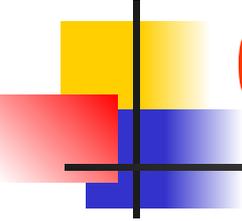


Часть 3. Моделирование бизнеса



Тема 3.1. Классификация моделей

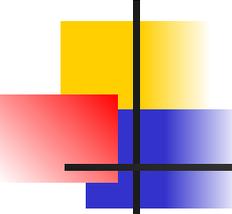
Тема 3.2. Структурные методологии

Тема 3.3. Объектно-ориентированный язык UML

Тема 3.4. Имитационная методология

Тема 3.5. Интегрированная методология ARIS

Тема 3.6. Инструментальные средства



Понятие модели

Модель представляет искусственный, созданный человеком объект любой природы (умозрительный или материально реализованный), который замещает или воспроизводит исследуемый объект

Процесс построения, изучения и применения моделей называется **моделированием**

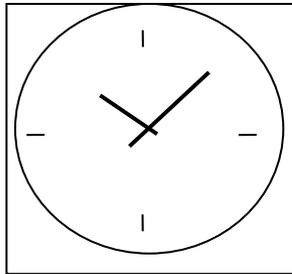
Модель - **упрощенный, приближенный** образ, который отражает наиболее существенные (с точки зрения цели моделирования) свойства оригинала

Соответствие модели оригиналу называется **адекватностью** модели.

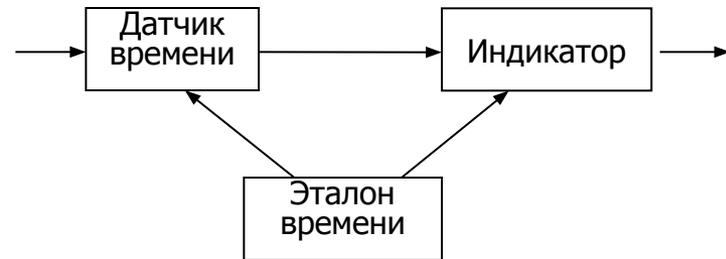
Адекватность включает требования полноты и точности (правильности). Требования должны выполняться в той мере, которая **достаточна для достижения цели**

Понятие модели

Для одного и того же объекта может быть построено множество различных моделей, отвечающих различным целям



модель внешнего вида часов

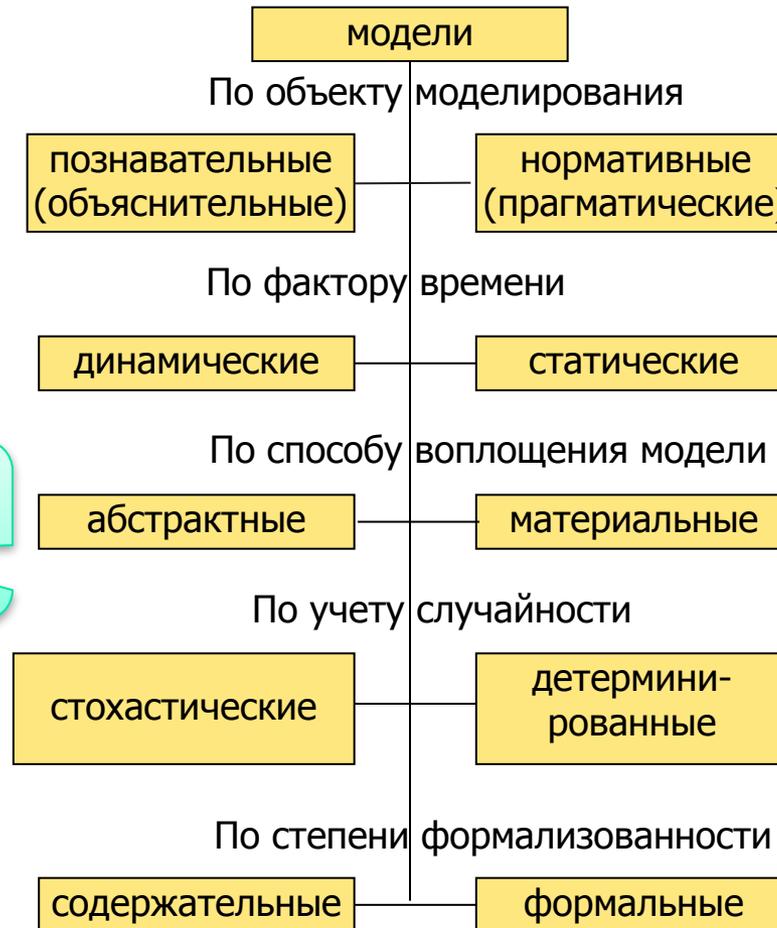


структурная схема часов

Виды подобия: *прямое* (макет, фотография), *косвенное* (подобие по аналогии), *условное* (на основе соглашений)

Процесс моделирования имеет свойство *динамичности*: модели развиваются, уточняются, переходят одна в другую

Классификация моделей



отражают уже существующие объекты

отражают изменения объекта во времени

идеальные конструкции созданные средствами мышления

отражают случайные процессы

В них сохраняется семантика моделируемого объекта

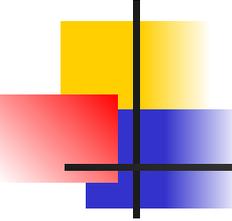
отражают идеальные (эталонные) объекты или которые должны быть осуществлены

не учитывают временной фактор

построены из реальных объектов

отражают процессы, не подверженные случайностям

могут не иметь смысловой интерпретации



Языки описания моделей

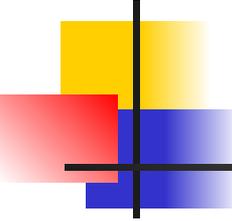
Языки описания моделей: аналитические, численные, логические, теоретико-множественные, лингвистические, графические

Графические модели (схемы, диаграммы, графики, чертежи) – наглядны

Нотация — система условных обозначений (знаков) и правил их использования, принятая в конкретной методологии

Требования к нотации :

- простота — простой знак предпочтительнее сложного;
- наглядность — хотя бы отдаленное сходство с оригиналом;
- индивидуальность — достаточное отличие от других обозначений;
- однозначность — нельзя обозначать одним символом различные объекты;
- определенность — четкие правила использования модели;
- учет устоявшихся традиций

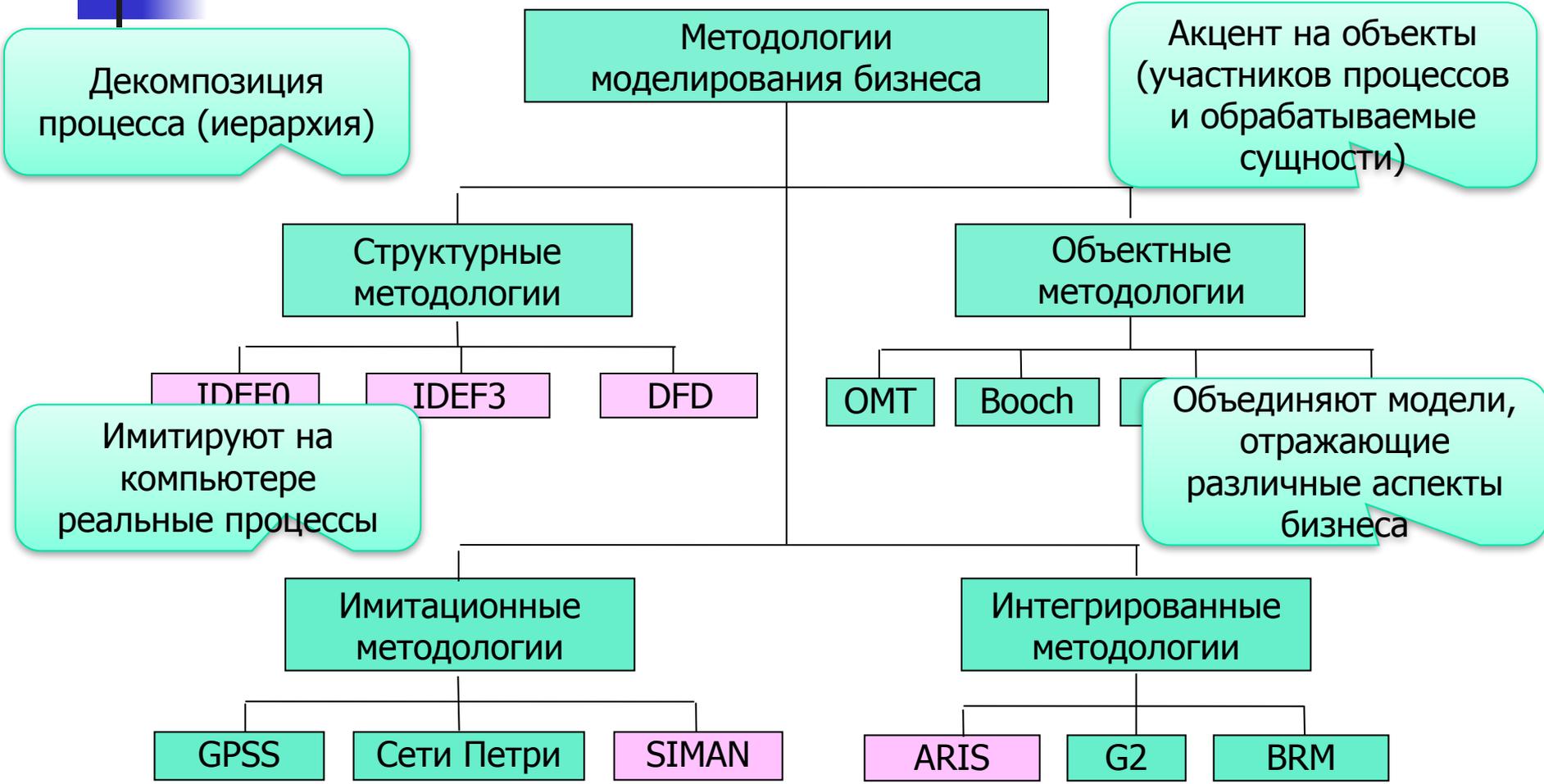


Содержание модели бизнеса

В модели бизнеса отражают:

- функции в мире, которые бизнес-система должна выполнять - что она делает, для кого, с какой целью;
- процессы, последовательность отдельных шагов процессов (работ, операций);
- ресурсы, используемые при выполнении процессов (исполнителей процессов, оборудование, инструменты, материалы и т.д.);
- материальные и информационные потоки, возникающие в ходе выполнения процессов
- данные, необходимые при выполнении процессов, и отношения между этими данными

Методологии моделирования бизнеса



Структурные методологии

Структурные методологии

Модели представлены в виде **иерархии диаграмм**. Основаны на декомпозиции процесса на все более мелкие подпроцессы (функции, работы, действия)

IDEF0

— **функциональные диаграммы (SADT)**.

Показывают, какими **элементами** (входными и выходными) обмениваются **функции** между собой и с окружением

IDEF3

— **диаграммы потоков работ (Work Flow Diagrams)**.

Показывают в какой **последовательности** выполняются **работы** (разветвления, слияния по типу И, ИЛИ)

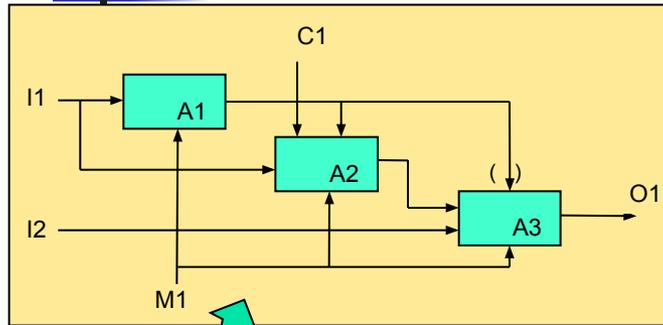
DFD

— **диаграммы потоков данных (Data Flow Diagrams)**.

Показывают как **процессы обработки информации** обмениваются **данными** друг с другом, с хранилищами данных и с внешними сущностями

IDEF0: декомпозиция

Диаграмма A0



Продажа мебели на заказ
A0

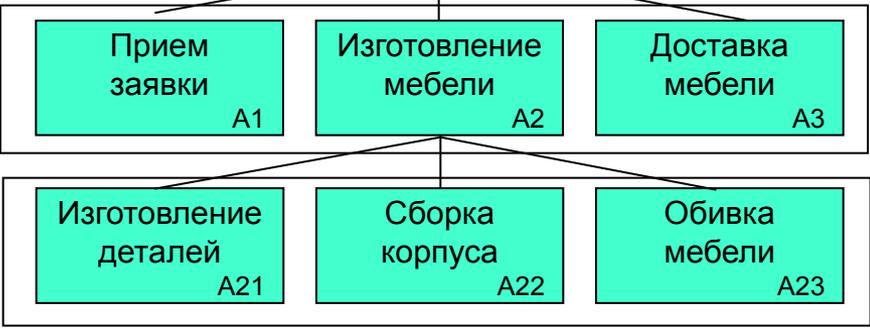


Диаграмма A-0

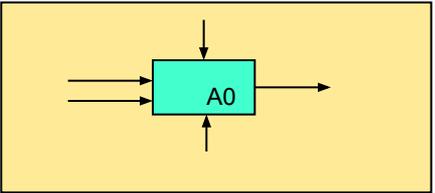
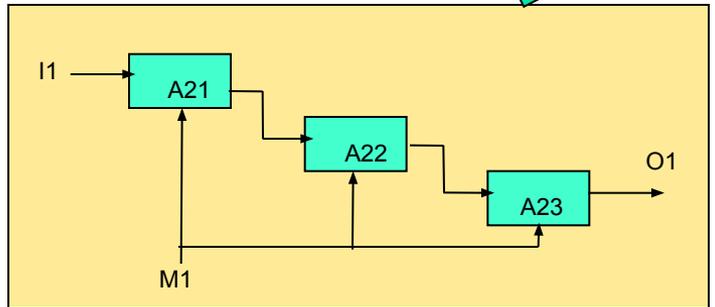


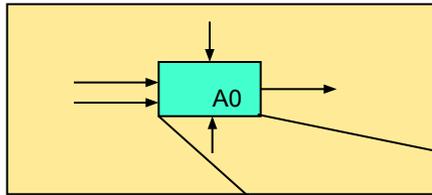
Диаграмма A2



Процесс декомпозируется на все более мелкие функции. Взаимодействие функций, полученных в результате декомпозиции одной функции, отображается на отдельной диаграмме.

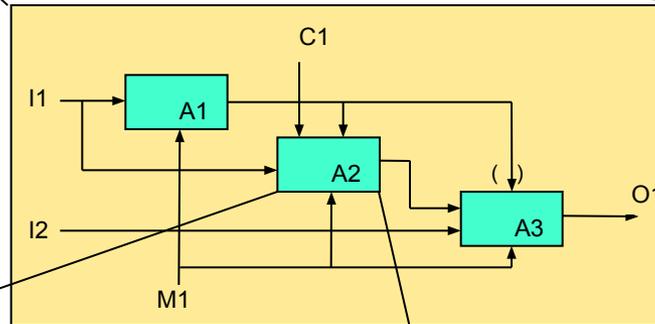
IDEF0: иерархия диаграмм

Диаграмма A-0



← Контекстная диаграмма

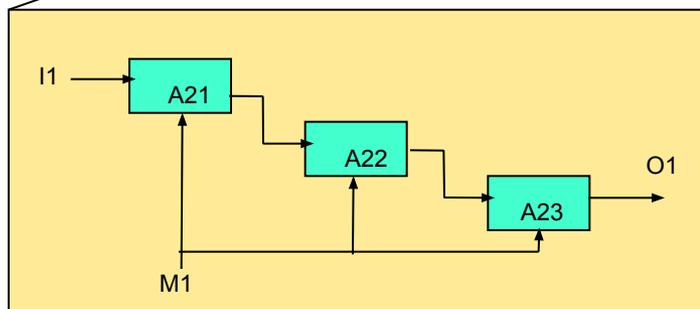
Диаграмма A0



← Диаграмма декомпозиции 1-го уровня

Диаграмма декомпозиции 2-го уровня

Диаграмма A2



Блоки – функции,
Дуги – объекты, используемые функциями и являющиеся результатами функций

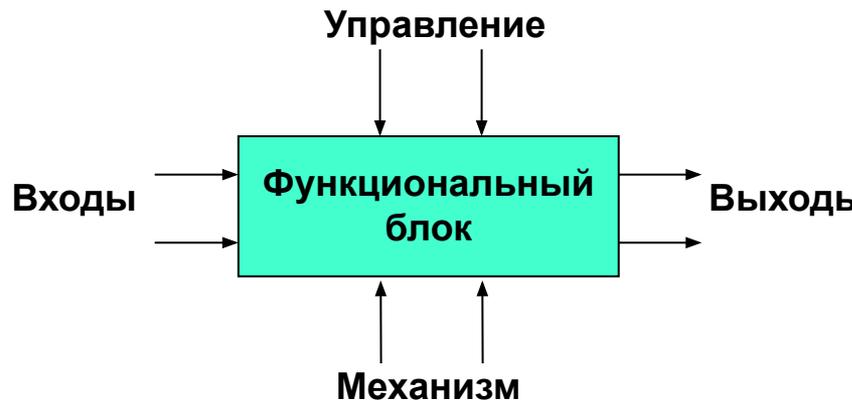
IDEFO: блок и дуги

Управление (Control) - информация, как происходит преобразование:
план, проект, инструкция

Функциональный блок (Activity) – процесс, работа, преобразование, действие: *изготовление продукта, обработка заказа*

Входы (Input) - объекты, которые преобразуются в выходы:
сырье, материалы, заявка

Выходы (Output) – результат преобразования:
изготовленный продукт, выполненная услуга, обработанная заявка

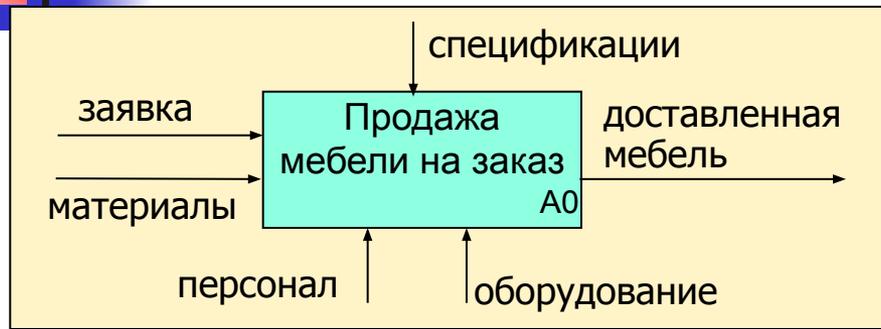


Дуги – это:

не только входные/выходные потоки. Входящие дуги — это необходимые **УСЛОВИЯ** (ограничения), для того чтобы действие могло произойти, выходящие — **РЕЗУЛЬТАТ** действия.

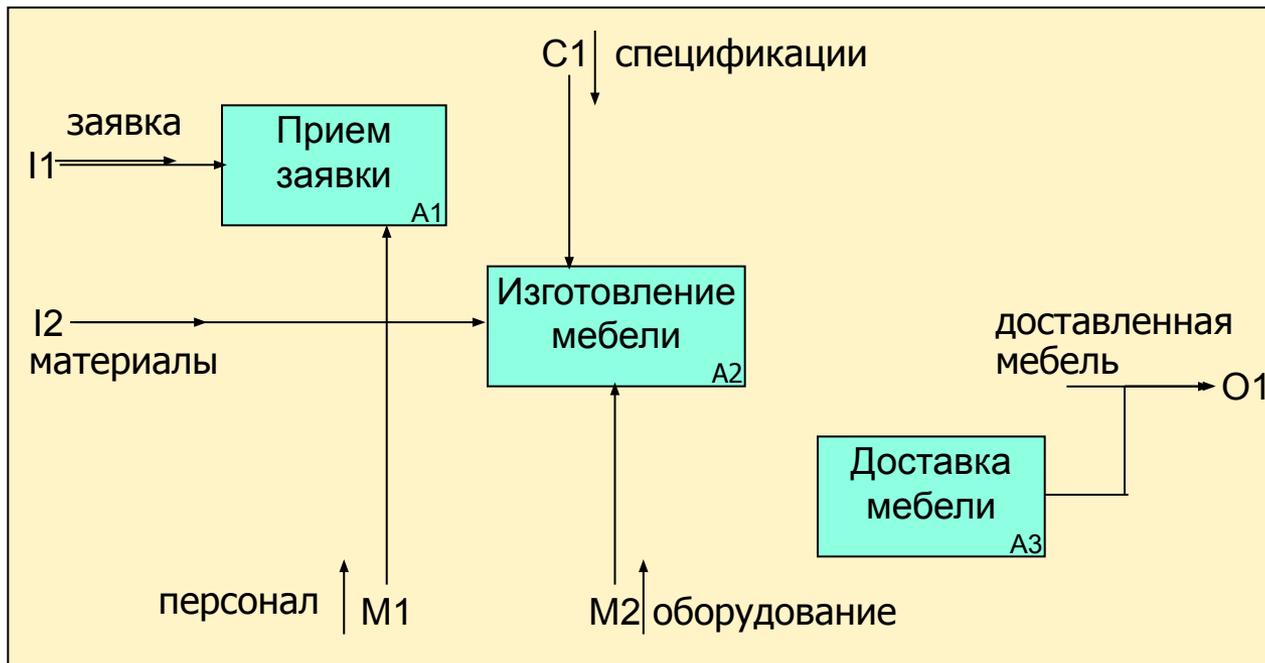
Механизм (Mechanism) – объекты, осуществляющие преобразование:
рабочий, цех, станок, инструмент

IDEF0: внешние дуги



При декомпозиции блока связанные с ним дуги переносятся на дочернюю диаграмму.

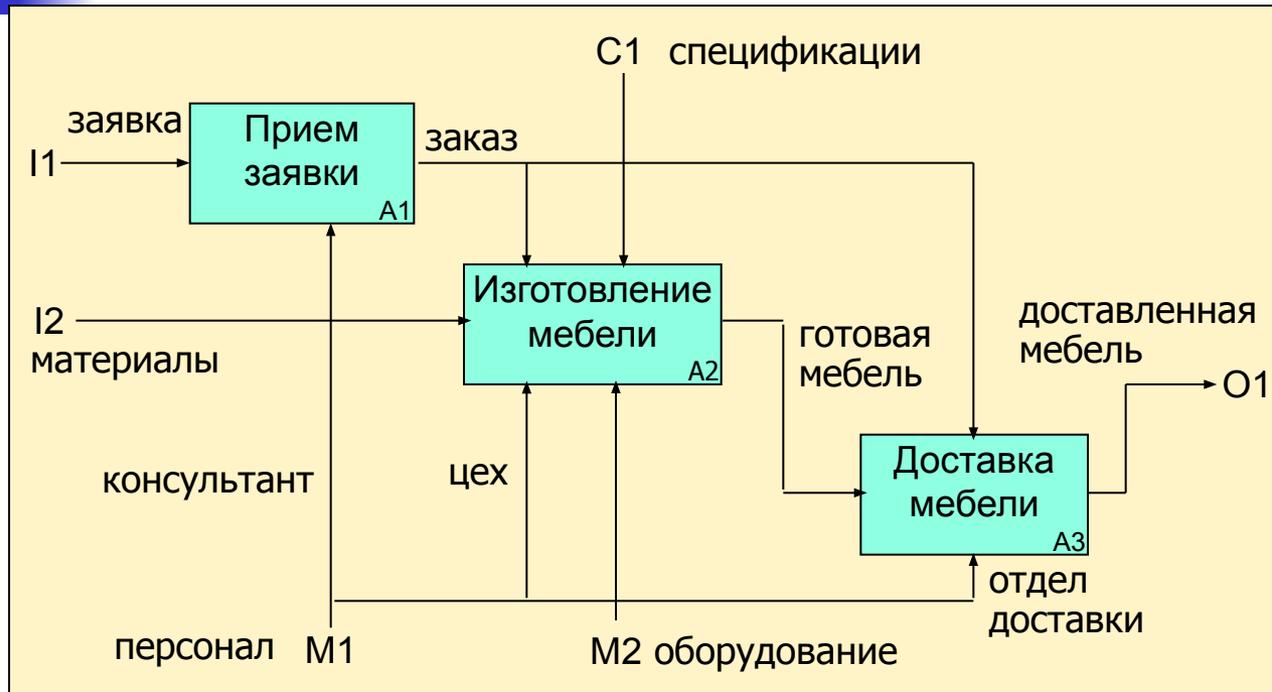
Это внешние дуги, которые имеют источник или получатель вне диаграммы.



