Методы визуального анализа и проектировангия систем

Диаграммы UML

Клевцов С.И. кафедра МПС

Диаграммы состояния

Диаграммы состояний (state machine diagrams) – это технология описания поведения системы.

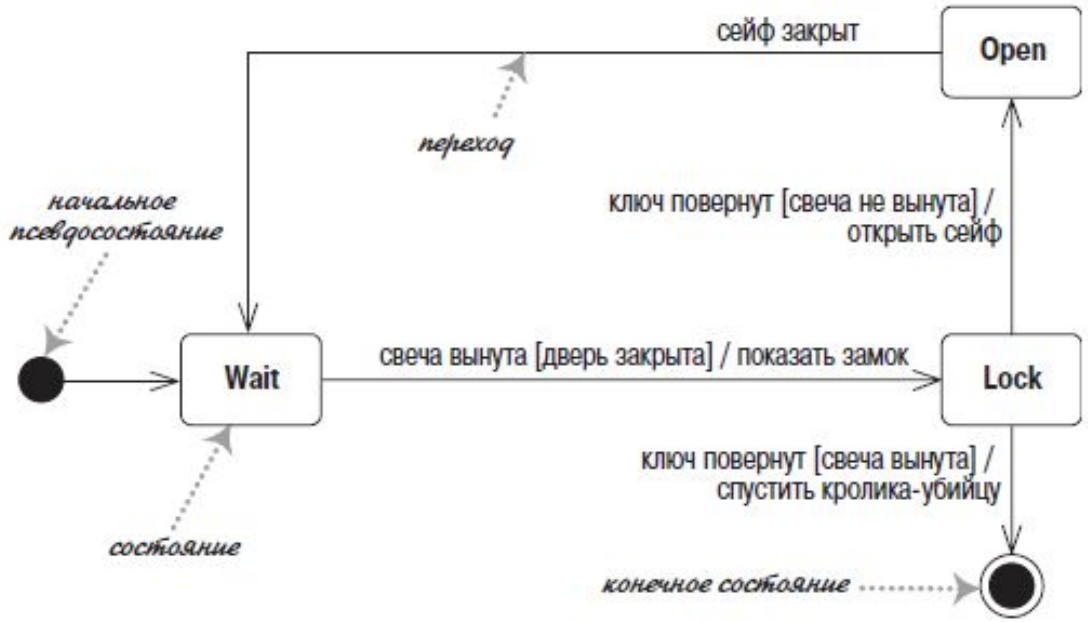
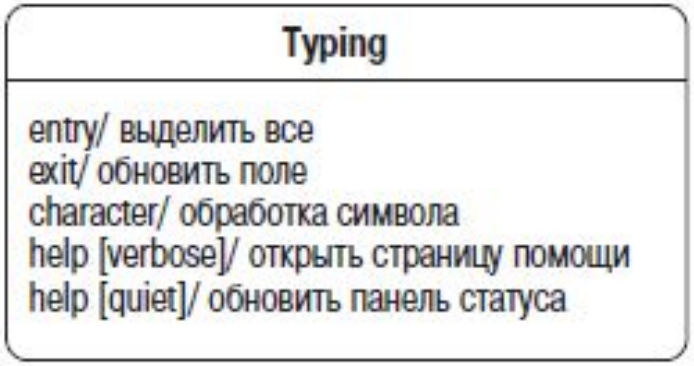


Диаграмма состояния начинается с состояния создаваемого объекта контроллера: состояния Wait (Ожидание). На диаграмме начало процесса обозначено с помощью начального псевдосостояния (initial pseudostate), которое не является состоянием. Контроллер может находиться в одном из трех состояний: Wait (Ожидание), Lock (Замок) , Open (Открыт).

Переход (transition) означает перемещение из одного состояния в другое. Каждый переход имеет свою метку, которая состоит из трех частей: Триггер-идентификатор [защита]/активность

Диаграммы состояния

Внутренние активности



Состояния могут реагировать на события без совершения перехода, используя внутренние активности (internal activities), и в этом случае событие, защита и активность размещаются внутри прямоугольника состояния.

Внутренняя активность подобна самопереходу (self-transition) – переходу, который возвращает в то же самое состояние.

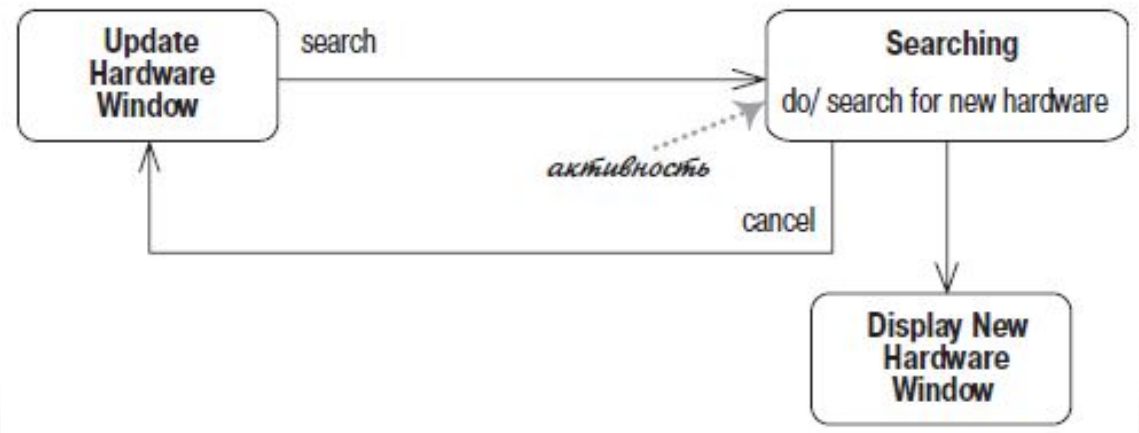
Имеются специальные активности: входная и выходная активности.
Входная активность выполняется всякий раз, когда вы входите в состояние; выходная активность – всякий раз, когда вы покидаете состояние.

Диаграммы состояния

Состояния активности

В состояниях, которые представлены ранее, объект молчит и ожидает следующего события, прежде чем что-нибудь сделать. Однако возможны состояния, в которых объект проявляет некоторую активность.

В состоянии активности (activity state) ведущаяся активность обозначается символом do/;
отсюда термин do activity (проявлять активность).

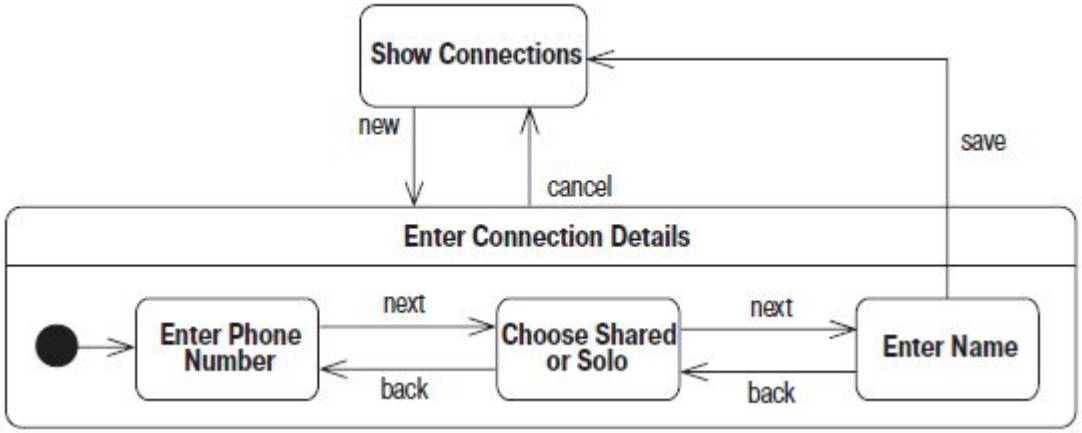


Диаграммы состояния

Суперсостояния

Часто бывает, что несколько состояний имеют общие переходы и внутренние активности.
В таких случаях можно их превратить в подсостояния (substates), а общее поведение перенести в суперсостояние (superstate)

