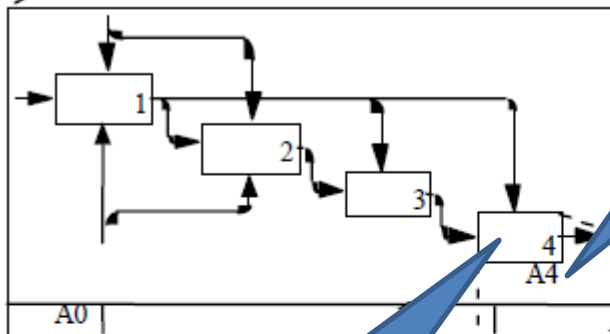
A3 Планировать выпуск продукции



Перечень (индекс) узлов

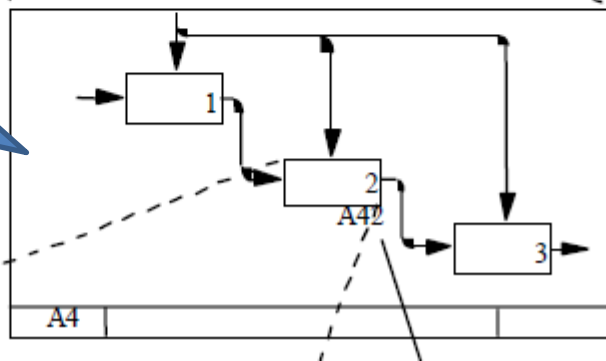
Декомпозиция – 6/6

Более общее представление



Номер узла показывает, что этот блок был декомпозирован. С-номер или номер листа дочерней диаграммы может использоваться вместо узлового номера

4 – родительский блок для диаграммы A4

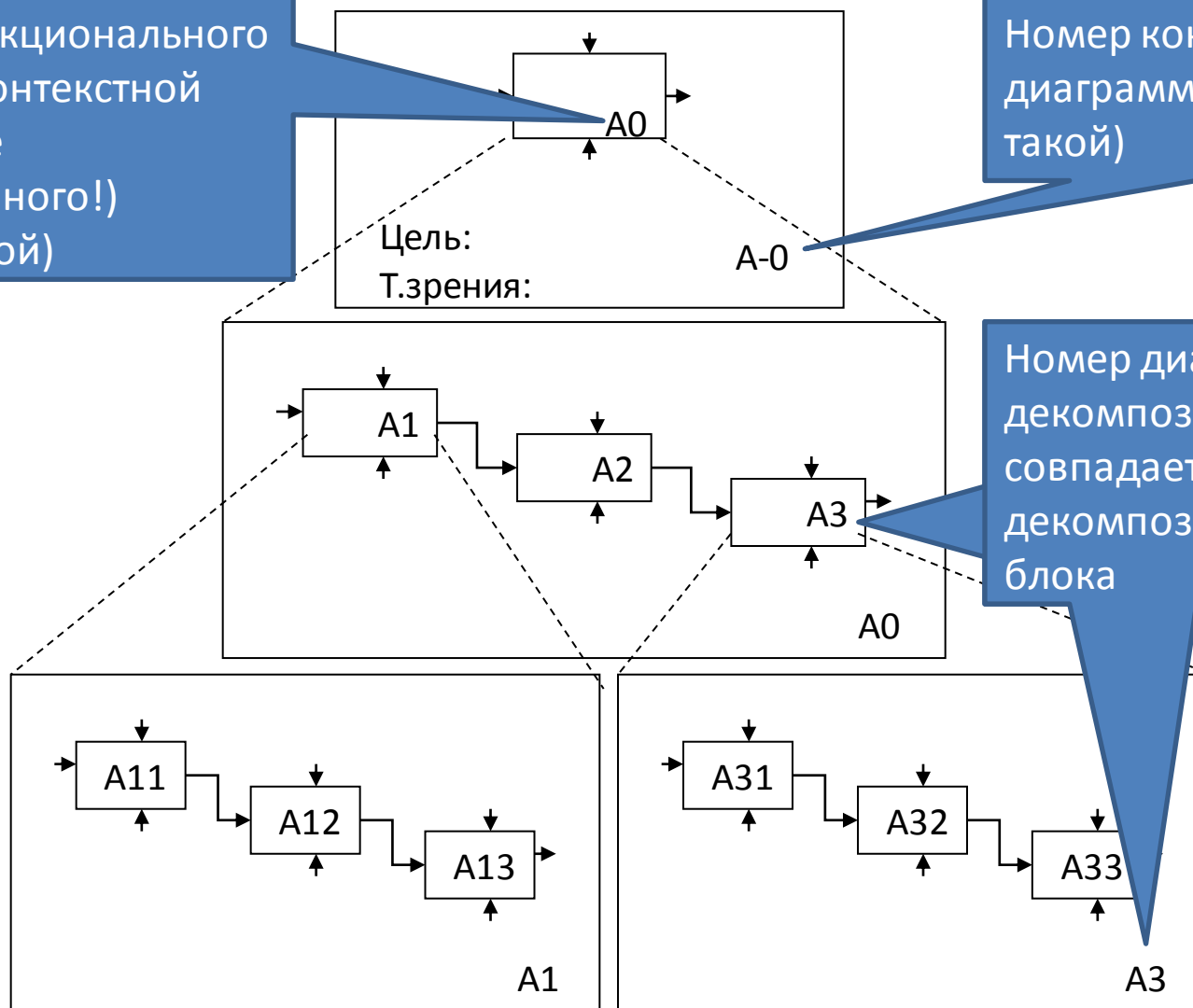


Более детальное представление

Нумерация диаграмм и функций – 1/2

Номер функционального блока на контекстной диаграмме (единственного!) (всегда такой)

Номер контекстной диаграммы (всегда такой)



Нумерация диаграмм и функций – 2/2

- Сложно [стандарт]:

A 6 1 * * * *

				_____	и т.д.
				_____	Номер блока на диаграмме A61
				_____	Номер блока на диаграмме A6
				_____	Номер блока на диаграмме A0
				_____	Имя блока A0

- Проще:
 - К номеру диаграммы добавляется номер функции на диаграмме
 - Номер диаграммы декомпозиции совпадает с номером декомпозируемого блока

Глоссарий – 1/2

- Глоссарий предназначен для определения аббревиатур (акронимов), ключевых слов и фраз, используемых в качестве имен и меток на диаграммах
- Глоссарий определяет понятия и термины, которые должны быть одинаково понимаемы всеми участниками разработки и пользователями модели, чтобы правильно интерпретировать ее содержание

Глоссарий – 2/2

- Глоссарий применим ко всем элементам IDEF0 диаграммы, в том числе к:
 - Функциональным блокам
 - Интерфейсным дугам

Дополнения к диаграммам – 1/2

- Любой диаграмме может быть поставлен в соответствие текст-комментарий содержания диаграммы
 - Используется для объяснений и уточнений характеристик, потоков, внутриблочных соединений и т.д.
 - Не должен использоваться для описания и без того понятных блоков и стрелок на диаграммах

Дополнения к диаграммам – 2/2

- Любой диаграмме может быть поставлена в соответствие диаграмма-иллюстрация (FEO-диаграмма For Exposition Only):
 - Поясняет особо интересные и тонкие аспекты диаграммы (диаграмм)
 - Диаграммы-иллюстрации не ограничены синтаксисом IDEF0

