

Введение в анализ ИС

Лекция 5

Составитель: Эверстов В.В.

Дата составления: 29.09.2010

Дата модификации: 30.09.2011

События

- Событие – это происшествие, случившееся в определенный момент времени, например нажатие пользователем левой кнопки мыши и т.п.
- По определению событие происходит мгновенно, по крайней мере во временном масштабе приложения.
- Событие это происшествие, которое рассматривается как атомарное и скоротечное.

События

- Одно событие может логически предшествовать другому или следовать за ним. События могут быть и несвязанными друг с другом.
- Если временная задержка при передаче информации между двумя точками превышает временной интервал между событиями, эти события должны быть параллельными, поскольку не могут влиять друг на друга.

События

- К событиям относятся не только нормальные происшествия, но и ошибочные ситуации.
- Ошибочная ситуация ничем не отличается от любого другого события. Только в нашей интерпретации она становится ошибкой.

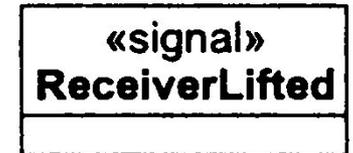
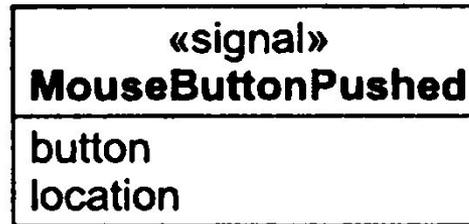
Сигнал

- Сигнал – это явная односторонняя передача информации от одного объекта другому. Объект передающий сигнал другому объекту, может рассчитывать на получение ответа, но этот ответ будет отдельным сигналом, и его отправка (или задержка) будет целиком зависеть от второго объекта.

Сигнал

- Событие сигнала – это событие получения или отправки сигнала. Обычно более важным считается получение сигнала, т.к. оно влияет на объект-получатель.
- Каждая передача является уникальным событием, но в UML их группируют в классы сигналов и именуют подчеркивая общую структуру и поведение.

Классы сигналов



Событие изменения

- Событие изменения – это событие, вызванное выполнением логического выражения. Суть события состоит в том, что его значение изменяется с «ложно» на «истинно», осуществляется событие изменения.
- В UML событие изменения обозначается ключевым словом `when`, за которым следует логическое выражение в круглых скобках.

Пример событий изменения

- When(температура в комнате < установка нагрева)
- When(температура в комнате > установка охлаждения)
- When(заряд батареи < нижнее ограничение)
- When(давление в шинах < минимальное давление)

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 значение в круглых скобках.
- Временной интервал обозначается ключевым словом `after`, за которым следует выражение, результатом вычисления которого является временным интервалом.

Пример

- when (дата = 1 января 2000 г.)
- after (10 секунд)

Состояния

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

Состояние

Solvent

Insolvent

Waiting

Dialing

Powered

BelowFreezing

- На рисунке обозначена система обозначений состояния в UML. Прямоугольник с округленными углами, в котором ставится необязательное название состояния. Название выделяем полужирным шрифтом, центрируем его и пишем с заглавной буквы.

Состояния

- Абстрактно события и состояния можно изобразить следующим образом: события это точки на линии времени, а состояния – это интервалы между двумя событиями.



Пример описания

Состояние: *AlarmRinging*

Описание: alarm on watch is ringing to indicate target time

Событие, приводящее к данному состоянию:

setAlarm (targetTime)

any sequence not including *clearAlarm*

when (*currentTime = targetTime*)

Условие, характеризующее данное состояние:

alarm = on, alarm set to *targetTime*, *targetTime* \leq *currentTime* \leq

targetTime + 20 seconds, and no button has been pushed since *targetTime*

События, возможные в данном состоянии:

Событие	Отклик	Следующее состояние
when (<i>currentTime = targetTime</i> + 20)	<i>resetAlarm</i>	<i>normal</i>
<i>buttonPushed</i> (any button)	<i>resetAlarm</i>	<i>normal</i>

Переход

- Это мгновенная смена одного состояния другим. Переход запускается (осуществляется) при смене исходного состояния целевым.
- Переход осуществляется когда происходит связанное с ним событие. Выбор целевого состояния зависит от исходного состояния, так и от полученного события. Событие может вызвать переходы на множество объектов.

Сторожевое условие

- Это логическое выражение, которое должно быть истинным, чтобы переход мог запуститься. Переход со сторожевым условием запускается в тот момент, когда осуществляется соответствующее событие, но только если в этот момент выполнено его сторожевое условие.

