

Диаграмма состояний (Statechart Diagram)

Для общего представления требуемой функциональности моделируемой системы предназначены *диаграммы вариантов использования*. Они на концептуальном уровне описывают поведение, требуемое от системы в целом или от её подсистем.

Диаграмма состояний описывает процесс *изменения состояний* только одного класса, точнее — одного экземпляра определенного класса. Она моделирует все возможные изменения в состоянии конкретного объекта.

Изменения состояния объекта могут быть вызваны внутренними причинами или внешними воздействиями (со стороны других объектов системы или извне).

1. Автомат

Автомат (state machine) в языке UML представляет собой некоторый формализм для моделирования поведения элементов модели и системы в целом. С другой стороны, автомат описывает поведение отдельного объекта в форме последовательности состояний, которые охватывают все этапы его жизненного цикла, начиная от создания объекта и заканчивая его уничтожением. Каждая диаграмма состояний представляет некоторый автомат.

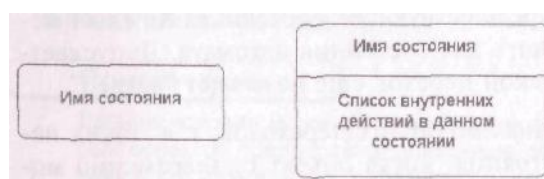
Простейший пример — ситуация с исправностью технического устройства, например, компьютера. Вводятся два самых общих состояния: "исправен" и "неисправен" и два перехода: "выход из строя" и "ремонт". В виде диаграммы состояний это отображается как:



Основные понятия формализма автомата — *состояние* и *переход*. Длительность нахождения системы в отдельном состоянии существенно превышает время, которое затрачивается на переход из одного состояния в другое.

2. Состояние

Понятие *состояния* (state) является фундаментальным в метамодели языка UML и в прикладном системном анализе. В языке UML под состоянием понимается абстрактный метакласс, используемый для *моделирования отдельной ситуации*, в течение которой имеет место выполнение некоторого условия.



Состояние на диаграмме изображается прямоугольником со скругленными вершинами (см. рис.). Прямоугольник может быть разделен на две секции горизонтальной линией. В верхней (или единственной) записывается *имя состояния*, в нижней — список внутренних действий или переходов в данном состоянии. Действие — атомарная операция, выполнение которой приводит к изменению состояния или возврату некоторого значения (например, "истина" или "ложь").

Имя состояния

Имя состояния — строка текста, раскрывающая содержательный смысл данного состояния. Имя записывается с заглавной буквы. Рекомендуется в качестве имени использовать глаголы в настоящем времени (звонит, печатает, ожидает) или причастия (занят, свободен, передано, получено).

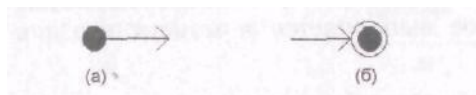
Список внутренних действий

Каждое действие записывается в виде отдельной строки и имеет следующий формат: $\langle \text{метка-действия} / \text{выражение-действия} \rangle$.

Метка действия указывает на обстоятельства или условия, при которых будет выполняться деятельность, определенная выражением действия. Если список выражений действия пустой, то разделитель в виде наклонной черты '/' может не указываться.

Начальное состояние

Начальное состояние представляет собой частный случай состояния, которое не содержит никаких внутренних действий (псевдосостояния).



Конечное состояние

Конечное (финальное) состояние представляет собой частный случай состояния. Не содержит никаких внутренних действий (псевдосостояния). В этом состоянии будет находиться объект по умолчанию после завершения работы автомата в конечный момент времени.

3. Переход

Простой переход (simple transition) представляет собой отношение между двумя последовательными состояниями, которое указывает на факт смены одного состояния другим. Переход осуществляется при наступлении *окончания выполнения деятельности* (do activity), *получении* объектом сообщения или *приеме* сигнала. На переходе указывается имя события. На переходе могут указываться действия, производимые объектом. Срабатывание перехода может зависеть от сторожевого условия.

Изображается сплошной линией со стрелкой, направленной в целевое состояние. Может быть помечен строкой текста, следующего формата:

$\langle \text{сигнатура события} \rangle$ '[' $\langle \text{сторожевое условие} \rangle$ ']' $\langle \text{выражение действия} \rangle$.

При этом *сигнатура события* описывает некоторое событие с необходимыми аргументами: <имя события> '(' <список параметров, разделенных запятыми> ')

Событие

Событие (event) представляет собой спецификацию некоторого факта, имеющего место в пространстве и во времени. Про события говорят, что они "происходят", при этом отдельные события должны быть упорядочены во времени. В языке UML события играют роль стимулов, которые инициируют переходы из одних состояний в другие.

Имя события идентифицирует каждый отдельный переход. В этом случае принято считать переход *триггерным*, т. е. таким, который специфицирует событие-триггер.

Если рядом со стрелкой перехода не указана никакая строка текста, то соответствующий переход является *нетриггерным*.

Сторожевое условие

Сторожевое условие (guard condition) представляет собой некоторое булевское выражение. Вычисление истинности сторожевого условия происходит только после возникновении ассоциированного с ним события-триггера, инициирующего соответствующий переход. Из одного состояния может быть несколько переходов с одним и тем же событием-триггером. Никакие два сторожевых условия не должны одновременно принимать значение "истина".