Для обозначения внешних дуг используются буквы: I (Input), C (Control), O (Output) M (Mechanism).

Внешние дуги соединяются с блоками

IDEF0: внутренние дуги

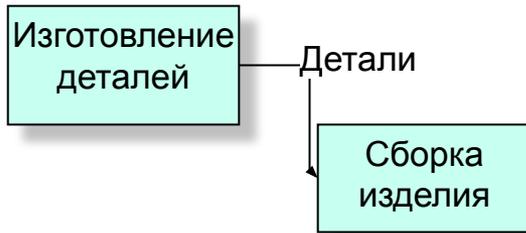


Внутренние дуги: выходы одних блоков могут являться входами (управлением, механизмом) других.

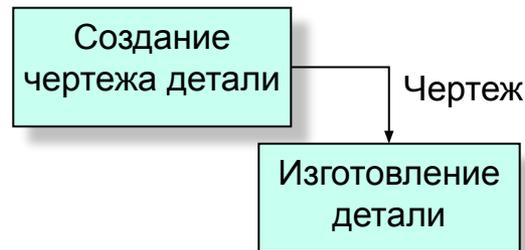
И внешние, и внутренние дуги могут **разветвляться**: одни и те же объекты могут использоваться сразу в нескольких других функциях. И внешние, и внутренние дуги могут **сливаться**: выходы нескольких функций могут использоваться в одном месте.

IDEF0: виды внутренних дуг

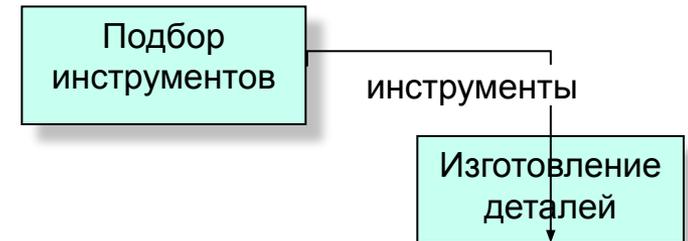
Выход-вход



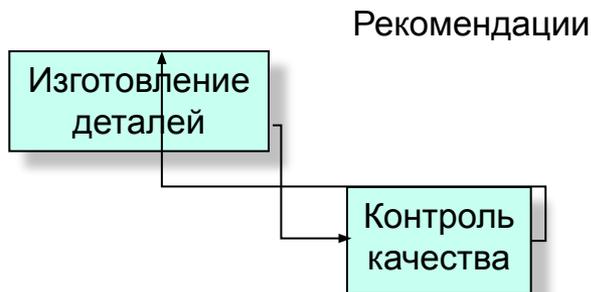
Выход-управление



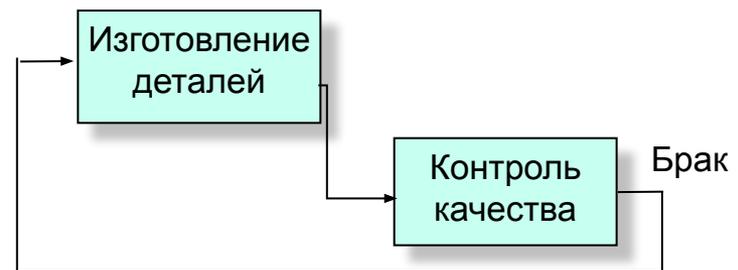
Выход-механизм



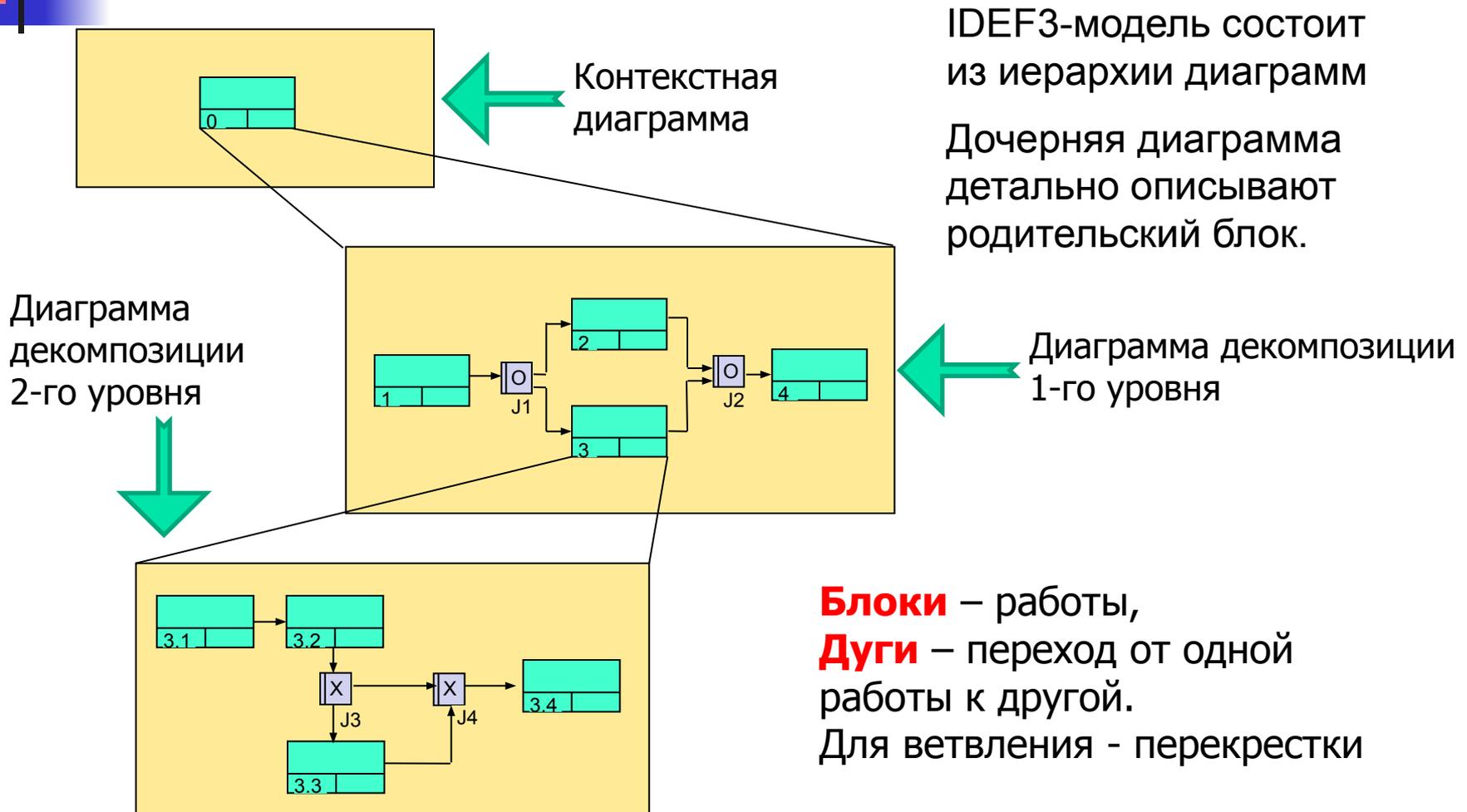
Обратная связь по управлению



Обратная связь по входу



IDEF3: иерархия диаграмм



IDEF3: элементы

Единица работы (Unit of work) - действие, процесс

Связь приоритета (Precedence) – после окончания первой работы начнется вторая

Отношение - соединяет ссылку с работой



Ссылка – элемент, связанный с работой

Перекресток – слияние или разветвление

IDEF3: элементы

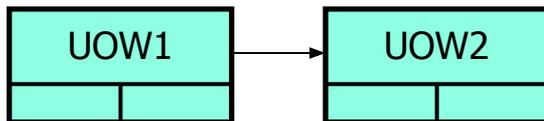


Единицы работ (Unit of work) - отображают действия, процессы, этапы выполнения работ. Единица работы может иметь только один вход и один выход

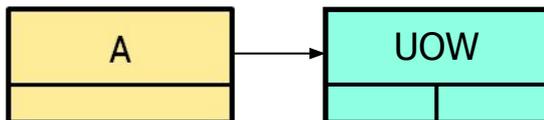


Ссылки (Referents):

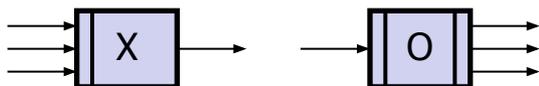
- необходимые элементы для выполнения процесса;
- результат процесса (изделие);
- активаторы процесса (клиент, поставщик).



Связь приоритета (Precedence) – после окончания первой работы начнется вторая



Отношения (Relational Link) - соединяют ссылку с работой

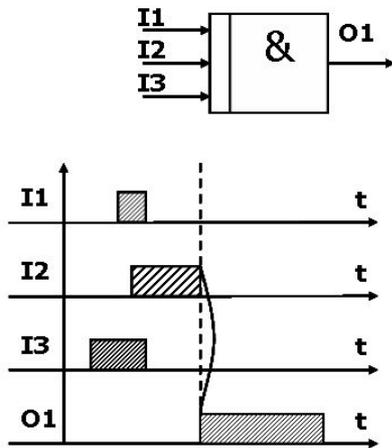


Перекрестки (Junctions). Типы перекрестков:

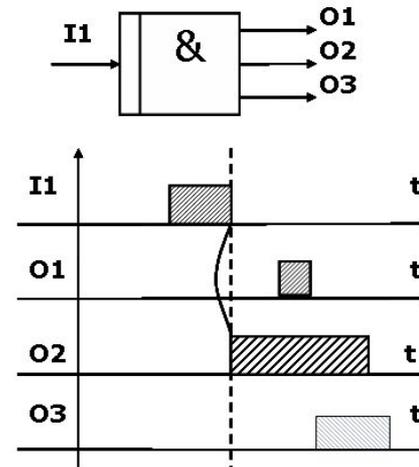
- слияния и разветвления,
- И (&), ИЛИ (O), Исключающее ИЛИ (X);
- синхронные, асинхронные

Типы перекрестков

1. Асинхронное И (Asynchronous AND)



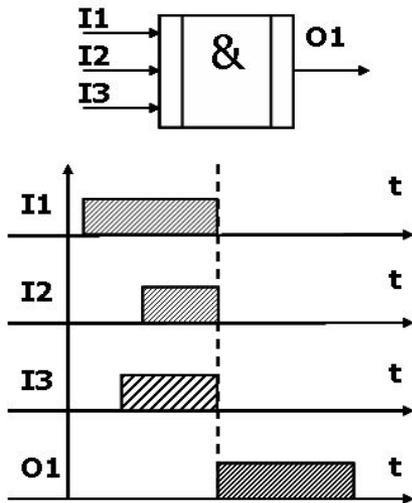
выходной процесс запустится, если завершились все входные процессы



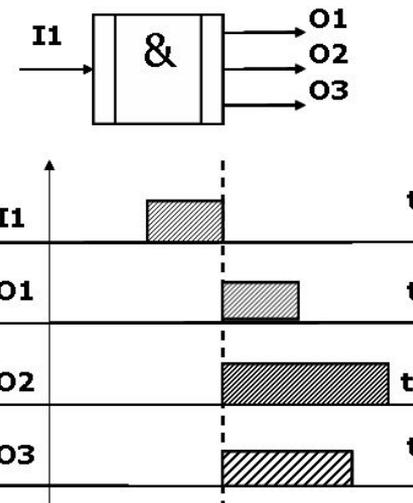
после завершения входного процесса запустятся все выходные процессы

Типы перекрестков

2. Синхронное И (Synchronous AND)



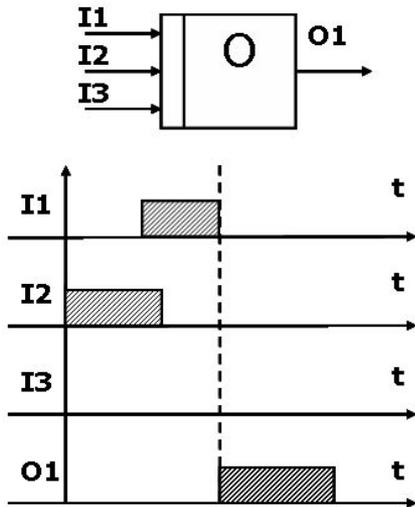
выходной процесс запустится, если завершились одновременно все входные процессы



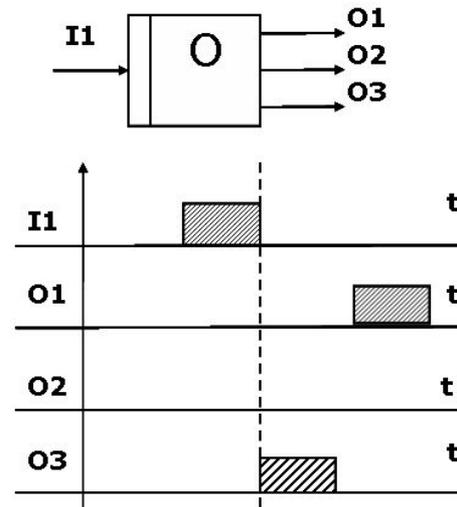
после завершения входного процесса запустятся все выходные процессы, причем запустятся одновременно

Типы перекрестков

3. Асинхронное ИЛИ (Asynchronous OR)



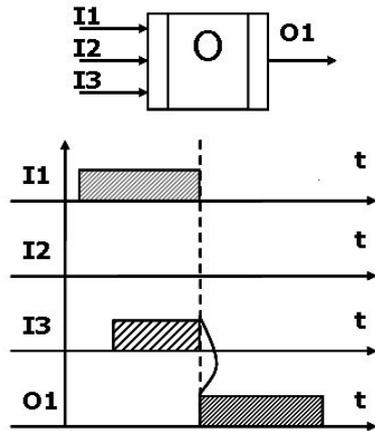
выходной процесс запустится, если завершится один или несколько входных процессов



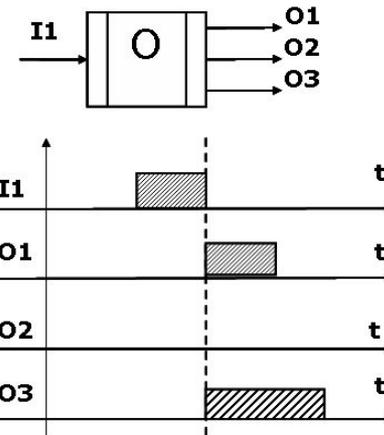
после завершения входного процесса запустятся один или несколько выходных процессов

Типы перекрестков

4. Синхронное ИЛИ (Synchronous OR)



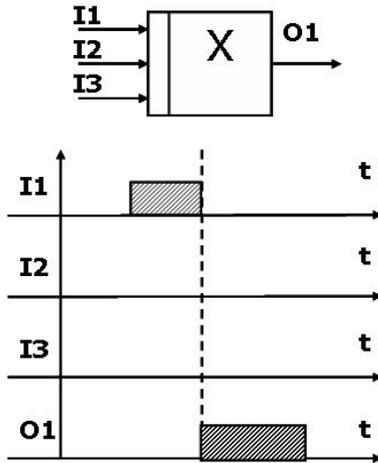
выходной процесс запустится, если завершились один или несколько входных процессов, причем завершились одновременно



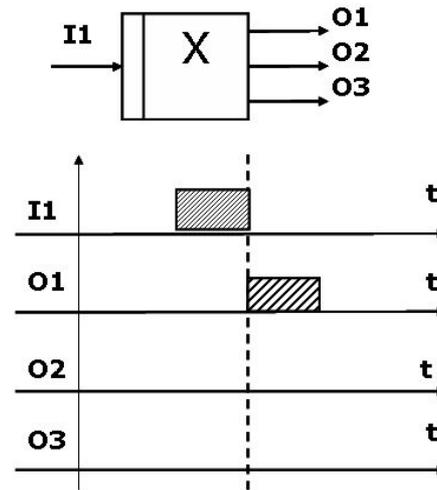
после завершения входного процесса запустится один или несколько выходных процессов, причем запустятся одновременно

Типы перекрестков

5. Исключающее ИЛИ (XOR, Exclusive OR)



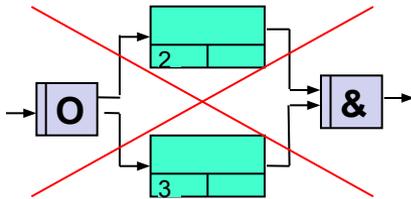
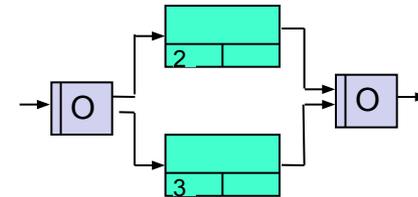
выходной процесс запустится, если завершился только один входной процесс



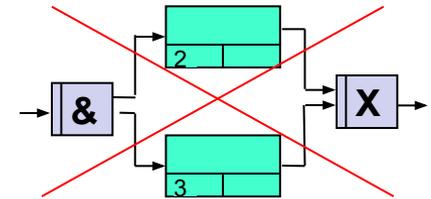
после завершения входного процесса запустится только один выходной процесс

Правила создания перекрестков

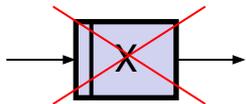
1. Каждому перекрестку слияния должен предшествовать перекресток ветвления.



2. Перекресток слияния «И» не может следовать за перекрестком ветвления типа «ИЛИ» (синхронного, асинхронного или исключаяющего).

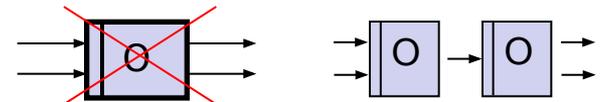


3. Перекресток слияния «Исключающее ИЛИ» не может следовать за перекрестком ветвления «И».

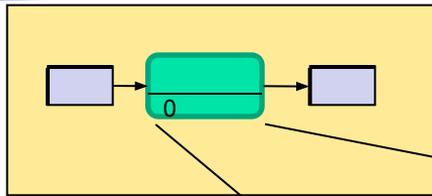


4. Перекресток, имеющий одну стрелку на одной стороне, должен иметь более одной стрелки на другой.

5. Перекресток не может быть одновременно перекрестком слияния и ветвления. При необходимости, вводится каскад перекрестков.



DFD: иерархия диаграмм



Контекстная диаграмма

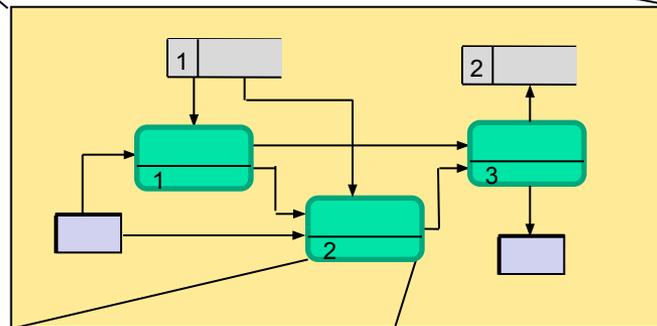
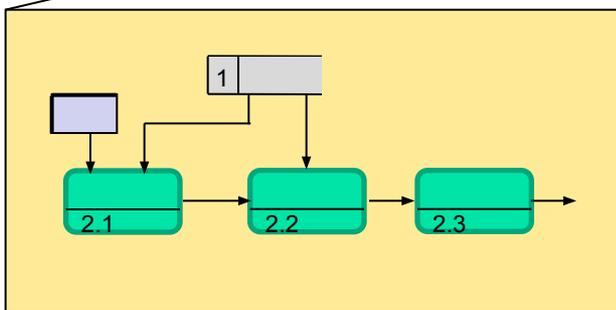


Диаграмма декомпозиции 1-го уровня

Диаграмма декомпозиции 2-го уровня



DFD-модель состоит из иерархии диаграмм

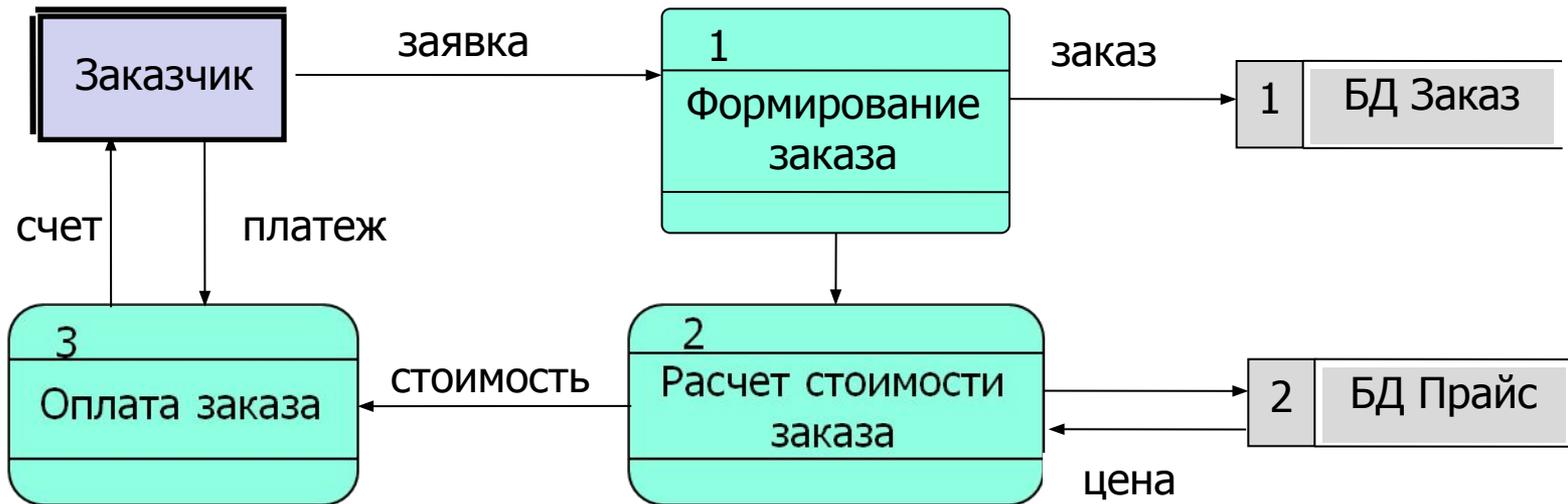
Дочерняя диаграмма детально описывают родительский блок.