Обозначение переходов

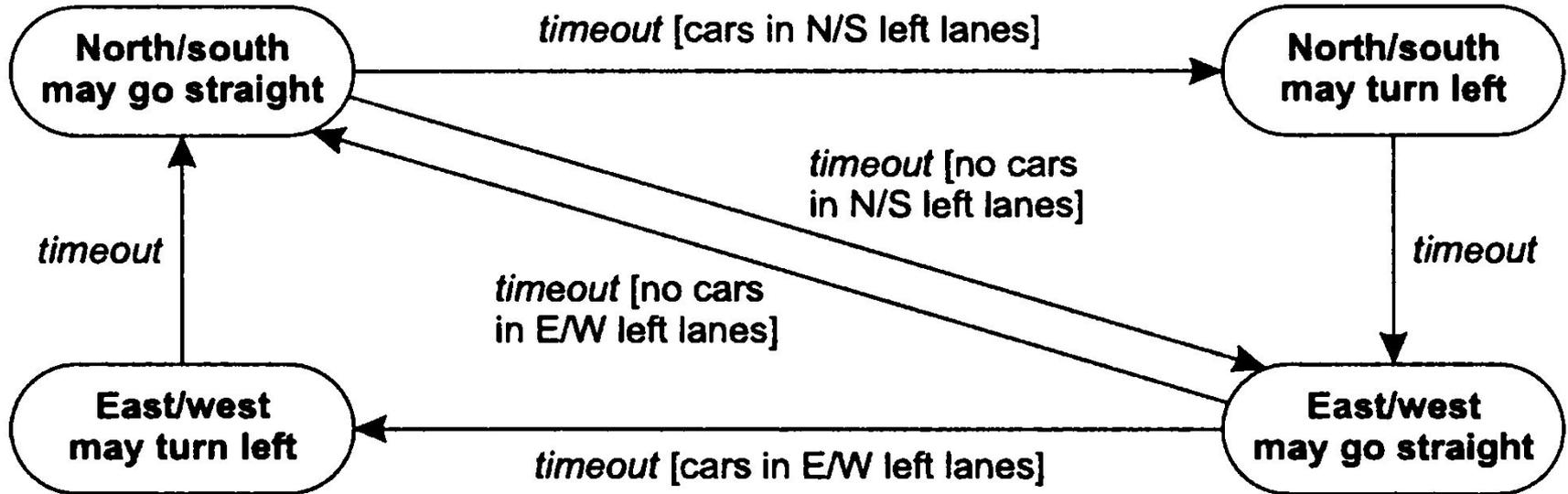


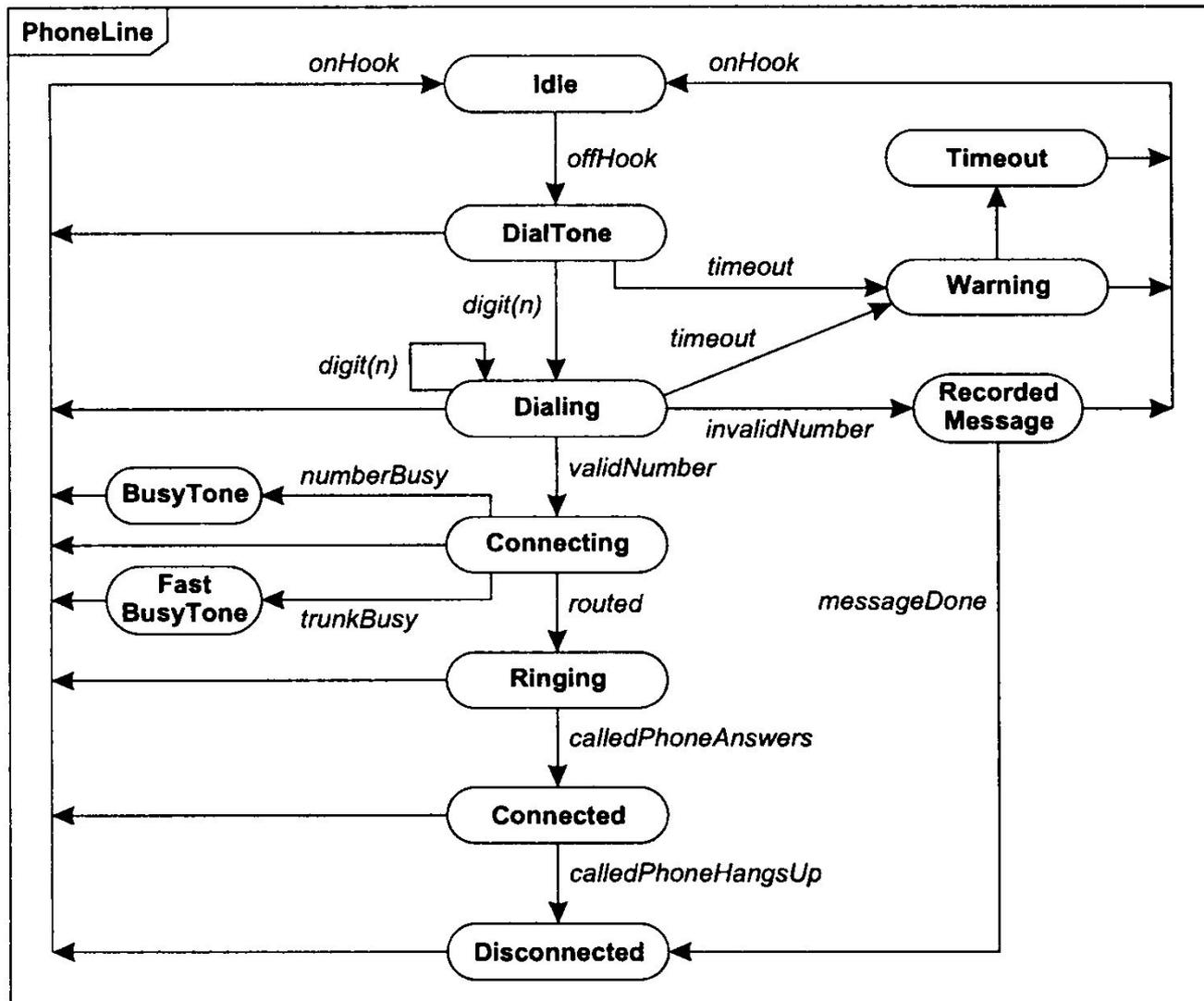
Диаграмма состояний

- Это граф, узлами которого являются состояния, а направленными дугами – переходы между состояниями. Диаграмма состояний описывает последовательности состояний, вызываемые последовательностями событий. Названия состояний должны быть уникальными в рамках диаграммы.

Модель состояний

- 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). Отдельные диаграммы взаимодействуют друг с другом посредством передачи событий, а так же косвенно, через сторожевые условия. Некоторые события и сторожевые условия присутствуют только на одной диаграмме, тогда как другие на нескольких.

