

## **Диаграммы последовательности**

### **Диаграммы взаимодействия**

Диаграммы взаимодействия предназначены для моделирования динамических аспектов системы.

Диаграмма взаимодействия показывает взаимодействие, включающее набор объектов и их отношений, а также пересылаемые между объектами сообщения.

Существуют две разновидности диаграммы взаимодействия — диаграмма последовательности и диаграмма сотрудничества.

Диаграмма последовательности — это диаграмма взаимодействия, которая выделяет упорядочение сообщений по времени.

Диаграмма сотрудничества — это диаграмма взаимодействия, которая выделяет структурную организацию объектов, посылающих и принимающих сообщения.

Элементами диаграмм взаимодействия являются участники взаимодействия — объекты, связи, сообщения.

### ***Диаграммы последовательности***

Отражая сценарий поведения в системе, эта диаграмма обеспечивает более наглядное представление порядка передачи сообщений. Правда, она не позволяет показать такие детали, которые видны на диаграмме сотрудничества (структурные характеристики объектов и связей).

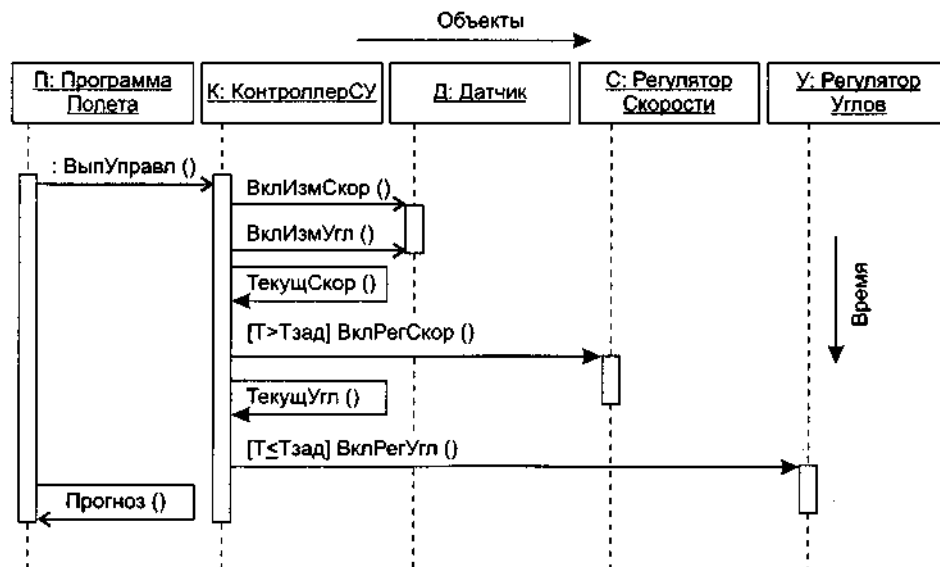
На диаграмме последовательности изображаются исключительно те объекты, которые непосредственно участвуют во взаимодействии и не показываются возможные статические ассоциации с другими объектами. Для диаграммы последовательности ключевым моментом является именно динамика взаимодействия объектов во времени.

При этом диаграмма последовательности имеет как бы два измерения. Одно — слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии.

Графически каждый объект изображается прямоугольником и располагается в верхней части своей линии жизни (рис. 1).

Внутри прямоугольника записываются имя объекта и имя класса, разделенные двоеточием. При этом вся запись подчеркивается, что является признаком объекта, который, как известно, представляет собой экземпляр класса.

Графически диаграмма последовательности — разновидность таблицы, которая показывает объекты, размещенные вдоль оси  $X$ , и сообщения, упорядоченные по времени вдоль оси  $Y$ .



**Рис. 1** Диаграмма последовательности системы управления полетом

Как показано на рис. 1, объекты, участвующие во взаимодействии, помещаются на вершине диаграммы, вдоль оси  $X$ .

Обычно слева размещается объект, инициирующий взаимодействие, а справа — объекты по возрастанию подчиненности.