Блоки – процессы обработки информации, **дуги** – данные. Данные могут храниться в хранилищах и могут поступать (передаваться) от внешних сущностей

DFD: элементы модели

Внешние сущности – элементы, обменивающиеся информацией с системой

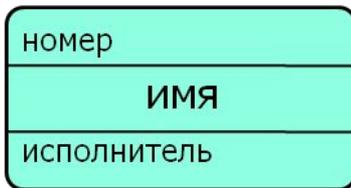
Процессы (функции, действия), которые обрабатывают информацию



Потоки данных - данные которые передаются

Хранилища данных - данные, которые хранятся

DFD: элементы модели



1. **Процессы** (функции, операции, действия), которые обрабатывают и изменяют информацию. Процессы показывают, каким образом входные потоки данных преобразуются в выходные
2. **Потоки данных**, которые обозначают взаимодействие процессов с внешним миром и между собой. Поток данных соединяет выход процесса (объекта) с входом другого процесса (объекта).
3. **Хранилища данных** - представляют собой собственно данные, к которым осуществляется доступ. Эти данные могут быть созданы или изменены процессами.
4. **Внешние сущности** - определяют внешние элементы, которые участвуют в процессе обмена информацией с системой. Они изображают входы в систему (источники информации) и/или выходы из системы (приемники информации). Примеры: *заказчик, персонал, поставщик, клиент, склад, банк*

Язык UML

UML-модель бизнеса

Прецедентная модель

Описывает **бизнес-процессы** (прецеденты) и окружение

Диаграмма вариантов использования

Описывает **взаимодействие** процессов с окружением

Диаграмма деятельности

Описывает **последовательность шагов** процесса

Объектная модель

Описывает **объекты** бизнес-процессов (исполнителей, заказы, услуги и т.д.),

Диаграмма классов

Описывает **классы** (типы) объектов и их связи

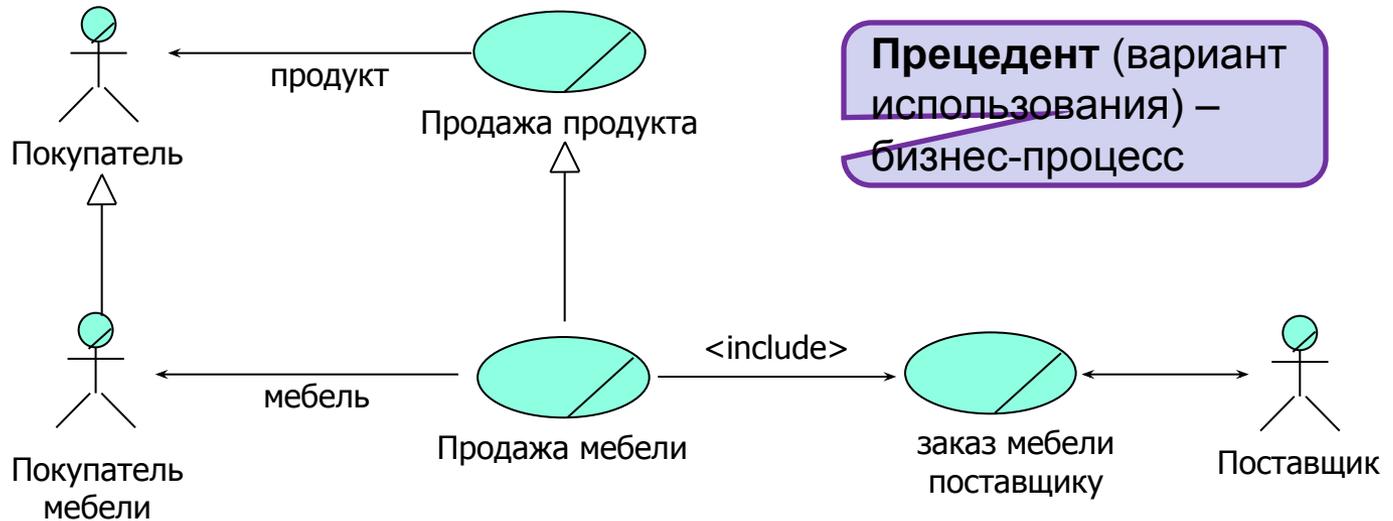
Диаграмма последовательности

Описывает **последовательность взаимодействия объектов** в ходе выполнения процесса

Диаграмма вариантов использования

ИСПОЛЬЗОВАНИЯ

Диаграмма вариантов использования (Use Case Diagram) отражает основные бизнес-процессы, их взаимодействие с окружением.



Прецедент (вариант использования) – бизнес-процесс

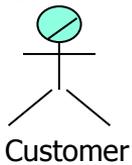
Отношение обобщения

Актор - субъект окружения бизнеса

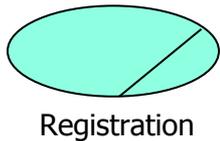
Отношение коммуникации

Отношение включения

Элементы диаграммы Use Case



Актор (действующее лицо, business actor) - субъект окружения бизнеса. Примеры акторов: Клиент, Покупатель, Поставщик, Партнер, Акционер, Заказчик.



Прецедент (вариант использования, business use case) - относительно законченная последовательность действий в рамках некоторого бизнес-процесса, приносящая ощутимый результат конкретному актору . Примеры прецедентов: Производство продукта Продажа продукта, Сервисное обслуживание, Разработка продукта, Маркетинг и сбыт.



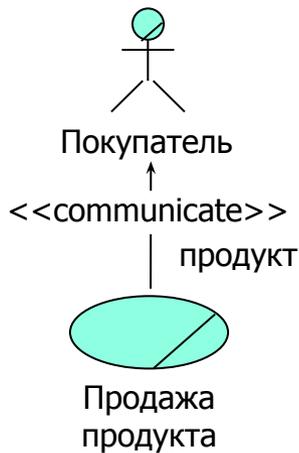
Экземпляр (реализация) прецедента – конкретный вариант хода событий **класс** прецедентов - обобщенный прецедент.

Для акторов тоже различают понятия класса и экземпляра.

Акторы разных классов могут иметь общие характеристики или общие обязательства.

Можно ввести **обобщенный класс** акторов. Между обобщенным типом актора и более конкретным устанавливается **отношение обобщения**

Элементы диаграммы Use Case



Между прецедентами и акторами устанавливаются **отношения коммуникации** (отношения ассоциации со стереотипом *communicate*).

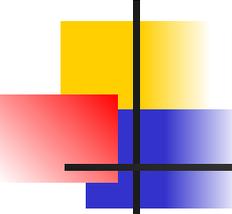
Они моделируют взаимосвязи прецедентов с окружением (информационные и материальные потоки)

Между прецедентами, как правило, устанавливаются только отношения зависимости а также отношения, структурирующие прецеденты – отношения *обобщения*, *включения* (зависимости со стереотипом *include*), *расширения* (зависимости со стереотипом *extend*).

Для каждого из элементов модели составляется спецификация.

В спецификации актора: наименование, стереотип (business actor), описание, список атрибутов, список обязательств и др.

В спецификации прецедента: наименование, стереотип (business use case), краткое описание, перечень связанных с прецедентом поддиаграмм и документов



Поток событий прецедента

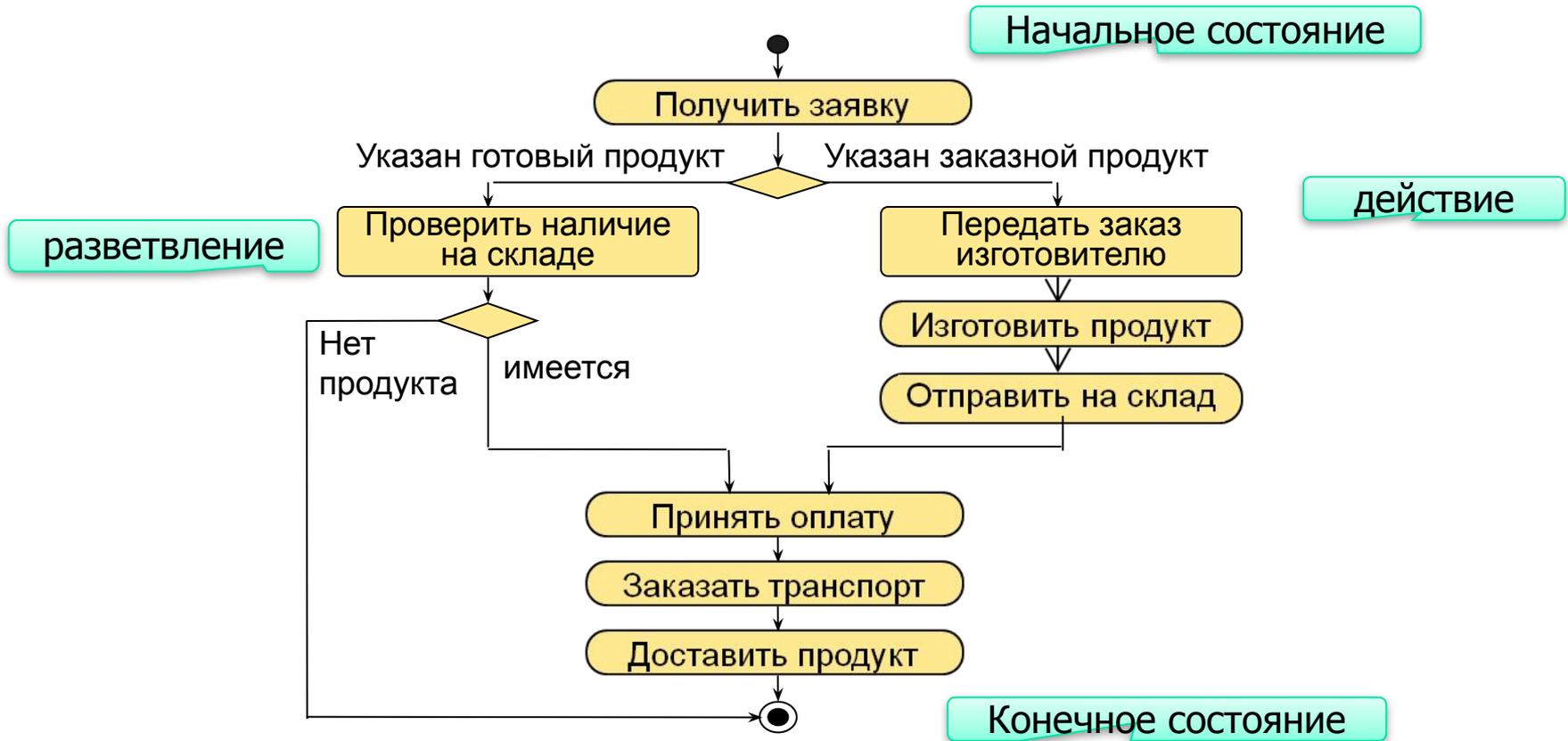
Поток событий - описание прецедентов последовательностью шагов

Поток событий прецедента «Продажа продукта»:

1. Продавец получает заявку клиента
2. Если в заявке указан готовый продукт, то Продавец проверяет наличие продукта на складе. Если продукта нет в наличии, прецедент заканчивается. Если продукт есть на складе, то прецедент продолжается с шага 6.
3. Если в заявке указывается заказной продукт, то Продавец формирует заказ и передает его Изготовителю продукта.
4. Изготовитель изготавливает продукт в соответствии с требованиями клиента и сообщает о готовности Продавцу.
5. Изготовитель отправляет продукт на Склад.
6. Продавец сообщает Клиенту о готовности продукта и принимает от Клиента оплату.
7. Продавец сообщает Отправителю количество продукта и адрес клиента и заказывает транспорт.
8. Отправитель получает продукт со склада и доставляет его клиенту.

Диаграмма деятельности

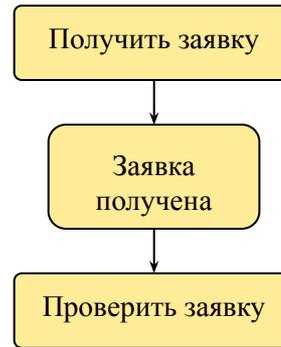
Диаграмма деятельности отражает последовательность действий



Элементы диаграммы деятельности

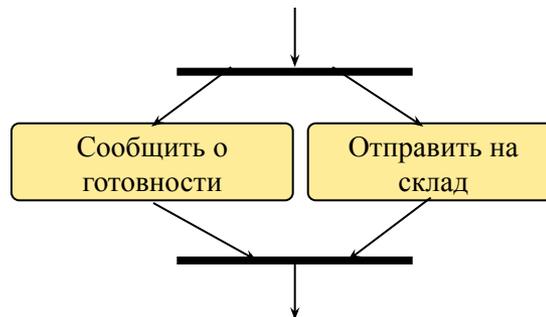
-  начальное состояние
-  конечное состояние
-  действие
-  переход
-  ветвление
-  синхронизация
-  состояние

Чередование событий и состояний



Каждый шаг (действие) переводит прецедент в новое состояние. В свою очередь, новое состояние является стимулом для выполнения следующего шага. Т.о. прецедент –это **машина состояний-событий**

Распараллеливание потока



Разветвление потока

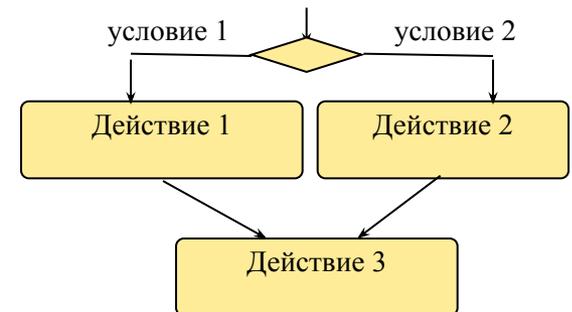
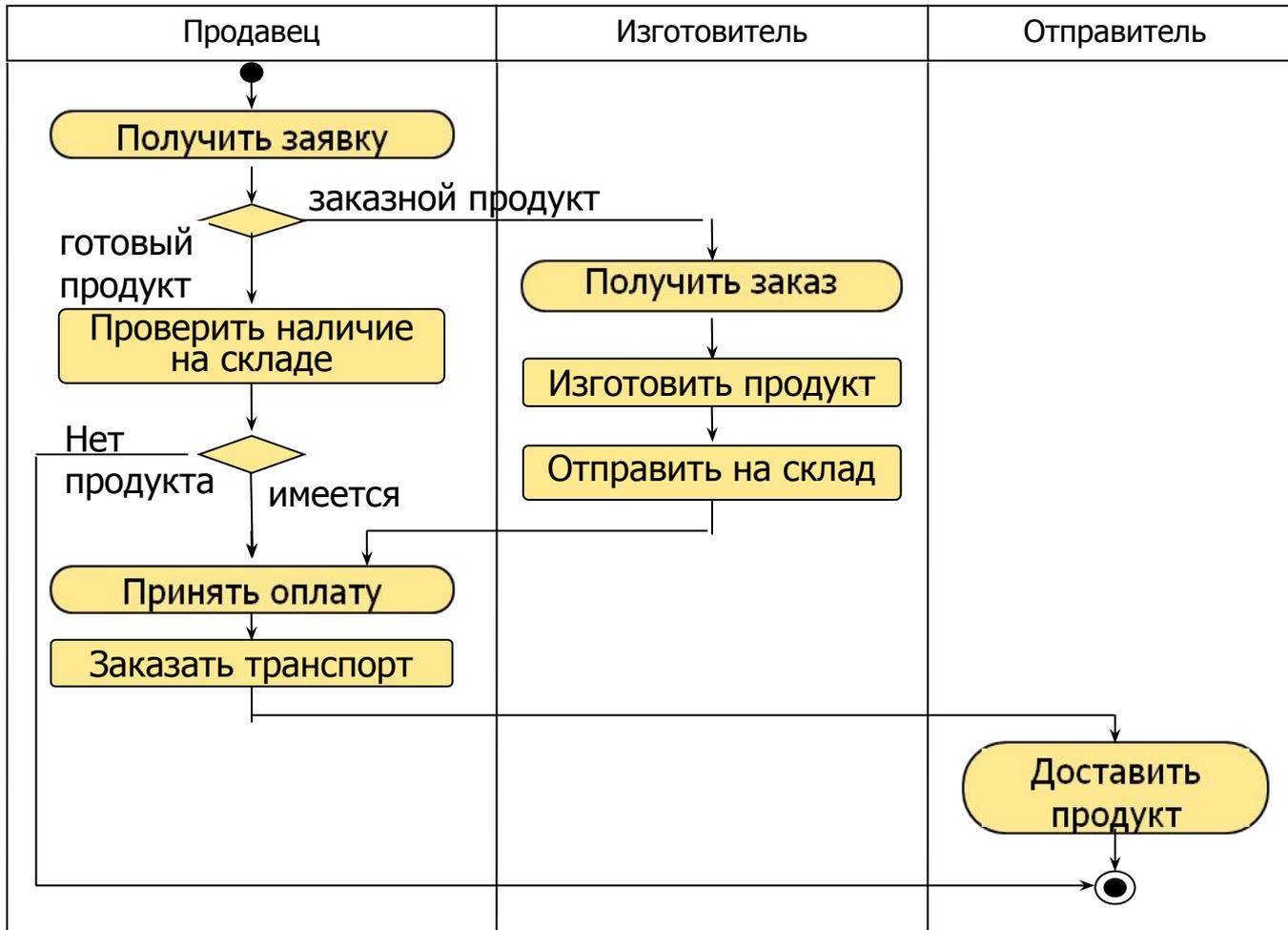
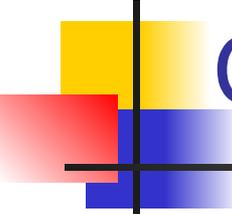


Диаграмма деятельности



Дорожки:
 Если в выполнении прецедента участвуют несколько объектов, то действия, выполняемые каждым объектом, размещаются на соответствующей дорожке



Структурирование прецедентов

Чтобы упростить описание прецедента, необходимо его **структурировать**. Рассмотрим два способа структурирования.

1. Выделение фрагментов

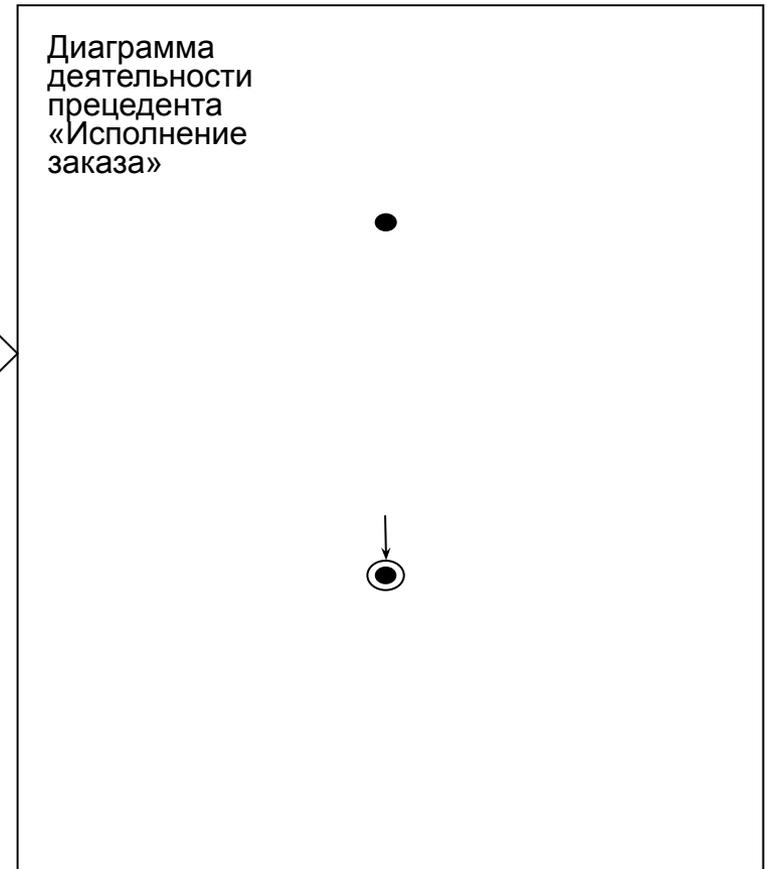
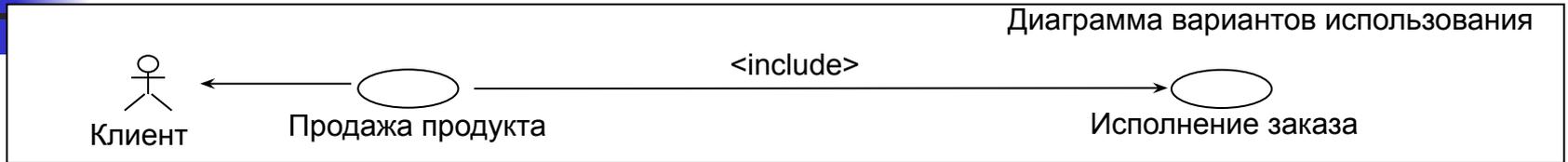
Если из описания прецедента с альтернативными потоками событий можно выделить **фрагмент**, представляющий собой относительно законченную последовательность событий, то данный фрагмент рассматривается как отдельный прецедент. Между выделенным прецедентом и базовым устанавливается **отношения включения (include)**.

Иногда используют **отношение расширения (extend)**. Оно устанавливается между базовым прецедентом и прецедентом, содержащим некоторое дополнительное поведение, выполняемое при определенных условиях.