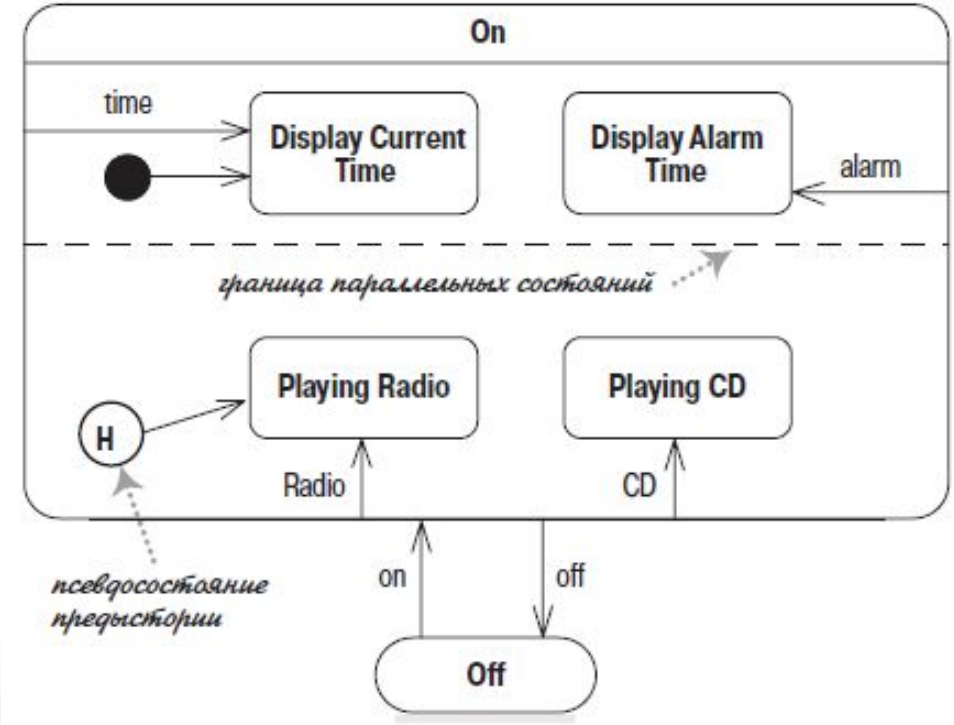
Суперсостояние с вложенными подсостояниями



Диаграммы состояния

Параллельные состояния

Состояния могут быть разбиты на несколько параллельных состояний, запускаемых одновременно.



Здесь также используется псевдосоотояние предыстории (history pseudostate).

Диаграммы состояния

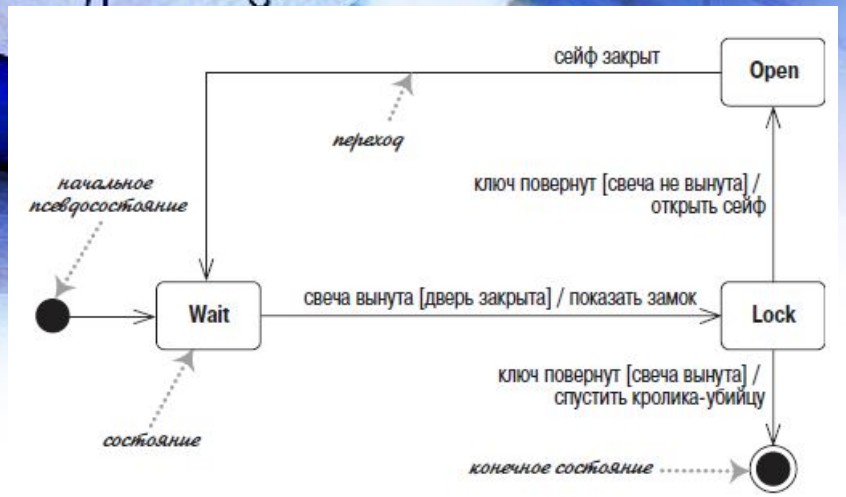
Реализация диаграмм состояний

Диаграмму состояний можно реализовать тремя основными способами:

- с помощью вложенного оператора switch или аналогичной конструкции на языке программирования
- паттерна State
- таблицы состояний.

Самый прямой подход в работе с диаграммами состояний – это, например, вложенный оператор switch на C# или аналогичная конструкция на другом языке

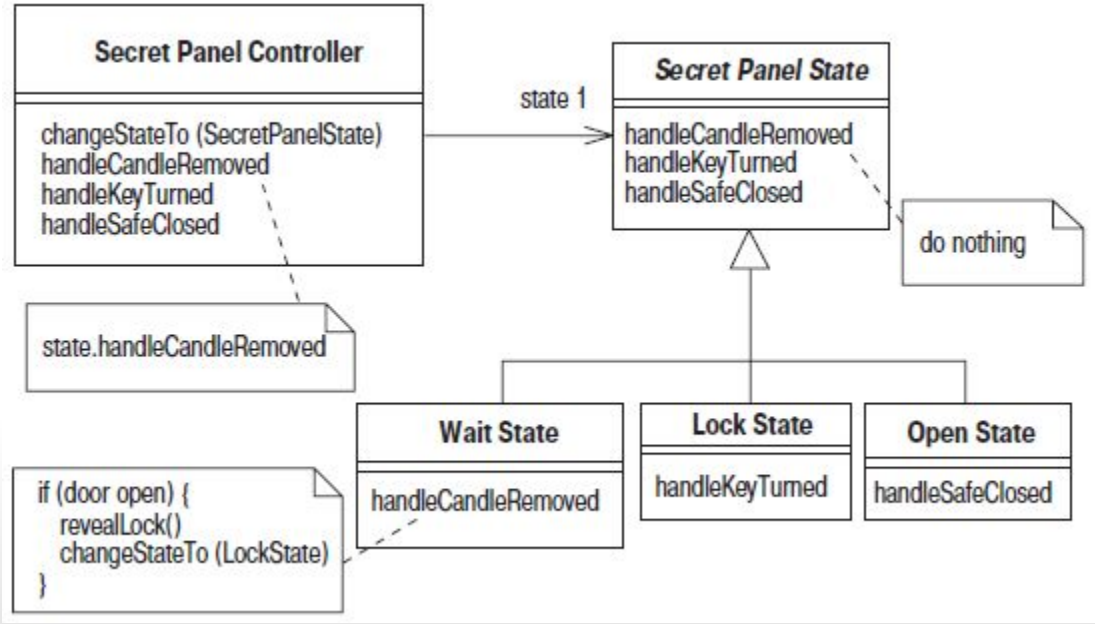
Диаграммы состояния



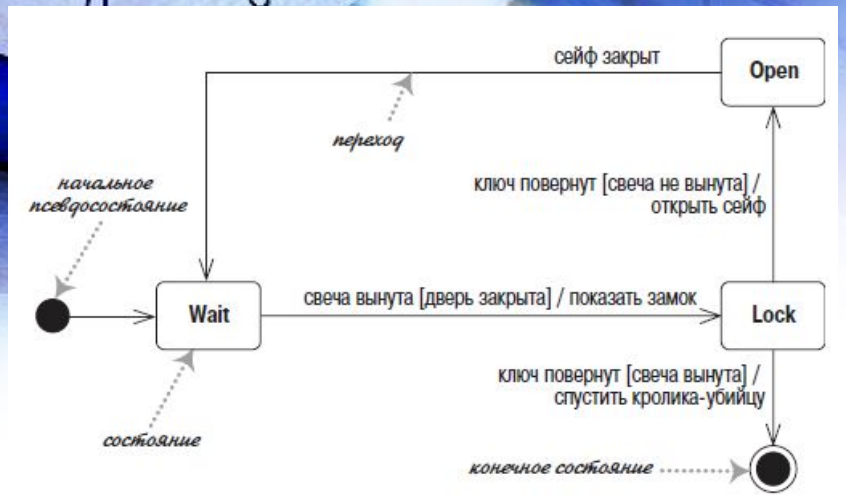
Реализация диаграмм состояний

Паттерн «Состояние» (State pattern) представляет иерархию классов состояний для обработки поведения состояний.
Каждое состояние на диаграмме имеет свой подкласс состояния.
Например, контроллер имеет методы для каждого события, которые просто перенаправляют к классу состояния.

