

Для моделирования поведения на логическом уровне в языке UML могут использоваться сразу несколько канонических диаграмм: состояний, деятельности, последовательности и кооперации, каждая из которых фиксирует внимание на отдельном аспекте функционирования системы. В отличие от других диаграмм диаграмма состояний описывает **процесс изменения состояний только одного класса**, а точнее - одного экземпляра определенного класса, т. е. моделирует все возможные изменения в состоянии конкретного объекта. При этом изменение состояния объекта может быть вызвано внешними воздействиями со стороны других объектов или извне. Именно для описания реакции объекта на подобные внешние воздействия и используются диаграммы состояний.

Главное предназначение этой диаграммы - описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла. Диаграмма состояний представляет динамическое поведение сущностей, на основе спецификации их реакции на восприятие некоторых конкретных событий. Системы, которые реагируют на внешние действия от других систем или от пользователей, иногда называют реактивными. Если такие действия инициируются в произвольные случайные моменты времени, то говорят об асинхронном поведении модели.

Хотя диаграммы состояний чаще всего используются для описания поведения отдельных экземпляров классов (объектов), но они также могут быть применены для спецификации функциональности других компонентов моделей, таких как варианты использования, актеры, подсистемы, операции и методы.

Диаграмма состояний по существу является графом специального вида, который представляет некоторый автомат. Понятие **автомата** в контексте UML обладает довольно специфической семантикой, основанной на теории автоматов.

V(Вершинами) - состояния и некоторые другие типы элементов автомата (псевдосостояния), которые изображаются соответствующими графическими символами.

E (Дуга) - служат для обозначения переходов из состояния в состояние.

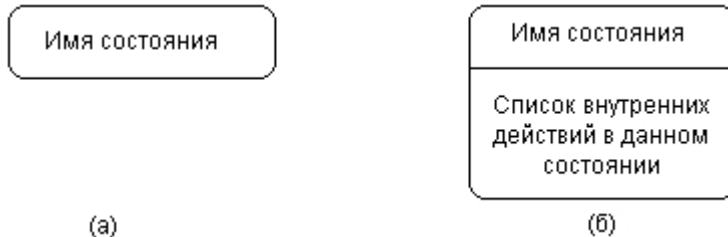
Диаграммы состояний могут быть вложены друг в друга, образуя вложенные диаграммы более детального представления отдельных элементов модели.

Состояния

В языке UML под состоянием понимается абстрактный метакласс, используемый для моделирования отдельной ситуации, в течение которой имеет место выполнение некоторого условия. Состояние может быть задано в виде набора конкретных значений атрибутов класса или объекта, при этом изменение их отдельных значений будет отражать изменение состояния моделируемого класса или объекта.

Состояние на диаграмме изображается прямоугольником со скругленными вершинами (рис. 6.2). Этот прямоугольник, в свою очередь, может быть разделен на две секции горизонтальной линией. Если указана лишь одна секция, то в ней записывается только

имя состояния (рис. 6.2, а). В противном случае в первой из них записывается имя состояния, а во второй - список некоторых внутренних действий или переходов в данном состоянии (рис. 6.2, б). При этом под действием в языке UML понимают некоторую атомарную операцию, выполнение которой приводит к изменению состояния или возврату некоторого значения (например, "истина" или "ложь").



Имя состояния

Имя состояния представляет собой строку текста, которая раскрывает содержательный смысл данного состояния.

Имя всегда записывается с заглавной буквы. Поскольку состояние системы является составной частью процесса ее функционирования, рекомендуется в качестве имени использовать глаголы в настоящем времени (звенит, печатает, ожидает) или соответствующие причастия (занят, свободен, передано, получено).

Список внутренних действий

<метка-действия '/' выражение-действия>

метка действия указывает на обстоятельство действия

выражение действия может использовать любые атрибуты и связи, которые принадлежат области имен или контексту моделируемого действия.

Перечень фиксированных меток действия:

- entry - эта метка указывает на действие, специфицированное следующим за ней выражением действия, которое выполняется в момент входа в данное состояние
- exit - эта метка указывает на действие, специфицированное следующим за ней выражением действия, которое выполняется в момент входа из данного состояния
- do - метка специфицирует выполняющуюся деятельность ("do activity"), которая выполняется в течении всего времени, пока объект находится в данном состоянии, или до тех пор, пока не закончится вычисление, специфицированное следующим за ней выражением действия. **При завершении события генерируется соответствующий результат.**
- include - метка используется для обращения к подавтомату, при этом следующее за ней выражение действия содержит имя этого подавтомата.

Ввод пароля
entry / установить символы невидимыми
exit / установить символы видимыми
символ / получить символ
помощь / показать помощь

Начальное и конечное состояние

Начальное состояние служит для указания графической области, от которой начинается процесс изменения состояний. Конечное - где заканчивается.

Переход (transition)

Переход означает перемещение из одного состояния в другое. Каждый переход имеет свою метку:

триггер-идентификатор [защита] / активность (trigger-signature[guard]/activity)

Все части не обязательны.

- триггер-идентификатор - событие, которое может вызвать изменение состояния.
- защита - логическое условие, которое должно быть выполнено, чтобы переход имел место.
- активность - некоторое поведение системы во время перехода. Это может быть любое поведенческое выражение.

Пропуск активности - означает, что ничего не происходит во время перехода.

Пропуск защиты означает, что переход всегда осуществляется.