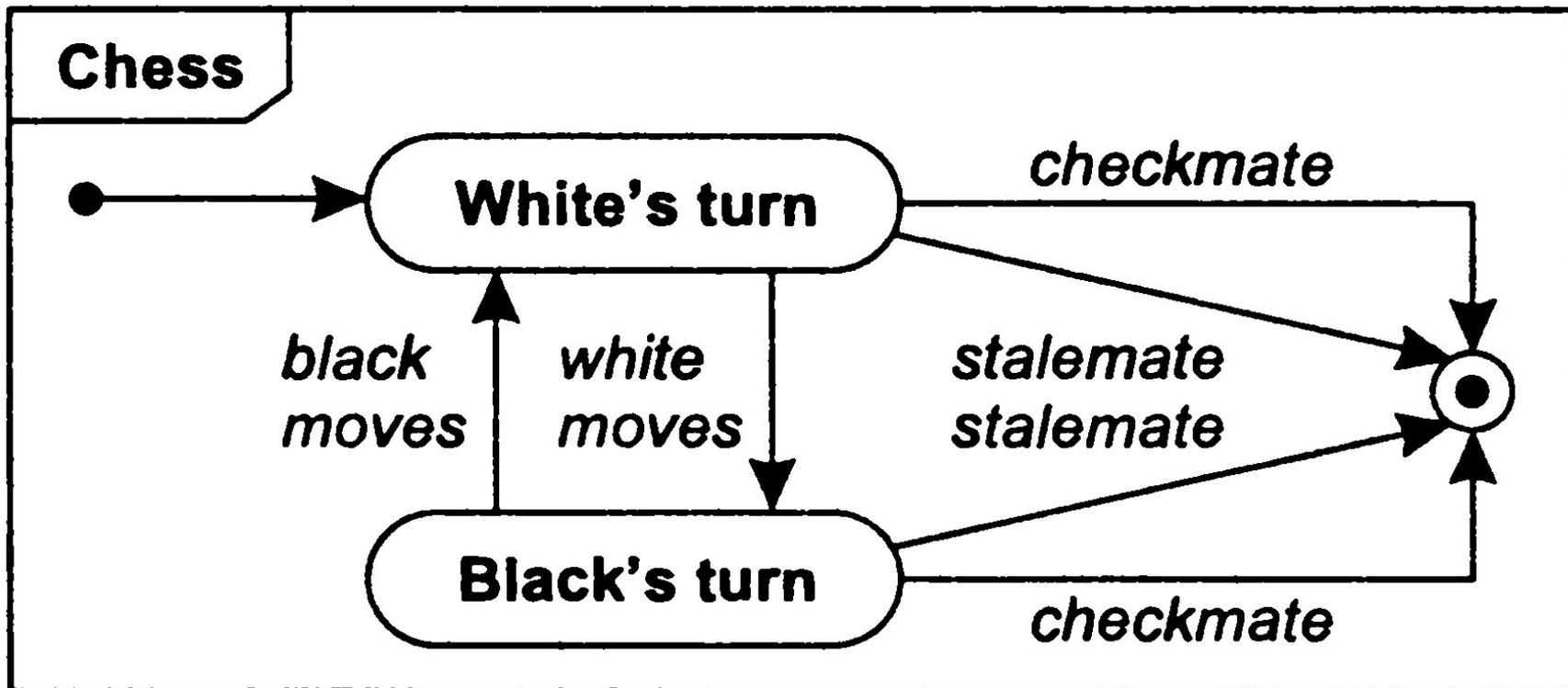
Пример диаграммы состояний



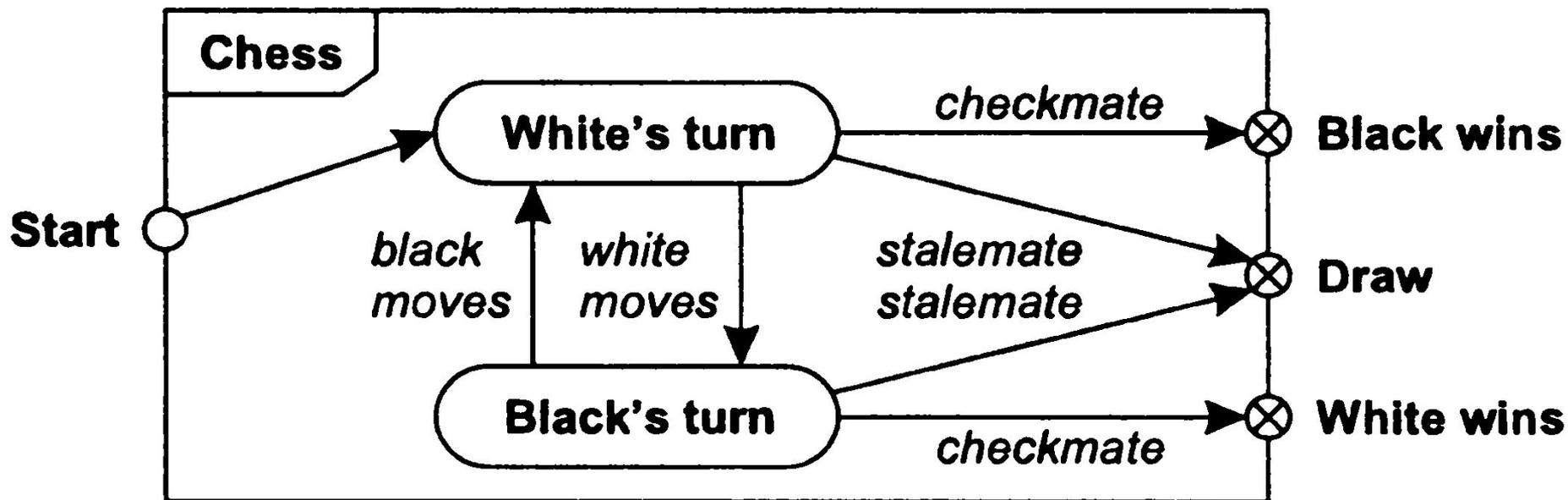
Одноразовые диаграммы состояний

- Диаграммы состояний могут описывать непрерывные циклы или одноразовые жизненные циклы.
- Одноразовые диаграммы состояний описывают объекты с конечным сроком существования. Такие диаграммы имеют начальное и конечное состояния. Сразу после объект оказывается в начальном состоянии. Вход в конечное состояние означает уничтожение объекта.

Пример



Пример



Действия и деятельность

- Действие – это ссылка на поведение выполняемое в ответ на произошедшее событие.
- Деятельность – это фактическое поведение, которое может вызваться любым количеством действий.