Правила построения диаграмм – 1/11

Правила построение диаграмм задаются разделом 8 стандарта. Нумерация приведена по стандарту, группировка моя «Идентификационные» правила:

1. В составе модели должна присутствовать контекстная диаграмма A-0, которая содержит только один блок. Номер единственного блока на контекстной диаграмме A-0 должен быть «A0»

Правила построения диаграмм – 2/11

2. Блоки на диаграмме должны располагаться по диагонали – от левого верхнего угла диаграммы до правого нижнего в порядке присвоенных номеров
 - Примечание: Это номера присваиваются в порядке изображения, а не наоборот!
 - Блоки на диаграмме, расположенные вверху слева «доминируют» над блоками, расположенными внизу справа:
 - «Доминирование» – влияние, которое блок оказывает на другие блоки → Расположение блоков на листе диаграммы отражает авторское понимание влияния

Правила построения диаграмм – 3/11

3. Все диаграммы кроме контекстной должны содержать 3 – 6 блоков (почти что $7 \pm 2!$)
 - 1-2 блока на диаграмме \rightarrow а надо ли было вводить такой уровень разбиения?
 - Более 6 блоков \rightarrow сложно воспринять читателю + сложно модифицировать создателю
 - А почему не 7 ± 2 ? Потому что 7 ± 2 – это не количество объектов на схеме, а количество элементов. В качестве элементов тут выступают блоки и стрелки

Правила построения диаграмм – 4/11

4. Блоки неконтекстной диаграммы нумеруются от 1 до 6 от верхнего левого к нижнему правому. Номер пишется в правом нижнем углу
5. Декомпозируемый блок, должен иметь ссылку на дочернюю диаграмму. Ссылка помещается под правым нижним углом блока
6. Имена блоков и стрелок уникальны →
Если метки стрелок совпадают, то стрелки отображают тождественные данные

Правила построения диаграмм – 5/11

«Рисовальные» правила:

7. Желательно повторять метки на стрелках сложной формы

Во избежание путаницы:

8. Расстояние между блоками и поворотами стрелок следует делать максимальным

11. Расстояние между параллельными стрелками следует делать максимальным

17. Уменьшить число пересечений, петель и поворотов стрелок

Правила построения диаграмм – 6/11

«Стрелочные» правила:

9. Блок всегда должен иметь хотя бы одну управляющую и одну выходную стрелку.

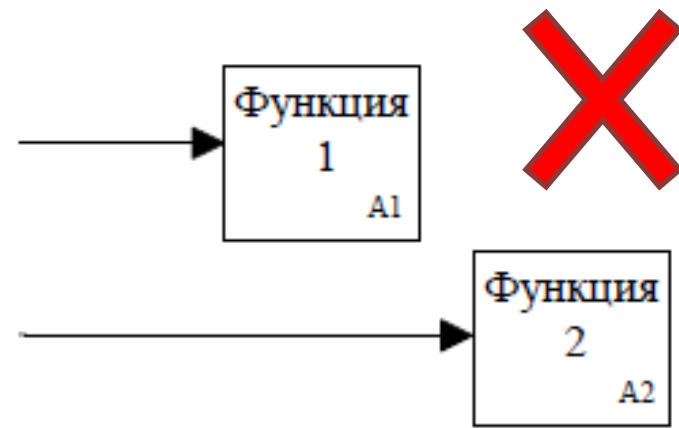
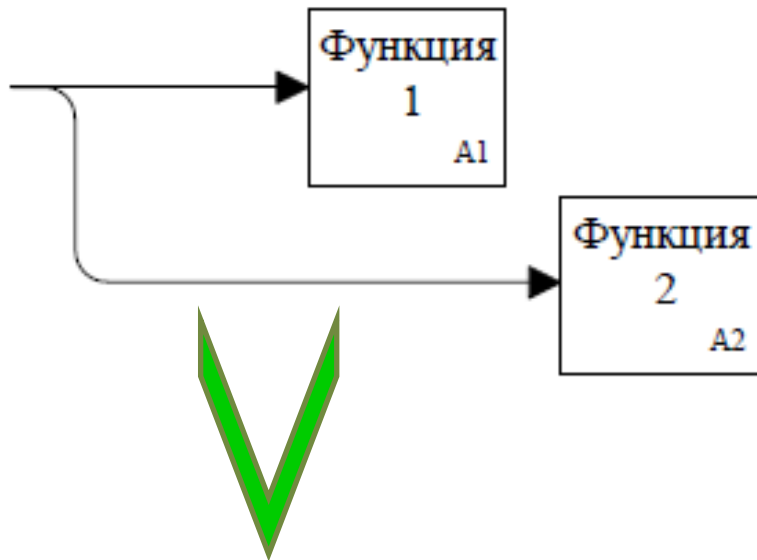
Блок может не иметь входных стрелок

10. Если одни и те же данные служат и для управления, и для входа, вычерчивается только стрелка управления:

- Подчеркивается управляющий характер данных
- Уменьшается сложность диаграммы

Правила построения диаграмм – 7/11

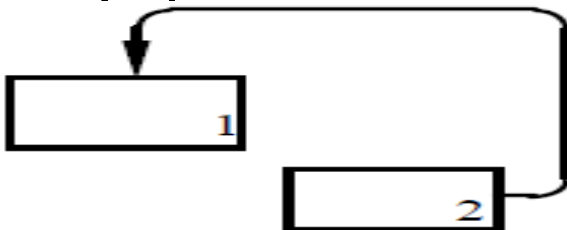
12. Стрелки связываются (сливаются), если они представляют сходные данные и их источник не указан на диаграмме



Правила построения диаграмм – 8/11

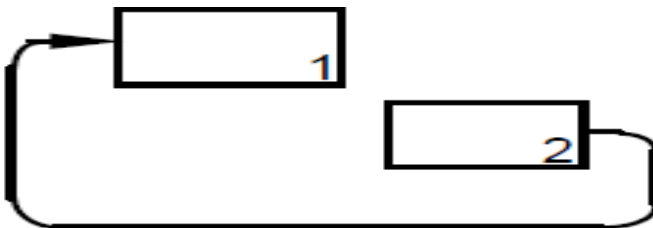
13. Обратные связи:

- По управлению – показываются «вверх и над»



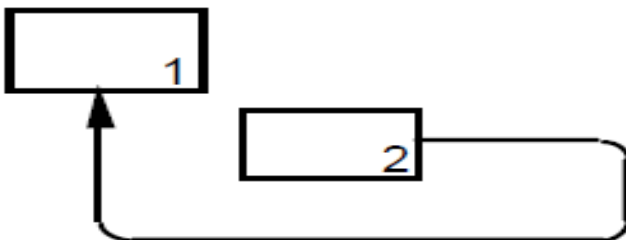
Выход нижестоящей работы управляет вышестоящей работой

- По входу – показываются «вниз и под»



Как правило, используется для описания циклов

- По механизму – показываются «вниз и под»