Паттерн «Состояние», реализующий диаграмму



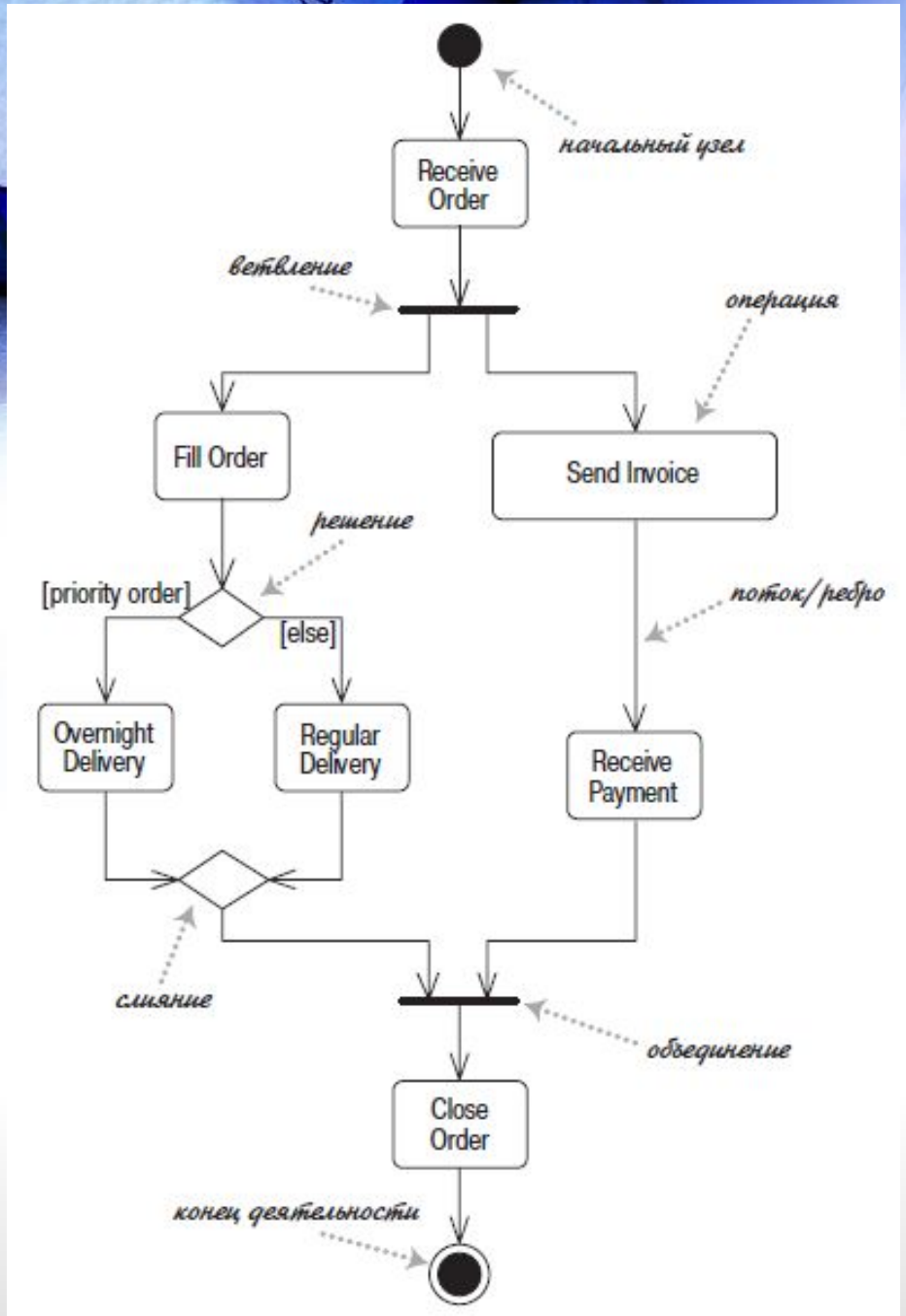
Диаграммы состояния



Реализация диаграмм состояний

Таблица состояний представляет диаграмму состояний в виде данных

Исходное состояние	Целевое состояние	Событие	Защита	Процедура
Wait	Lock	Candle removed (свеча удалена)	Door close дверь закрыта	Reveal lock (показать замок)
Lock	Open	Key turned (ключ повернут)	Candle in (свеча на месте)	Open safe (открыть сейф)
Lock	Final	Key turned (ключ повернут)	Candle out (свеча удалена)	Release killer rab-bit (освободить убийцу-кролика)
Open	Wait	Safe closed (сейф закрыт)		



Диаграммы деятельности

Диаграммы деятельности – это технология, позволяющая описывать логику процедур, бизнес-процессы и потоки работ.

Диаграмма деятельности позволяет выбирать порядок действий.

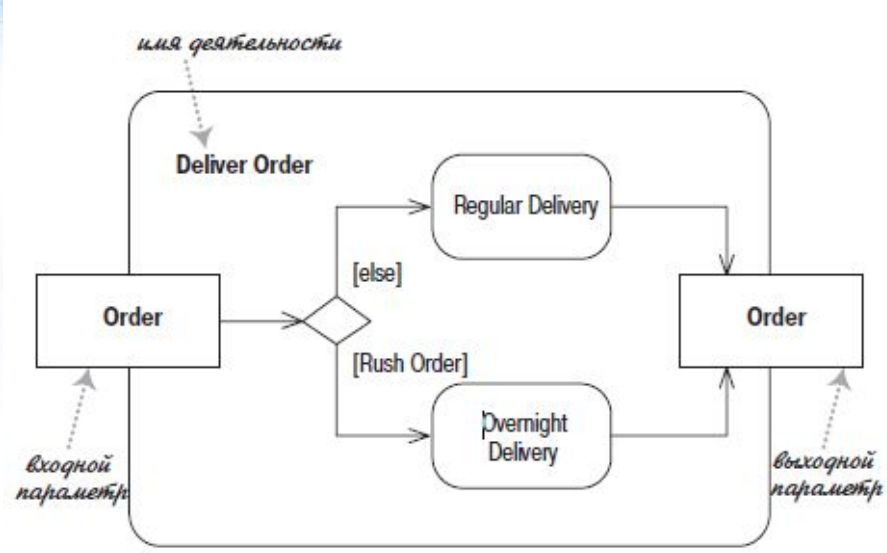
Другими словами, диаграмма только устанавливает правила обязательной последовательности действий, которым необходимо следовать.

Это важно для моделирования бизнес-процессов, поскольку эти процессы часто выполняются параллельно.

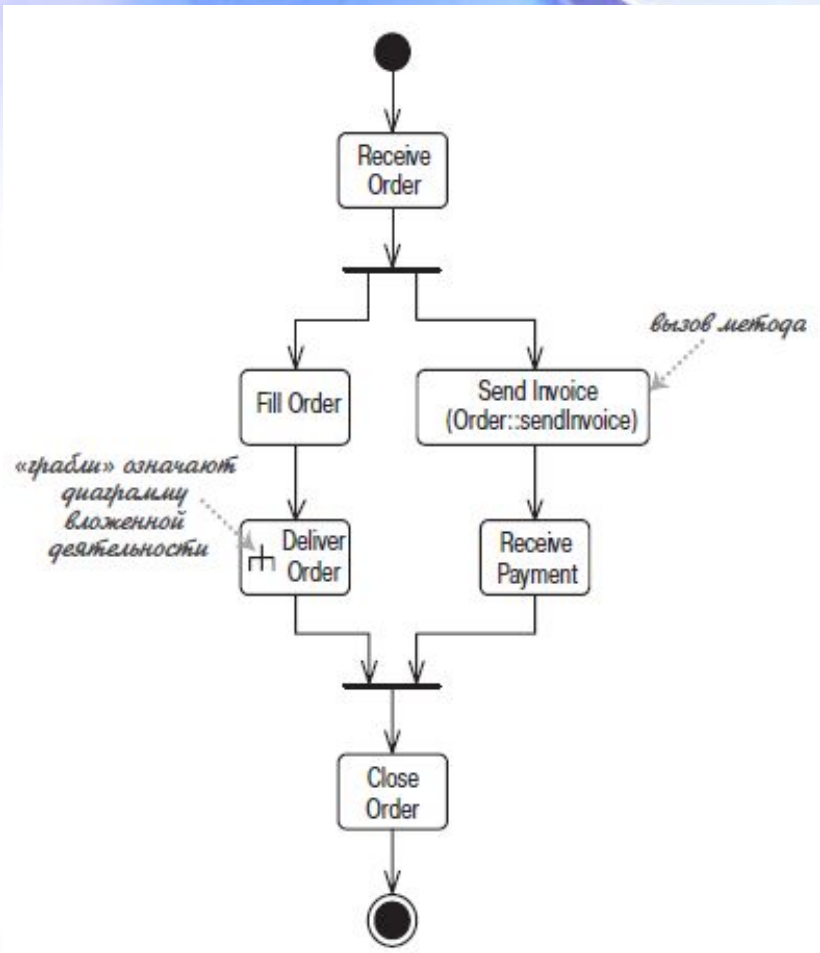
Такие диаграммы также полезны при разработке параллельных алгоритмов, в которых независимые потоки могут выполнять работу параллельно.