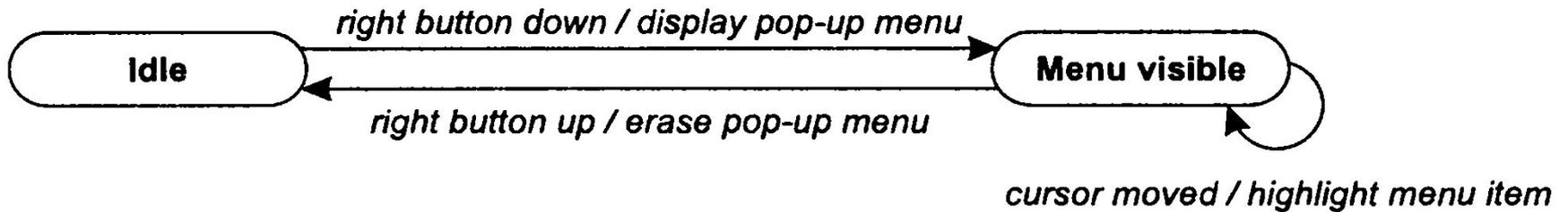
Деятельность

- Деятельность может выполняться при переходе, при входе в состояние или при выходе из него, а также при наступлении какого-либо иного события в состоянии.
- Деятельность может описывать внутренние управляющие операции, например установку атрибутов или порождение других событий. Эта деятельность не имеет аналогов в реальном мире и предназначена для структурирования при реализации. Например, программа может увеличивать внутренний счетчик на единицу каждый раз при осуществлении какого-либо события.

Деятельность

- Деятельность обозначается косой чертой (/), после которой ставится название или описание деятельности. Ключевое слово **do** используется для обозначения текущей деятельности и не может использоваться в качестве имени события.