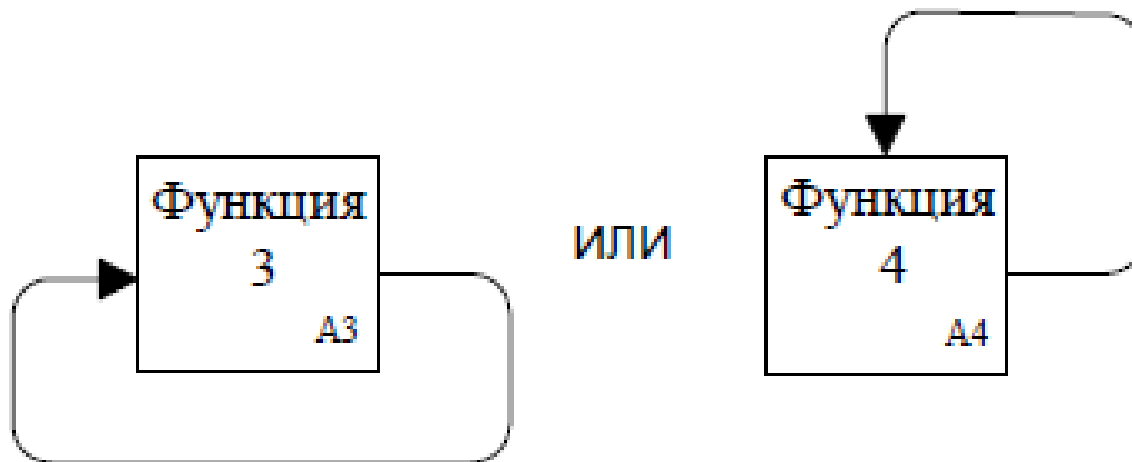


Выход нижестоящей работы создает ресурсы, выполняющие вышестоящую работу

Правила построения диаграмм – 9/11

14. Циклические обратные связи для одного и того же блока изображаются только для того, чтобы их подчеркнуть

- По умолчанию обратные связи изображают на диаграмме декомпозиции (т.е. на уровень ниже)



Правила построения диаграмм – 10/11

15. Стрелки объединяются, если они имеют общий источник или приемник, или они представляют связанные данные

- Общее название лучше описывает суть данных
- Цель: минимизация числа стрелок, касающихся каждой стороны блока
- Ограничение: нельзя сливать стрелки, если природа данных слишком разнородна

Правила построения диаграмм – 11/11

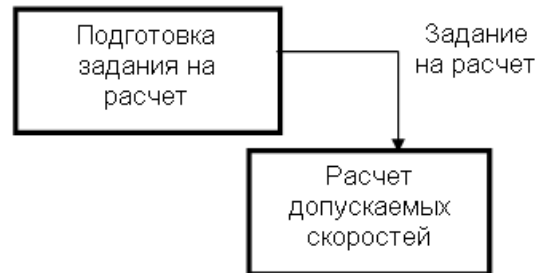
- Стрелки разделяются, если одни и те же данные (ресурсы, и др.) передаются из одного источника в несколько приемников одновременно. При этом:
 - Если стрелка представляет собой данные, то они могут «размножиться» (скопироваться в несколько приемников одновременно)
 - Если стрелка представляет собой ресурсы, то надо понимать, что они либо:
 - Делятся между приемниками, либо
 - Пример: исходные материалы, детали
 - Используются приемниками последовательно
 - Пример: персонал, инструменты, тестовые платформы, помещения

Виды связей – 1/10

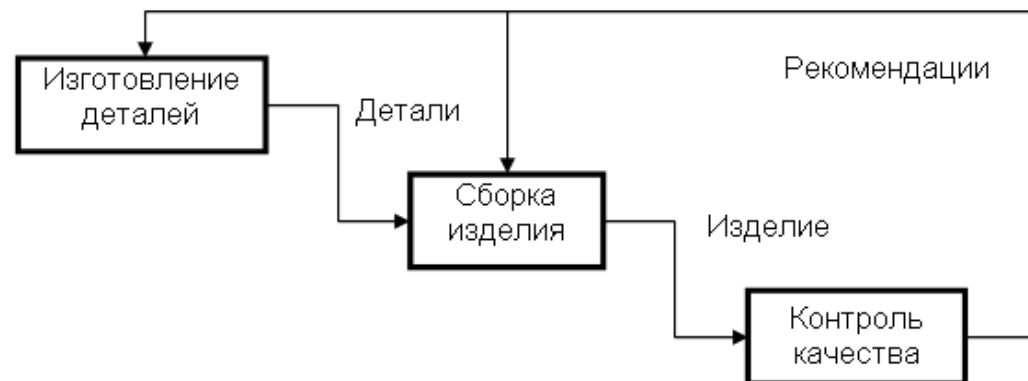
- Иерархическая – связь между функцией и подфункциями, из которых она состоит (см. декомпозиция)
- Регламентирующая (управляющая, подчиненная) – отражает зависимость одной функции от другой
 - Выход одной работы есть управление другой
 - Функцию, из которой выходит управление, называют регламентирующей или управляющей
 - Функция, в которую входит управление называют подчиненной

Виды связей – 2/10

- Регламентирующая (продолжение)
 - Прямая связь по управлению: управление передается с вышестоящей работы на нижестоящую

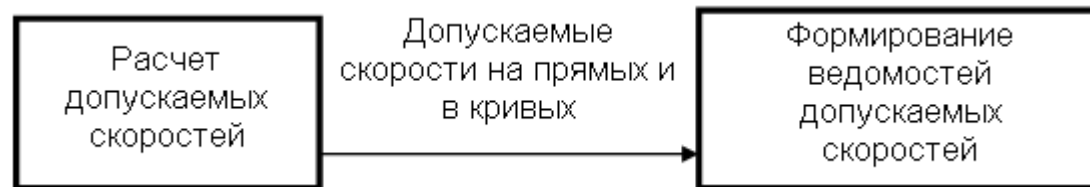


- Обратная связь по управлению: управление передается от нижестоящей к вышестоящей

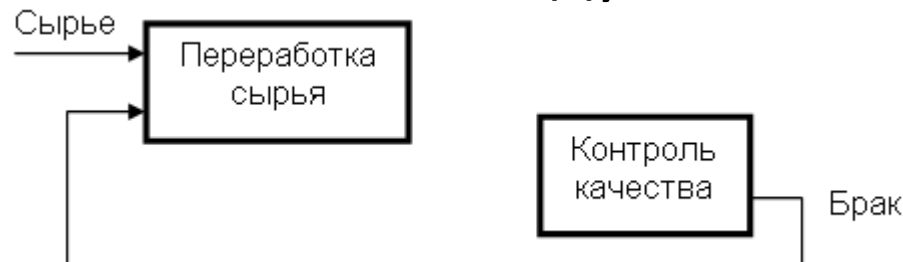


Виды связей – 3/10

- Функциональная (технологическая) – отражает связь функций «вход-выход»
 - Отражает технологию (последовательность работ) обработки потока материальных или нематериальных объектов
 - Прямая связь по входу

