State chart

Диаграмма состояний (state chart) - диаграмма, которая представляет конечный автомат.

Главное назначение диаграммы состояний - описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение моделируемой системы в течение всего ее 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.

Диаграммы состояний могут быть вложены друг в друга, образуя вложенные диаграммы для более детального представления состояний отдельных элементов модели. Для понимания семантики конкретной диаграммы состояний необходимо представлять особенности поведения моделируемой сущности, а также иметь общие сведения из теории конечных автоматов.

Конечный автомат

Конечный автомат (state machine) - модель для спецификации поведения объекта в форме последовательности его состояний, которые описывают реакцию объекта на внешние события, выполнение объектом действий, а также изменение его отдельных свойств.

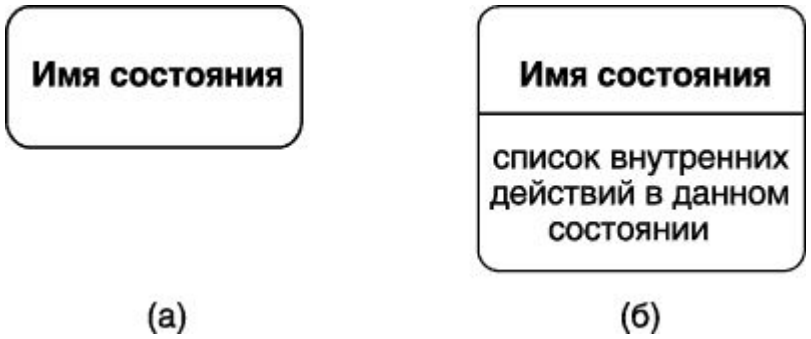
Основными понятиями, характеризующими конечный автомат, являются состояние и переход. Ключевое различие между ними заключается в том, что длительность нахождения системы в отдельном состоянии существенно превышает время, которое затрачивается на переход из одного состояния в другое. Предполагается, что в пределе время перехода из одного состояния в другое равно нулю (если дополнительно ничего не сказано). Другими словами, переход объекта из состояния в состояние происходит мгновенно.

В общем случае конечный автомат представляет динамические аспекты моделируемой системы в виде ориентированного графа, вершины которого соответствуют состояниям, а дуги - переходам. При этом поведение моделируется как последовательное перемещение по графу состояний от вершины к вершине по связывающим их дугам с учетом их ориентации.

Состояние

Состояние (**state**) - условие или ситуация в ходе жизненного цикла объекта, в течение которого он удовлетворяет логическому условию, выполняет определенную деятельность или ожидает события. Состояние может быть задано в виде набора конкретных значений атрибутов объекта некоторого класса, при этом изменение отдельных значений этих атрибутов будет отражать изменение состояния моделируемого объекта в целом.

Состояние на диаграмме изображается прямоугольником со скругленными вершинами. 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 . В противном случае в первой из них записывается имя состояния, а во второй - список некоторых внутренних действий или переходов в данном состоянии.



Действия

Действие (action) – выполнимое атомарное вычисление, в результате которого изменяется состояние системы или возвращается значение.

Для ряда состояний может потребоваться дополнительно указать действия, которые должны быть выполнены моделируемым элементом. Для этой цели служит добавочная секция в обозначении состояния, содержащая перечень внутренних действий или деятельность, которые производятся в процессе нахождения моделируемого элемента в данном состоянии. Каждое действие записывается в виде отдельной строки и имеет следующий формат:

<метка действия '/ ' выражение действия>

Пример состояния с непустой секцией внутренних действий

Действие

Перечень меток действий в языке UML фиксирован, причем эти метки не могут быть использованы в качестве имен событий:

Входное действие (entry action) - действие, которое выполняется в момент перехода в данное состояние. Обозначается с помощью ключевого слова - метки действия **“entry”**, которое указывает на то, что следующее за ней выражение действия должно быть выполнено в момент входа в данное состояние.

Действие выхода (exit action) - действие, производимое при выходе из данного состояния. Обозначается с помощью ключевого слова - метки действия **“exit”**, которое указывает на то, что следующее за ней выражение действия должно быть выполнено в момент выхода из данного состояния.

Внутренняя деятельность (do activity) - выполнение объектом операций или процедур, которые требуют определенного времени. Обозначается с помощью ключевого слова - метки деятельности **“do”**, выполняемую в течение всего времени, пока объект находится в данном состоянии, или до тех пор, пока не будет прервано внешним событием. При нормальном завершении внутренней деятельности генерируется соответствующее событие.