Сообщения, посылаемые и принимаемые объектами, помещаются вдоль оси  $Y$  в порядке возрастания времени от вершины к основанию диаграммы. Используются те же синтаксис и обозначения синхронизации, что и в диаграммах сотрудничества. Таким образом, обеспечивается простое визуальное представление потока управления во времени.

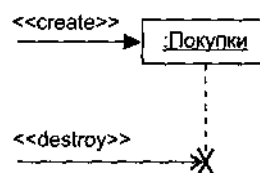
Диаграммы последовательности отличаются две важные характеристики.

Первая характеристика — *линия жизни* объекта.

Линия жизни объекта — это вертикальная пунктирная линия, которая обозначает период существования объекта. Большинство объектов существуют на протяжении всего взаимодействия, их линии жизни тянутся от вершины до основания диаграммы.

Впрочем, объекты могут создаваться в ходе взаимодействия. Их линии жизни начинаются с момента приема сообщения «create».

Кроме того, объекты могут уничтожаться в ходе взаимодействия. Их линии жизни заканчиваются с момента приема сообщения «destroy». Как представлено на рис. 2, уничтожение линии жизни отмечается пометкой  $X$  в конце линии:



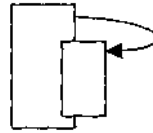
**Рис.2** Создание и уничтожение объекта

Вторая характеристика — *фокус управления*.

Фокус управления — это высокий тонкий прямоугольник, отображающий период времени, в течение которого объект выполняет действие (свою или подчиненную процедуру).

Вершина прямоугольника отмечает начало действия, а основание — его завершение. Момент завершения может маркироваться сообщением возврата, которое показывается пунктирной стрелкой.

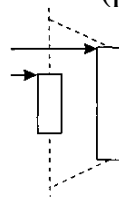
Можно показать вложение фокуса управления (например, рекурсивный вызов собственной операции). Для этого второй фокус управления рисуется немного правее первого (рис. 3).



**Рис. 3.** Вложение фокусов управления

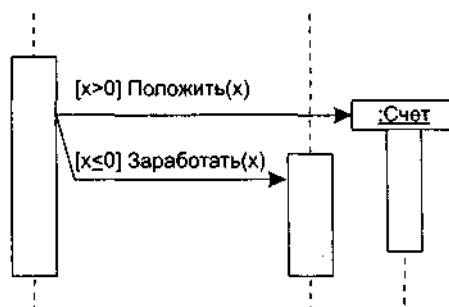
*Замечания.*

1. Для отображения «условности» линия жизни может быть разделена на несколько параллельных линий жизни. Каждая отдельная линия соответствует условному ветвлению во взаимодействии. Далее в некоторой точке линии жизни могут быть снова слиты (рис.4).



**Рис. 4.** Параллельные линии жизни

2. Ветвление показывается множеством стрелок, идущих из одной точки. Каждая стрелка отмечается сторожевым условием (рис. 5).



**Рис. 5.** Ветвление

Для изображения ветвления рисуются две или более стрелки, выходящие из одной точки фокуса управления объекта.

При этом соответствующие условия должны быть явно указаны рядом с каждой из стрелок в форме сторожевого условия.

Как нетрудно представить, если условие записано в форме булевского выражения, то ветвление будет содержать только две ветви. В любом случае

условия должны взаимно исключать одновременную передачу альтернативных сообщений.

С помощью ветвления можно изобразить и более сложную логику взаимодействия объектов между собой.

Если условий более двух, то для каждого из них необходимо предусмотреть ситуацию единственного выполнения.

Рассматриваемый пример относится к моделированию взаимодействия программной системы обслуживания клиентов в банке. На этом примере диаграммы последовательности объект 1 передает управление одному из трех других объектов.

Как было отмечено выше, цель взаимодействия в контексте языка UML заключается в том, чтобы специфицировать коммуникацию между множеством взаимодействующих объектов.

Каждое взаимодействие описывается совокупностью сообщений, которыми участвующие в нем объекты обмениваются между собой.