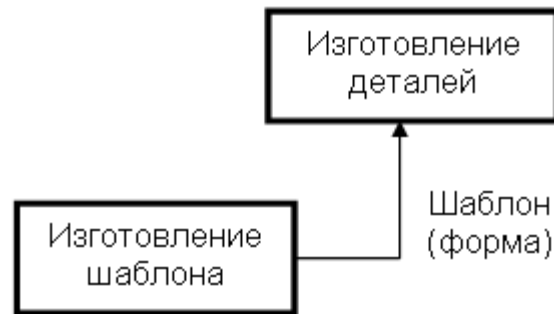


- Обратная связь по входу



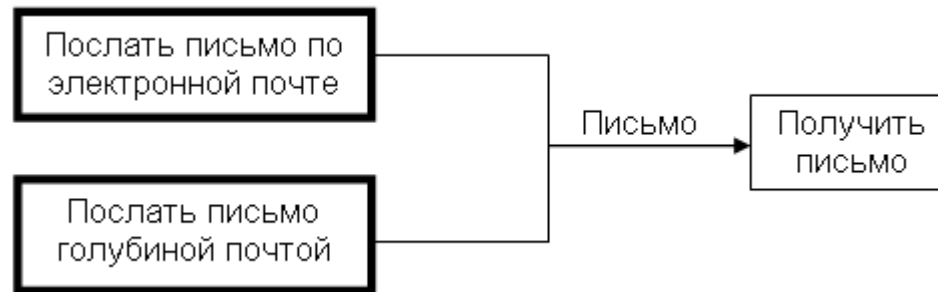
Виды связей – 4/10

- Потребительская – когда выход одной функции служит механизмом для следующей функции
 - Одна функция потребляет ресурсы, вырабатываемые другой



Виды связей – 5/10

- Логическая – связь между логически однородными функциями, выполняющими, [как правило,] одну и ту же работу, но разными (альтернативными) способами или используя разные исходные данные (материалы)

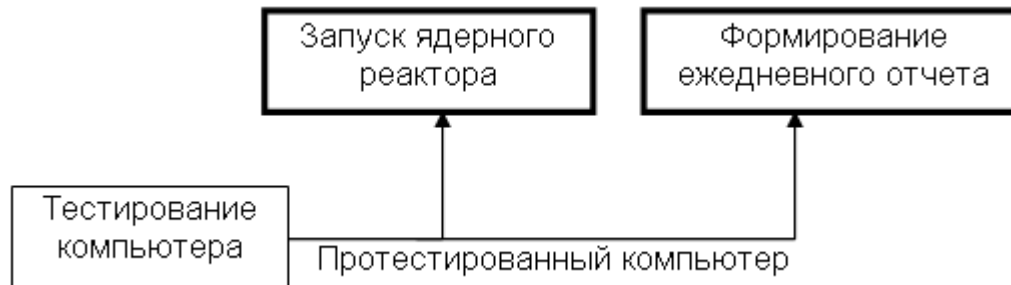


Виды связей – 6/10

- Коллегиальная (методическая) – связь между функциями, алгоритм работы которых определяется одним и тем же управлением. Например:
 - Совместная работа сотрудников одного отдела, подчиняющихся начальнику, который отдает указания и приказы (управляющие сигналы)
 - Алгоритмы работы функций определяются одним и тем же методическим обеспечением (СНИП, ГОСТ, официальными нормативными материалами и т. д.)

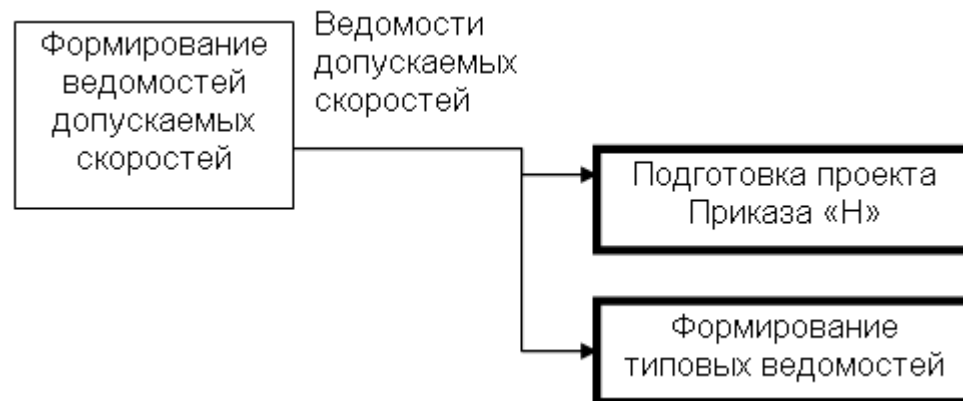
Виды связей – 7/10

- Ресурсная – связь, возникающая между функциями, использующими для своей работы одни и те же ресурсы
 - Связанные такой связью функции, как правило, не могут выполняться одновременно



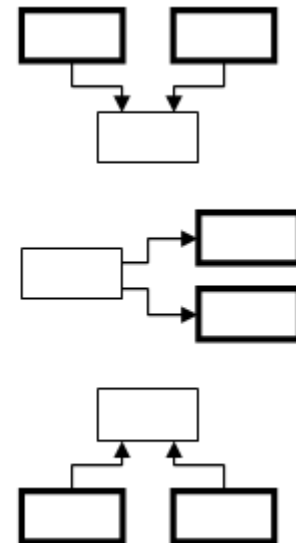
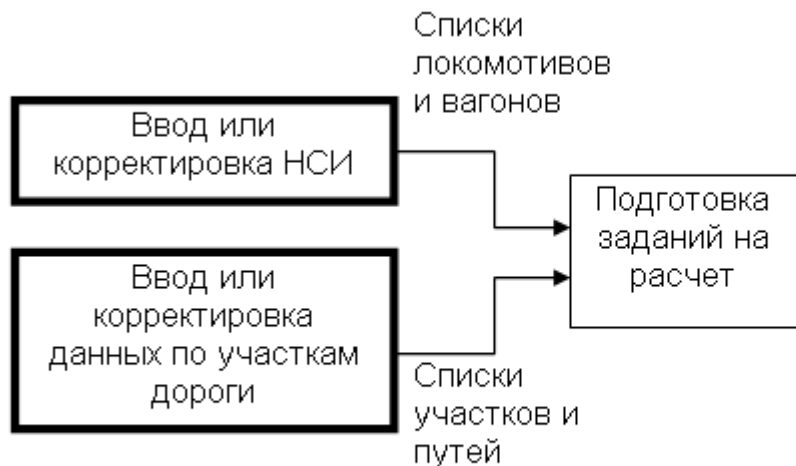
Виды связей – 8/10

- Информационная – связь между функциями, использующими в качестве входных данных одну и ту же информацию



Виды связей – 9/10

- **Временная (синхронизационная)** – связь между функциями, которые должны выполняться одновременно до или одновременно после другой функции
 - Может возникать между одинаковыми или разными дугами