Деятельность



Текущая деятельность

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

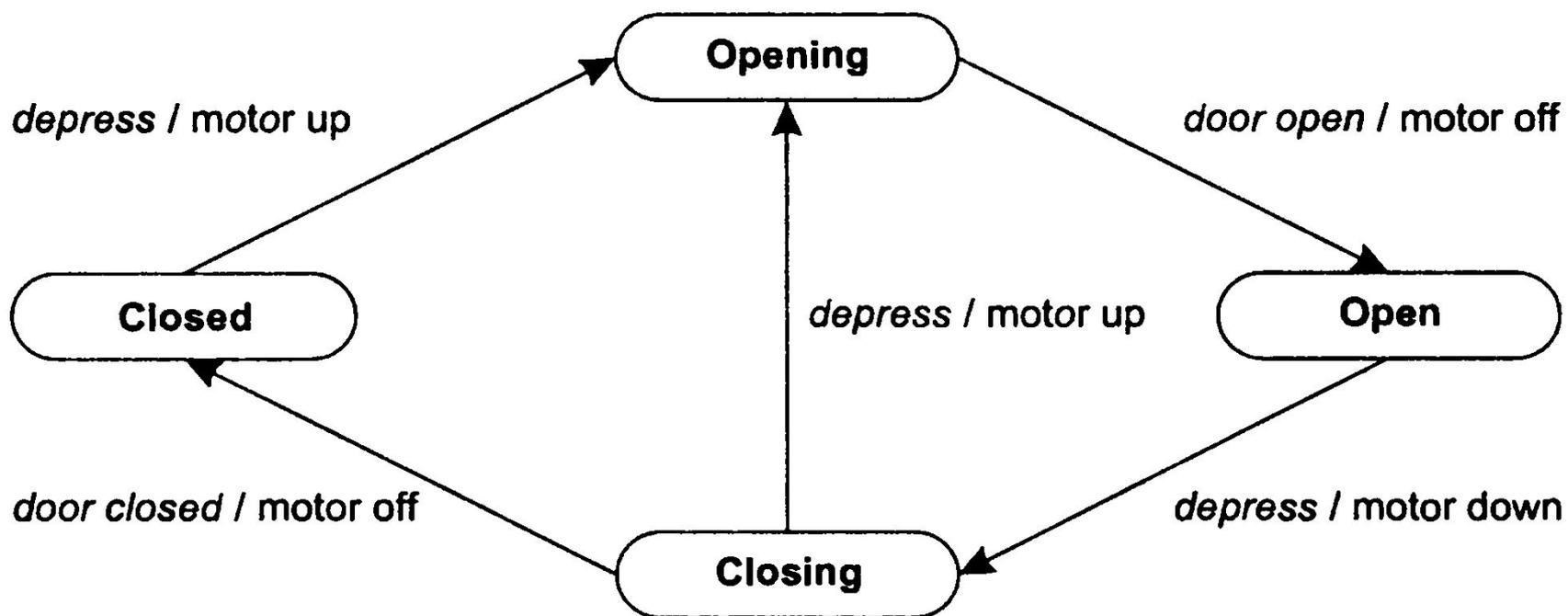
Текущая деятельность

Paper jam
do / flash warning light

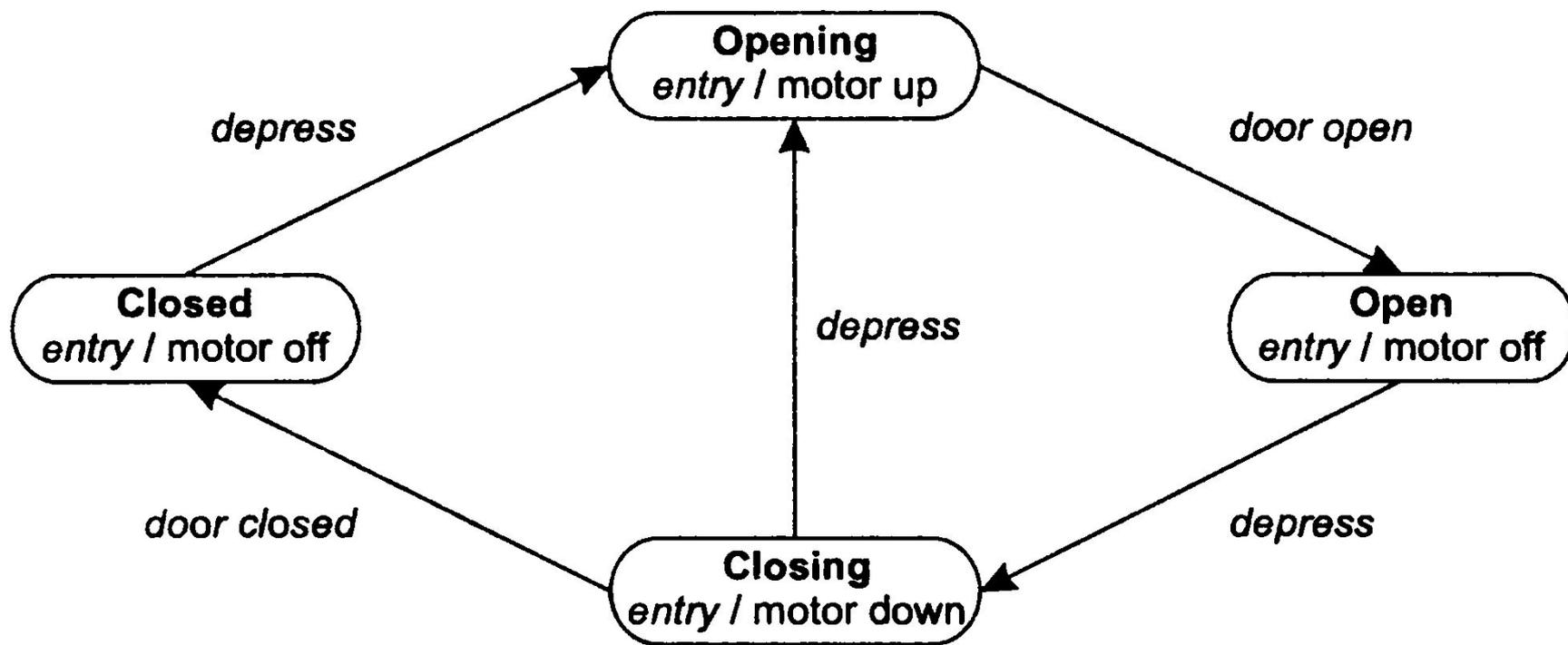
Деятельность

- Может быть прикреплена не только к переходу, но и ко входу в состояние или к выходу из него. Никаких отличий в возможностях двух систем обозначений нет, однако часто при всех переходах в одно и то же состояние выполняется одинаковая деятельность, которую, в таком случае, удобнее привязывать к самому событию.