Диаграммы деятельности

Декомпозиция операции



Дополнительная диаграмма деятельности

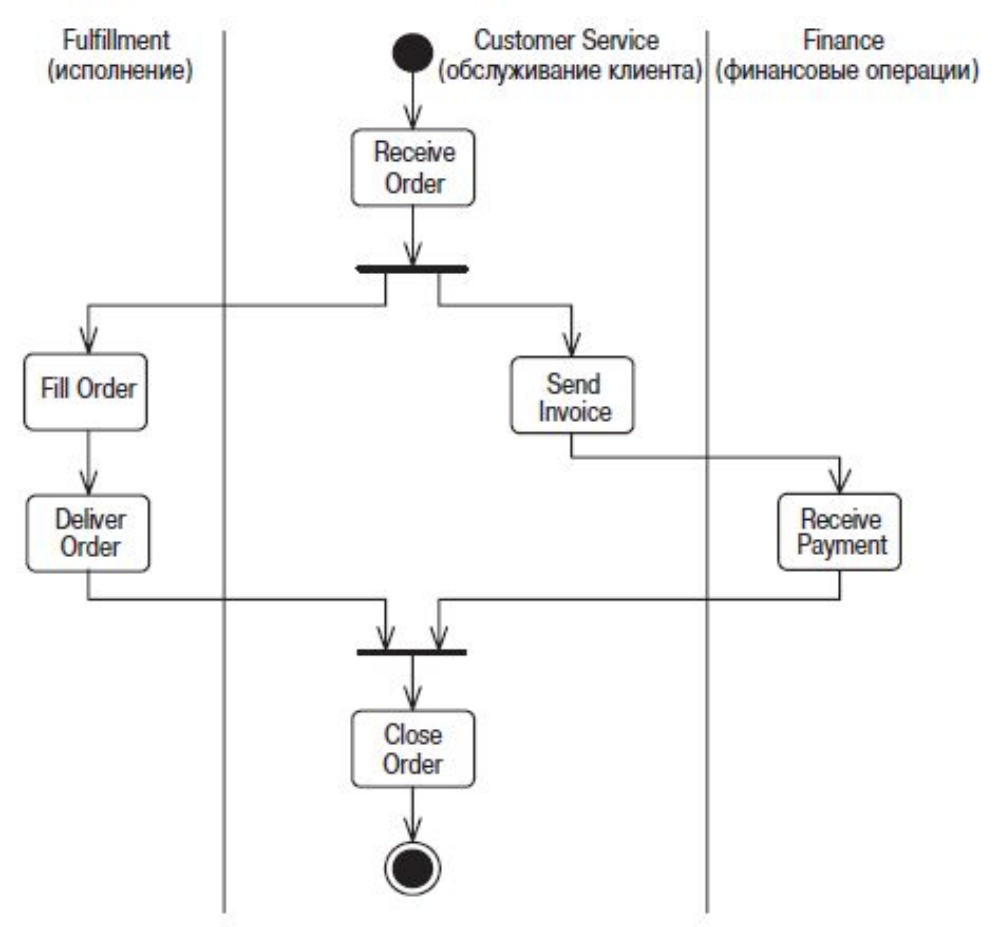


Деятельность из предыдущего слайда модифицирована для вызова деятельности из Дополнительной диаграммы

Диаграммы деятельности

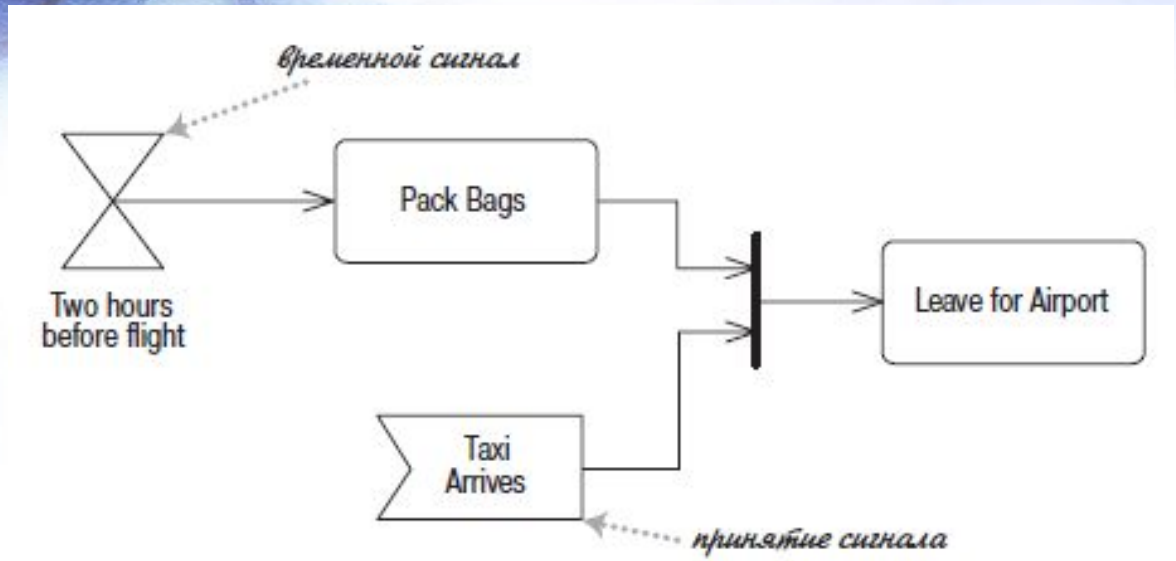
Разделы

Можно разбить диаграмму деятельности на разделы (partitions), чтобы показать, кто что делает, то есть какие операции выполняет тот или иной класс или, для представленной диаграммы, подразделение предприятия



Диаграммы деятельности

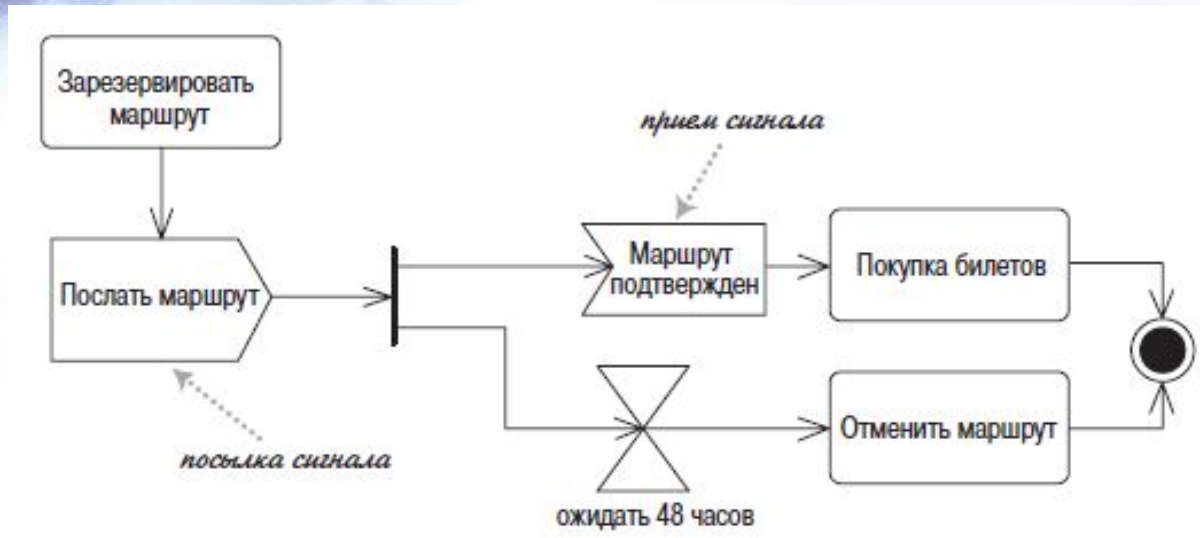
Сигналы



Временной сигнал (time signal) приходит по прошествии времени. Такие сигналы могут означать конец месяца в отчетном периоде или приходить каждую секунду в контроллере реального времени