Виды связей – 10/10

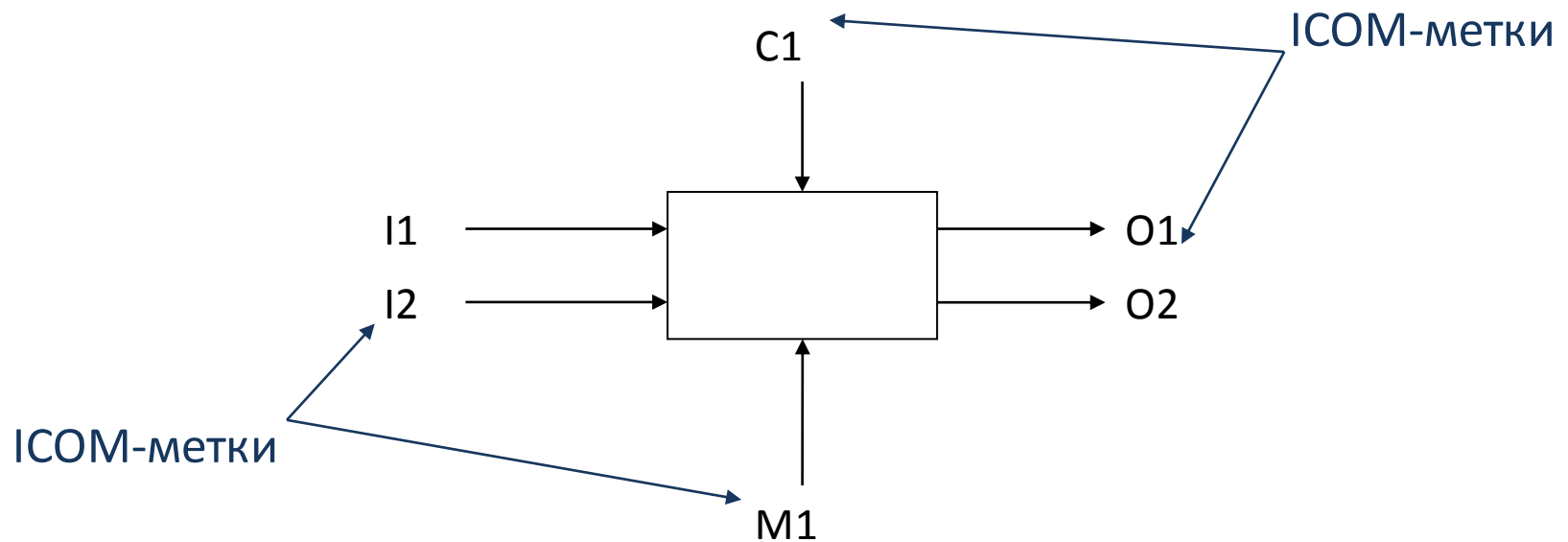
- Случайная связь возникает ~~на отдыхе~~
~~после принятия спиртных напитков~~, когда конкретная связь между функциями мала или полностью отсутствует (0_0)



Граничные стрелки – 1/2

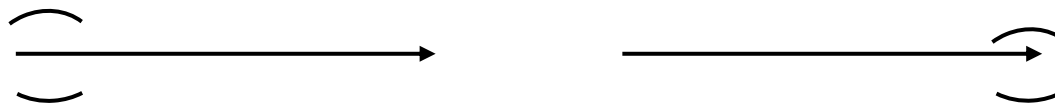
- Стрелки на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром
 - Они начинаются у границы диаграммы и заканчиваются у функционального блока или наоборот
 - Такие стрелки называются граничными
 - Граничные стрелки помечаются с помощью ICOM-меток (Input, Control, Output,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

Граничные стрелки – 2/2



Тоннельные стрелки

- При необходимости отобразить стрелки, которые значимы на данном уровне и не значимы на родительской диаграмме, используют тоннельные стрелки
 - Пример: некоторые данные используются только на данном уровне и не используются на других
 - По умолчанию малозначащая стрелка появится на всех уровнях модели, что затруднит чтение диаграмм



Оформление диаграмм – 1/4

- Диаграммы оформляются на стандартном бланке
 - Раньше: облегчало подшивку и копирование
 - Сейчас: упрощает понимание из-за унификации внешнего вида
- Бланк разделен на поля:
 - Рабочей информации
 - Содержит данные для отслеживания работы с диаграммой
 - Сообщений
 - Собственно область рисования диаграммы
 - Идентификации
 - Содержит идентификацию диаграммы и ее место в иерархии

Оформление диаграмм – 2/4

USED AT:	AUTHOR: FIO	DATE: 27.02.2009	<input checked="" type="checkbox"/>	WORKING	READER	DATE	CONTEXT: TOP
	PROJECT: model1	REV: 27.02.2009	<input type="checkbox"/>	DRAFT			
			<input type="checkbox"/>	RECOMMENDED			
			<input type="checkbox"/>	PUBLICATION			
NOTES: 1 2 3 4 5 6 7 8 9 10							

Поле рабочей информации

Поле сообщений

Поле идентификации

NODE:

A-0

TITLE:

NUMBER:

Оформление диаграмм – 3/4

USED AT:	AUTHOR: FIO PROJECT: model1 NOTES: 1 2 3 4 5 6 7 8	DATE: 27.02.2009 REV: 27.02.2009	<input type="checkbox"/> WORKING <input type="checkbox"/> DRAFT <input type="checkbox"/> RECOMMENDED <input type="checkbox"/> PUBLICATION	READER	DATE	CONTEXT: TOP
----------	--	-------------------------------------	--	--------	------	------------------------

Сведения о модели:

- Автор
- Название проекта
- Замечания
- Дата создания и последнего обновления

Статус проекта (один из):

- Рабочая версия (Working) – диаграмма с большим числом изменений на стадии разработки
- Эскиз (Draft) – меньше изменений и свидетельствует о достижении некоторого согласия ряда читателей
- Рекомендовано (Recommended) – сопутствующие тексты утверждены
- Публикация (Publication) – материал готов к выпуску

Сведения о читателях-экспертах и дате экспертизы

Ссылка на родительский проект

NODE: A-0	TITLE:	NUMBER:
---------------------	--------	---------

Оформление диаграмм – 4/4

USED AT:	AUTHOR: FIO	DATE: 27.02.2009	WORKING	READER	DATE	CONTEXT: TOP
	PROJECT: model1	REV: 27.02.2009	DRAFT			
			RECOMMENDED			
			PUBLICATION			
NOTES: 1 2 3 4 5 6 7 8 9 10						

Номер (идентификатор)
диаграммы

Название диаграммы
(совпадает с названием
блока на родительской
диаграмме)

