

Тема: Моделирование состояний

Цель: Изучить основы моделирования состояний в объектно-ориентированном подходе.

План:

1. Моделирование событий
2. Моделирование состояний
3. Переходы и условия
4. Диаграммы состояний
5. Поведение на диаграммах состояний

Литература:

1. Заботина, Н. Н. Методы и средства проектирования информационных систем: учебное пособие / Н.Н. Заботина. — Москва: ИНФРА-М, 2020. — 175-170. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-16-104187-1. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1043093>. С.
2. Смирнова Г.Н. Проектирование экономических информационных систем: учебник / Г.Н. Смирнова, А.А. Сорокин, Ю.Ф. Тельнов. М., 2003. С. 351 – 373.
3. Грекул В.И. Проектирование информационных систем: учеб. пособие / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. М., 2008. С. 196 – 209.
4. Хорев П.Б. Технологии объектно-ориентированного программирования: учеб. пособие для студ. высш. учеб. заведений/П.Б. Хорев.-2-е изд.-М.: ИЦ "Академия", 2008.-448с.
5. Рамбо Дж. UML 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо, М. Блаха. СПб.: Питер, 2007. С. 116-156.

1. Моделирование событий

Модель состояний описывает последовательности операции, происходящих в системе в ответ на внешние воздействия. Модель состояний состоит из нескольких диаграмм состояний, по одной на каждый класс, поведение которого во времени важно для приложения.

Диаграммы состояний используются для описания поведения сложных систем. Они определяют все возможные состояния, в которых может находиться объект, а также процесс смены состояний объекта в результате некоторых событий. Диаграмма состояний – это стандартная концепция из информатики (графическое представление конечного автомата), связывающая события и состояния. События представляют внешние воздействия, а состояния – значения объектов. [2, 7]

Событие – это происшествие, случившееся в определенный момент времени. Например, *нажатие клавиши левой кнопки мыши, вылет рейса 123 из Москвы.*

Неявным атрибутом события является момент его осуществления. Продолжительные изменения, осуществляющиеся в течение некоторого промежутка времени, хорошо описываются с помощью концепции состояния.

Параллельными событиями называются несвязанные события.

События бывают разных видов. Чаще всего встречаются события сигналов, события изменения и события времени.

Событие сигнала

Сигнал – это явная односторонняя передача информации от одного объекта к другому.

Событие сигнала – это событие получения или отправки сигнала.

Каждая передача сигнала является уникальным происшествием, но мы группируем их в классы сигналов и даем каждому классу имя, подчеркивая общую структуру и поведение. Например, *вылет рейса 123 из Москвы 9.11.09* – это экземпляр класса сигналов *ВылетРейса*.

Некоторые сигналы являются обычными происшествиями, но большинство из них характеризуется атрибутами, в которых хранятся передаваемые этими сигналами значения.

Например, класс *ВылетРейса* обладает атрибутами: авиакомпания, номерРейса, город, дата.

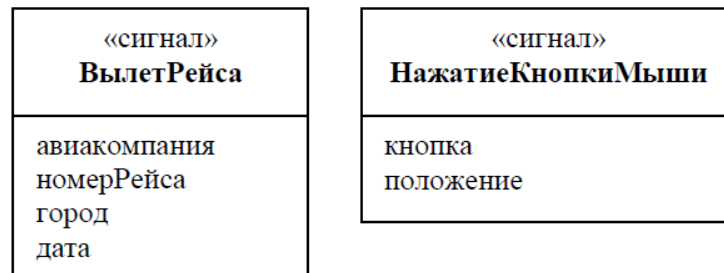


Рис. 1. Классы сигналов и их атрибуты

В UML сигнал обозначается ключевым словом «сигнал», который ставится над именем сигнала в верхнем разделе прямоугольника.

События изменения

Событие изменения – это событие, вызванное выполнением логического выражения. Суть события состоит в том, что некоторое выражение постоянно проверяется, и как только его значение изменяется с «ложно» на «истинно», осуществляется событие изменения.

В UML событие изменения обозначается словом *when*, за которым следует логическое выражение в круглых скобках [7].

when заряд батареи < нижнее значение

when давление в шинах < минимальное значение

События времени

Событие времени – это событие, вызванное достижением момента абсолютного времени или истечением временного интервала. В UML момент абсолютного времени обозначается ключевым словом *when*, за которым следует временное выражение в круглых

скобках. Временной интервал обозначается ключевым словом after, за которым следует выражение, результатом вычисления которого является временной интервал.

when дата = 1 января 2010 г.

after 10 секунд

2. Моделирование состояний

Состояние – это абстракция значений и связей объекта. Множества значений и связей группируется в соответствии с массовым поведением объектов.