Деятельность



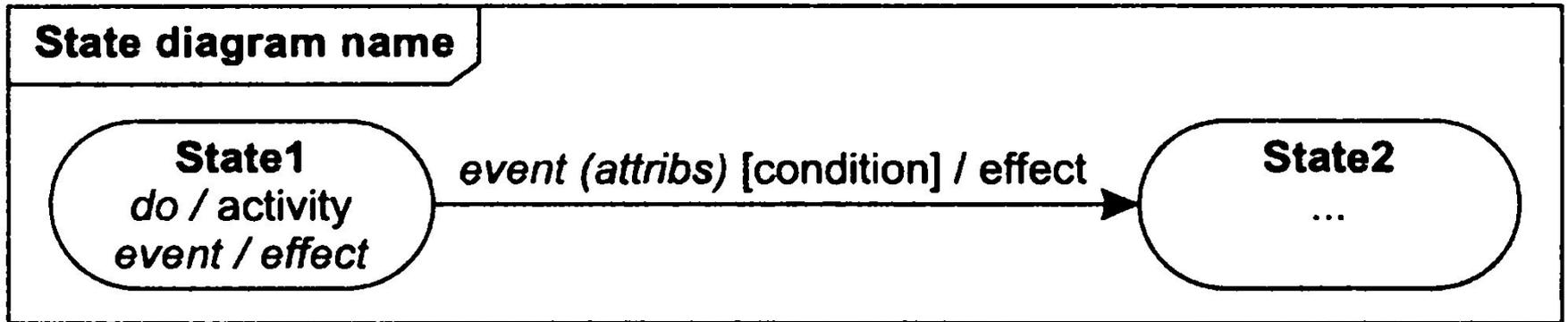
Деятельность



Отправка сигнала

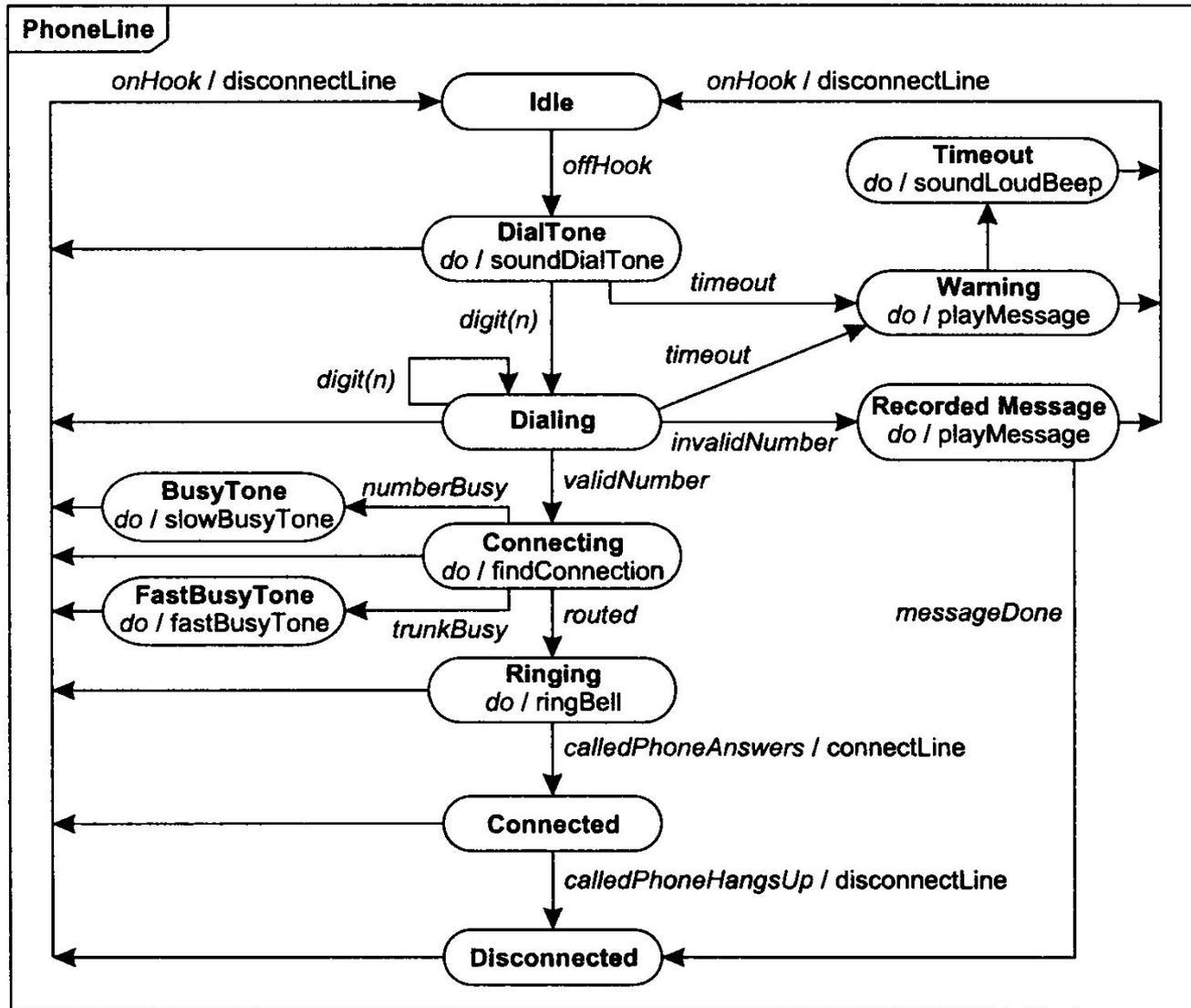
- Объект может выполнять деятельность, заключающуюся в отправке сигнала другому объекту. Система объектов взаимодействует обмениваясь сигналами.
- Деятельность `send target.S(attributes)` передает сигнал `S` с заданными атрибутами целевому объекту (или группе объектов) `target`.