Уникальный номер
версии диаграммы

NODE:
A-0

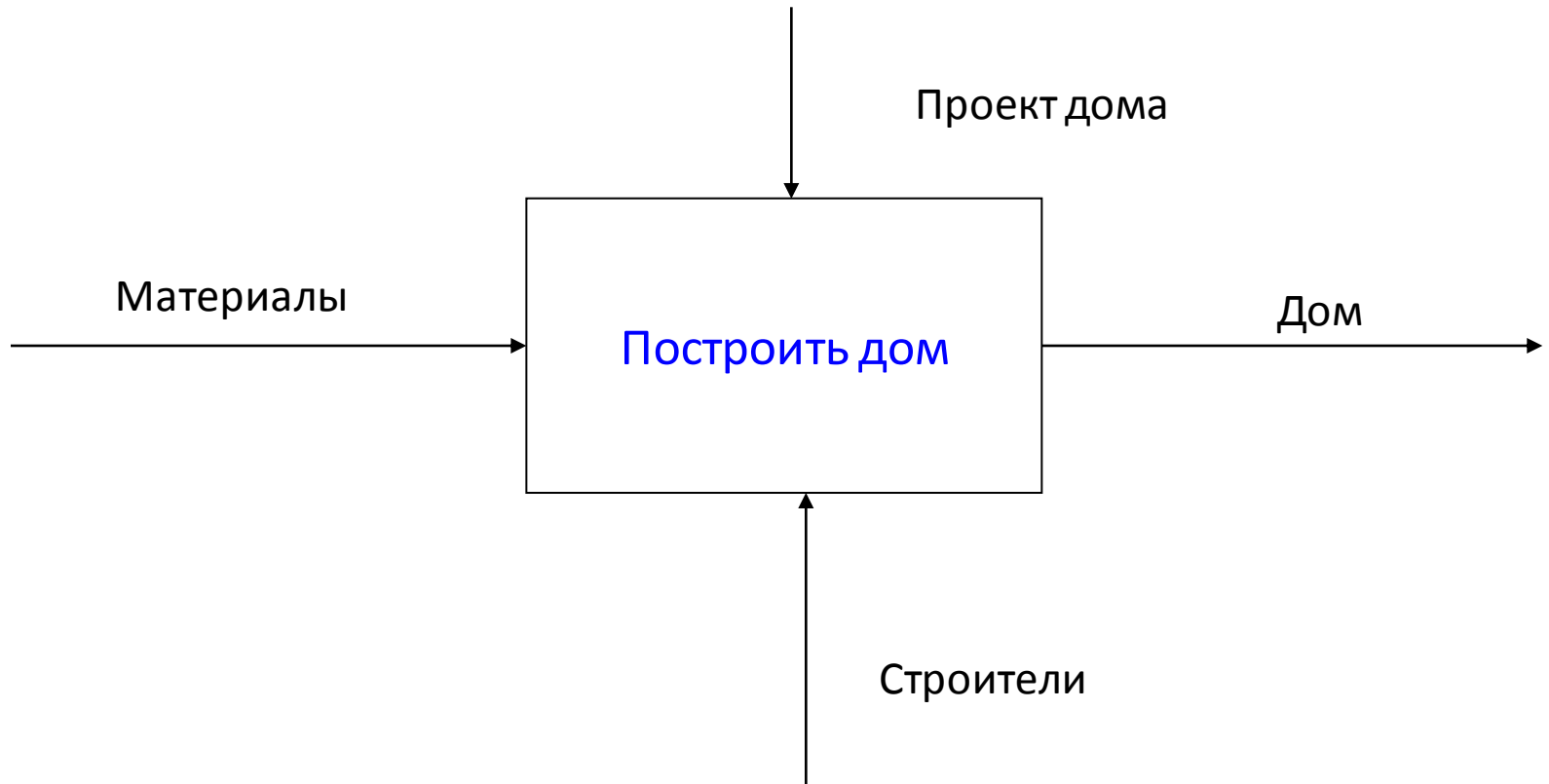
TITLE:

NUMBER:

IDEFO – ПРИМЕР ПОСТРОЕННОГО ОПИСАНИЯ

Пример модели процесса постройки садового домика

1. Строим контекстную диаграмму.