Выражение действия

Выражение действия (action expression) — атомарная операция. Выполняется сразу *после* срабатывания соответствующего перехода *до начала* каких бы то ни было действий в целевом состоянии. Атомарность действия означает, что это достаточно простое вычисление и оно не может быть прервано никаким другим действием до тех пор, пока не закончится его выполнение.

Выражение действия может содержать целый список отдельных действий, разделенных символом ";".

4. Составное состояние и подсостояние

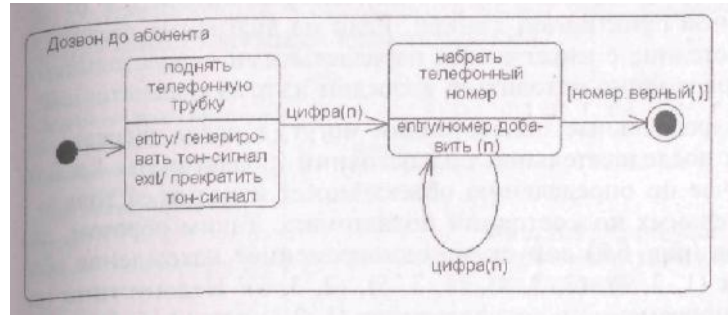
Составное состояние (composite state) — сложное состояние, состоящее из других вложенных в него состояний. Последние будут выступать по отношению к первому как *подсостояния* (substate). Изображаются внутри символа составного состояния (см. рис).



Составное состояние может содержать два или более параллельных подавтомата или несколько последовательных подсостояний. Каждое сложное состояние может уточняться только одним из указанных способов.

Последовательные подсостояния

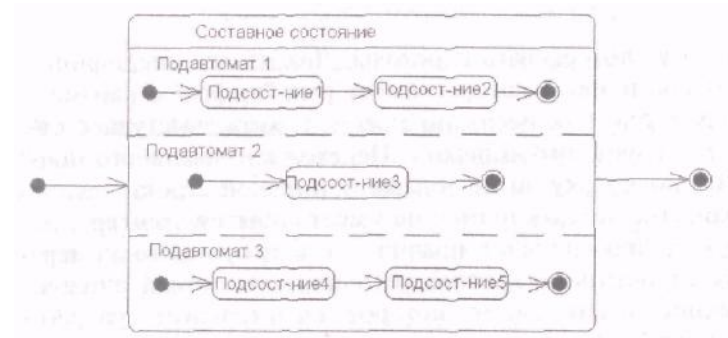
Последовательные подсостояния (sequential substates) используются для моделирования такого поведения объекта, во время которого в каждый момент времени объект может находиться в одном и только одном подсостоянии.



Параллельные подсостояния

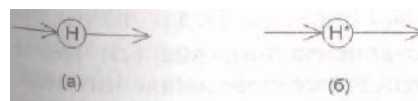
Параллельные подсостояния (concurrent substates) позволяют специфицировать два и более подавтомата, которые могут выполняться параллельно внутри составного события. Отеляются горизонтальной пунктирной линией.

Составное состояние с вложенными параллельными подсостояниями, определяет что объект может одновременно находиться в каждом из этих подсостояний.



5. Историческое состояние

Историческое состояние (history state) применяется в контексте составного состояния. Оно используется для запоминания того из последовательных подсостояний, которое было текущим в момент выхода из составного состояния.



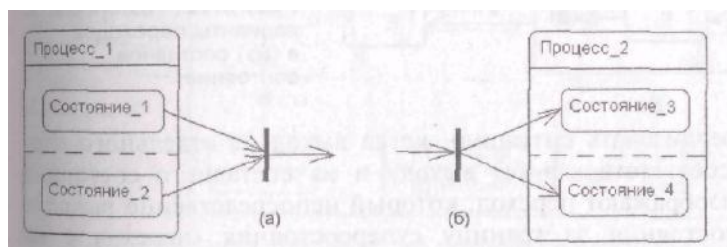
Недавнее историческое состояние (shallow history state) (см. рис., а) является первым подсостоянием в составном состоянии, и переход извне в это составное состояние должен вести непосредственно в это историческое состояние.

Запомненное состояние также может являться составным состоянием. *Давнее историческое состояние* (deep history state) служит для запоминания всех подсостояний любого уровня вложенности для текущего подавтомата.

6. Сложные переходы

Переходы между параллельными состояниями

Переход может иметь несколько состояний–источников и несколько целевых состояний. Такой переход получил специальное название — *параллельный* переход. Если параллельный переход имеет две или более входящих дуг (см. рис., а), то его называют *соединением* (join). Если же он имеет две или более исходящих дуг (см. рис., б), то его называют *ветвлением* (fork).



Переходы между составными состояниями

Переход, стрелка которого соединена с границей некоторого составного состояния, обозначает переход в составное состояние (переход **b** на рис.). Он эквивалентен переходу в начальное состояние каждого из подавтоматов (возможно, единственному), входящих в состав данного суперсостояния. Переход, выходящий из составного состояния (переходы **f** и **g** на рис.) относится к каждому из вложенных подсостояний. Это означает, что объект может покинуть составное суперсостояние, находясь в любом из его подсостояний. Для этого вполне достаточно выполнения события и сторожевого условия.

Синхронизирующие состояния

Синхронизирующее состояние (synch state) используется чтобы явно указать события в других подавтоматах, оказывающие непосредственное влияние на поведение данного подавтомата.