Псевдосостояния

Псевдосостояние (pseudo-state) - вершина в конечном автомате, которая имеет форму состояния, но не обладает поведением.

Начальное состояние (start state) - разновидность псевдосостояния, обозначающее начало выполнения процесса изменения состояний конечного автомата или нахождения моделируемого объекта в составном состоянии. В этом состоянии находится объект по умолчанию в начальный момент времени. Графически начальное состояние в языке UML обозначается в виде закрашенного кружка, из которого может только выходить стрелка-переход.

Конечное состояние (final state) - разновидность псевдосостояния, обозначающее прекращение процесса изменения состояний конечного автомата или нахождения моделируемого объекта в составном состоянии. В этом состоянии должен находиться моделируемый объект или система по умолчанию после завершения работы конечного автомата. Графически конечное состояние в языке UML обозначается в виде закрашенного кружка, помещенного в окружность, в которую может только входить стрелка-переход.

(а) начальное состояние

(б) конечное состояние

Переход

Переход (transition) - отношение между двумя состояниями, которое указывает на то, что объект в первом состоянии должен выполнить определенные действия и перейти во второе состояние.

Переход осуществляется при наступлении некоторого события: окончания выполнения деятельности (do activity), получении объектом сообщения или приемом сигнала. Переход может быть направлен в то же состояние, из которого он выходит. В этом случае его называют переходом в себя. Этот переход изображается петлей со стрелкой и отличается от внутреннего перехода.

На диаграмме состояний переход изображается сплошной линией со стрелкой, которая выходит из исходного состояния и направлена в целевое состояние.

Событие

Событие (event) - спецификация существенных явлений в поведении системы.

В языке UML события играют роль стимулов, которые инициируют переходы из одних состояний в другие. В качестве событий можно рассматривать сигналы, вызовы, окончание фиксированных промежутков времени или моменты окончания выполнения определенных действий. В зависимости от вида происходящих событий - стимулов в языке UML различают два типа переходов: **триггерные** и **нетриггерные**.

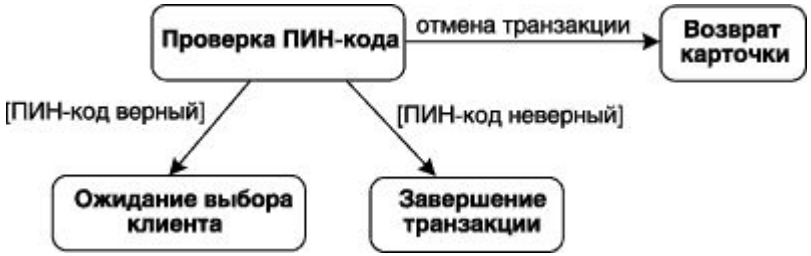
Переход называется **триггерным**, если его специфицирует событие-триггер, связанное с внешними условиями по отношению к рассматриваемому состоянию. В этом случае рядом со стрелкой триггерного перехода обязательно указывается имя события в форме строки текста, начинающейся со строчной буквы. Наиболее часто в качестве имен триггерных переходов задают имена операций, вызываемых у тех или иных объектов системы.

Переход называется **нетриггерным**, если он происходит по завершении выполнения ду-деятельности в данном состоянии. Для них рядом со стрелкой перехода не указывается никакого имени события, а в исходном состоянии должна быть описана внутренняя ду-деятельность, по окончании которой произойдет тот или иной нетриггерный переход.

Сторожевое условие

Сторожевое условие (guard condition) - логическое условие, записанное в прямых скобках и представляющее собой булевское выражение.

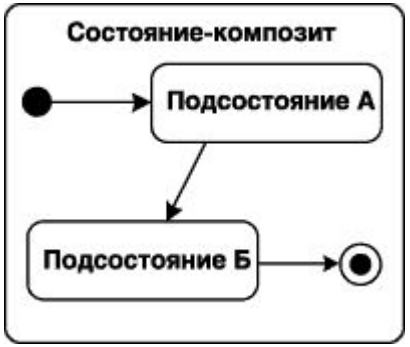
При этом булевское выражение должно принимать одно из двух взаимно исключающих значений: "истина" или "ложь". Из контекста диаграммы состояний должна явно следовать семантика этого выражения, а для записи выражения может использоваться обычный язык, псевдокод или язык программирования.



Составное состояние

Составное состояние (composite state) - сложное состояние, которое состоит из других вложенных в него состояний.