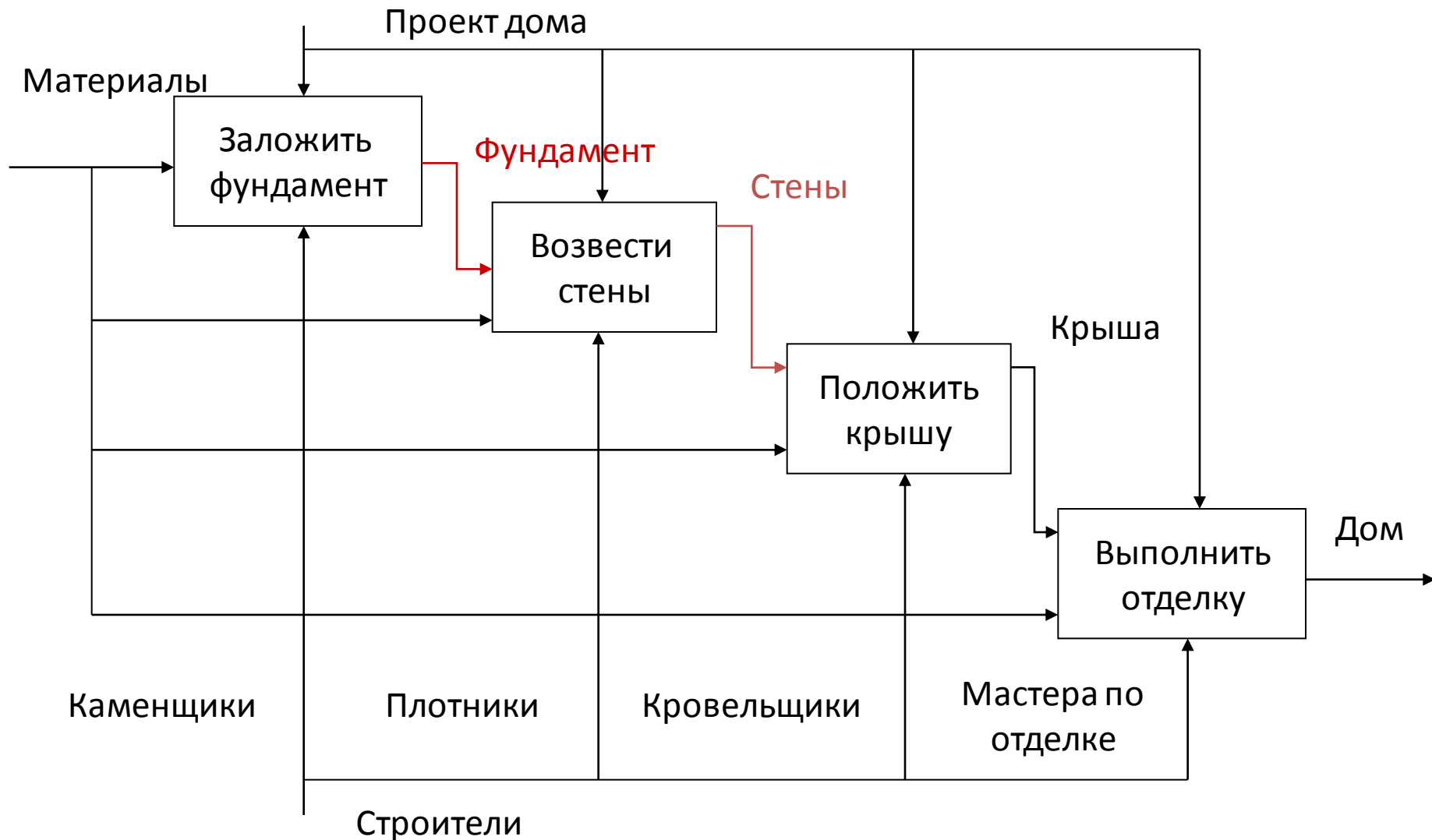


Цель: Определить действия, необходимые для постройки дачного домика

Точка зрения: владельца дачного участка

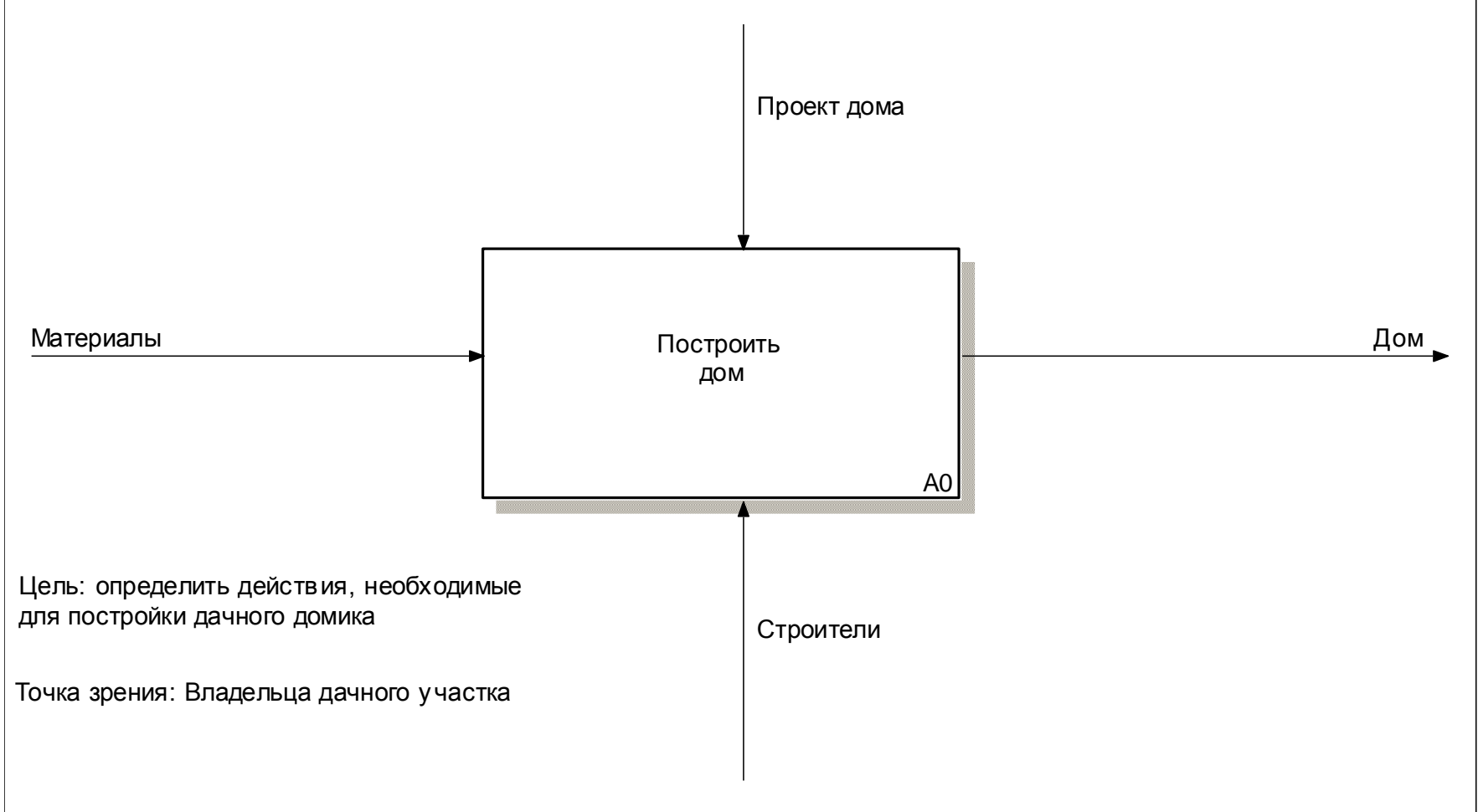
Пример модели процесса постройки садового домика

2. Декомпозируем контекстную диаграмму



Пример модели, построенной с использованием CASE-средства BPWin

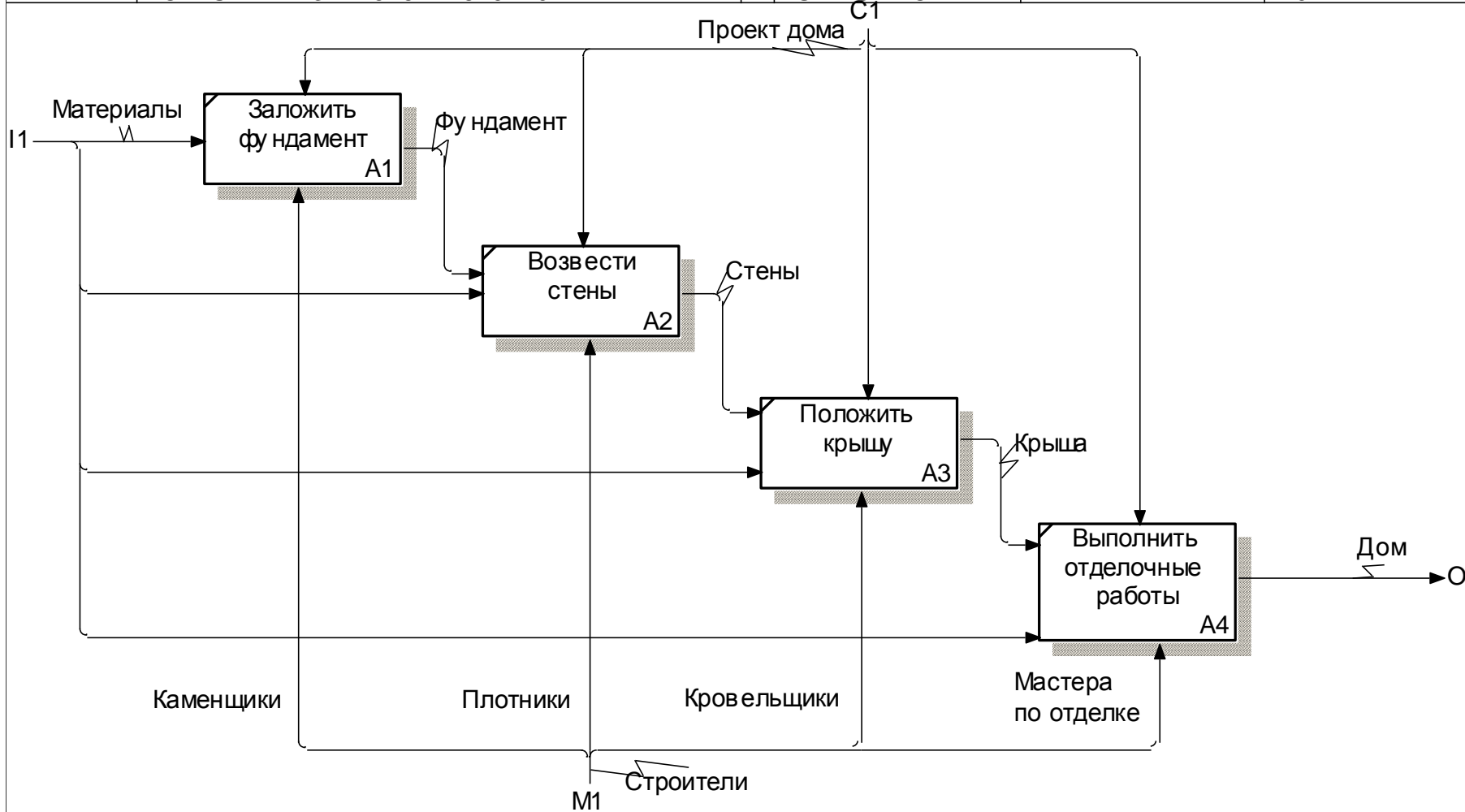
USED AT:	AUTHOR: Шилина М.А.	DATE: 27.02.2009	WORKING	READER	DATE	CONTEXT: TOP
	PROJECT: Постройка дома	REV: 27.02.2009	DRAFT			
			RECOMMENDED			
			PUBLICATION			
	NOTES: 1 2 3 4 5 6 7 8 9 10					