Основные обозначения



- **Состояние.** Обозначается прямоугольником со скругленными углами, в котором может быть указано имя состояния. Для начальных и конечных состояний имеются специальные обозначения.
- **Переход.** Изображается линией, соединяющей исходное состояние с целевым. Стрелка ставится около конца линии указывающего на целевое состояние. Линия может состоять из нескольких сегментов, т.е. быть ломаной.
- **Событие.** Событие сигнала изображается меткой на переходе. После названия события в круглых скобках можно указать атрибуты. Событие изменения обозначается ключевым словом *when*, после которого в круглых скобках указывается логическое выражение. Событие времени также обозначается ключевым словом *when*, после которого в круглых скобках указывается временное выражение или ключевым словом *after*, после которого в круглых скобках указывается интервал времени.
- **Диаграмма состояний.** Заключается в прямоугольную рамку. Название диаграммы указывается в небольшом пятиугольном теге в верхнем левом углу рамки.
- **Сторожевое условие.** Может быть указано в квадратных скобках после события.
- **Действия.** Могут прикрепляться к переходу или состоянию. Указываются после символа «/». Действия если их несколько отделяются друг от друга запятыми. И выполняются параллельно. (Если действия необходимо выполнить последовательно их нужно разделить промежуточными состояниями.)

Пример



Модель состояний

- Некоторые объекты предметной области за время своего жизненного цикла сменяют несколько качественно различных состояний. В этих состояниях они могут иметь разные ограничения на значения атрибутов, разные ассоциации или кратности, выполнять различные операции или иметь разное поведение и т.д. Часто бывает полезно построить диаграммы состояний для таких классов.
- Большинство классов предметной области не требуют использования диаграмм состояний. Модель состояний может помочь в понимании поведения тех классов, которые могут находиться в существенно разных состояниях.

Модель состояний

- Построение модели состояний может быть разбито на несколько этапов:
 - Выявление классов, обладающих разными состояниями,
 - Выделение состояний классов,
 - Выделение событий,
 - Построение диаграмм состояний,
 - Проверка диаграмм состояний.

Выявление классов

- Изучите список классов предметной области, характеризующихся четко определенным жизненным циклом. Ищите классы, которые развиваются или имеют циклическое поведение. Идентифицируйте значимые состояния в жизненном цикле каждого из объектов.

Выделение состояний

- Перечислите состояния для каждого из классов из предыдущего этапа.
Охарактеризуйте объекты в каждом состоянии: укажите значения атрибутов, которые могут иметь объекты, ассоциации.
Дайте каждому состоянию осмысленное название. Название не должно описывать как объект перешел в это состояние, оно должно обозначать самое состояние.
- Рассмотрев события и переходы между состояниями вы можете найти недостающие состояния.

Выделение событий

- Получив предварительный список состояний, займитесь поиском событий, которые вызывают переходы между этими состояниями. Подумайте о внешних воздействиях, которые вызывают изменения состояний. Во многих ситуациях событие может быть рассмотрено как завершение текущей деятельности. Завершение деятельности может вызывать альтернативные переходы, которые могут добавляться в процессе совершенствования модели.

Выделение событий

- Другие события можно обнаружить, поразмыслив о том каким образом объект может попасть в определенное состояние.
- Внутри состояний могут происходить и события, не вызывающие переходов. Для модели состояний предметной области важны только те события, которые вызывают переходы.
- Информацию, содержащуюся в событии, следует в форме списка его параметров.

Диаграмма состояний

- Распределите события по состояниям, к которым они относятся. Добавьте переходы. Если событие инициирует целевое состояние, вы должны рассмотреть в каких состояниях это событие может происходить и добавить в диаграмму переходы из этих состояний в целевое.
- Обычно деятельность не слишком важна для диаграммы состояний классов предметной области. Если же объекты классов выполняют деятельность при переходах, добавьте эту деятельность на диаграмму.

Проверка

- Проверьте все диаграммы состояний. Все ли состояния достижимы? Особенное внимание нужно уделить маршрутам по диаграмме.
 - Если она описывает класс с развивающимся поведением, есть ли на диаграмме маршрут, соединяющий начальное состояние с конечным? Присутствуют ли на ней ожидаемые отклонения от основной последовательности?
 - Если диаграмма описывает класс с циклическим поведением, есть ли на ней основная петля цикла? Есть ли тупиковые состояния, завершающие цикл?
- Ищите недостающие маршруты на основании своих знаний о предметной области. Иногда отсутствие маршрута указывает на отсутствие нужных состояний.