В этом смысле сообщение (message) представляет собой законченный фрагмент информации, который отправляется одним объектом другому.

При этом прием сообщения инициирует выполнение определенных действий, направленных на решение отдельной задачи тем объектом, которому это сообщение отправлено.

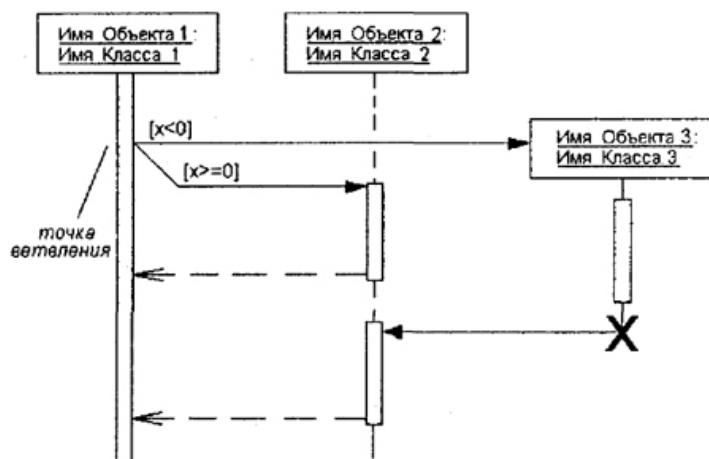
Таким образом, сообщения не только передают некоторую информацию, но и требуют или предполагают от принимающего объекта выполнения ожидаемых действий.

Сообщения могут инициировать выполнение операций объектом соответствующего класса, а параметры этих операций передаются вместе с сообщением.

На диаграмме последовательности все сообщения упорядочены по времени своего возникновения в моделируемой системе.

В таком контексте каждое сообщение имеет направление от объекта, который инициирует и отправляет сообщение, к объекту, который его получает. Иногда отправителя сообщения называют клиентом, а получателя — сервером.

При этом сообщение от клиента имеет форму запроса некоторого сервиса, а реакция сервера на запрос после получения сообщения может быть связана с выполнением определенных действий или передачи клиенту необходимой информации тоже в форме сообщения.



В языке UML могут встречаться несколько разновидностей сообщений, каждое из которых имеет свое графическое изображение (рис..

- Первая разновидность сообщения является наиболее распространенной и используется для вызова процедур, выполнения операций или обозначения отдельных вложенных потоков управления. Начало этой стрелки всегда соприкасается с фокусом управления или линией жизни того объекта-клиента, который инициирует это сообщение. Конец стрелки соприкасается с линией жизни того объекта, который принимает это сообщение и выполняет в ответ определенные действия. При этом принимающий объект зачастую получает и фокус управления, становясь активным.

- Вторая разновидность сообщения используется для обозначения простого (не вложенного) потока управления. Каждая такая стрелка указывает на прогресс одного шага потока. При этом соответствующие сообщения обычно являются асинхронными, т. е. могут возникать в произвольные моменты времени. Передача такого сообщения обычно сопровождается получением фокуса управления объектом, его принявшим.

- Третья разновидность явно обозначает асинхронное сообщение между двумя объектами в некоторой процедурной последовательности. Примером такого сообщения может служить прерывание операции при возникновении исключительной ситуации. В этом случае информация о такой ситуации передается вызывающему объекту для продолжения процесса дальнейшего взаимодействия.

- Наконец, последняя разновидность сообщения используется для возврата из вызова процедуры. Примером может служить простое сообщение о завершении некоторых вычислений без предоставления результата расчетов объекту-клиенту. В процедурных потоках управления эта стрелка может быть опущена, поскольку ее наличие неявно предполагается в конце активизации объекта. В то же время считается, что каждый вызов процедуры имеет свою пару — возврат вызова. Для непроцедурных потоков управления,

включая параллельные и асинхронные сообщения, стрелка возврата должна указываться явным образом.

Обычно сообщения изображаются горизонтальными стрелками, соединяющими линии жизни или фокусы управления двух объектов на диаграмме последовательности.