NODE: A-0	TITLE: Построить дом	NUMBER:
--------------	-------------------------	---------

Пример модели, построенной с использованием CASE-средства BPWin

USED AT:	AUTHOR: Шилина М.А.	DATE: 27.02.2009		WORKING	READER	DATE	CONTEXT:
	PROJECT: Постройка дома	REV: 10.03.2010		DRAFT			
				RECOMMENDED			
	NOTES: 1 2 3 4 5 6 7 8 9 10			PUBLICATION			A-0

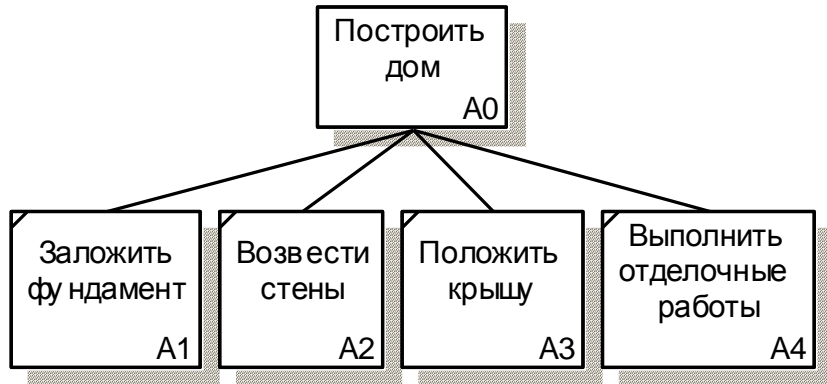


NODE:	TITLE: Построить дом	NUMBER:
A0		

Дерево узлов

USED AT:	AUTHOR: Шилина М.А.	DATE: 27.02.2009	WORKING	READER	DATE	CONTEXT: TOP A-0
	PROJECT: Постройка дома	REV: 27.02.2009	DRAFT			
			RECOMMENDED			
			PUBLICATION			

NOTES: 1 2 3 4 5 6 7 8 9 10

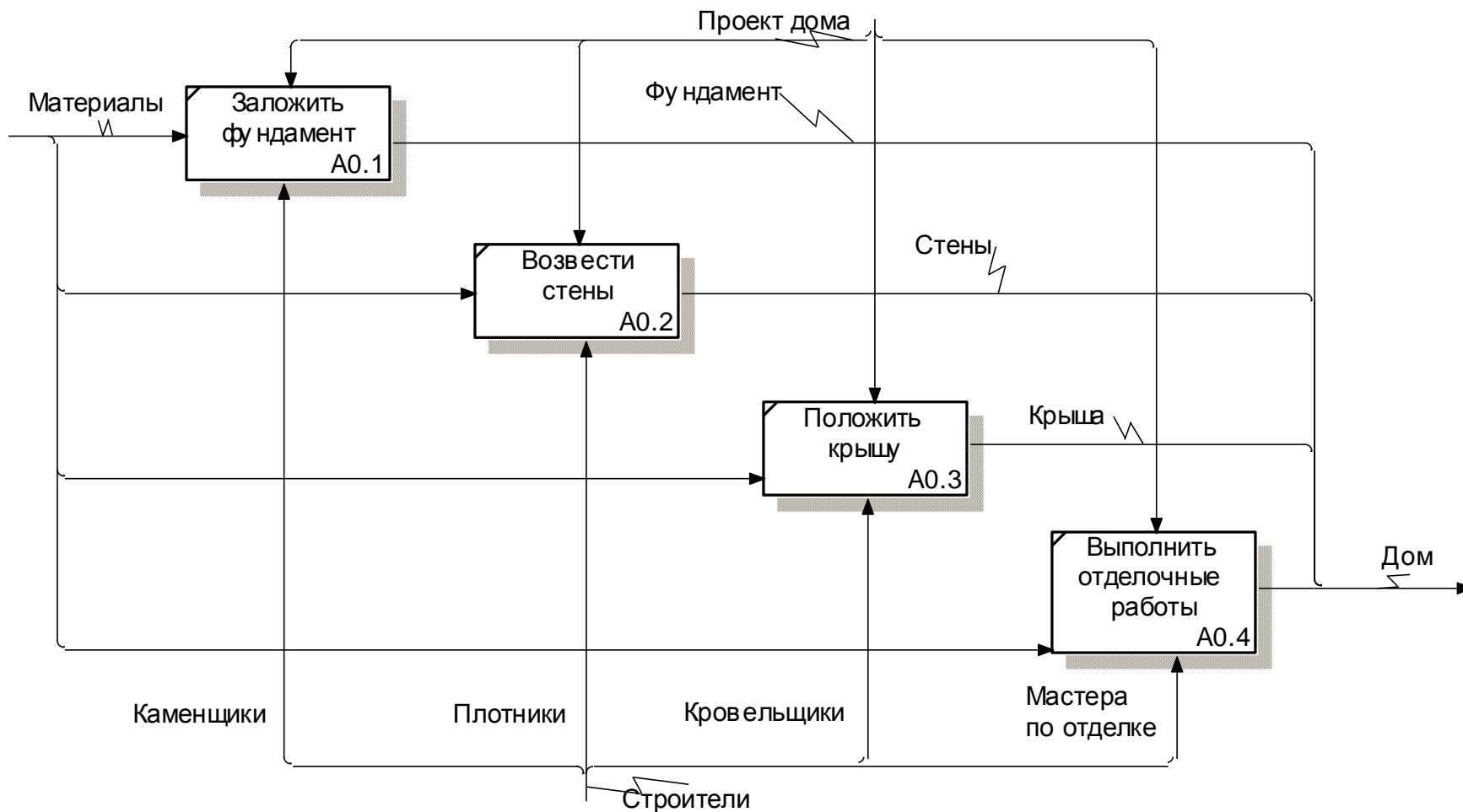


NODE: A0	TITLE: Построить дом	NUMBER: <input type="text"/>
-------------	-------------------------	---------------------------------

FEO-страница

USED AT:	AUTHOR: Шилина М.А.	DATE: 27.02.2009		WORKING	READER	DATE	CONTEXT: A-0
	PROJECT: Постройка дома	REV: 27.02.2009		DRAFT			
				RECOMMENDED			
				PUBLICATION			

NOTES: 1 2 3 4 5 6 7 8 9 10



NODE: A0F	TITLE: Построить дом	NUMBER:
---------------------	--------------------------------	---------