Диаграммы деятельности

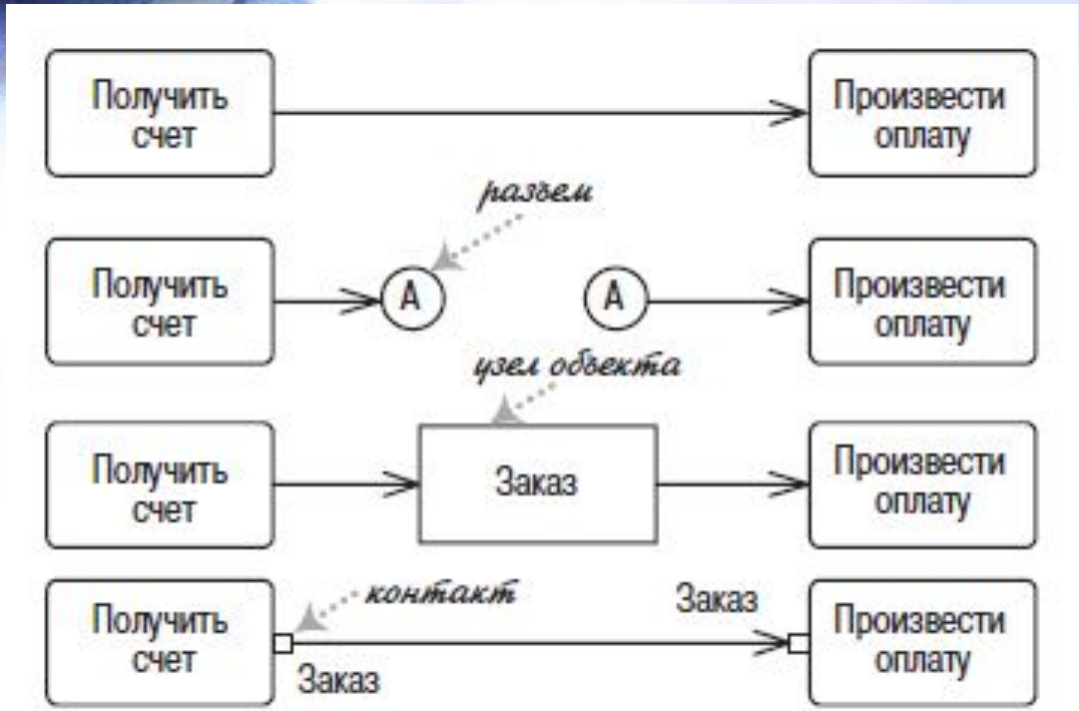
Сигналы



Можно как принимать сигналы, так и посылать их.
Это полезно, когда необходимо послать сообщение, а затем нужно ожидать ответа, перед тем как продолжить

Диаграммы деятельности

Потоки и ребра



В UML 2 параллельно употребляются термины поток (flow) и ребро (edge) для обозначения связи между двумя операциями.

Самый простой вид ребра – это обычная стрелка между двумя операциями.

При возникновении трудностей с разводкой линий можно воспользоваться разъемами (connectors), которые позволят не рисовать линии на всем их протяжении.

Разъемы изображаются парами: один для входного и один для выходного потоков, при этом они должны иметь одну и ту же метку.

Диаграммы деятельности

Контакты и преобразования

Процедуры, как и методы, могут иметь параметры.

Показывать на диаграмме деятельности информацию о параметрах не обязательно, но при желании можно отобразить параметры с помощью контактов (pins).

Если процедура разбивается на части, то контакты должны соответствовать прямоугольникам параметров на разделенной диаграмме.

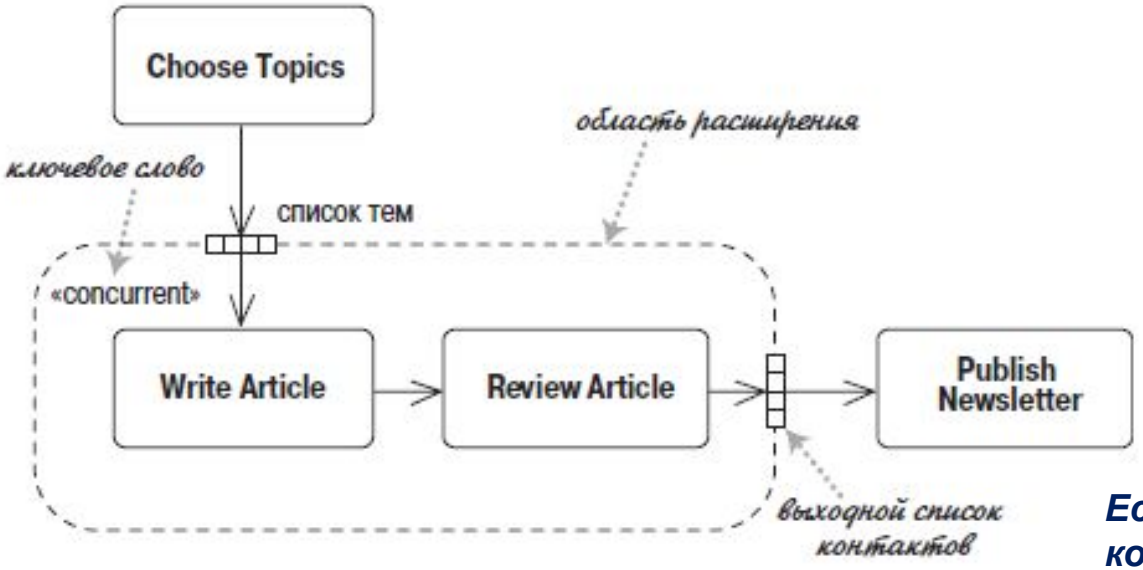
Если требуется нарисовать точную диаграмму деятельности, то необходимо обеспечить соответствие выходных параметров одной процедуры входным параметрам другой.

Если они не совпадают, то можно указать преобразование (transformation) для перехода от одной процедуры к другой.

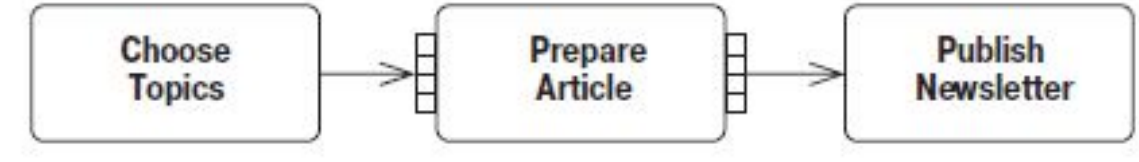
Диаграммы деятельности

Области расширения

Область расширения (expansion region)
отмечает область диаграммы деятельности, где операции выполняются один раз для каждого элемента коллекции.



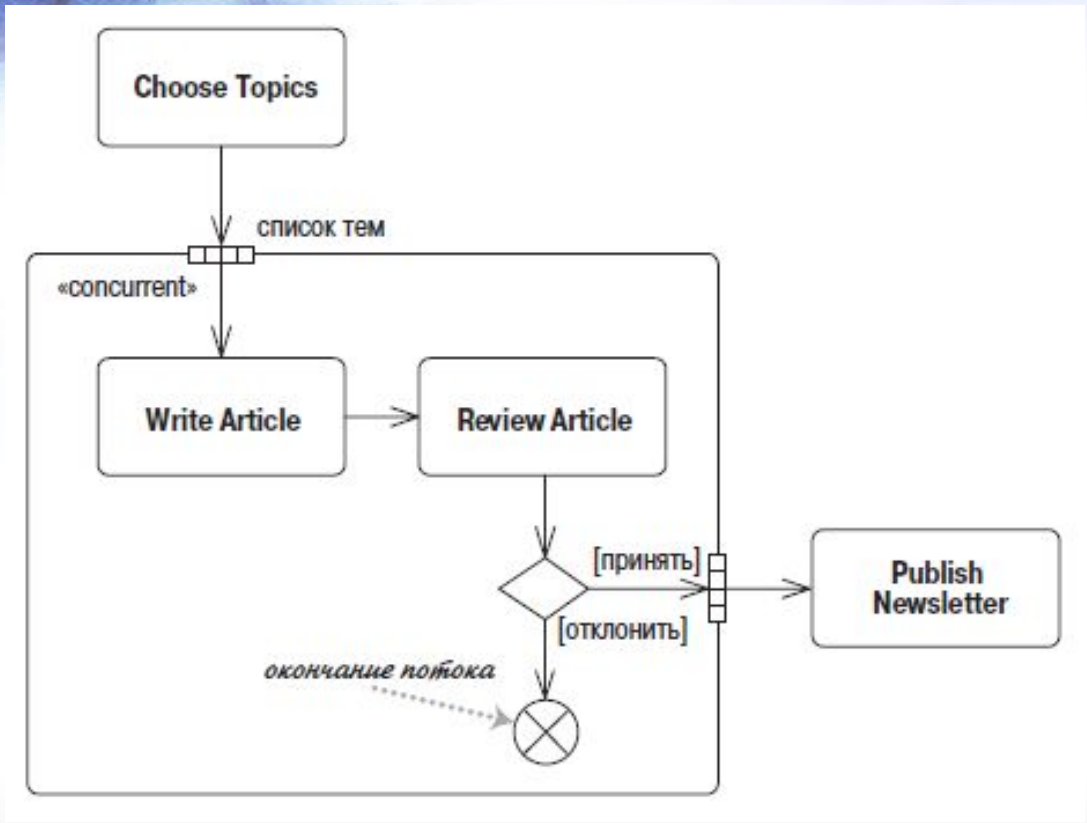
Если есть только одна операция, которую надо вызывать несколько раз, то применяется нотация, показанная ниже