При этом неявно предполагается, что время передачи сообщения достаточно мало по сравнению с процессами выполнения действий объектами.

Считается также, что за время передачи сообщения с соответствующими объектами не может произойти никаких событий. Другими словами, состояния объектов остаются без изменения. Если же это предположение не может быть признано справедливым, то стрелка сообщения изображается под некоторым наклоном, так чтобы конец стрелки располагался ниже ее начала.

В отдельных случаях объект может посылать сообщения самому себе, иницируя так называемые рефлексивные сообщения.

Такие сообщения изображаются прямоугольником со стрелкой, начало и конец которой совпадают.

Подобные ситуации возникают, например, при обработке нажатий на клавиши клавиатуры при вводе текста в редактируемый документ, при наборе цифр номера телефона абонента.

Таким образом, в языке UML каждое сообщение ассоциируется с некоторым действием, которое должно быть выполнено принявшим его объектом.

При этом действие может иметь некоторые аргументы или параметры, в зависимости от конкретных значений которых может быть получен различный результат.

Соответствующие параметры будет иметь и вызывающее это действие сообщение.

Более того, значения параметров отдельных сообщений могут содержать условные выражения, образуя ветвление или альтернативные пути основного потока управления.

### **Стереотипы сообщений**

В языке UML предусмотрены некоторые стандартные действия, выполняемые в ответ на получение соответствующего сообщения. Они могут быть явно указаны на диаграмме последовательности в форме стереотипа рядом с сообщением, к которому они относятся. В этом случае они записываются в кавычках. Используются следующие обозначения для моделирования действий:

- "call" (вызвать) — сообщение, требующее вызова операции или процедуры принимающего объекта. Если сообщение с этим стереотипом рефлексивное, то оно иницирует локальный вызов операции у самого пославшего это сообщение объекта;

- "return" (возвратить) — сообщение, возвращающее значение выполненной операции или процедуры вызвавшему ее объекту. Значение результата может инициировать ветвление потока управления;
- "create" (создать) — сообщение, требующее создания другого объекта для выполнения определенных действий. Созданный объект может получить фокус управления, а может и не получить его;
- "destroy" (уничтожить) — сообщение с явным требованием уничтожить соответствующий объект. Посылается в том случае, когда необходимо прекратить нежелательные действия со стороны существующего в системе объекта, либо когда объект больше не нужен и должен освободить задействованные им системные ресурсы;
- "send" (послать) — обозначает посылку другому объекту некоторого сигнала, который асинхронно инициируется одним объектом и принимается (перехватывается) другим. Отличие сигнала от сообщения заключается в том, что сигнал должен быть явно описан в том классе, объект которого инициирует его передачу.

Кроме стереотипов, сообщения могут иметь собственное обозначение операции, вызов которой они инициируют у принимающего объекта. В этом случае рядом со стрелкой записывается имя операции с круглыми скобками, в которых могут указываться параметры или аргументы соответствующей операции. Если параметры отсутствуют, то скобки все равно должны присутствовать после имени операции. Примерами таких операций могут служить следующие: "выдать клиенту наличными сумму (п)", "установить соединение между абонентами (a,b)", "сделать вводимый текст невидимым ()", "подать звуковой сигнал тревоги ()".

### **Временные ограничения на диаграммах последовательности**

В отдельных случаях выполнение тех или иных действий на диаграмме последовательности может потребовать явной спецификации временных ограничений, накладываемых на сам интервал выполнения операций или передачу сообщений.

В языке UML для записи временных ограничений используются фигурные скобки. Временные ограничения могут относиться как к выполнению определенных действий объектами, так и к самим сообщениям, явно специфицируя условия их передачи или приема. Важно понимать, что в отличие от условий ветвления, которые должны выполняться альтернативно, временные ограничения имеют обязательный или директивный характер для ассоциированных с ними объектов.

Временные ограничения могут записываться рядом с началом стрелки соответствующего сообщения.