Например, состояние банка может быть либо «платежеспособен», либо «банкрот» в зависимости от того, что больше активы или обязательства.

В UML состояние обозначается прямоугольником со скругленными углами, в котором указывается название состояния.



Рис. 2. Состояния

Объекты класса обладают конечным числом возможных состояний. В конкретный момент времени каждый объект может находиться ровно в одном состоянии. В конкретный момент времени разные объекты класса могут охватывать широкий спектр состояний.

Состояние описывает отклик объекта на получаемые события. В конкретном состоянии игнорируются любые события за исключением тех, поведение при получении которых описано явным образом. Отклик на событие может быть вызовом поведения или изменением состояния.

Например, если в состоянии *Гудок* нажать на кнопку с цифрой на телефоне, телефонная линия переходит в состояние *НорборНомера*. Если же в состоянии *Гудок* повесить трубку, линия отключается и переходит в состояние *Свободно*.

Способ описания состояния

<p><i>Состояние:</i> ЗвонокБудильника <i>Описание:</i> Будильник звонит, чтобы указать момент времени <i>Событие, приводящее к данному состоянию:</i> установкаБудильника произвольные действия, не приводящие к его сбрасыванию when текущее время = заданному времени <i>Условие, характеризующее данное состояние:</i> будильник = включен, если в будильнике установлено заданное время, такое, что заданное время < текущее время < заданное время + 20 сек. <i>События, возможные в данном состоянии:</i></p>		
<i>Событие</i>	<i>Отклик</i>	<i>Следующее состояние</i>

when тек. время = зад. вр. + 20 с.	сбросБудильника	нормальное
нажата кнопка	сбросБудильника	нормальное

3. Переходы и условия

Переход – мгновенная смена одного состояния другим. Переход запускается при смене исходного состояния целевым, когда происходит связанное с ним событие.

Сторожевое условие – это логическое выражение, которое должно быть истинным, чтобы переход мог запуститься.

Например, на рис. 3 показаны переходы со сторожевыми условиями для светофоров на перекрестке.

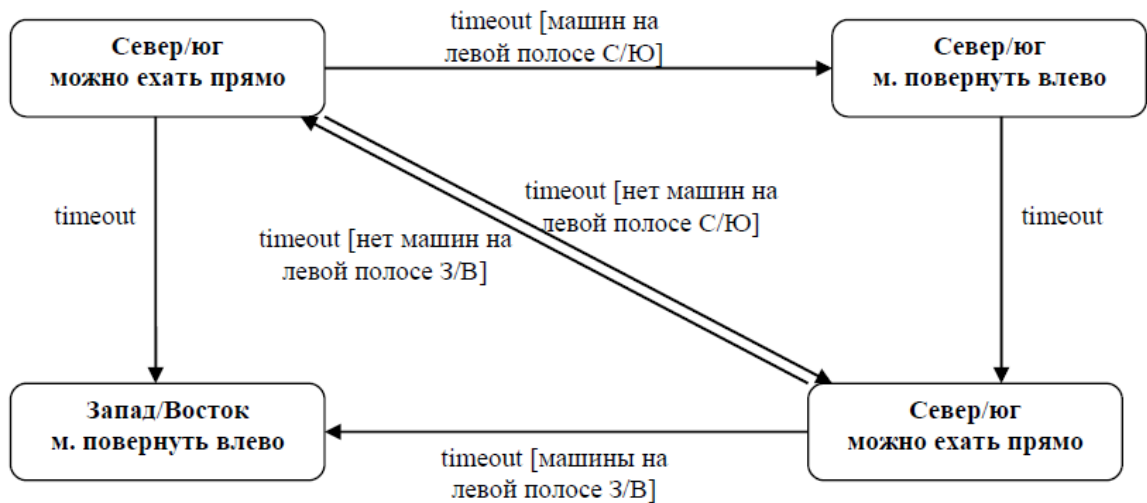


Рис. 3. Переходы со сторожевыми условиями [5]

4. Диаграммы состояний

Диаграмма состояний – это граф, узлами которого являются состояния, а направленными дугами – переходы между состояниями.

Название состояния должно быть уникальным в рамках диаграммы.

Модель состояний состоит из множества диаграмм состояний, по одной на каждый класс, поведение которого с течением времени важно для приложения.

Диаграммы состояний должны быть согласованы по интерфейсам (событиям и сторожевым условиям).

Класс, имеющий несколько состояний, характеризуется важным поведением во времени. Если же класс обладает одним состоянием, его поведение во времени можно игнорировать. Диаграммы состояний с одним состоянием можно описать в простой форме без всякой графики, а именно в виде таблицы воздействий и откликов, в которой будут приводиться события и сторожевые условия, а также вызываемое ими поведение.