Диаграммы деятельности

Окончание потока

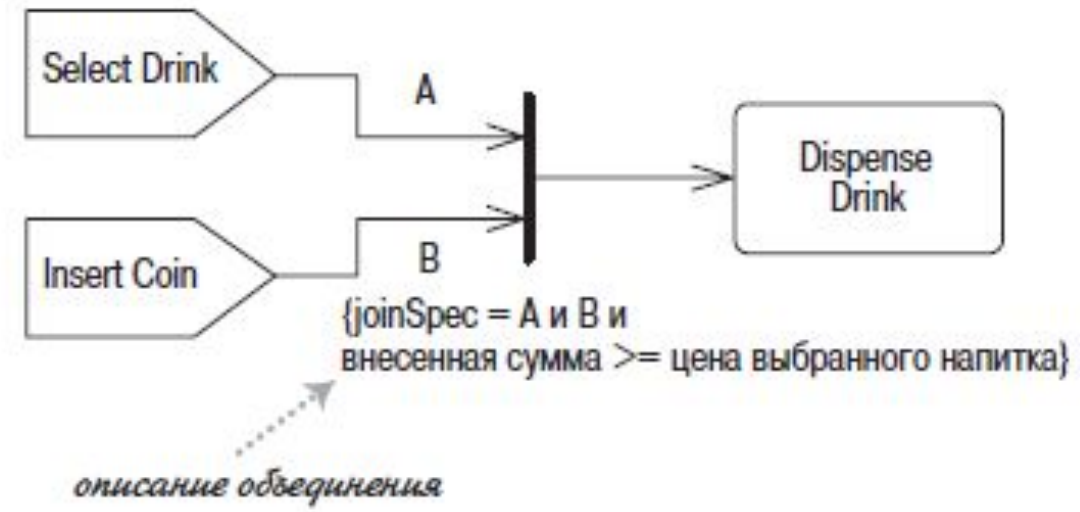
Окончание потока (flow final) означает завершение конкретного потока без завершения всей активности



Диаграммы деятельности

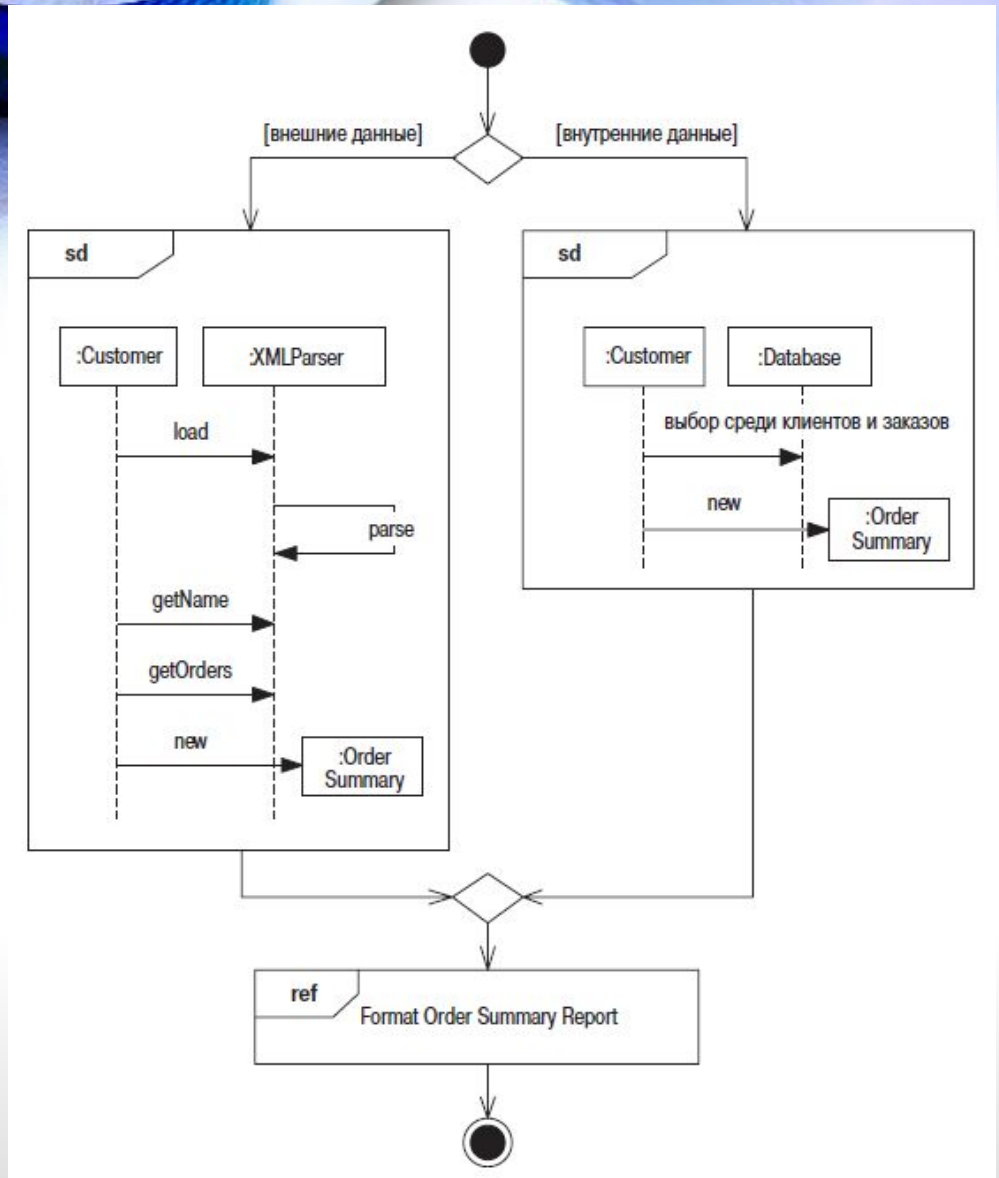
Описания объединений

По умолчанию объединение разрешает выполнение выходного потока, когда все входные потоки достигли объединения



Описание объединения (join specification) – это логическое выражение, присоединенное к объединению.

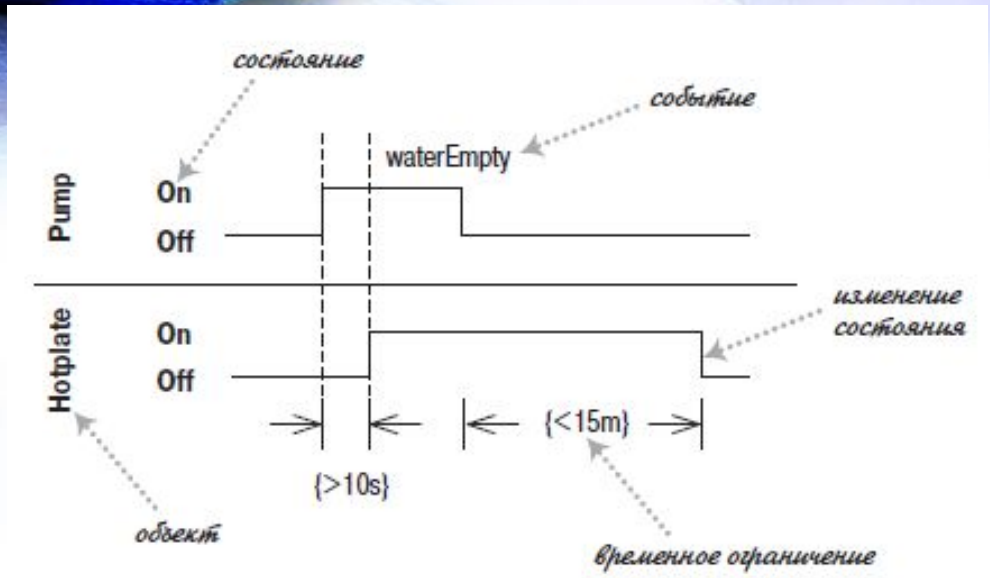
Диаграммы обзора взаимодействия



Диаграммы обзора взаимодействия – это комбинация диаграмм деятельности и диаграмм последовательности.