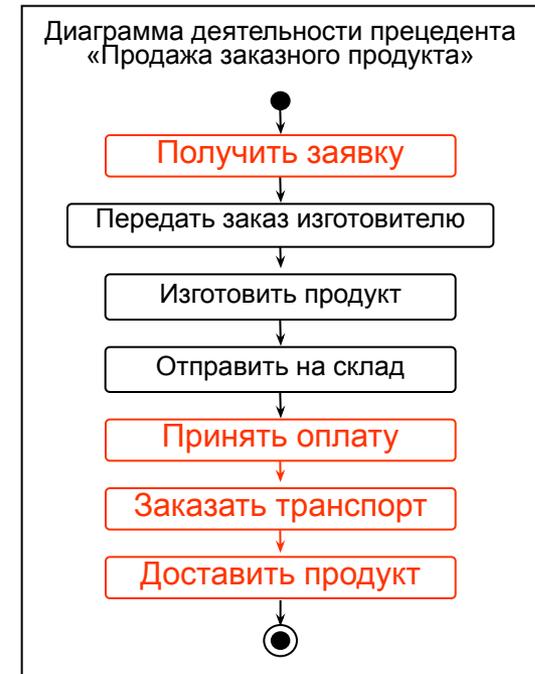
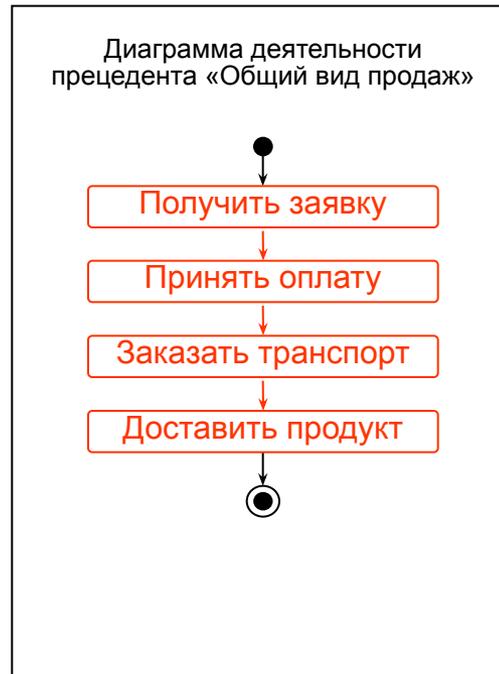
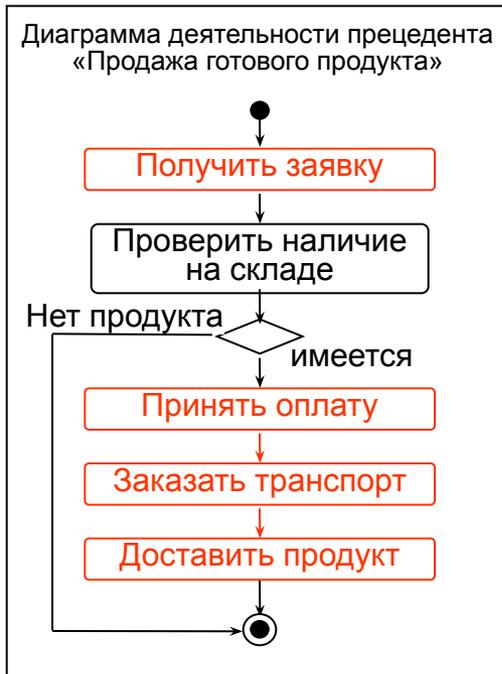
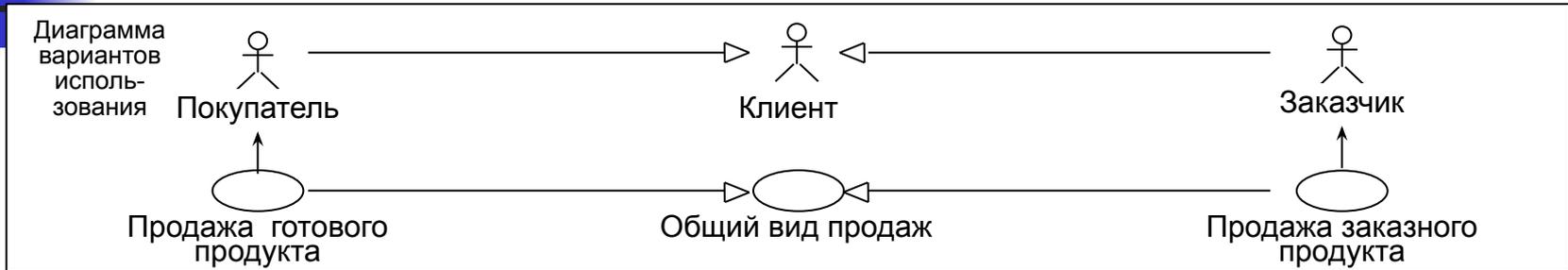
2. Обобщение

Если несколько прецедентов имеют похожее поведение, то следует выделить **общее** поведение в отдельный прецедент (родительский). Между каждым из частных прецедентов и родительским устанавливается **отношение обобщения (generalization)**.

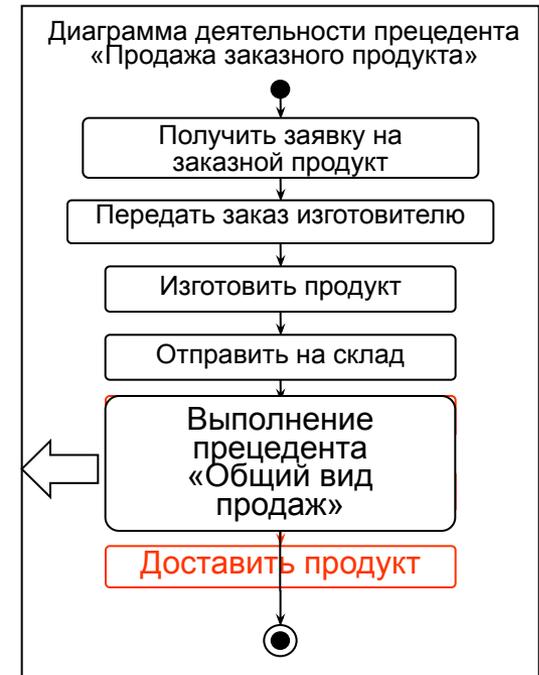
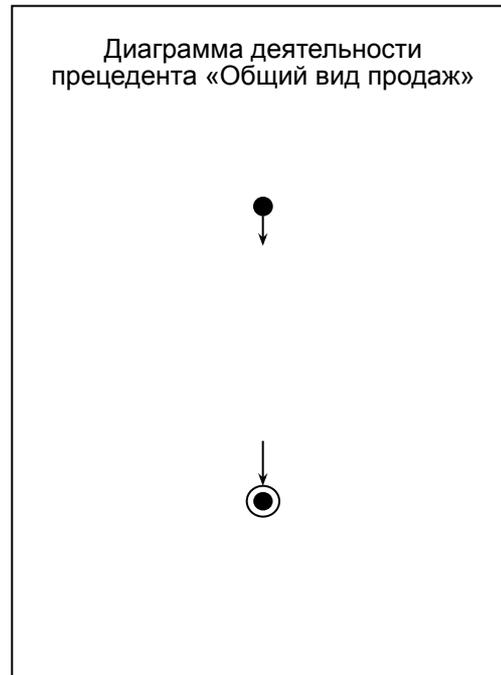
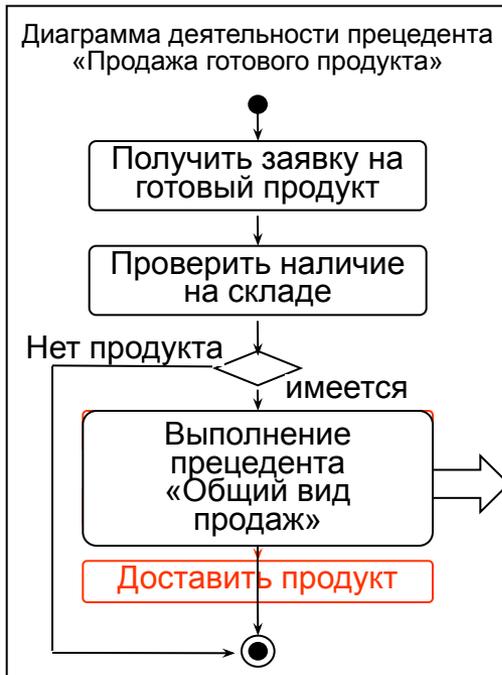
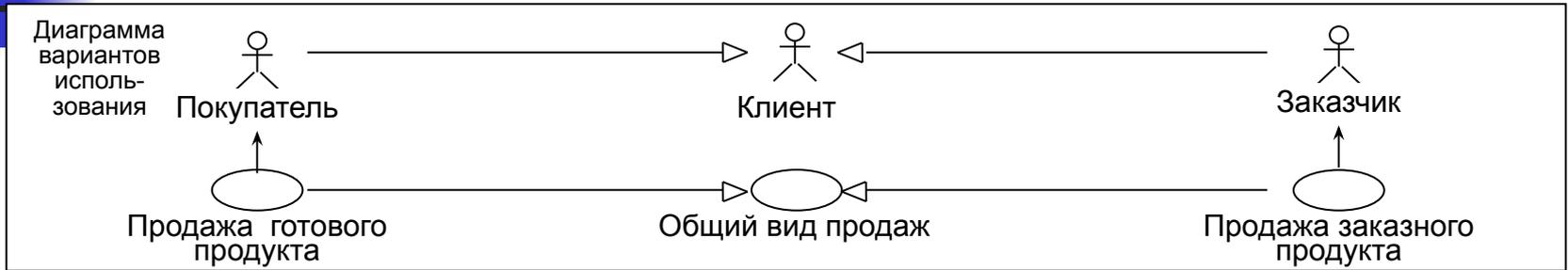
Структурирование прецедентов выделением фрагментов



Структурирование прецедентов обобщением



Структурирование прецедентов обобщением



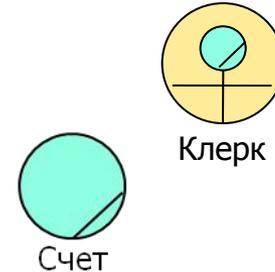
Объектная модель бизнес-процесса

Раскрывает **внутреннее** устройство бизнеса: какие виды ресурсов используются для реализации прецедентов и каким образом они взаимодействуют.

Классы объектов модели бизнеса:

активные – **бизнес-работники** (стереотип business worker), например, Продавец, Изготовитель, Разработчик;

пассивные - **сущности** (стереотип business entity), например, Продукт, Заказ, Счет.



Класс – некоторый *тип* объектов (множество похожих объектов),

Экземпляр – конкретный объект (представитель класса).

Объекты имеют:

ИМЯ (через двоеточие может быть указано имя класса)

СВОЙСТВА - описываются с помощью **атрибутов**

поведение - представляется с помощью **операций**

<u>Продавец1: Продавец</u>
ФИО: Иванов И.П. Стаж: 5
Получить заказ Принять оплату

У объектов одного класса состав атрибутов и операций одинаков.

Они отличаются значениями атрибутов,

Диаграмма классов

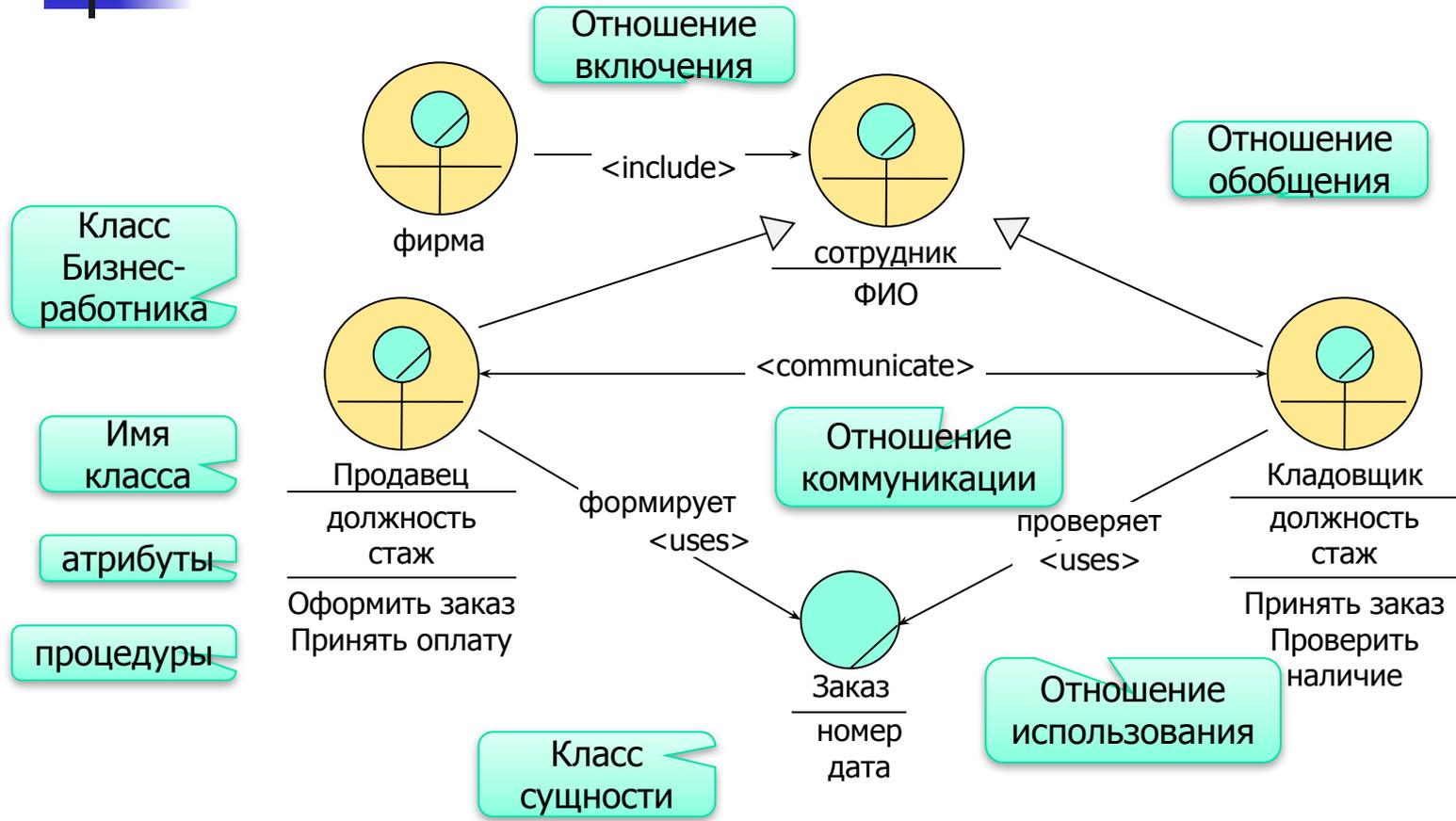
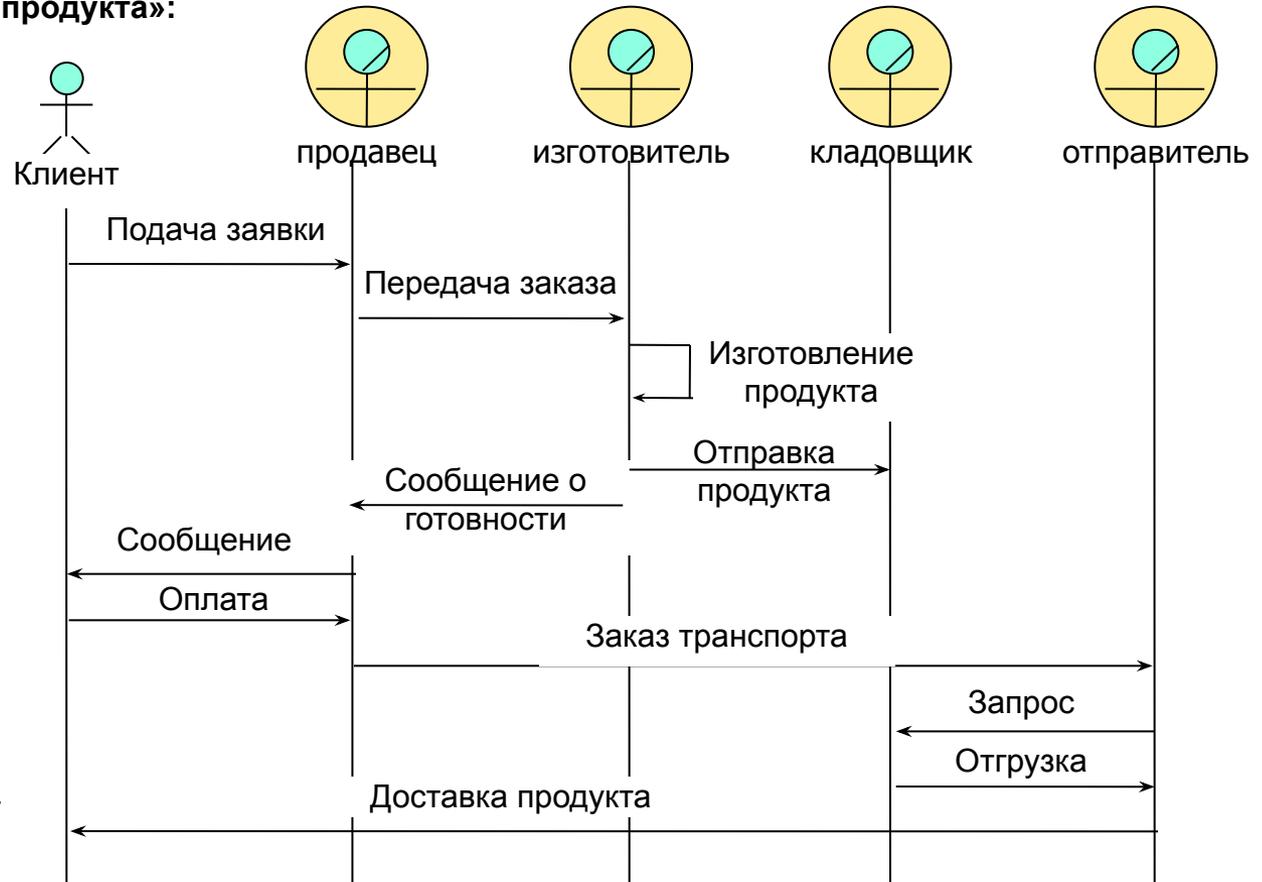


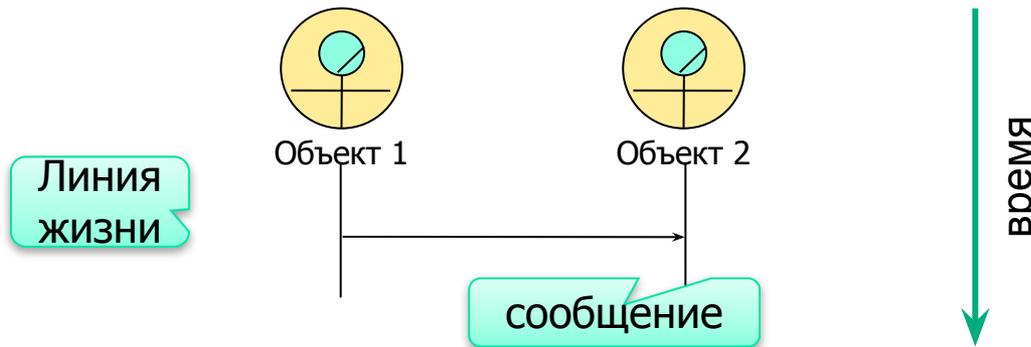
Диаграмма последовательности

Прецедент «Продажа заказного продукта»:

1. Продавец получает заявку клиента
2. Продавец формирует заказ и передает его Изготовителю продукта.
4. Изготовитель изготавливает продукт.
5. Изготовитель отправляет продукт на Склад и сообщает о готовности Продавцу.
6. Продавец сообщает Клиенту о готовности продукта и принимает от Клиента оплату.
7. Продавец сообщает Отправителю адрес клиента и заказывает транспорт.
8. Отправитель получает продукт со склада и доставляет его клиенту.



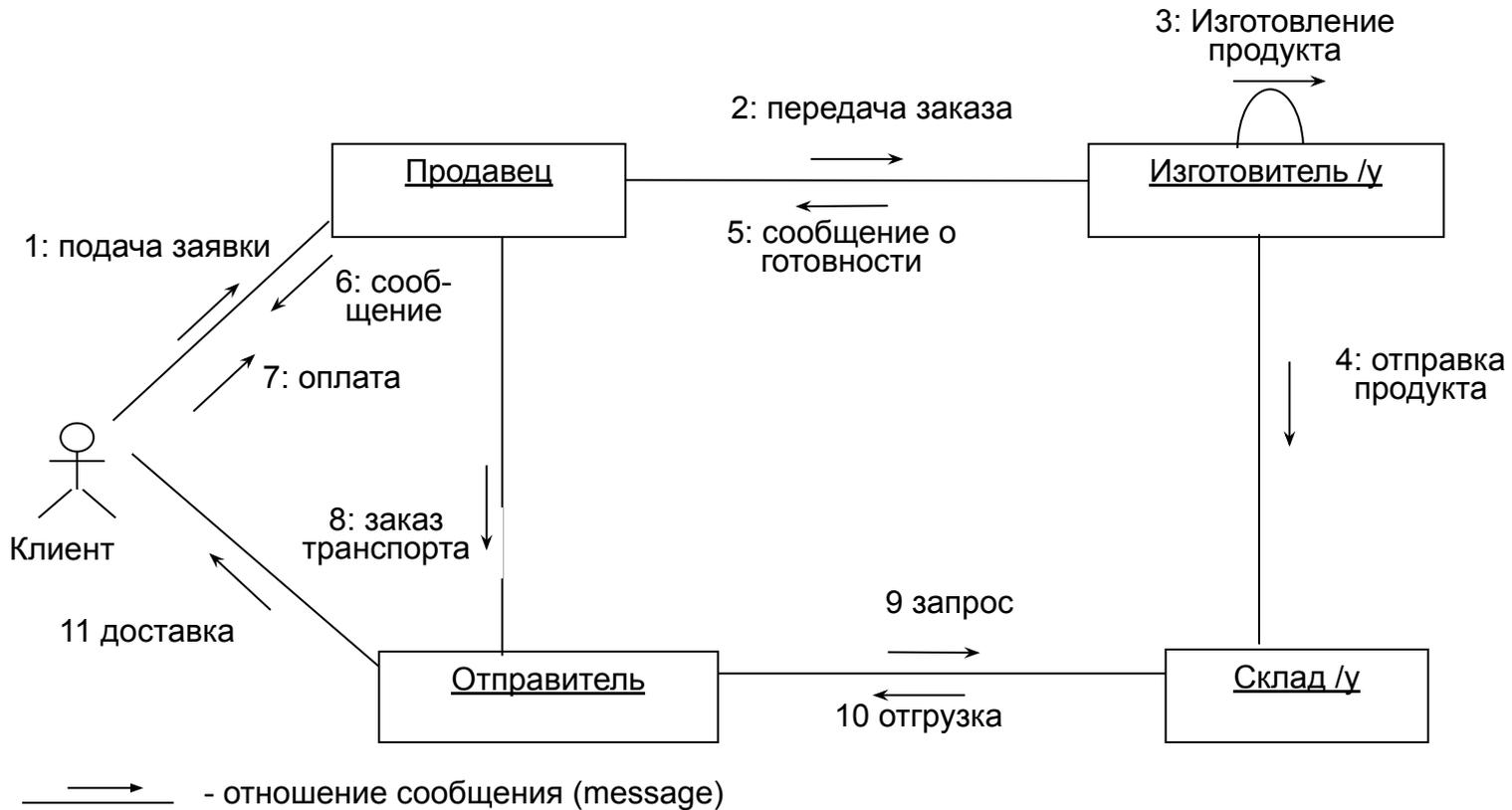
Элементы диаграммы последовательности



Отношение сообщения моделирует материальный или информационный поток. Прием сообщений инициирует выполнение некоторого действия получателем

Сообщения упорядочены по времени: первое сообщение изображается вверху диаграммы, следующее – ниже, следующее – еще ниже и т.д. Однако диаграмма не содержит метрики времени