Вложенные состояния выступают по отношению к составному состоянию как подсостояния (substate). И хотя между ними имеет место отношение композиции, графически все вершины диаграммы, которые соответствуют вложенным состояниям, изображаются внутри символа составного состояния.

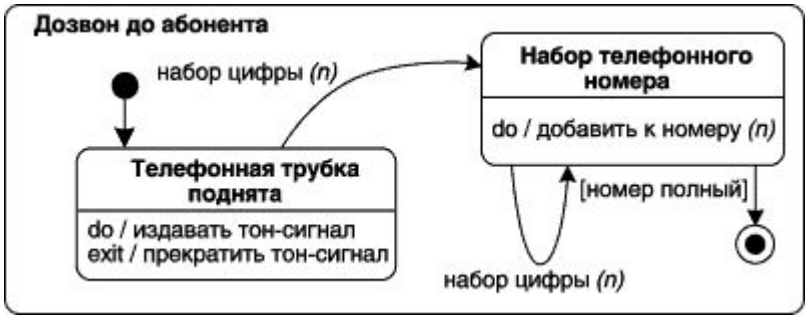


Составное состояние может содержать или несколько последовательных подсостояний, или несколько параллельных конечных подавтоматов. При этом любое из подсостояний, в свою очередь, может содержать внутри себя другие вложенные подсостояния. Количество уровней вложенности составных состояний в языке UML не фиксировано.

Последовательные подсостояния

Последовательные подсостояния (sequential substates) - вложенные состояния состояния-композита, в рамках которого в каждый момент времени объект может находиться в одном и только одном подсостоянии.

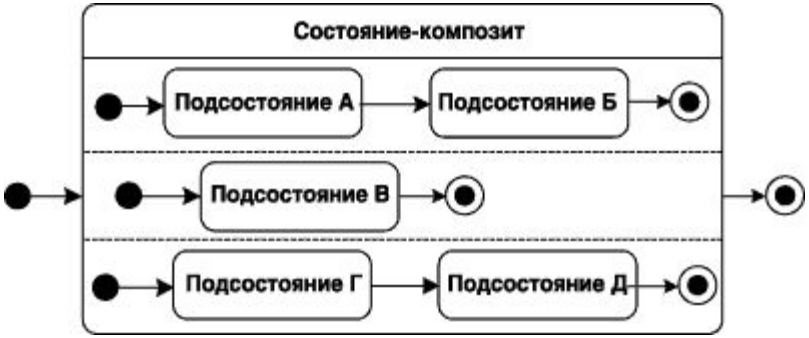
Поведение объекта в этом случае представляет собой последовательную смену подсостояний, от начального до конечного . Моделируемый объект или система продолжает находиться в составном состоянии, тем не менее, введение в рассмотрение последовательных подсостояний позволяет учесть более тонкие логические аспекты его внутреннего поведения.



Параллельные подсостояния

Параллельные подсостояния (concurrent substates) - вложенные состояния, используемые для спецификации двух и более конечных подавтоматов, которые могут выполняться параллельно внутри составного состояния.

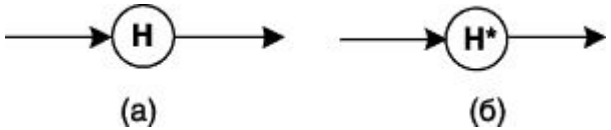
Каждый из конечных подавтоматов занимает некоторую графическую область внутри составного состояния, которая отделяется от остальных горизонтальной пунктирной линией. Если на диаграмме состояний имеется составное состояние с вложенными параллельными подсостояниями, то объект может одновременно находиться в каждом из этих подсостояний.



Исторические состояния

Историческое состояние (history state) - псевдосостояние, используемое для запоминания того из последовательных подсостояний, которое было текущим в момент выхода из составного состояния.

Историческое состояние применяется только в контексте составного состояния. При этом существует две разновидности исторического состояния: неглубокое или недавнее и глубокое или давнее.