Можно считать диаграммы обзора взаимодействия диаграммами деятельности, в которых деятельности заменены небольшими диаграммами последовательности

Временные диаграммы

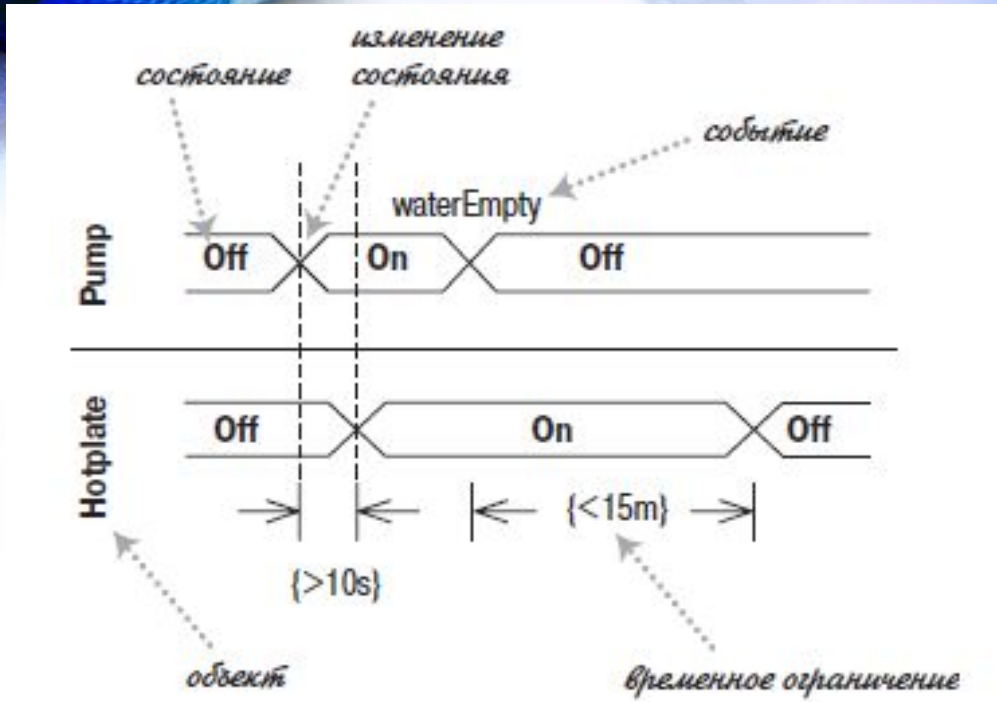


Временные диаграммы –

это еще одна форма диаграмм взаимодействия, которая акцентирована на временных ограничениях: либо для одиночного объекта, либо, что более полезно, для группы объектов.

Временная диаграмма, на которой состояния представлены в виде линий

Временные диаграммы

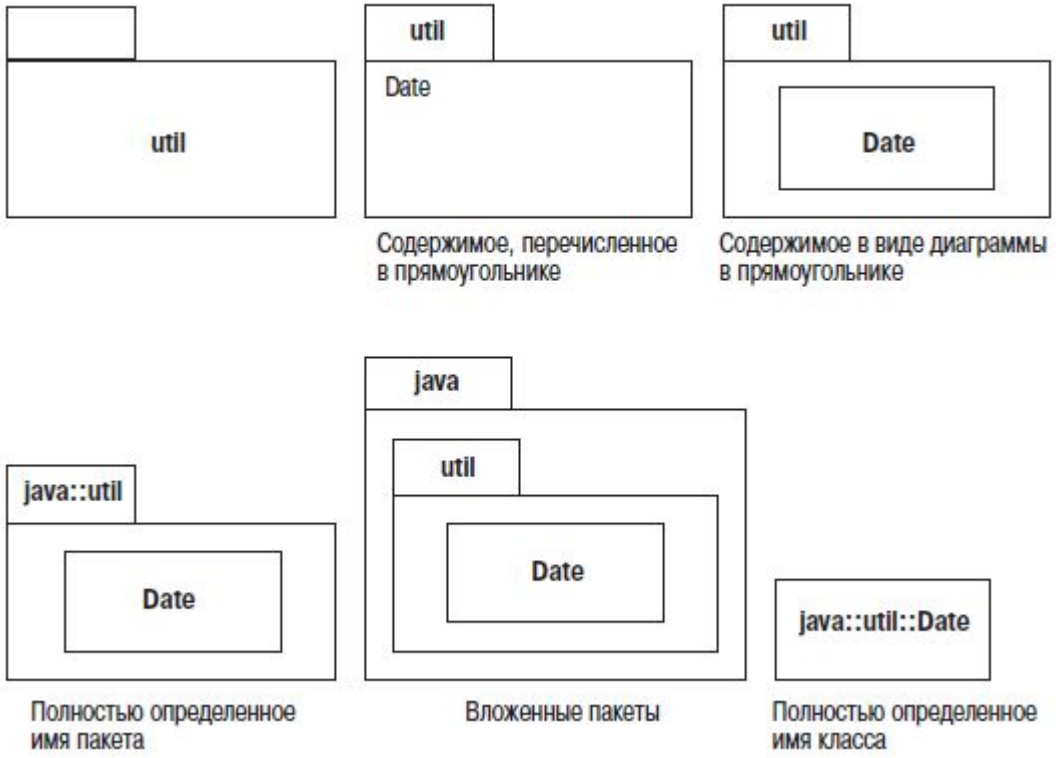


Временные диаграммы –

это еще одна форма диаграмм взаимодействия, которая акцентирована на временных ограничениях: либо для одиночного объекта, либо, что более полезно, для группы объектов.

Временная диаграмма, на которой состояния представлены в виде областей

Диаграммы пакетов

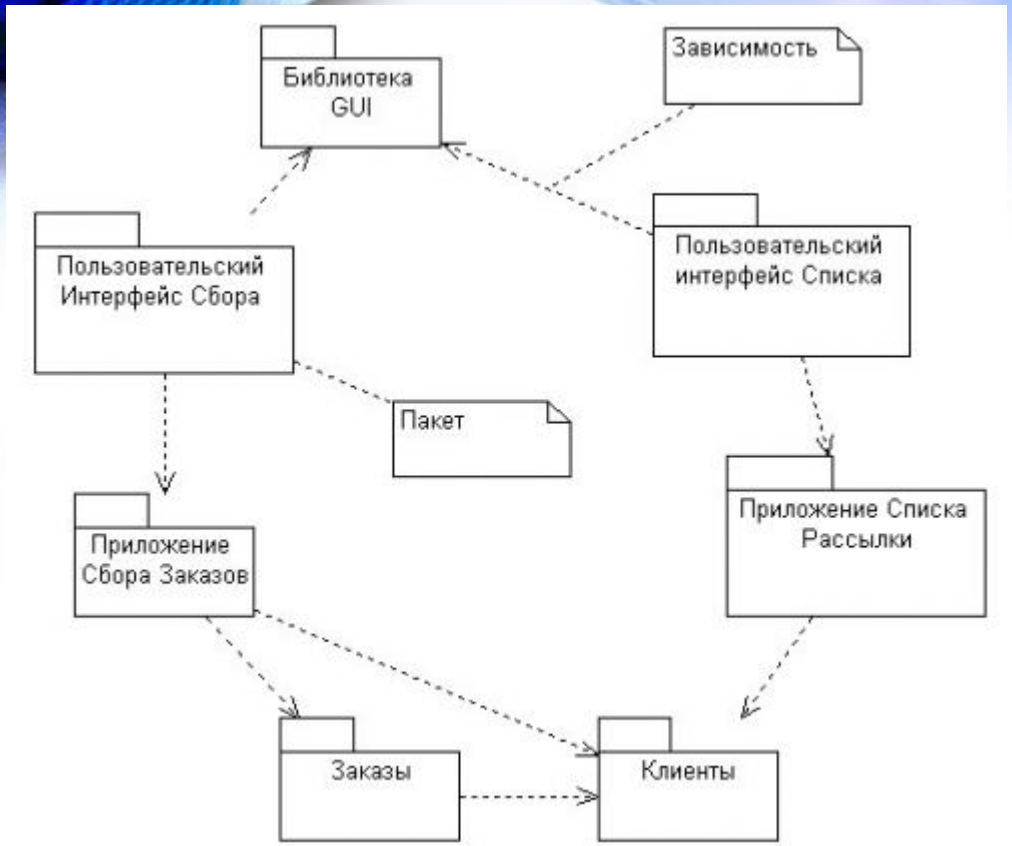


Способы изображения пакетов на диаграммах

Пакет (package) – это инструмент группирования, который позволяет взять любую конструкцию UML и объединить ее элементы в единицы высокого уровня.