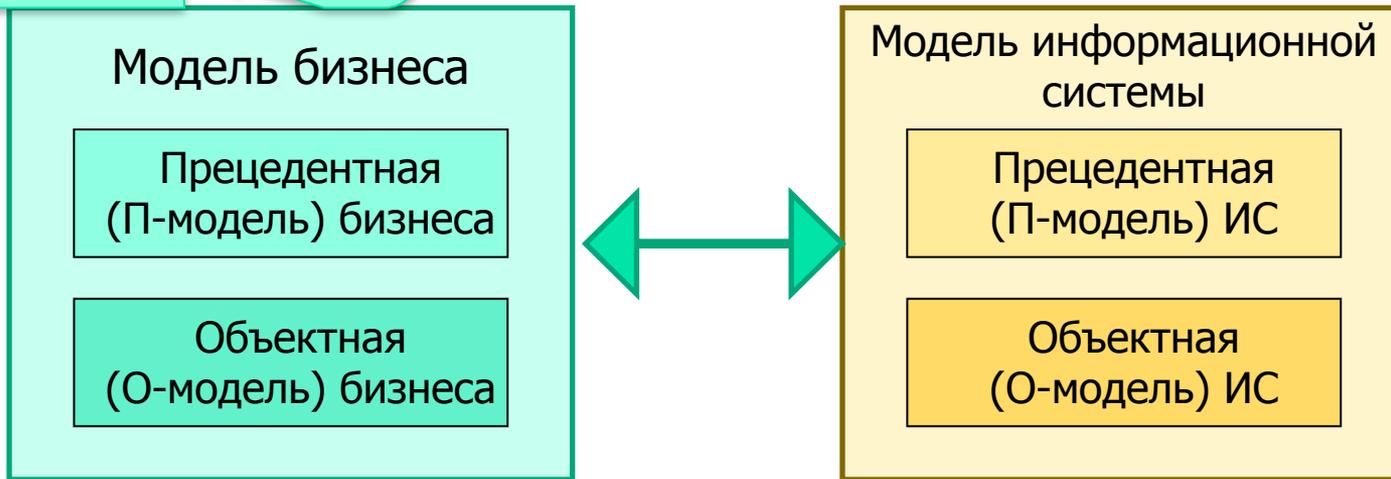
Диаграмма кооперации (Collaboration Diagram)



Построение ИС поддержки

Модель бизнес-процесса «Как должно быть» описывает выполнение процесса с помощью ИС

Модель ИС строится на основе модели бизнес-процесса «Как должно быть»



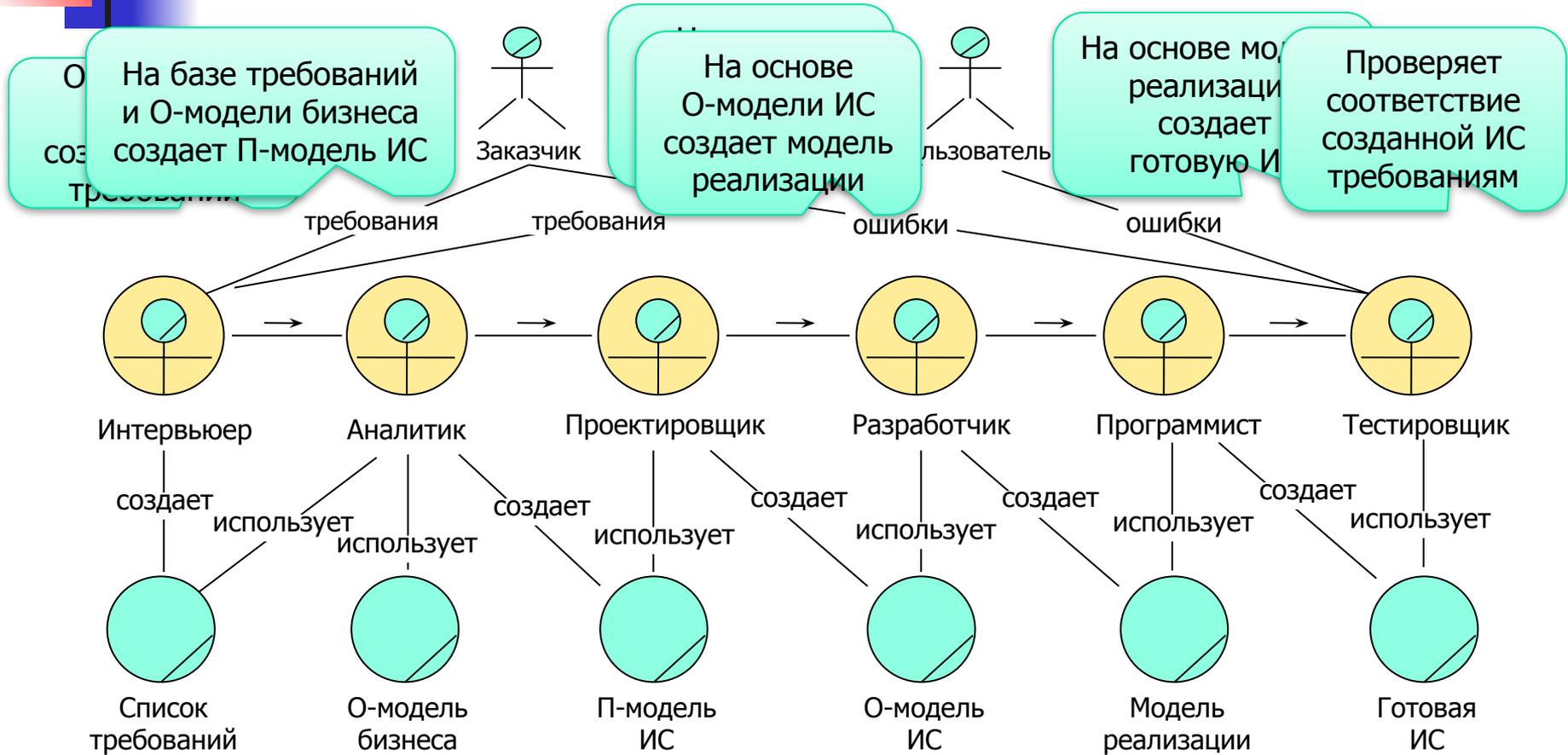
На протяжении всей разработки необходимо предусмотреть несколько итераций между формированием модели бизнеса и созданием ИС

Прецедент «Разработка ИС»



1. **Сбор требований.** Опрос потенциальных Пользователей ИС и Заказчиков, формирование Списка требований.
2. **Анализ требований.** Определяются функции системы и ее интерфейсы. Создается П-модель ИС и макет
3. **Идеальное проектирование** (логическое). Проектировщик создает О-модель ИС (выделяет объекты ИС, описывает их структуру, взаимодействие) и уточненный прототип системы.
4. **Реальное проектирование** (физическое) Разработчик адаптирует О-модель ИС к реальным условиям, создает физическую структуру БД, спецификации компонент, макеты экранных форм и т.д.
5. **Реализация.** Программист на основании модели реализации разрабатывает программу, создает базы данных, сопроводительную документацию и т.д.
6. **Тестирование.** Тестировщик проверяет соответствие созданной ИС предъявляемым к ней требованиям

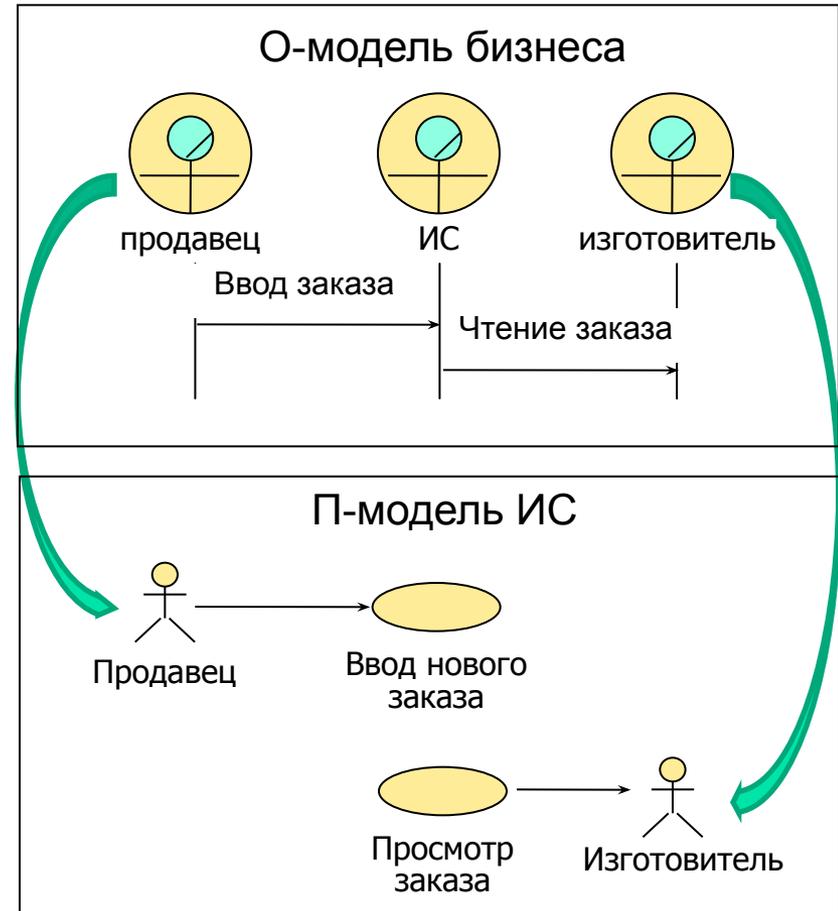
Диаграмма кооперации объектов прецедента «Разработка ИС»



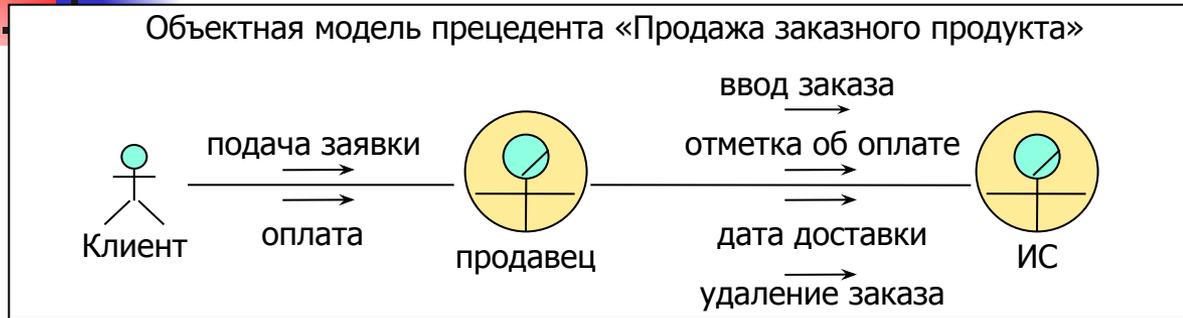
Связь О-модели бизнеса и П-модели ИС

В О-модели бизнеса Информационная система (ИС) является активным объектом (business worker), взаимодействующим (принимаящим и передающим сообщения) с другими активными объектами

В П-модели информационной системы активные объекты бизнеса (business worker), взаимодействующие с ИС, становятся акторами (пользователями) ИС, взаимодействующими с прецедентами ИС

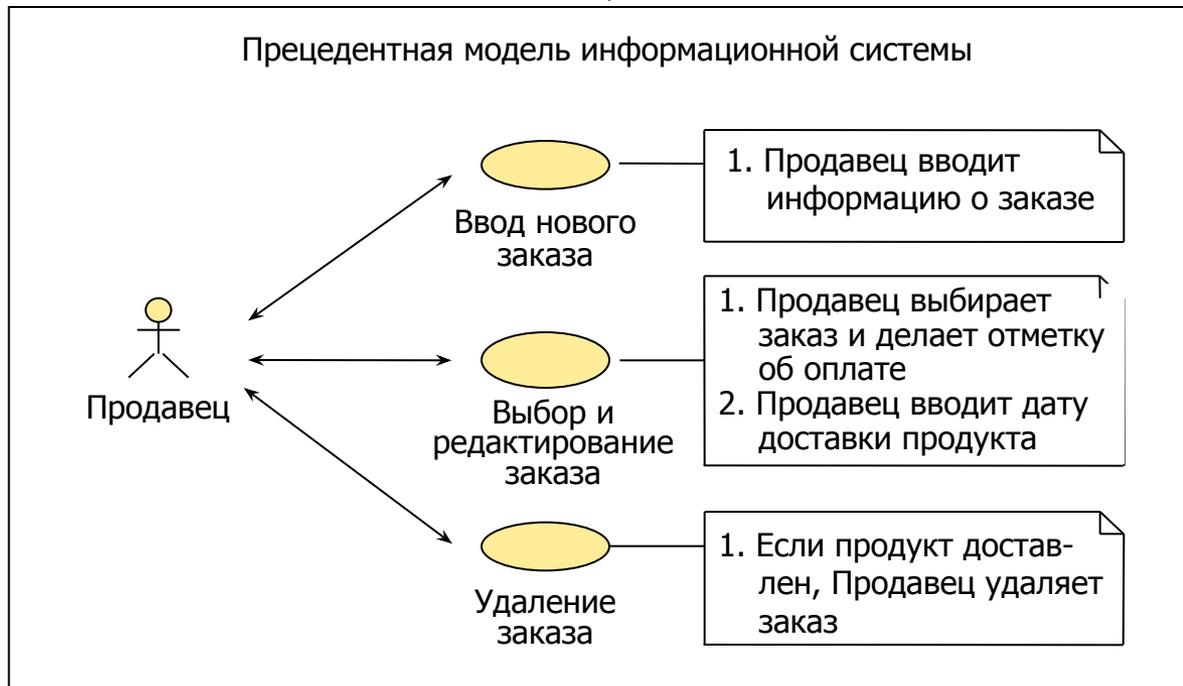


Построение П-модели ИС



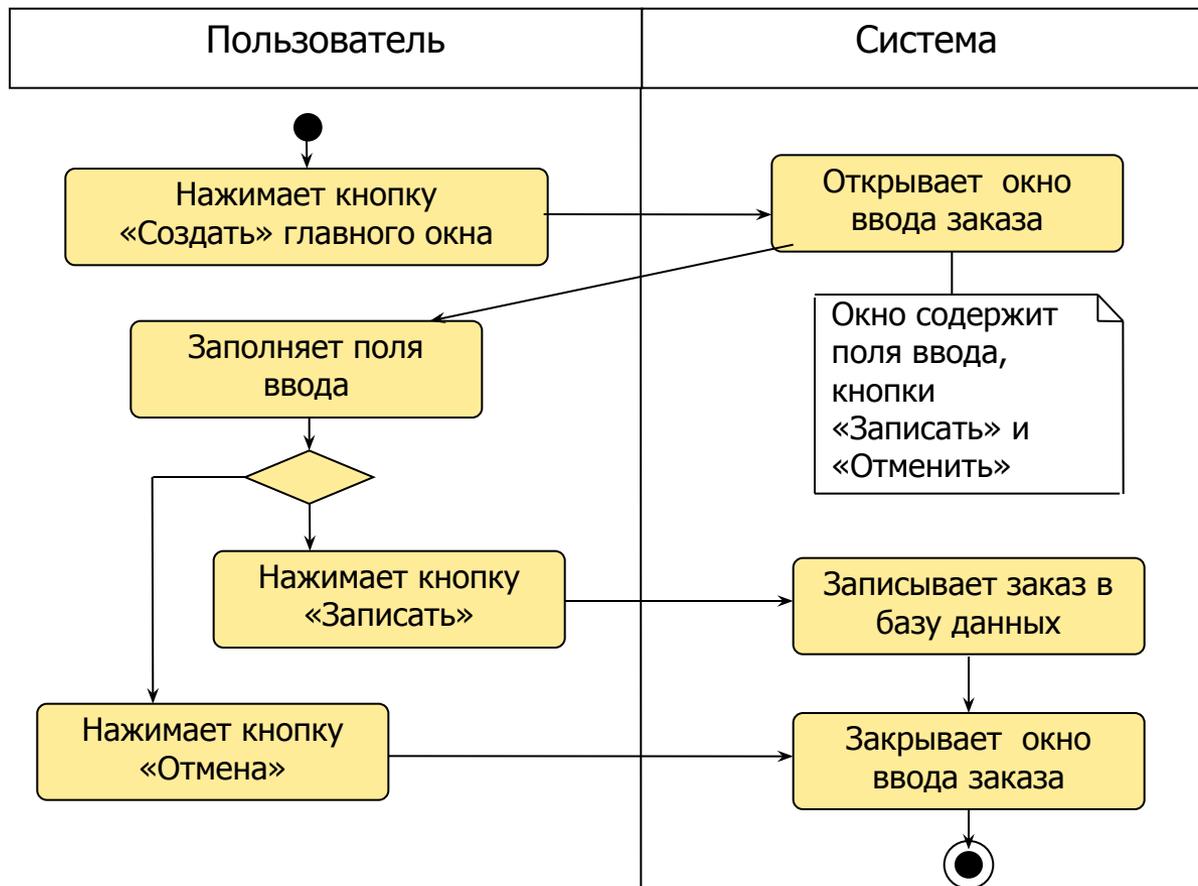
Последовательность шагов :

1. Выбирается активный объект (или актер) О-модели бизнеса использующий ИС в своей деятельности
2. В П-модели ИС ему сопоставляется актер – пользователь ИС
3. Если в О-модели бизнеса некоторое обязательство объекта выполняется с помощью ИС, то оно вносится в описание прецедента П-модели ИС
4. Если рассмотрены не все объекты О-модели бизнеса, использующие ИС, то все шаги повторяются для очередного объекта



Поток событий прецедента ИС

Диаграмма деятельности прецедента ИС «Ввод нового заказа»



Имитационное моделирование

Имитационное моделирование позволяет **воспроизводить процесс** функционирования системы во времени, осуществлять многократные испытания модели с разными входными данными

Позволяет анализировать **случайные процессы** и выявлять их интегральные характеристики

Пример обслуживания клиента в банке.

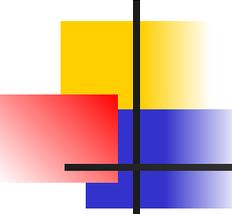
Время прихода клиентов в банк - раз в 15-20 минут.

Требуемые клиентами операции: операция1 – вероятность 0.25, операция2 – вероятность 0.40, операция3 – вероятность 0.35.

Время выполнения операций: для операции 1 - 20 ± 5 мин, для операции 2 - 40 ± 10 мин, для операции 3 - 120 ± 15 мин.

Распределение операций по кассам: операция 1 – касса 1, операция 2 – кассы 2 и 3, операция 3 – кассы 4 и 5.

Нужно определить среднее время нахождения клиента в системе, минимальное и максимальное время ожидания в очереди.



Система Arena

Инструментальное средство имитационного моделирования Arena позволяет воспроизводить процессы массового обслуживания.

Пользователь может:

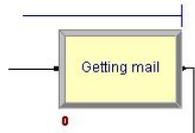
- создать модель – создать графическую диаграмму (схему), описывающую последовательность выполнения процесса
- «проиграть» модель – задать время имитации и запустить процесс
- сформировать отчеты по результатам «проигрывания» модели
- проанализировать результаты моделирования.

Система Arena использует язык визуального имитационного моделирования **SIMAN (SIMulation Analysis)**

Элементы модели SIMAN



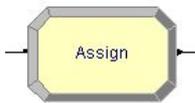
создает
сущности



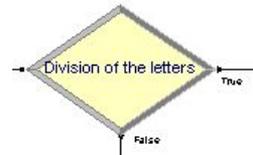
обрабатывает
сущности



удаляет
сущности



задает атрибут
сущности



разветвляет процесс
по условию

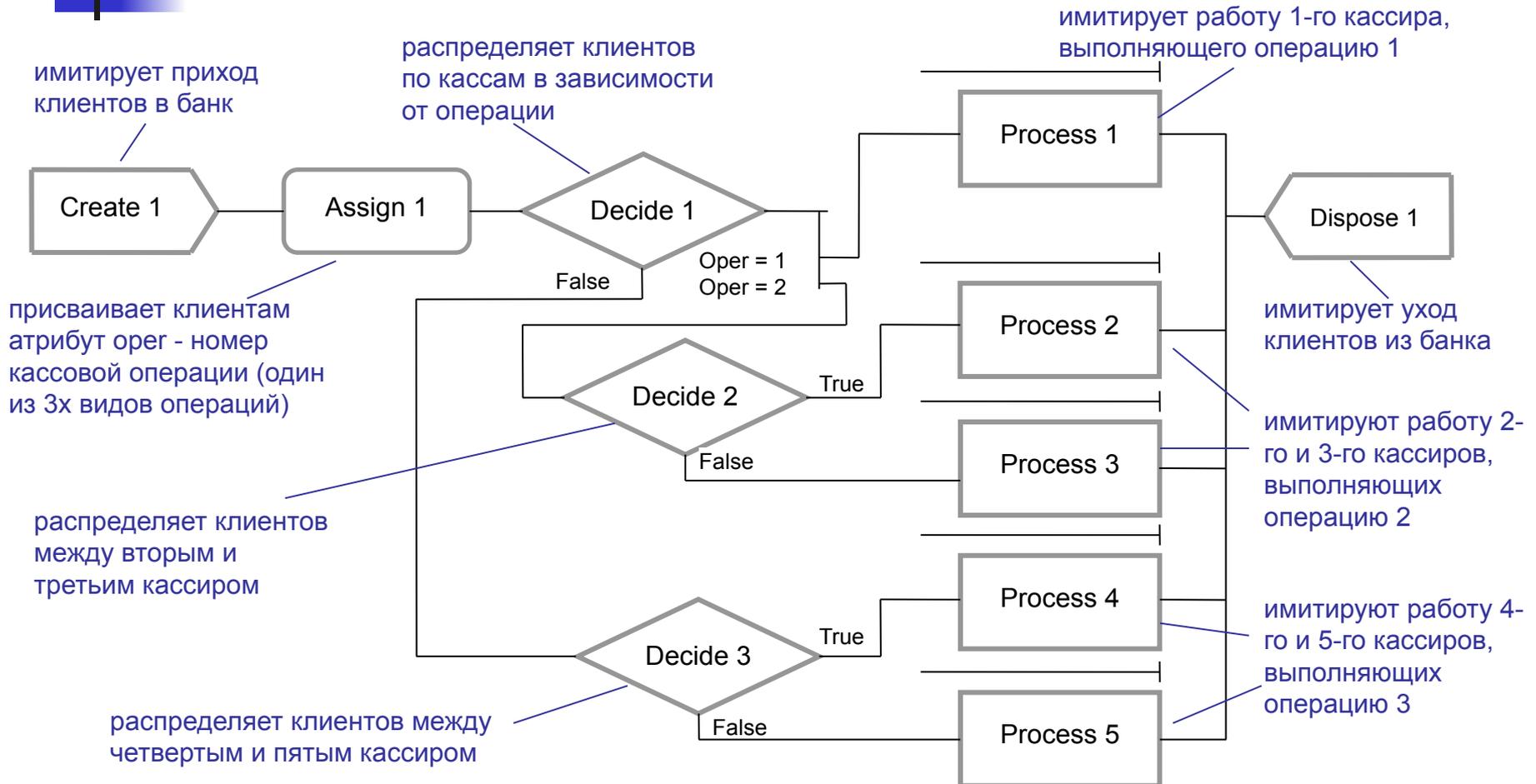
Процессы – работы, операции, действия. Изображаются в виде графических модулей. Они соединяются в общую схему.