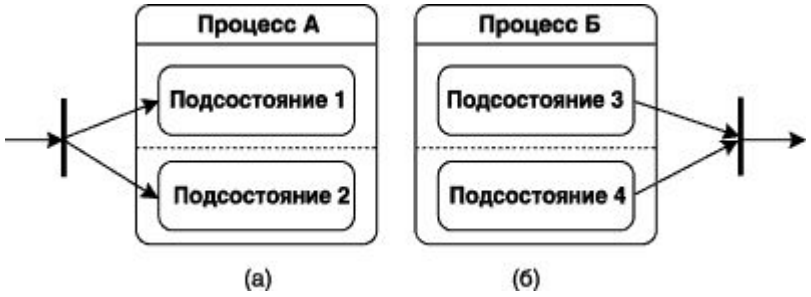
Неглубокое историческое состояние (shallow history state) обозначается в форме небольшой окружности, в которую помещена латинская буква “H”. Историческое состояние теряет свою историю в тот момент, когда конечный подавтомат доходит до своего конечного состояния. При этом неглубокое историческое состояние запоминает историю только того конечного подавтомата, к которому оно относится.

Глубокое историческое состояние (deep history state) также обозначается в форме небольшой окружности, в которую помещена латинская буква “H” с символом “*”, и служит для запоминания всех подсостояний любого уровня вложенности для исходного составного состояния.

Параллельный переход

В отдельных случаях возникает необходимость явно показать ситуацию, когда переход может иметь несколько исходных состояний или целевых состояний. Введение в рассмотрение параллельных переходов может быть обусловлено необходимостью синхронизировать отдельные процессы управления на параллельные нити без спецификации дополнительной синхронизации в параллельных конечных подавтоматах.

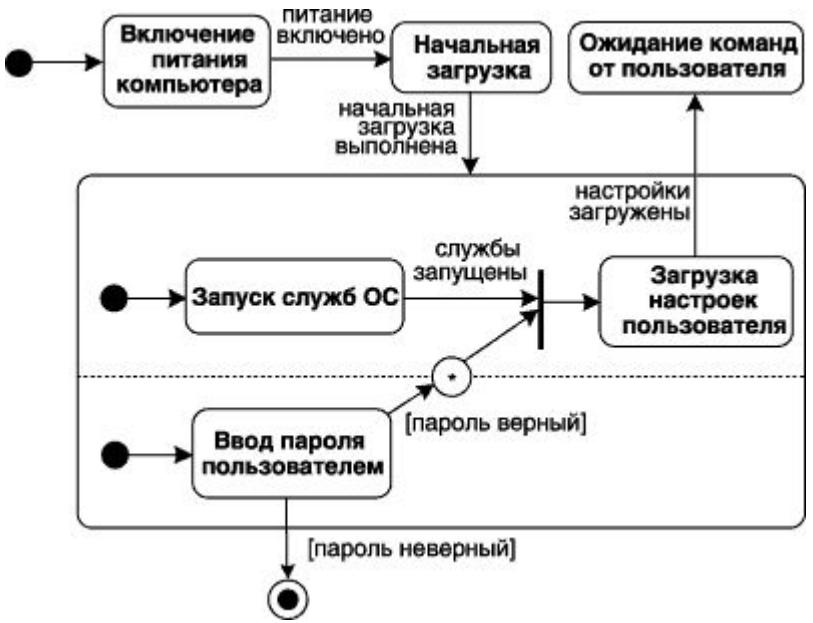
Графически такой переход изображается вертикальной черточкой, аналогично обозначению перехода в известном формализме сетей Петри. Если параллельный переход имеет две или более исходящих из него дуг, то его называют разделением (fork). Если же он имеет две или более входящие дуги, то его называют слиянием (join).



Состояние синхронизации

В общем случае поведение параллельных конечных подавтоматов происходит независимо друг от друга, что позволяет, например, моделировать многозадачность в программных системах. Однако в отдельных ситуациях может возникнуть необходимость учесть в модели синхронизацию наступления отдельных событий и срабатывание соответствующих переходов. Для этой цели в языке UML имеется псевдосостояние, которое называется синхронизирующим состоянием или состоянием синхронизации.

Состояние синхронизации (synch state) - псевдосостояние в конечном автомате, которое используется для синхронизации параллельных областей конечного автомата. Синхронизирующее состояние обозначается небольшой окружностью, внутри которой помещен символ звездочки “*”.



Задача

Обычные алгебраические выражения можно записывать также в обратной польской нотации - записи без скобок (предложил польский математик Ян Лукашевич). Например, A+(B-C)*D-F/(G+H) преобразуется в ABC-D*+FGH+/- или (A+B)*C-D+E/F/(G+H) в AB+C*D-EF/GH+/+. Требования:

- в выражении используются только натуральные числа;
- выражение заканчивается знаком "=".
- допустимые операторы и их приоритет:

Оператор	Приоритет
*, /	3
+, -	2
()	1
=	0

Разработать диаграмму состояний для решения задачи.