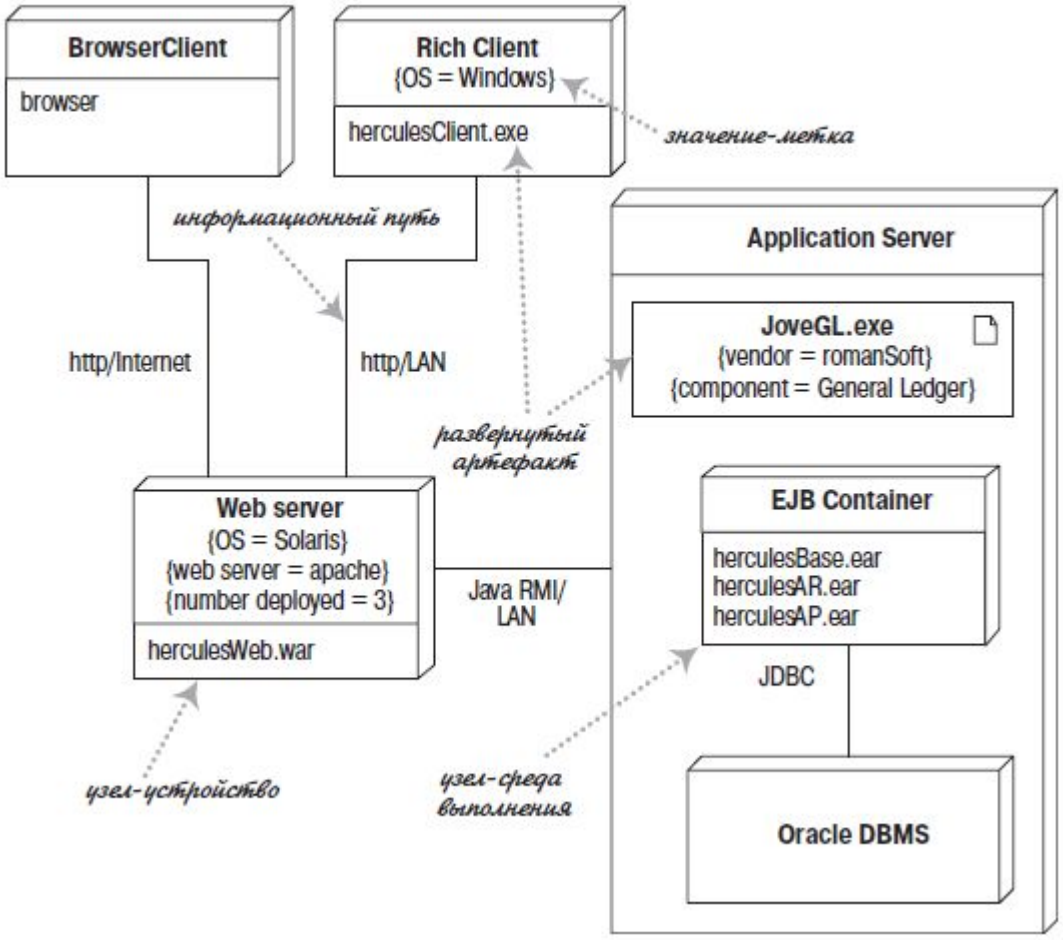
Диаграммы пакетов

Пакеты и зависимости



Классы предметной области, моделирующие деятельность организации

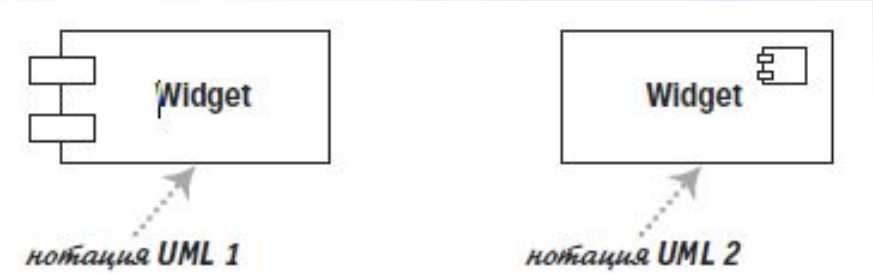
Диаграммы развертывания



Диаграммы развертывания ***представляют физическое расположение системы, показывая, на каком физическом оборудовании запускается***

та или иная составляющая программного обеспечения

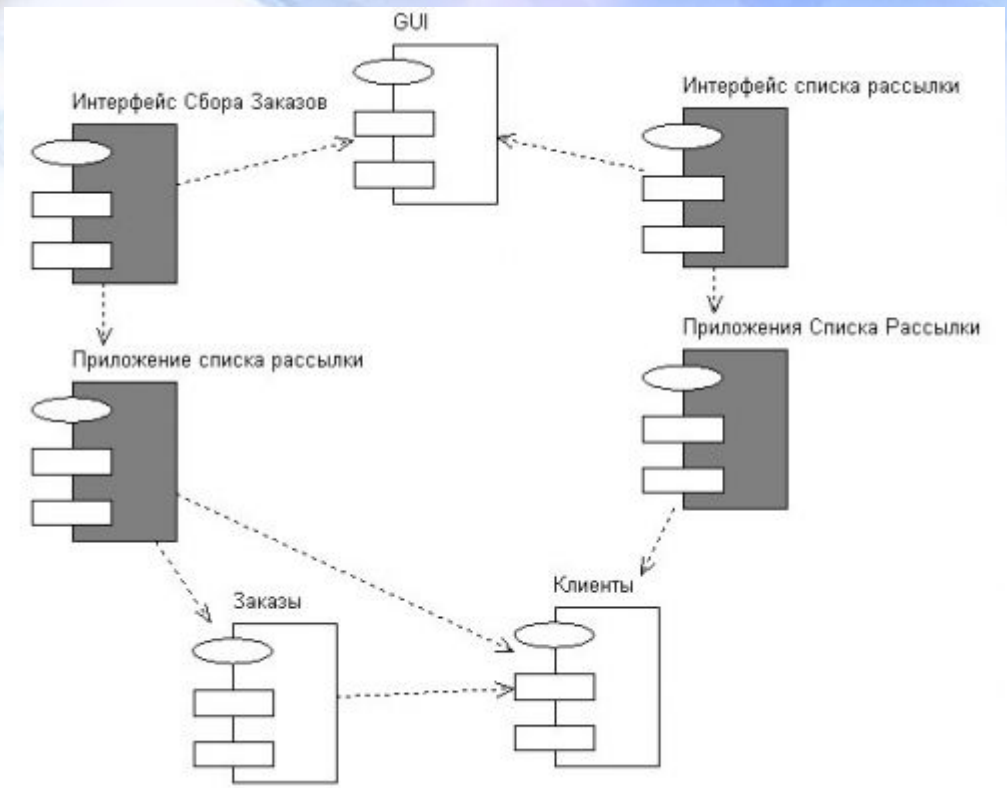
Диаграммы компонентов



Компоненты на диаграмме компонентов представляют собой физические модули программного кода.

Обычно они в точности соответствуют пакетам на диаграмме пакетов; таким образом, диаграмма компонентов отражает выполнение каждого пакета в системе.

Диаграммы компонентов



Зависимости между компонентами должны совпадать с зависимостями между пакетами. Эти зависимости показывают, каким образом одни компоненты взаимодействуют с другими.

Диаграммы компонентов

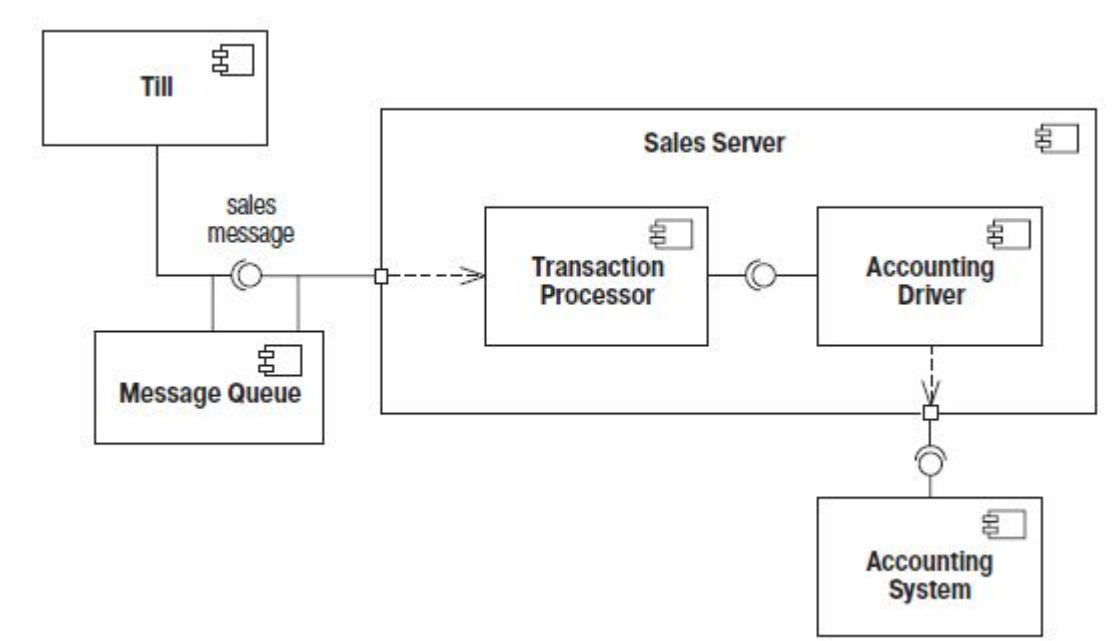


Диаграмма компонентов в нотации UML2