

25 Диаграмма кооперации (collaboration diagram). Кооперация. Объекты. Связи. Сообщения.

Диаграммы кооперации: <http://www.intuit.ru/studies/courses/32/32/lecture/1012?page=1>

Альтернативный вариант: http://www.info-system.ru/designing/methodology/uml/theory/collaboration_diagram_theory.html

UML: <http://ru.wikipedia.org/wiki/UML>

Диаграмма кооперации предназначена для описания поведения системы на уровне отдельных объектов, которые обмениваются между собой сообщениями, чтобы достичь нужной цели или реализовать некоторый вариант использования.

Главная особенность диаграммы кооперации заключается в возможности графически представить не только последовательность взаимодействия, но и все структурные отношения между объектами, участвующими в этом взаимодействии.

Прежде всего, на диаграмме кооперации в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов. Далее, как и на диаграмме классов, указываются ассоциации между объектами в виде различных соединительных линий. При этом можно явно указать имена ассоциации и ролей, которые играют объекты в данной ассоциации. Дополнительно могут быть изображены динамические связи - потоки сообщений. Они представляются также в виде соединительных линий между объектами, над которыми располагается стрелка с указанием направления, имени сообщения и порядкового номера в общей последовательности инициализации сообщений.

На этой диаграмме не указывается время в виде отдельного измерения. Поэтому последовательность взаимодействий и параллельных потоков может быть определена с помощью порядковых номеров.

Кооперация (collaboration) — спецификация множества объектов отдельных классов, совместно взаимодействующих с целью реализации отдельных вариантов использования в общем контексте моделируемой системы.

Понятие кооперации – одно из фундаментальных в языке UML. Цель самой кооперации состоит в том, чтобы специфицировать особенности реализации отдельных вариантов использования или наиболее значимых операций в системе. Кооперация определяет структуру поведения системы в терминах взаимодействия участников этой кооперации.

Объекты и их графическое изображение

Объект (object) — сущность с хорошо определенными границами и индивидуальностью, которая инкапсулирует состояние и поведение.

В контексте языка UML любой объект является экземпляром класса, описанного в модели и представленного на диаграмме классов. Объект создается на этапе реализации модели или выполнения программы. Он имеет собственное имя и конкретные значения атрибутов.

Для диаграмм кооперации полное имя объекта в целом представляет собой строку текста, разделенную двоеточием и записанную в формате:

<собственное имя объекта>/'<Имя роли класса>:<Имя класса>.

Если указано собственное имя объекта, то оно должно начинаться со строчной буквы. В то же время имя объекта, имя роли с символом " / " или имя класса могут отсутствовать. Однако двоеточие всегда должно стоять перед именем класса, а косая черта – перед именем роли.

Таким образом, на диаграммах кооперации могут встретиться следующие варианты возможных записей полного имени объекта:

- o : C – объект с собственным именем o, экземпляр класса C.
- : C – анонимный объект, экземпляр класса C.
- o:(или просто o) – объект -сирота с собственным именем o.
- o / R : C – объект с собственным именем o, экземпляр класса C, играющий роль R.
- / R : C – анонимный объект, экземпляр класса C, играющий роль R.
- o / R – объект -сирота с собственным именем o, играющий роль R.
- / R – анонимный объект и одновременно объект -сирота, играющий роль R.

Примеры изображения объектов на диаграммах кооперации приводятся на [рис. 7.1](#).



Рис. 7.1. Примеры графических изображений объектов на диаграммах кооперации уровня примеров

Если собственное имя объекта отсутствует, то такой объект принято называть **анонимным**. Однако в этом случае обязательно ставится двоеточие перед именем соответствующего класса ([рис. 7.1](#), в). Отсутствовать может и имя класса – такой объект называется **сиротой**. Для него записывается только собственное имя объекта, двоеточие не ставится, имя класса не указываются ([рис. 7.1](#), г). Если для объектов указываются атрибуты, то в большинстве случаев они принимают конкретные значения ([рис. 7.1](#), б). Для отдельных объектов ([рис. 7.1](#), д, е) могут быть дополнительно указаны роли, которые они играют в кооперации.

В контексте языка UML все объекты делятся на две категории: пассивные и активные. Пассивный объект оперирует только данными и не может инициировать деятельность по управлению другими объектами. Однако пассивные объекты могут посылать сигналы в процессе выполнения запросов, которые они обрабатывают. На диаграмме кооперации пассивные объекты изображаются обычным образом без дополнительных стереотипов.

Активный объект (active object) имеет собственный процесс управления и может инициировать деятельность по управлению другими объектами .

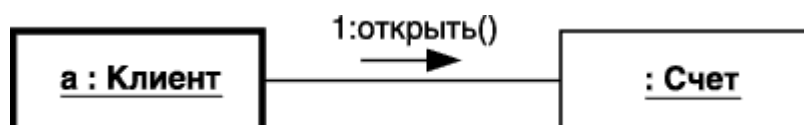


Рис. 7.2. Графическое изображение активного объекта (слева) на диаграмме кооперации

Мультиобъект (multiobject) представляет собой множество анонимных объектов, которые могут быть образованы на основе одного класса.