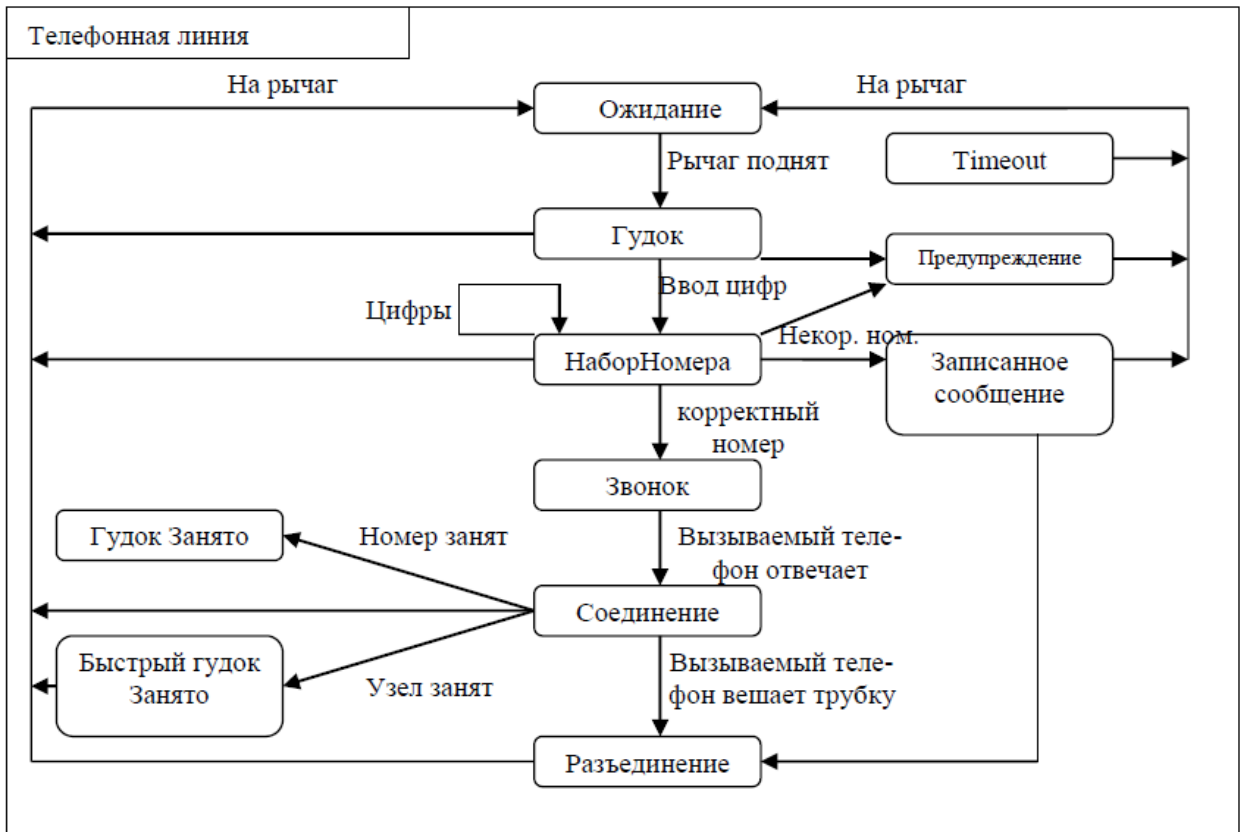


Рис. 4. Диаграмма состояний телефонной линии [5]

Одноразовые диаграммы состояний описывают объекты с конечным сроком существования. Такие диаграммы имеют начальное и конечное состояние. Сразу после создания объект оказывается в начальном состоянии. Вход в конечное состояние означает уничтожение объекта.