Для каждого процесса задаются параметры, например:

- ресурсы, выполняющие процессы – люди (продавец, кассир) или оборудование (станок, компьютер);
- время обработки сущности в виде статистической функции (равномерное распределение, экспоненциальное, ...);

Сущности – заказы, документы, заготовки изделий, клиенты. Они обрабатываются процессами. Например: приход клиента (создание сущности), присвоение номера операции (задание атрибута), выбор кассы (проверка условия и переход), выполнение операции (обработка сущности), уход клиента (удаление сущности). Перед процессами могут образовываться **очереди** из сущностей, ожидающих обработки, если процессы в данный момент заняты

Имитационная модель



Проигрывание модели

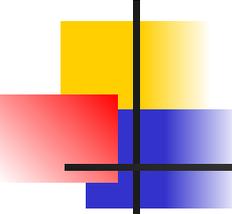
Пользователь задает **условия окончания** эксперимента - общее время проигрывания или количество сущностей, которые должны пройти через систему. Задает характеристики системы, по которым требуется получить **статистику**, и запускает имитацию.

Режим продвижения модельного времени **от события к событию**.

Сначала часы модельного времени - в 0. Затем часы продвигаются ко времени следующего ближайшего события и т.д. пока не будет выполнено условие останова.

Виды отчетов, формируемые ПП «Arena»:

- по сущностям – общее время нахождения в системе, время ожидания, (среднее, максимальное и минимальное значение) и др.;
- по очередям – среднее, максимальное и минимальное время ожидания в очереди, количество сущностей, ожидающих в очереди;
- по процессам – статистика по характеристикам времени и стоимости (среднее, максимальное и минимальное значение);
- по ресурсам – статистика по затраченным ресурсам.



Методология ARIS

ARIS (Architecture of Integrated Information System) или **АРИС (Архитектура Интегрированных информационных систем)** разработана в 1990-х годах профессором А.-В. Шеером

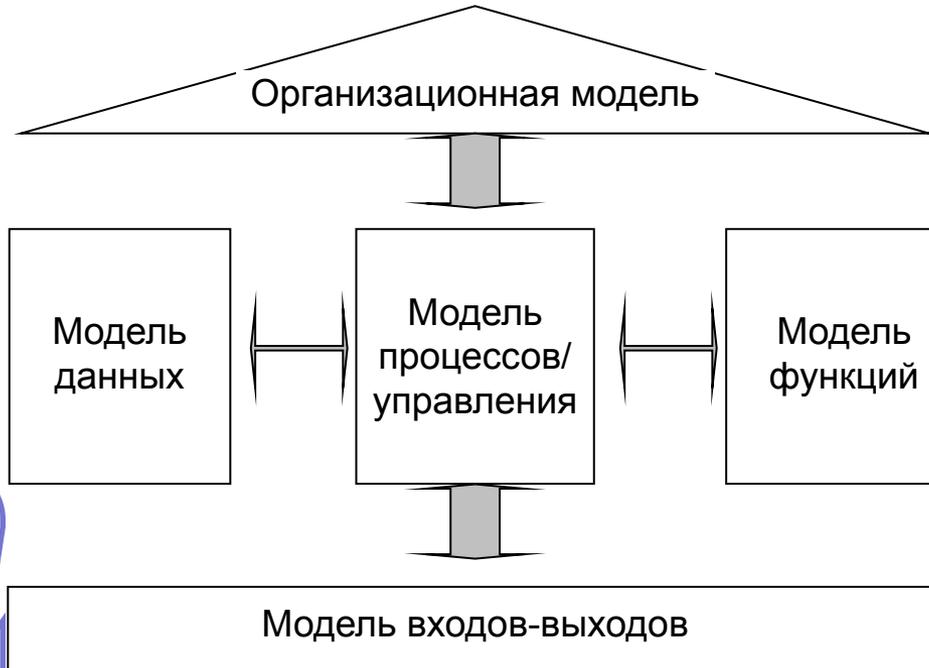
Преимущества:

- *Возможность рассматривать объект с различных точек зрения; дифференцируемый взгляд на анализируемый объект (организацию, систему управления)*
- *Богатство методов моделирования*
- *Единый репозиторий обеспечивает построение интегрированной и целостной модели предметной области*
- *Модели имеют широкое применение (анализ и совершенствование бизнес-процессов, проектирование ИС, разработка систем менеджмента качества)*
- *Возможность анализа и сравнения разных вариантов моделей*

5 типов представлений

структура информации, необходимой для реализации функций системы

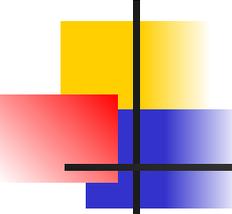
комплексный взгляд на реализацию деловых процессов в рамках системы



структура организации (иерархия подразделений и должностей)

иерархия функций (целей), выполняемых в организации

потоки материальных и нематериальных входов и выходов



Типы моделей

Для каждого из 5 представлений можно построить несколько типов моделей

Общее количество типов диаграмм – свыше 100 .

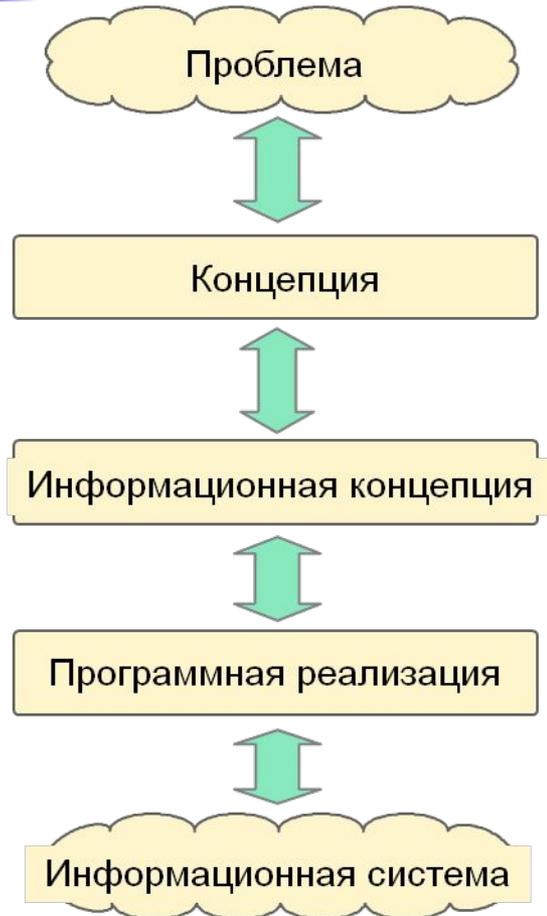
Модели классифицируются при помощи методологических фильтров

Методологические фильтры – регулируемые, переключаемые наборы моделей:

- **простой фильтр (easy filter);**
- **стандартный фильтр (standard filter);**
- **расширенный фильтр (extended standard filter);**
- **фильтр для модуля имитации (Simulation);**
- **фильтр для модуля функционально-стоимостного анализа (ABC)**

и др.

Уровни описания ИС



Анализ проблем бизнеса, которые предполагается решить с помощью информационной системы

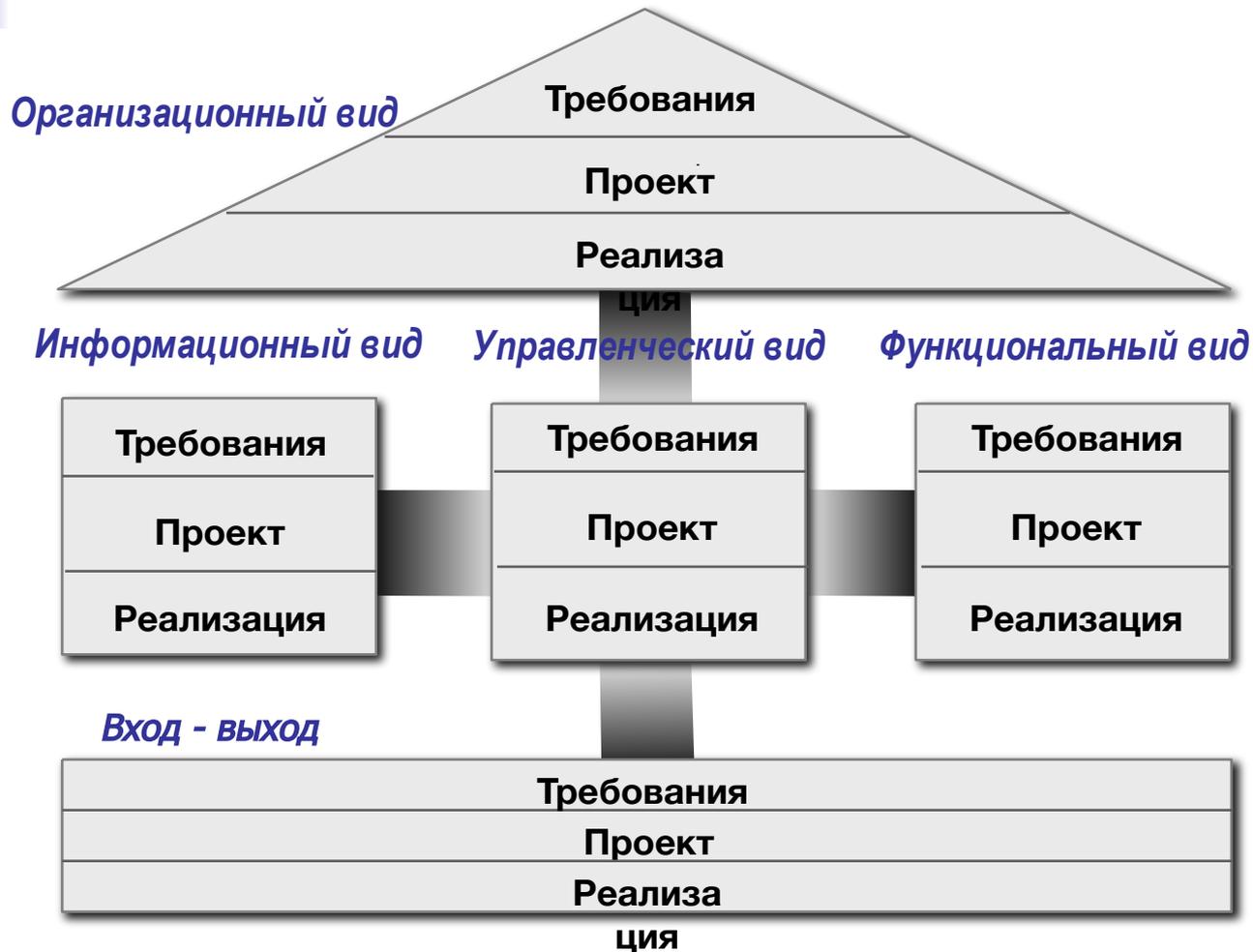
Определение требований к прикладной информационной системе, создаваемой для решения проблемы бизнеса

Спецификация проекта – отображение требований в описания в терминах информационных технологий

Описание реализации: спецификация проекта преобразуется в конкретные аппаратные и программные компоненты.

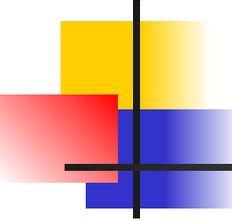
Разработка информационной системы

Уровни описания ИС



Классы моделей

Модели, входящие в полный фильтр (Entire Method)		Простой фильтр (Easy Filter)	Стандартный фильтр (Standard Filter)	Расширенный фильтр (Extended Standard Filter)	SAP-фильтр (SAP Filter)	Фильтр для модуля ARIS for MySAP.com	Фильтр для модуля ARIS Simulation and ARIS ABC	Фильтр для модуля Balanced Scorecard
Организационные модели. Уровень определения требований								
1	Организационная схема Organizational chart							
2	Календарь смен Shift calendar							
Организационные модели. Уровень спецификации проекта								
1	Топология сети Network topology							
Организационные модели. Уровень описания реализации								
1	Диаграмма сети Network diagram							
2	Технические ресурсы Technical resources							



Элементы моделей

Модель

ARIS: Содержит

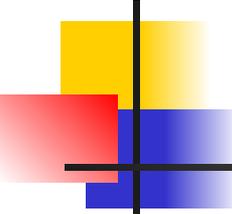
- Объекты
- Связи

Включает

- Внешние встроенные объекты
- Текст
- Геометрические фигуры

Каждый элемент (объект, связь) и сама модель имеет:

- Тип
- Имя
- Свойства (Атрибуты)
- Внешний вид



ПРОСТОЙ МЕТОДОЛОГИЧЕСКИЙ ФИЛЬТР. ОБЗОР МОДЕЛЕЙ

ОРГАНИЗАЦИОННЫЕ МОДЕЛИ

- *Организационная схема — Organizational chart*

ФУНКЦИОНАЛЬНЫЕ МОДЕЛИ

- *Дерево функций — Function Tree*

МОДЕЛИ ДАННЫХ

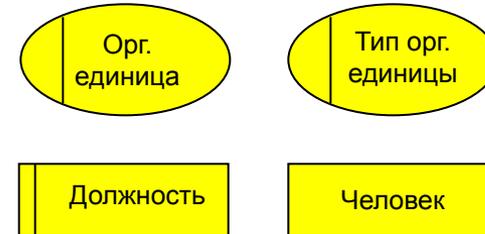
- *Модель технических терминов — Technical Term Models*

МОДЕЛИ ПРОЦЕССОВ/ УПРАВЛЕНИЯ

- *Событийная цепочка процесса — Extended event driven process chain (eEPC)*
- *Диаграмма окружения функции — Function allocation diagram*
- *Производственный и офисный процессы — Industrial and Office process*
- *Диаграмма цепочек добавленного качества — Value-added chain diagram (VAD)*

Организационная схема

Основные типы объектов этой модели:

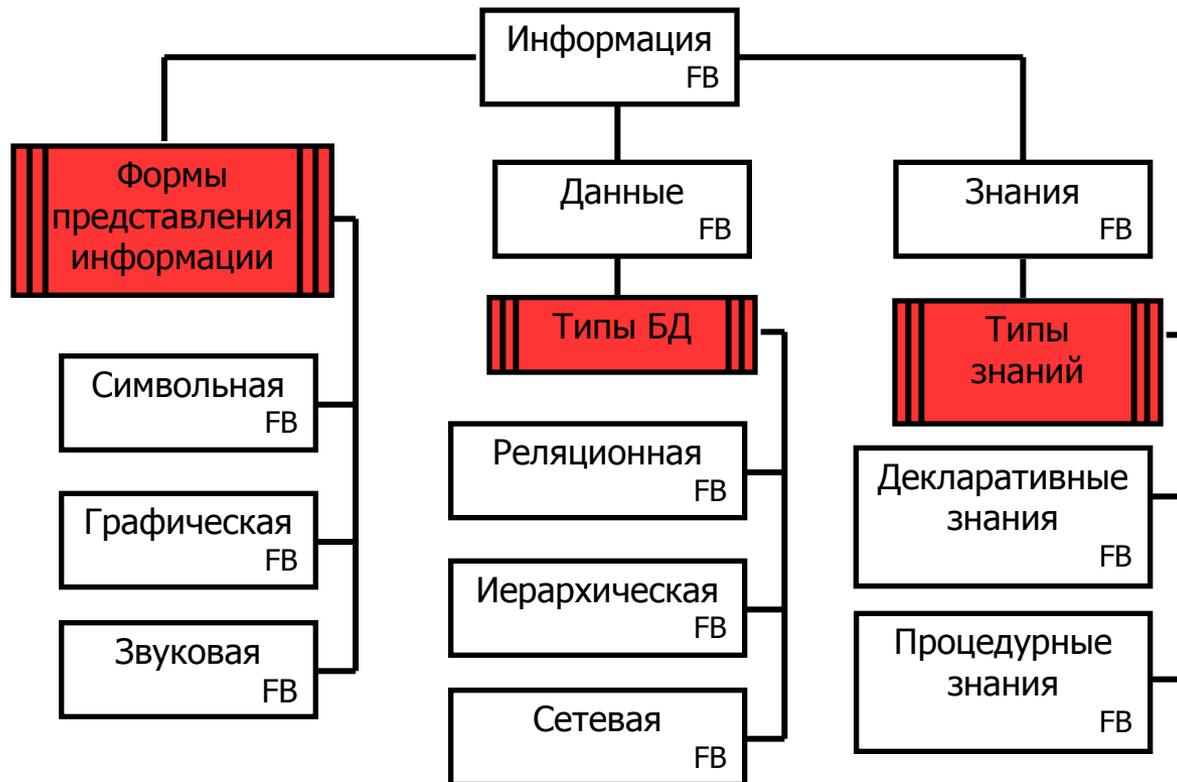


Типы отношений:

Имеет в подчинении, Имеет в своем составе, Взаимодействует с, ...

Модель строится иерархически — от верхнего уровня структуры к нижнему. Низшим уровнем является описание подразделений на уровне должностей — штатных единиц, занимаемых конкретными сотрудниками.

Модель технических терминов



Эта модель отображает многочисленные термины, определяющие различные объекты в организациях. Термины могут быть взаимосвязаны и иерархически упорядочены.

Типы объектов модели:

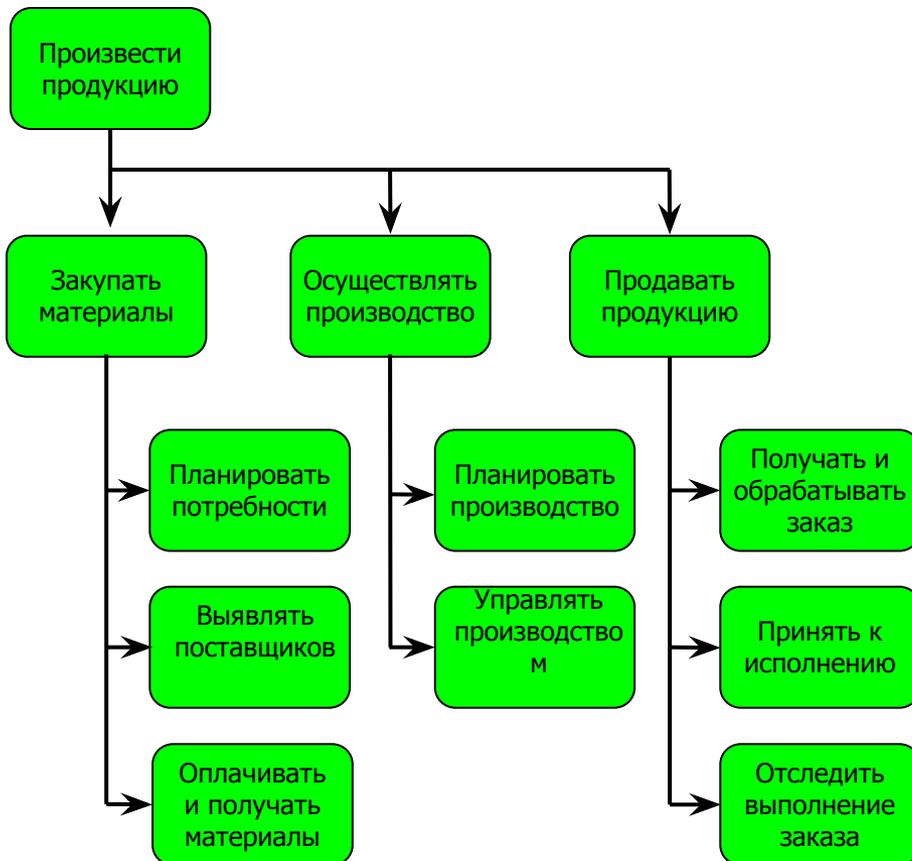
Технический термин

кластер

Типы отношений:

Содержит, Отображает
Является, Классифицирует
и др.

Дерево функций



Используется только один тип объекта — *функция* (работа, действие, этап в рамках процесса).

Типы отношений:

Подчиняется по процессу,

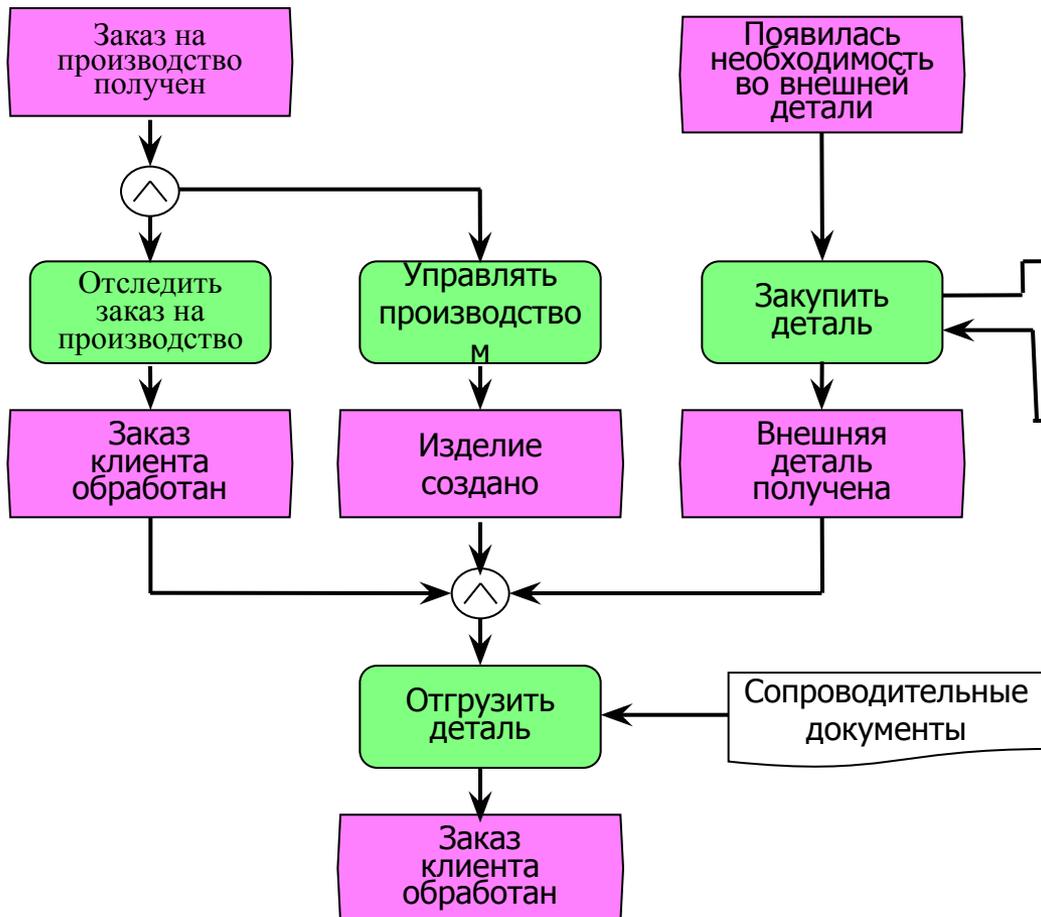
Подчиняется по объекту,

Подчиняется по способу выполнения.

На верхнем уровне функции представляют собой бизнес-процессы. Детализация функций образует иерархическую структуру.

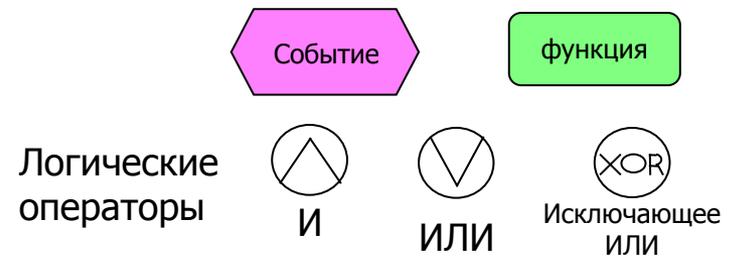
Самый нижний уровень представляют *базовые функции* (которые уже не могут быть разделены на составные элементы).