Но наиболее часто они записываются слева от этой стрелки на одном уровне с ней.

Если временная характеристика относится к конкретному объекту, то имя этого объекта записывается перед именем характеристики и отделяется от нее точкой.

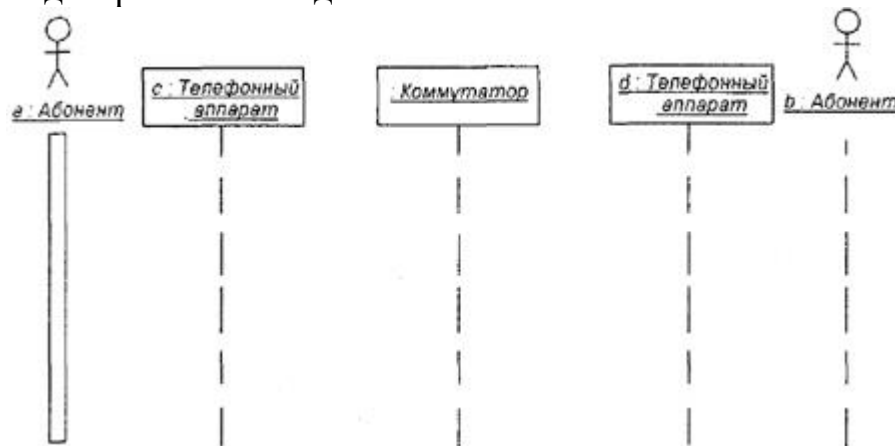
Примерами таких ограничений на диаграмме последовательности могут служить ситуации, когда необходимо явно специфицировать время, в течение которого допускается передача сообщения от клиента к серверу или обработка запроса клиента сервером:

- {время\_приема\_сообщения время\_отправки\_сообщения < 1 сек.}
- {время\_ожидания\_ответа < 5 сек.}
- {время\_передачи\_пакета < 10 сек.}
- {объект\_1. время\_подачи\_сигнала\_тревоги > 30 сек.}

В качестве примера рассмотрим построение диаграммы последовательности для моделирования процесса телефонного разговора с использованием мобильной телефонной сети. Объектами в этом примере являются: два абонента а и b, два телефонных аппарата end, коммутатор и сам разговор как объект моделирования. При этом как коммутатор, так и разговор являются анонимными объектами.

На первом этапе располагаем выбранные объекты на предполагаемой диаграмме (рис. 6). Заметим, что абонентов мы будем рассматривать как актеров, причем первый из них — а — играет активную роль, а второй — b — пассивную роль.

Поэтому первый получает фокус управления сразу после своего появления в системе, а второй имеет только линию жизни. Коммутатор также имеет постоянную активность, что изображается его фокусом управления. Разговор как объект появляется только после установки соединения и уничтожается с его прекращением. Поэтому он будет изображен позже на этой же диаграмме последовательности.



**Рис. 6.** Начальный фрагмент диаграммы последовательности для моделирования телефонного разговора



Процесс взаимодействия в этой системе начинается с поднятия трубки телефонного аппарата первым абонентом. Тем самым он посылает сообщение телефонному аппарату с, которое переводит этот аппарат в активное состояние и вызывает действие — подачу тонового сигнала в телефонную трубку для первого абонента.

Следующее действие также инициируется первым абонентом — набор цифр телефонного номера. Это представлено в форме итеративного сообщения со знаком "\*" слева от его имени.

Заметим, что поднятие телефонной трубки и набор цифр номера являются физическими действиями и поэтому изображаются в форме простых асинхронных сообщений.

После набора цифр или номера телефона аппарат с рекурсивно вызывает процедуру послыки коммутационных импульсов на коммутатор. Последний инициирует создание нового объекта в моделируемой системе — телефонного разговора.

Дополненный фрагмент диаграммы последовательности изображен на рис. 8.9.

После создания анонимный объект "разговор" сразу получает фокус активности и посылает сообщение телефонному аппарату d на выполнение действия — звонка вызова.