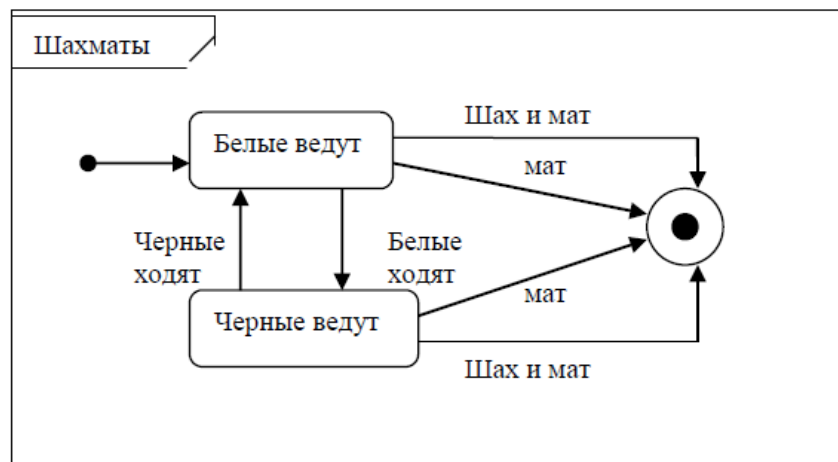


Рис. 5. Диаграмма состояний игры в шахматы

Основные обозначения для диаграммы состояний

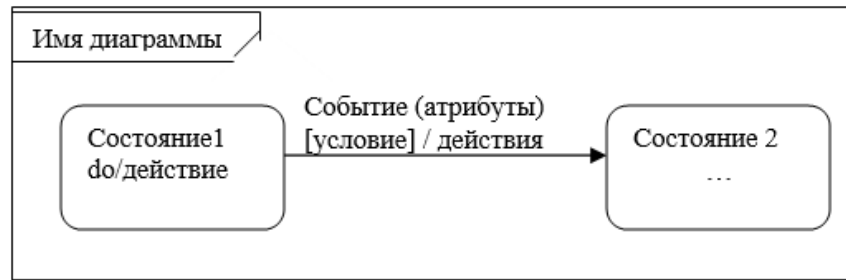


Рис. 6. [5]

Состояние обозначается прямоугольником со скругленными углами, в котором может быть указано имя состояния. Начальное состояние обозначается сплошным кружком, конечное – «бычьим глазом».

Переход изображается линией, соединяющей исходное состояние с целевым.

Событие сигнала изображается меткой на переходе. После названия события в круглых скобках можно указать атрибуты. События изменения обозначаются ключевым словом *when*, после которого в круглых скобках указывается логическое выражение. Событие времени также указывается ключевым словом *when*, после которого в круглых скобках указывается временное выражение, или ключевое слово *after*, после которого в круглых скобках указывается интервал времени.

Диаграмма состояний заключается в прямоугольную рамку. Название диаграммы указывается в небольшом прямоугольнике в левом верхнем углу рамки.

Сторожевые условия могут быть указаны в квадратных скобках после события.

Действия могут прикрепляться к переходу или состоянию. Указываются после символа «/».

5. Поведение на диаграммах состояний

Диаграммы состояний были бы не слишком полезны, если бы они описывали только события. Полное описание объекта должно указывать, что именно делает объект в ответ на событие.

Действие – это ссылка на поведение, выполняемое в ответ на произошедшее событие.

Деятельность – это фактическое поведение, которое может вызываться любым количеством действий. Например, деятельность *разъединить линию* может выполняться в ответ на событие на *рычаг*. Деятельность обозначается «/», после которой ставится название или описание деятельности. Деятельность ставится после вызывающего его события.

Ключевое слово *do* используется для обозначения текущей деятельности и не может использоваться в качестве имени события.

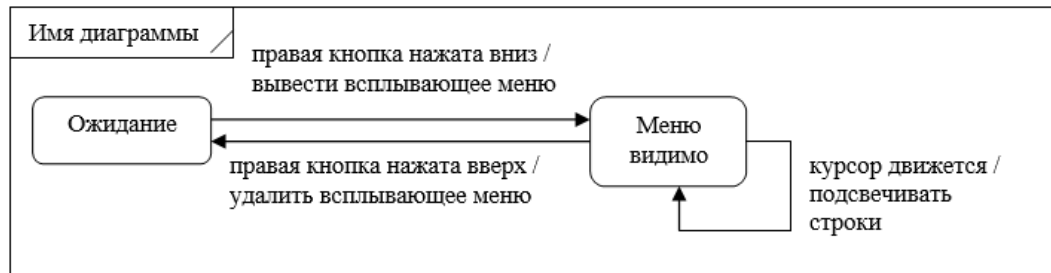


Рис. 7. Деятельность для всплывающего меню [5]

Текущей называется деятельность, занимающая некоторый промежуток времени. По определению такая деятельность может выполняться только в некотором состоянии и не может прикрепляться к переходу.

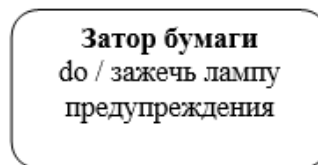


Рис. 8. Текущая деятельность для ксерокса

Вопросы для самопроверки:

1. В чем заключается назначение модели состояний?
2. Что такое событие?
3. Перечислите виды событий. Укажите их обозначение на UML.
4. Что такое состояние? переход? сторожевое условие?