Событийная цепочка процесса



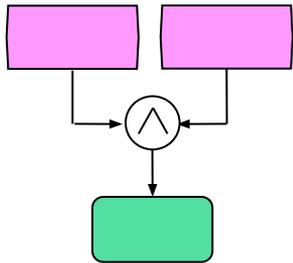
К моделям процессов/управления относится **Диаграмма eEPC** (extended Event driven Process Chain)

Основные типы объектов:

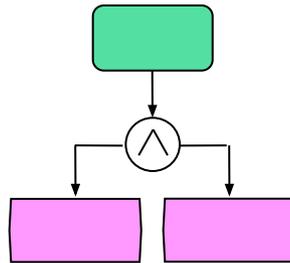


Элементы диаграммы eEPC

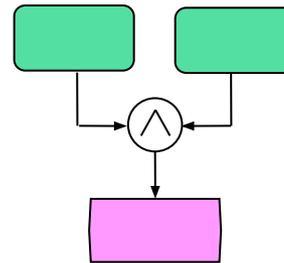
- ⇒ **Функция** – некоторое (шаг процесса). С функцией могут быть связаны: исполнители, входные и выходные документы, программное обеспечение и т.д.
- ⇒ **Событие** - какое-либо завершенное состояние объекта, которое влияет на дальнейший ход процесса. С одной стороны события являются стимулом к выполнению функций, с другой – их результатом.
- ⇒ Логические операторы (И, ИЛИ, XOR) показывают разветвления в потоке процесса. Примеры:



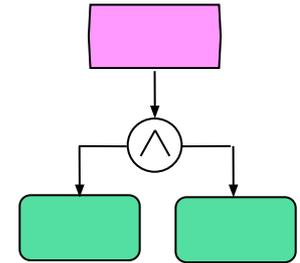
функция является результатом наступления нескольких событий



функция инициируют наступление нескольких событий

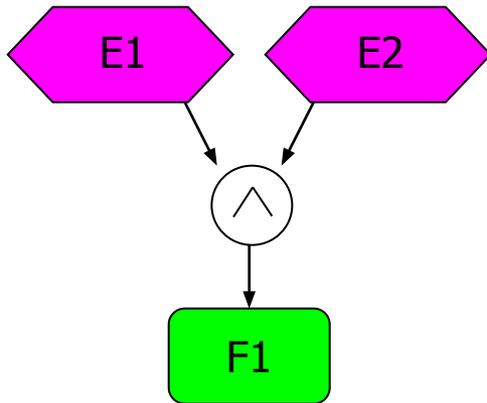


событие является результатом выполнения нескольких функций

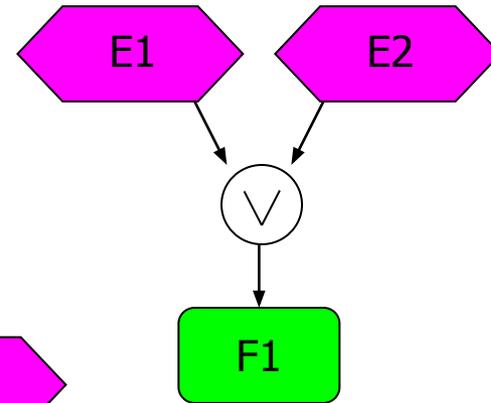


событие инициирует выполнение нескольких функций

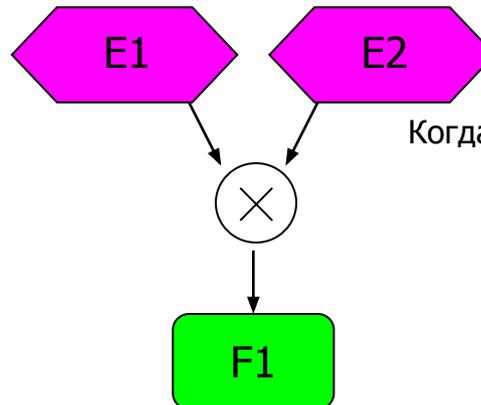
Логические операторы



Когда события E1 **И** E2 произошли, выполняется Функция F1

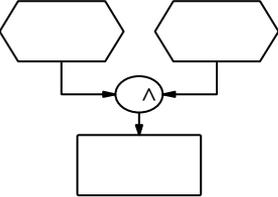
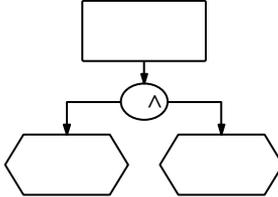
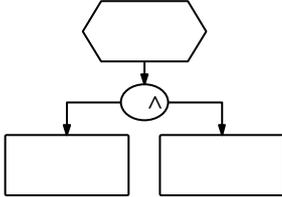
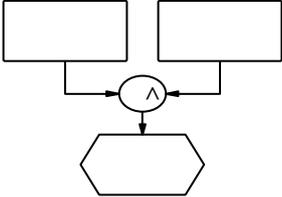
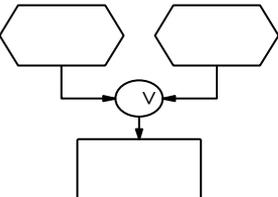
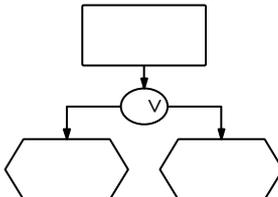
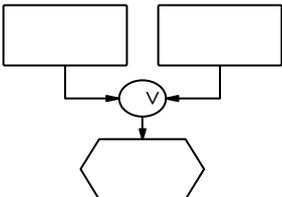
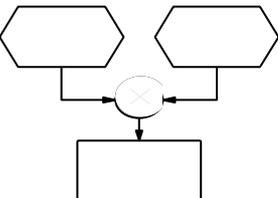
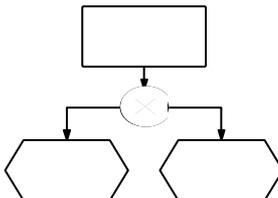
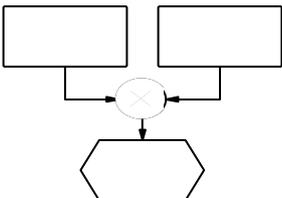


Когда произошли события E1 **ИЛИ** E2, выполняется Функция F1

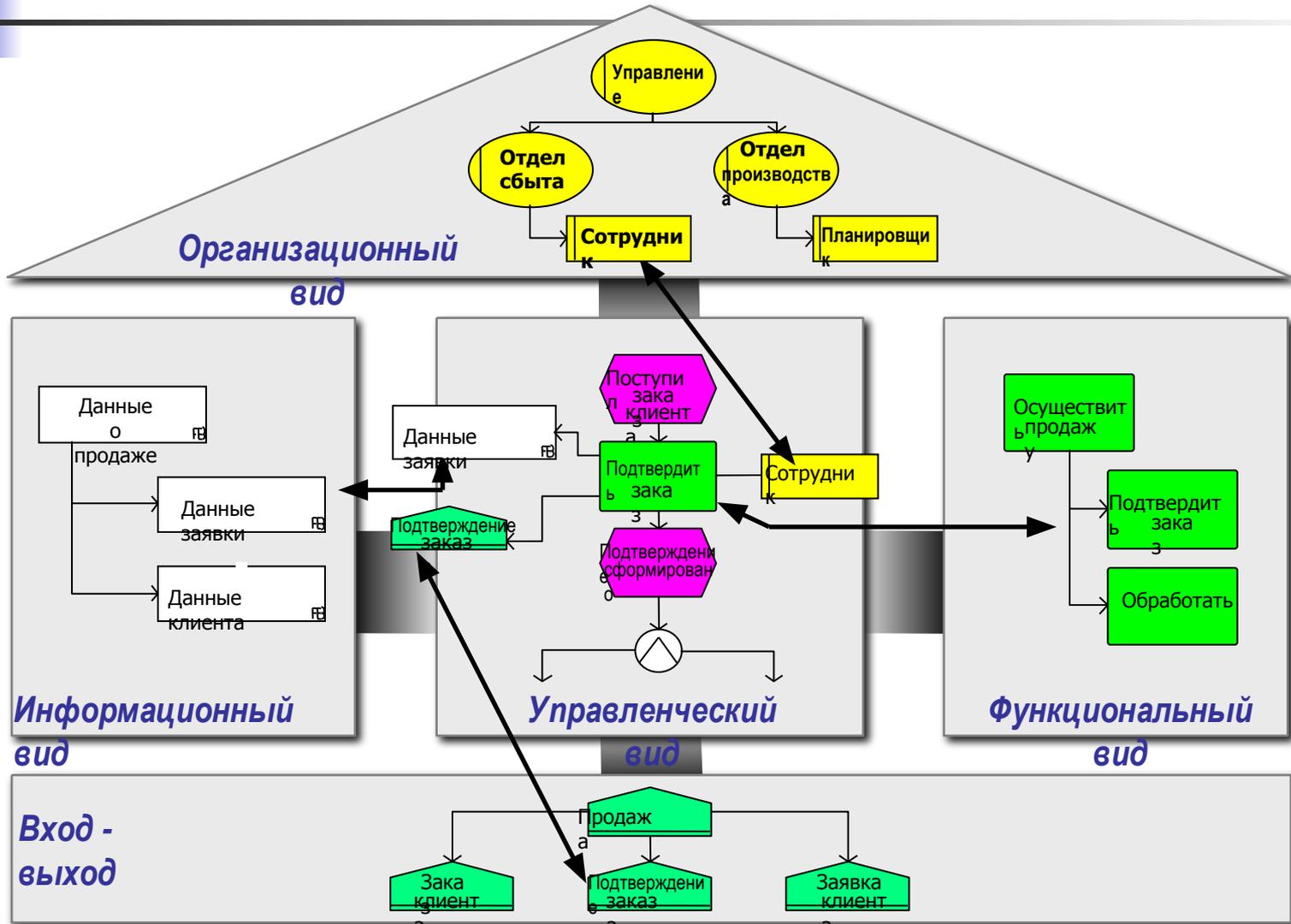


Когда произошло **ЛИБО** Событие E1 **ЛИБО** Событие E2, выполняется Функция F1

Правила

Соединение	Связывание событий		Связывание функций	
	инициирующие события	иницируемые события	инициирующие события	иницируемые события
 И				
 ИЛИ				
 XOR (исключающее ИЛИ)				

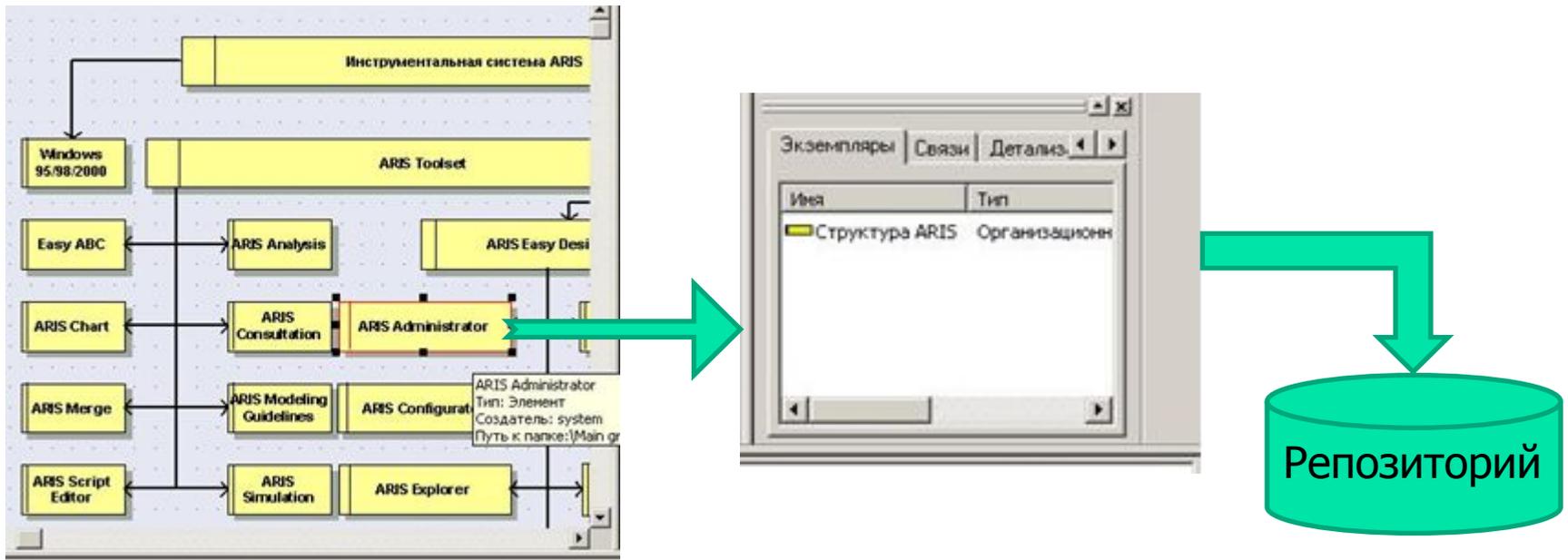
Интеграция моделей



Интеграция моделей

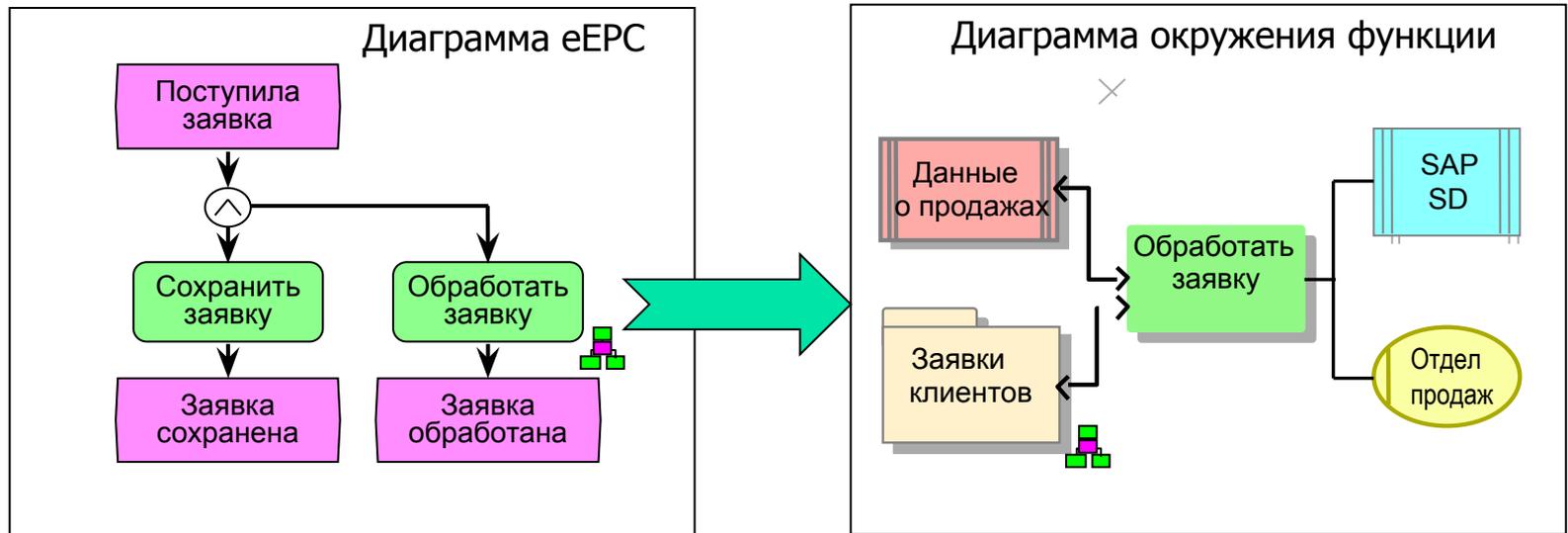
Реализуется благодаря хранению объектов в едином *репозитории* (специальной базе данных).

При создании нового объекта в репозитории появляется отдельная запись, задающая описание объекта. Объект можно скопировать из одной модели и вставить в другую с помощью команд Copy/Paste.



Детализация моделей

Для объектов текущей модели можно задавать **ссылки на другие модели**, являющиеся подробным описанием этого объекта.

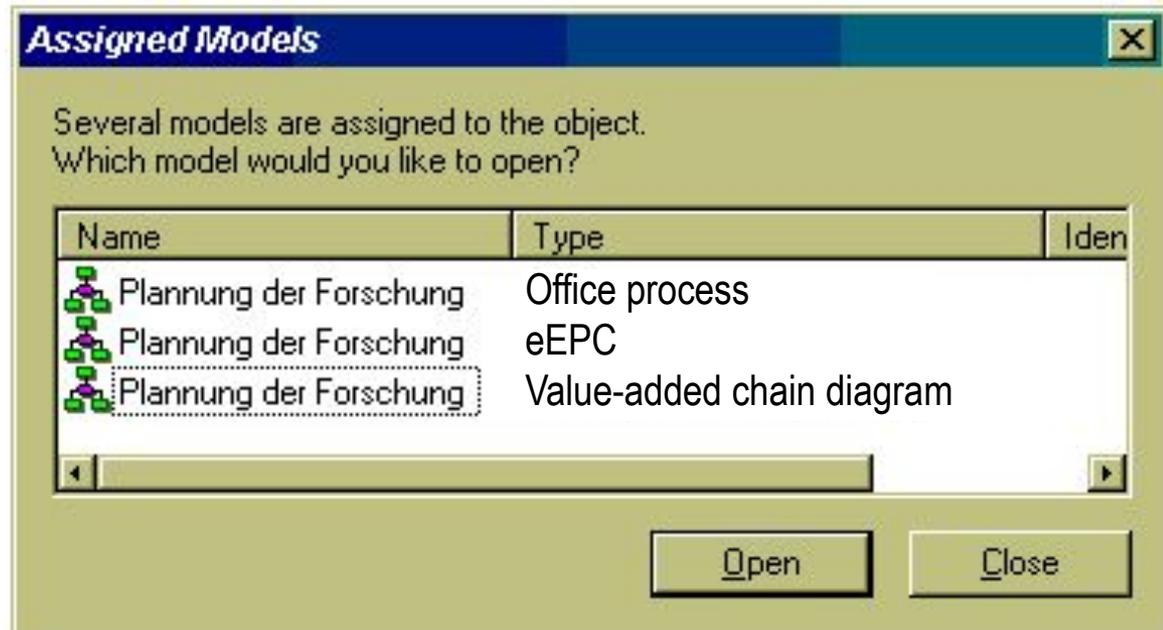
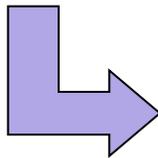


Механизм детализации позволяет избегать перегрузки моделей информацией, делая их более наглядными.

Детализация моделей

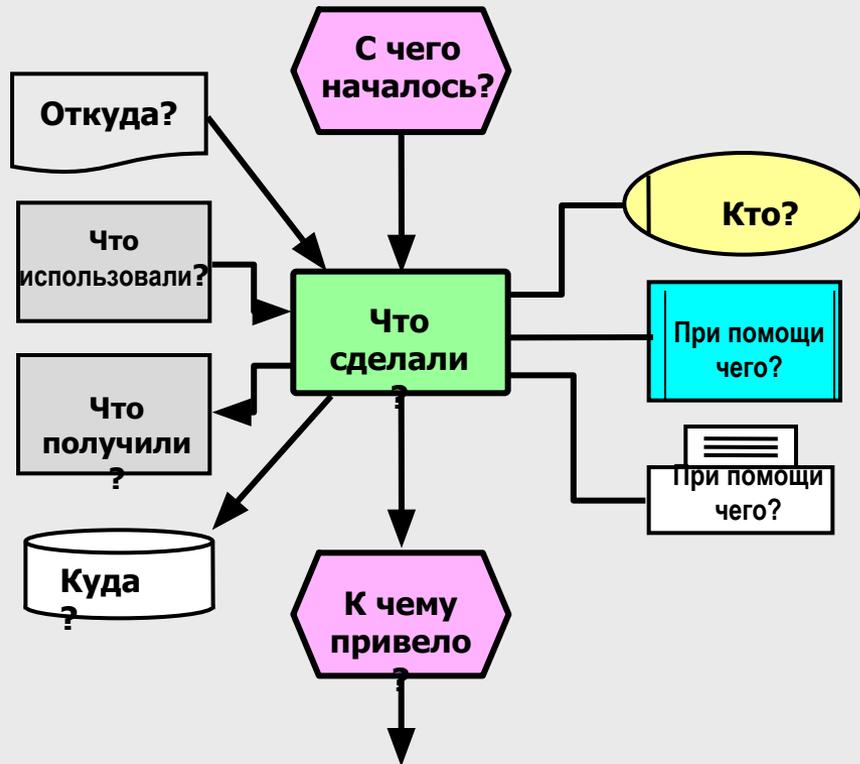
Типы детализации, разрешенные к использованию, зависят от типа объекта

Planung der Forschung

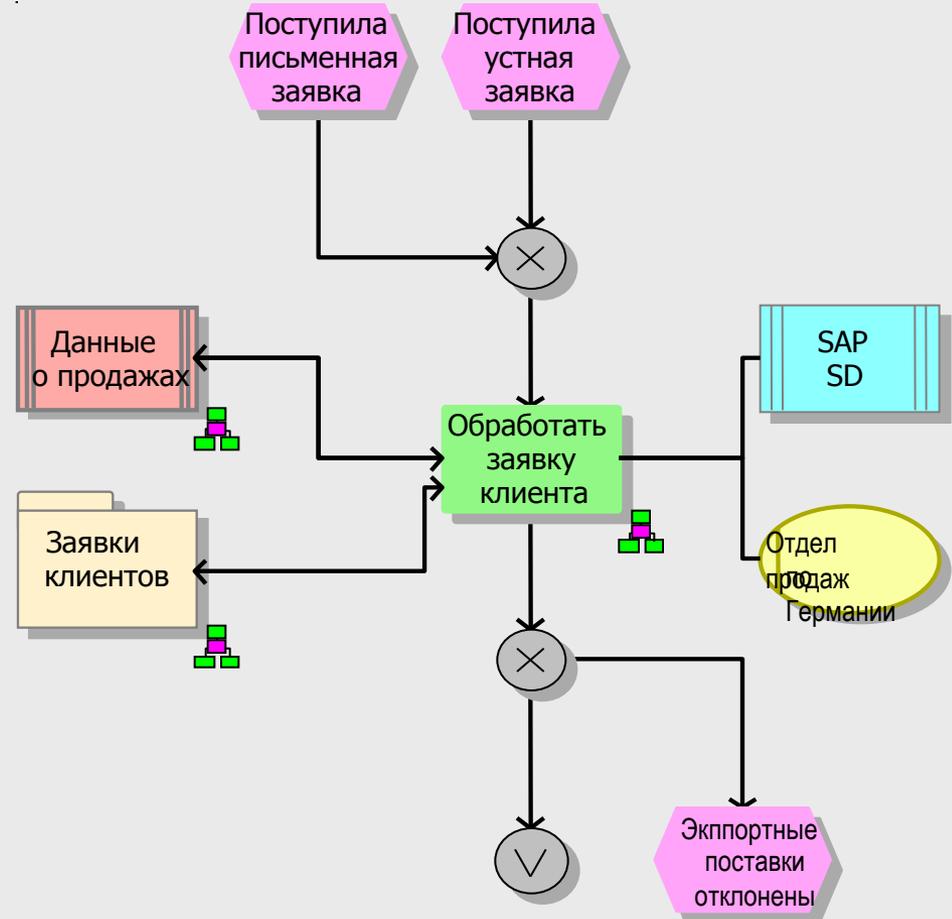


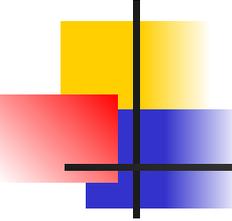
Детализация функции

Отображение



Пример





CASE-средства

Традиционно инструментальные средства для проектирования информационных систем назывались CASE-средствами

CASE (Computer Aided Software Engineering – компьютерная поддержка проектирования программного обеспечения) - это программно-технические средства для автоматизации разработки информационных систем.