При этом второй абонент снимает трубку (асинхронное сообщение), тем самым устанавливается прямое соединение между абонентами a и b. После того как абоненты опустят трубки, разговор заканчивается. Тем самым объект "разговор" уничтожается.

Окончательный вариант диаграммы последовательности может содержать некоторые временные ограничения и комментарии (рис. 7). Назначение отдельных сообщений соответствуют рассмотренным действиям.

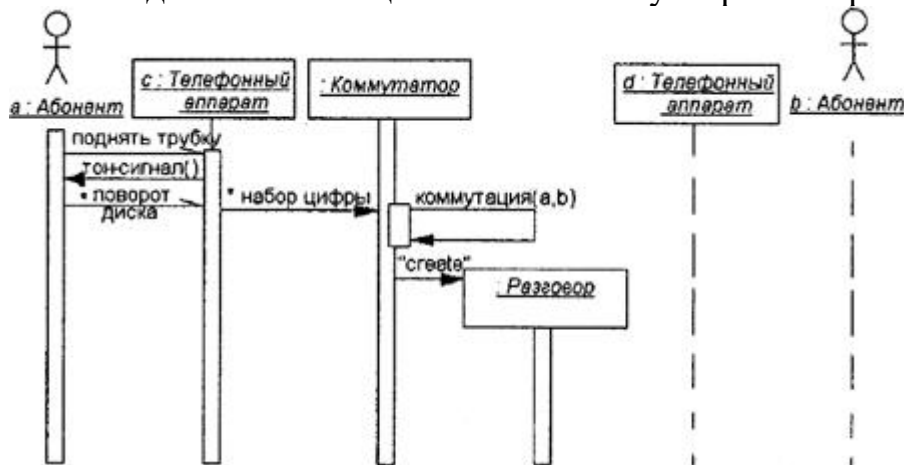
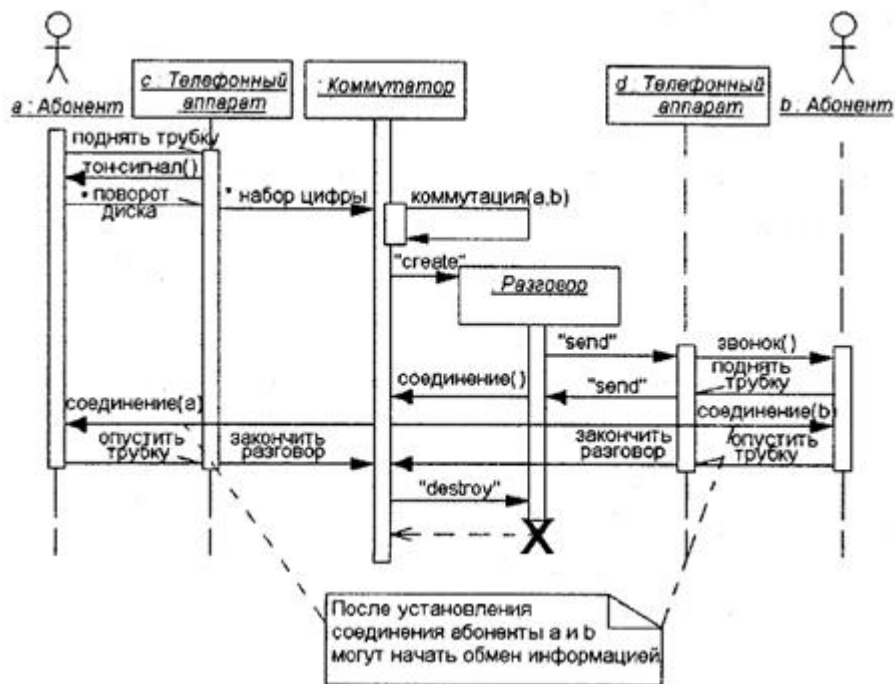


Рис. 7. Дополненный фрагмент диаграммы последовательности для моделирования телефонного разговора



**Рис 8.** Окончательный вариант диаграммы последовательности для моделирования телефонного разговора