На диаграмме кооперации мультиобъект используется для того, чтобы показать операции и сигналы, которые адресованы всему множеству анонимных объектов.

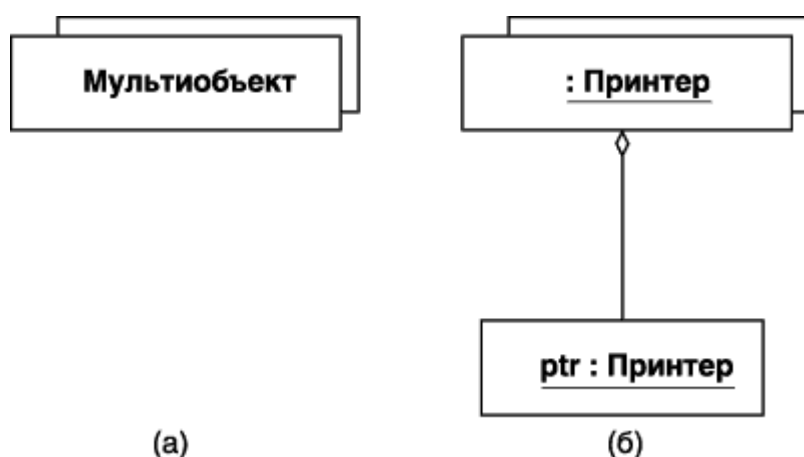


Рис. 7.3. Графическое изображение мультиобъектов на диаграмме кооперации

Связи на диаграмме кооперации

Связь (link) — любое семантическое отношение между некоторой совокупностью объектов .

Связь как элемент языка UML является экземпляром или примером произвольной ассоциации и может иметь место между двумя и более объектами. Бинарная связь на диаграмме кооперации изображается отрезком сплошной линии, соединяющей два прямоугольника объектов ([рис. 7.4](#)). На концах этой линии дополнительно могут быть явно указаны имена ролей соответствующей ассоциации.

Связи не имеют собственных имен, поскольку идентичны как экземпляры некоторой ассоциации. Другими словами, все связи на диаграмме кооперации могут быть только анонимными и при необходимости записываются без двоеточия перед именем ассоциации.



Сообщения и их графическое изображение

Сообщение (message) — спецификация передачи информации от одного элемента модели к другому с ожиданием выполнения определенных действий со стороны принимающего элемента.

При этом первый объект предполагает, что после получения сообщения вторым объектом последует выполнение некоторого действия. На диаграмме кооперации сообщение является причиной или стимулом начала выполнения операций, отправки сигналов, создания и уничтожения отдельных объектов. Связь обеспечивает канал для направленной передачи сообщений между объектами от объекта-источника к объекту-получателю.

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.

Сообщения в языке UML также специфицируют роли, которые играют объекты — отправитель и получатель сообщения. Сообщения на диаграмме кооперации изображаются дополнительными стрелками рядом с соответствующей связью или ролью ассоциации. Направление стрелки указывает на получателя сообщения. Внешний вид стрелки сообщения имеет определенный смысл. На диаграммах кооперации может использоваться один из трех типов стрелок для обозначения сообщений ([рис. 7.7](#)).

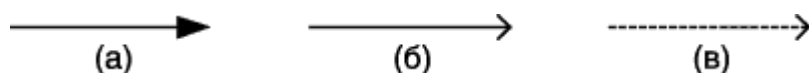


Рис. 7.7. Графическое изображение различных типов сообщений на диаграмме кооперации

- Сплошная линия с треугольной стрелкой ([рис. 7.7, а](#)) обозначает вызов процедуры (операции) или передачу потока управления. Сообщения этого типа могут быть использованы параллельно активными объектами, когда один из них передает сообщение этого типа и ожидает, пока не закончится некоторая последовательность

действий, выполняемая вторым объектом. Обычно все такие сообщения синхронны, т.е. инициируются по завершении деятельности или при выполнении определенного условия.

- Сплошная линия с V-образной стрелкой ([рис. 7.7, б](#)) обозначает асинхронное сообщение в простом потоке управления. В этом случае клиент передает асинхронное сообщение и продолжает выполнять свою деятельность, не ожидая ответа от сервера.
- Пунктирная линия с V-образной стрелкой ([рис. 7.7, в](#)) обозначает возврат из вызова процедуры. Стрелки этого типа зачастую отсутствуют на диаграммах кооперации, поскольку неявно предполагается их существование после окончания процесса выполнения операции или деятельности.

Каждое сообщение может быть помечено строкой текста, которая имеет следующий формат:

<Предшествующие сообщения> <Выражение последовательности> <Возвращаемое значение := имя сообщения> <(Список аргументов)>

В языке UML определены следующие стереотипы сообщений:

- <<call>> (вызвать) – сообщение, требующее вызова операции или процедуры объекта-получателя. Если сообщение с этим стереотипом рефлексивное, то оно инициирует локальный вызов операции у 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 стать активным (ему передается поток управления), а может остаться пассивным.
- <<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 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.