В настоящее время **CASE** все чаще расшифровывают как:

Computer Aided **System** Engineering - компьютерная поддержка проектирования **систем** (бизнес-систем, информационных систем, технических комплексов и др.)

Классификация CASE-средств по уровням проектирования ИС

Жизненный цикл создания ИС:

CASE-средства:

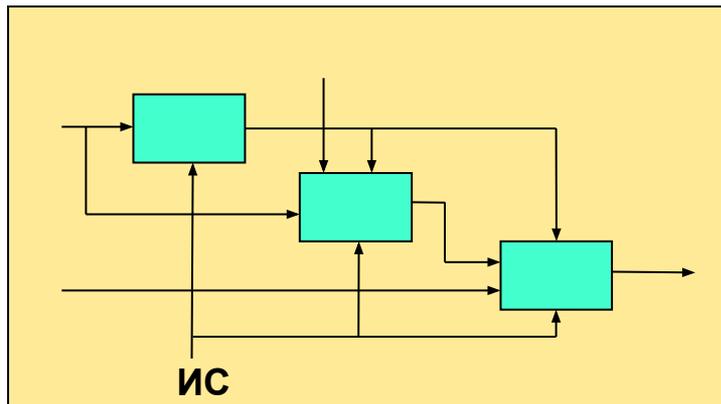


CASE-средства

Средства анализа предметной области (верхнего уровня).

Предназначены для описания **ЧТО** должна делать разрабатываемая информационная система, а **не как**.

Модель описывает бизнес-процессы (до автоматизации и после автоматизации), позволяет выделить функциональные требования к ИС.

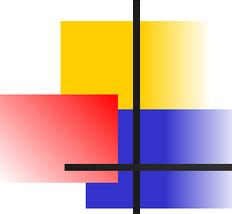


Используемые методологии:

IDEF0, DFD, IDEF3 и др.

Примеры CASE-средств:

Design/IDEF, BPwin, CASE Аналитик



CASE-средства

Средства анализа и проектирования (среднего уровня).

Предназначены и для описания **ЧТО** должна делать ИС (детально), и **как**.

Модель описывает: варианты использования ИС, прецеденты ИС, классы объектов, реализующих прецеденты, структуры данных (схемы баз данных), архитектуру ИС, спецификации компонент, интерфейсы.

Используемые методологии : UML, ERD, IDEF1X, и др.

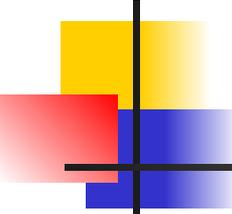
Примеры CASE-средств : Rational Rose, Erwin, Designer/2000

Соотношение анализа и проектирования



- В центре внимания - проблема
- отсутствие деталей
- описание структуры и поведения системы
- реализация функциональных требований в архитектуре системы
- модель относительно небольшого размера

- В центре внимания - решение
- детали - операции и атрибуты
- учет аспектов производительности
- приближение к реальному коду
- реализация нефункциональных требований
- крупная модель



CASE-средства

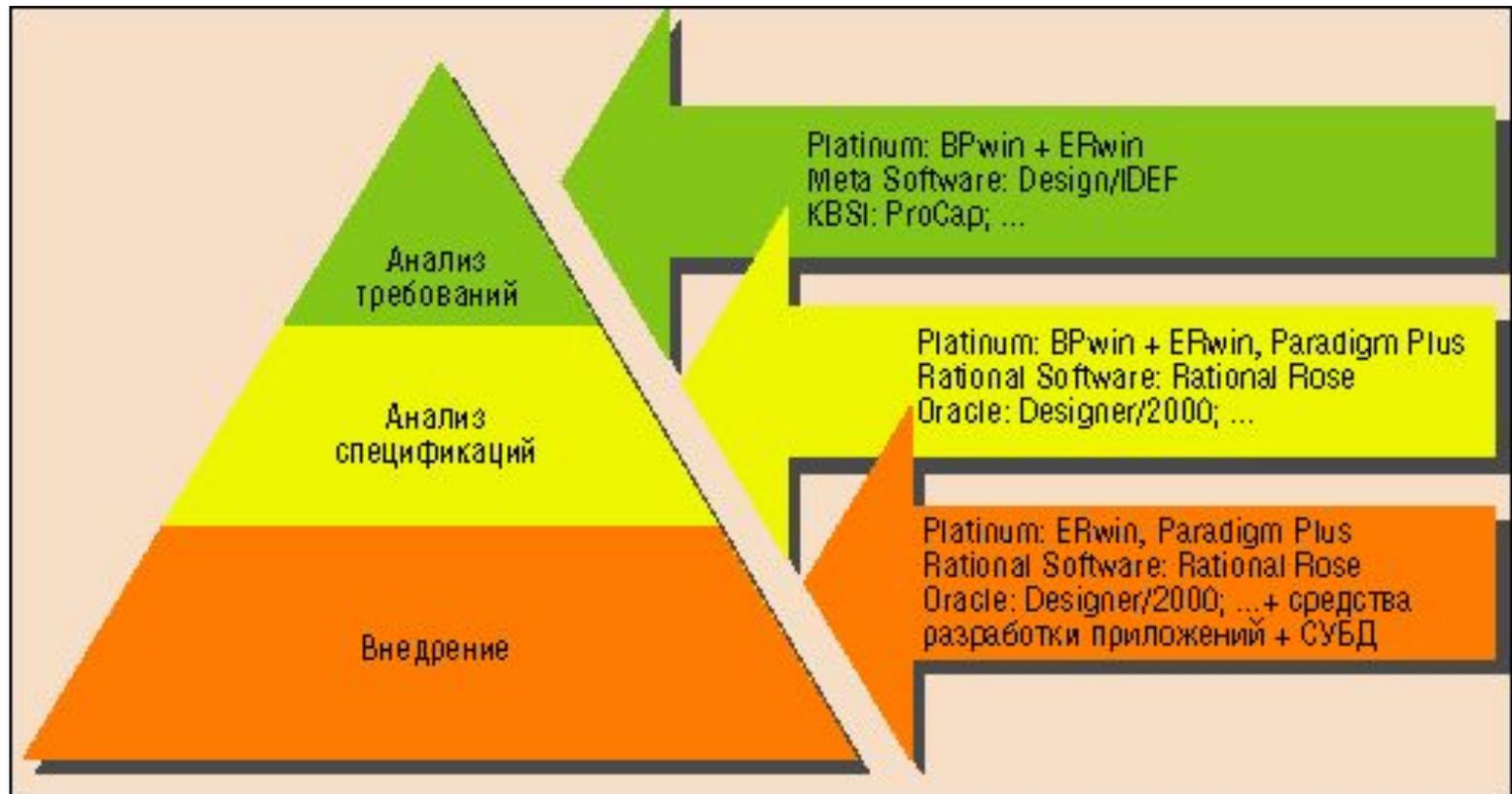
Средства разработки приложений (нижнего уровня).

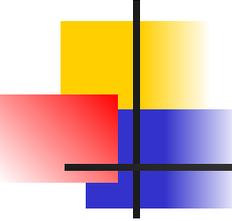
Предназначены для **реализации** информационной системы, построения модели реализации, генерации программного кода на различных языках программирования (C++, Object Pascal, Java, Visual Basic)

Примеры CASE-средств: Rational Rose, Oracle Designer/2000 + RAD-средства + СУБД, Paradigm Plus.

Существует еще один класс – ***вспомогательные*** CASE-средства. К ним относятся: **средства управления проектом** (для разработки календарных графиков выполнения проекта, распределения ресурсов), **средства тестирования**, **средства документирования** и др.

CASE-средства





Инструменты моделирования бизнеса

Инструментальные средства для **моделирования бизнеса** (в терминах CASE-средств это средства верхнего и среднего уровня) подразделяются в зависимости от **количества поддерживаемых методологий и методов** на:

- **локальные**, поддерживающие один-два типа моделей и методов (Design/IDEF, ProCap, S-Designor, "CASE. Аналитик");
- **малые интегрированные** средства моделирования, поддерживающие несколько типов моделей и методов (ERwin, BPwin);
- **средние интегрированные** средства моделирования, поддерживающие от 4 до 10-15 типов моделей и методов (Rational Rose, Paradigm Plus, Oracle Designer/2000);
- **крупные интегрированные** средства моделирования, поддерживающие более 15 типов моделей и методов (ARIS Toolset).

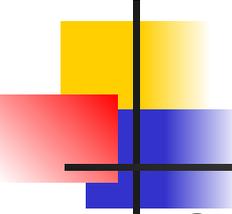
Инструменты моделирования бизнеса

По **типу методологий** моделирования выделяют:

- средства **структурного** моделирования (BPwin);
- средства **объектно-ориентированного** моделирования (Rational Rose);
- средства **комплексного** моделирования (ARIS Toolset).

Аспект	BPWin	Rational Rose	ARIS Toolset
Функции бизнеса	IDEF0, DFD	Use case	Дерево функций, дерево целей, ...
Логика бизнес-процессов	IDEF3	Диаграмма деятельности	Событийная цепочка EPC, офисный процесс, ...
Исполнители процессов, другие ресурсы	IDEF0	Д-ма последовательности, кооперации	Событийная цепочка д-ма окружения функции, организационная схема ...
Данные, информация	Связь с Erwin	Диаграмма классов	М. технических терминов, ER-модель,

Возможности инструментальных средств



Основные **функциональные возможности** средств моделирования:

визуальное моделирование - формирование диаграмм в интерактивном режиме с использованием визуальных средств;

проверка моделей – проверка соблюдения синтаксических и семантических правил, проверка согласованности моделей, обеспечение целостности проектных данных;

анализ характеристик бизнес-процесса – функционально-стоимостной анализ, имитационное моделирование, другие методы анализа;

документирование – оформление проектной документации, генерация технологических и рабочих инструкций;

ведение библиотеки типовых моделей – возможность сохранять типовые образцы в библиотеке и использовать их при построении новых моделей;

возможность групповой работы – возможность слияния моделей, созданных разными разработчиками.

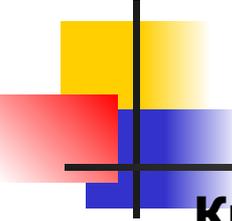
Возможности инструментальных средств

Возможности	BPWin	Rational Rose	ARIS Toolset
Построение диаграмм	+	+	+
Функционально-стоимостной анализ	+	-	+
Имитационное моделирование	+/-	-	+
Оформление проектной документации, инструкций	+/-	+	+
Ведение библиотеки типовых моделей	+/-	+/-	+
Контроль целостности проектных данных	+/-	+	+
Возможность групповой работы	+	+	+

«+» – да, «-» – нет, «+/-» – частичная реализация, требующая доработки иным инструментальным средством

Проблемы выбора средства моделирования бизнес-процессов

- Проблема выбора стандартной методики (компании продают инструменты, а не методы; обучают системам, а не методикам)
- Соответствие задачам проекта (трудно определить заранее, насколько выбранная методика подходит для конкретного проекта)
- Сложность регламентации использования инструментальных средств (много возможностей, чем гибче и мощнее система, тем труднее ее корректно использовать)
- Отсутствие стандартов и стандартной терминологии
- Отсутствие информации, возможностей для реального обмена опытом ведения проектов



Выбор инструментального средства

Критерии выбора инструментального средства:

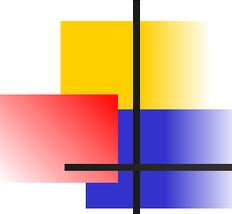
Функциональные возможности и методология.

Должны соответствовать потребностям. Не следует с «переплатой» приобретать инструмент с избыточными возможностями.

Если предполагается построение информационных систем поддержки бизнес-процессов, то больше подходят Oracle Designer, Power Designer и Rational Rose.

Если нужно средство моделирования бизнеса, небольшого по масштабу, не слишком сложного, то лучше подходит Bpwin.

Если выполняется крупный проект по реинжинирингу бизнеса, требующий тщательного анализа предполагаемых вариантов реконструкции бизнеса (обсчета временных, стоимостных и других количественных показателей) и последующей их оптимизации, то лучше подходит ARIS Toolset.



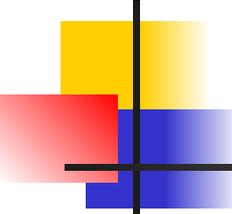
Выбор инструментального средства

Уровень подготовки персонала.

Как правило, чем шире возможности инструментальной среды, тем выше выдвигаются требования к пользователям.

Для работы со сложными в освоении инструментальными средствами (например, Rational Rose, ARIS), требуется обучить персонал или нанять уже подготовленных квалифицированных специалистов. Это требует дополнительных затрат и удлиняет сроки разработки.

Поэтому лучше использовать сравнительно простые в освоении средства (Design/IDEF, Vpwin), если их возможности соответствуют потребностям.



Выбор инструментального средства

Стоимость инструментальных средств

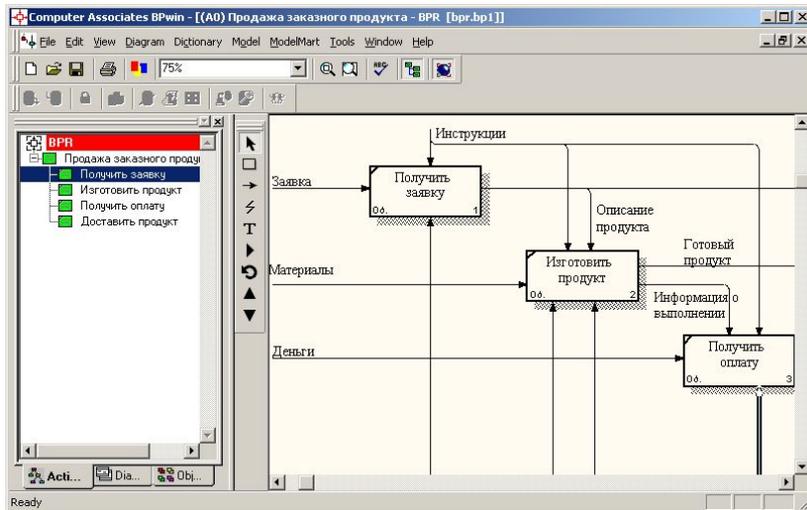
Цена инструментариев различных производителей может существенно различаться. При этом бюджетные затраты предприятия будут определяться не только начальными инвестициями на его приобретение, но и последующими затратами на техническую поддержку, обучение персонала, возможную модернизацию программно-аппаратной платформы, потенциальный «апгрейд» и т.д.

Характеристики программно-аппаратной платформы.

При выборе инструментальной среды необходимо оценить:

- требования к технической платформе;
- требования к общесистемному программному обеспечению;
- требования к телекоммуникационному обеспечению;
- возможности по обеспечению информационной безопасности;
- количество мест установки пользовательских приложений.

Инструментальное средство BPwin



Поддерживает **3 методологии**:

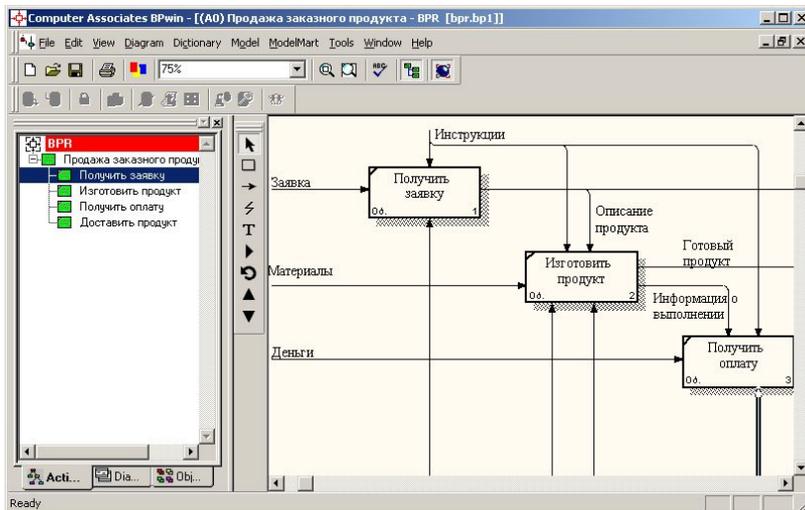
- **IDEFO**,
- **DFD** (поток данных)
- **IDEF3** (поток работ).

Имеет простой и понятный интерфейс.

Предоставляет **два инструмента** для оценки бизнес-процессов:

- **функционально-стоимостной анализ (ABC)**,
- **оценка свойств, определяемых пользователем (UDP)**.

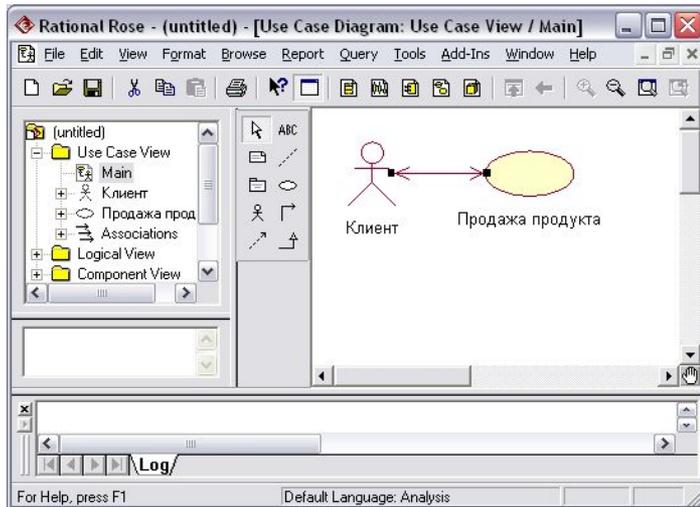
Инструментальное средство BPwin



- Осуществляет **проверку** целостности и согласованности модели, позволяет сливать и расщеплять модели.
- Имеет широкий набор средств **документирования** моделей, содержит собственный генератор отчетов.

- Интегрирован с ERwin (для моделирования БД), Paradigm Plus (для моделирования компонентов ПО) и др.
- Интегрирован со средством имитационного моделирования Arena.

CASE-средство Rational Rose

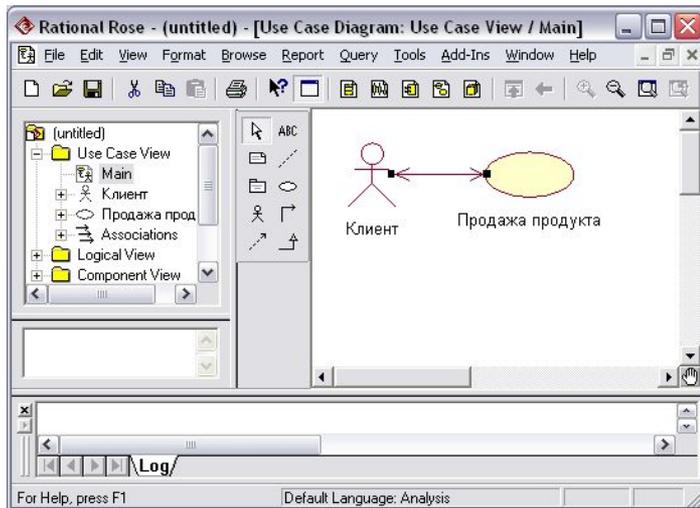


Полностью поддерживает компонентно-ориентированный процесс создания ИС.

Содержит **все диаграммы UML** (8), начиная от диаграмм вариантов использования и заканчивая диаграммами реализации.

Все **модели взаимосвязаны**: бизнес-модель, функциональная модель, модель проектирования, модель базы данных, модель компонентов и модель физического развертывания системы.

CASE-средство Rational Rose



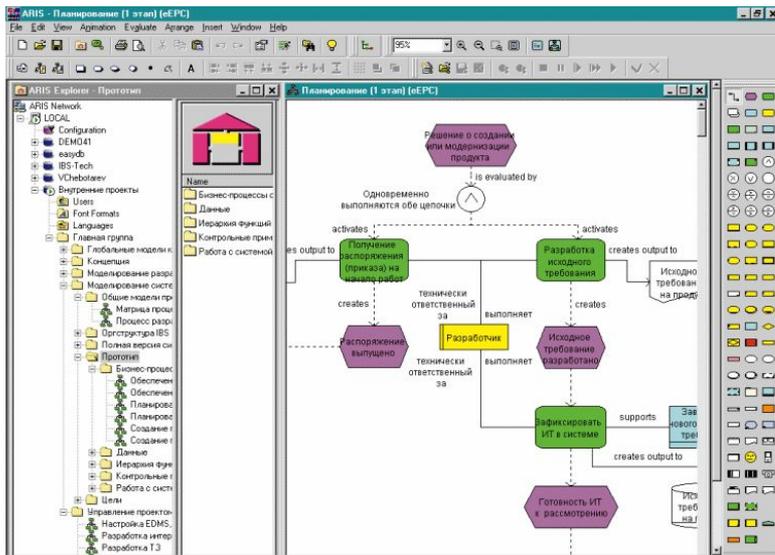
Есть возможность по созданию **шаблонов архитектурных решений (pattern)**, позволяющих использовать опыт, накопленный в предыдущих проектах.

Имеется возможность **генерации программного кода** (на языках C++, Java, Visual Basic, PowerBuilder и др.) на основе построенных моделей.

Интеграция с:

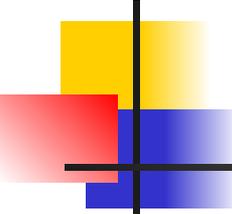
- Rational RequisitePro (анализ требований),
- Rational ClearCase (контроль версий),
- Rational SoDA (документирование, отчеты).

Программный пакет ARIS



Представляет собой **комплекс средств** анализа и моделирования деятельности предприятия. Позволяет отразить **различные взгляды** на бизнес-систему, которую мы можем оценить и рассмотреть с разных сторон, используя как собственные методы ARIS, так и различные известные методы (в частности ER и UML).

В системе ARIS есть **внутренняя база данных**, которая позволяет проверять модель на непротиворечивость, целостность, проводить верификацию модели.



Программный пакет ARIS

ППП ARIS состоит из **комплекса взаимосвязанных модулей**:

- *ARIS Designer* — конструктор моделей;
- *ARIS Explorer* — проводник;
- *ARIS Report* — генератор отчетов о элементах ARIS;
- *ARIS Semantic Check* — инструмент для семантических проверок и др.

Помимо моделирования ARIS предусматривает целый **комплекс операций** над моделями:

- проверка корректности моделей;
- оптимизация моделей по различным критериям;
- анализ моделей, проводимый по различным методикам, например, функционально-стоимостной анализ, стратегическое планирование;
- сравнение моделей;
- обмен информацией с другими программными системами;
- непрерывное улучшение модели и др.