

Лабораторная работа № 2.

Диаграмма деятельности

Цель работы: знакомство с основными понятиями построения диаграммы деятельности; изучение графических конструкций применяемых для построения модели; построение модели деятельности системы.

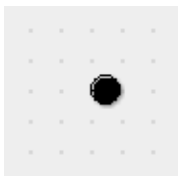
Теоретические сведения

Для моделирования процесса выполнения операций в языке UML используются диаграммы деятельности. Основным направлением использования диаграмм деятельности является визуализация особенностей реализации операций, когда необходимо представить алгоритмы их выполнения.

В контексте языка UML деятельность (activity) представляет собой совокупность отдельных вычислений, выполняемых автоматом, приводящих к некоторому результату или действию (action). На диаграмме деятельности отображается логика и последовательность переходов от одной деятельности к другой, а внимание аналитика фокусируется на результатах. Результат деятельности может привести к изменению состояния системы или возвращению некоторого значения.

Начальное состояние

Начальное состояние представляет собой частный случай состояния, которое не содержит никаких внутренних действий (псевдосостояние). В этом состоянии находится объект по умолчанию в начальный момент времени. Оно служит для указания на диаграмме графической области, от которой начинается процесс изменения состояний. Графически начальное состояние в языке UML обозначается в виде закрашенного кружка, из которого может только выходить стрелка, соответствующая переходу.



На самом верхнем уровне представления объекта переход из начального состояния может быть помечен событием создания (инициализации) данного объекта. В противном случае переход никак не помечается. Если этот переход не помечен, то он является первым переходом в следующее за ним состояние.

Конечное состояние

Конечное состояние представляет собой частный случай состояния, которое также не содержит никаких внутренних действий (псевдосостояние). В этом состоянии будет находиться объект по умолчанию после завершения работы автомата в конечный момент времени.



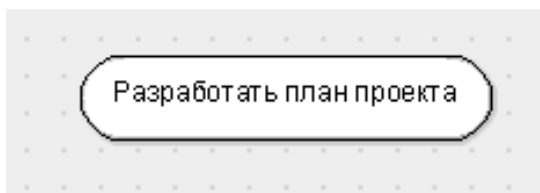
Оно служит для указания на диаграмме графической области, в которой завершается процесс изменения состояний или жизненный цикл данного объекта. Графически конечное состояние в языке UML обозначается в виде закрашенного кружка, помещенного в окружность, которую может только входить стрелка, соответствующая переходу.

Состояние действия

Состояние действия (action state) является специальным случаем состояния с некоторым входным действием и, по крайней мере, одним выходящим из состояния переходом. Этот переход неявно предполагает, что входное действие уже завершилось. Состояние действия не может иметь внутренних переходов.

дов, поскольку оно является элементарным. Обычное использование состояния действия заключается в моделировании одного шага выполнения алгоритма (процедуры) или потока управления.

Графически состояние действия изображается прямоугольником с закругленными углами. Внутри этого изображения записывается выражение действия (action-expression), которое должно быть уникальным в пределах одной диаграммы деятельности.



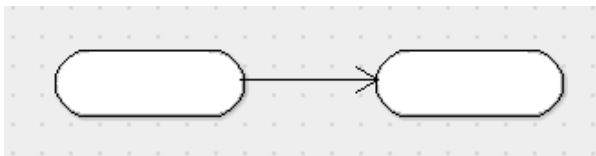
Действие может быть записано на естественном языке, некотором псевдокоде или языке программирования. Никаких дополнительных или неявных ограничений при записи действий не накладывается. Рекомендуется в качестве имени простого действия использовать глагол с пояснительными словами. Если же действие может быть представлено в некотором формальном виде, то целесообразно записать его на том языке программирования, на котором предполагается реализовывать конкретный проект.

Переходы

Простой переход

Простой переход (simple transition) представляет собой отношение между двумя последовательными состояниями, которое указывает на факт смены одного состояния объекта другим. Если пребывание моделируемого объекта в первом состоянии сопровождается выполнением некоторых действий, то переход во второе состояние будет возможен только после завершения

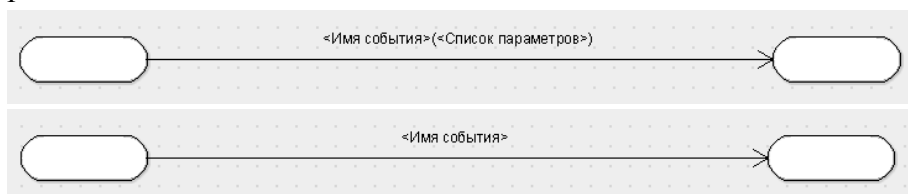
этих действий и, возможно, после выполнения некоторых дополнительных условий, называемых сторожевыми условиями.



На диаграмме состояний переход изображается сплошной линией со стрелкой, которая направлена в целевое состояние.

Событие

Событие (event) является самостоятельным элементом языка UML. Формально, событие 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. После наступления некоторого события уже нельзя вернуться к предыдущим событиям, если такая возможность не предусмотрена явно в модели.



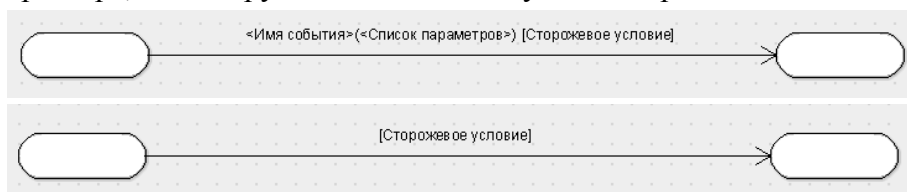
Семантика понятия события фиксирует внимание на внешних проявлениях качественных изменений, происходящих при переходе моделируемого объекта из состояния в состояние. В языке UML события играют роль стимулов, которые инициируют переходы из одних состояний в другие. В качестве событий можно рассматривать сигналы, вызовы, окончание фиксированных промежутков времени или моменты окончания выполнения определенных действий.

Имя события идентифицирует каждый отдельный переход на диаграмме состояний и может содержать строку текста, начинающуюся со строчной буквы. В этом случае принято считать переход триггерным, то есть таким, который специфицирует событие-триггер. Если рядом со стрелкой перехода не указана никакая строка текста, то соответствующий переход является нетриггерным, и в этом случае из контекста диаграммы состояний должно следовать, после окончания какой деятельности он выполняется. После имени события могут следовать круглые скобки для явного задания параметров соответствующего события-триггера. Если таких параметров нет, то список параметров со скобками может отсутствовать.

Сторожевое условие

Сторожевое условие (guard condition), если оно есть, всегда записывается в прямых скобках после события-триггера и представляет собой некоторое булевское выражение. Из контекста диаграммы состояний должна явно следовать семантика этого выражения, а для записи выражения может использоваться синтаксис языка объектных ограничений.

Введение для перехода сторожевого условия позволяет явно специфицировать семантику его срабатывания. Однако вычисление истинности сторожевого условия происходит только после возникновения ассоциированного с ним события-триггера, инициирующего соответствующий переход.



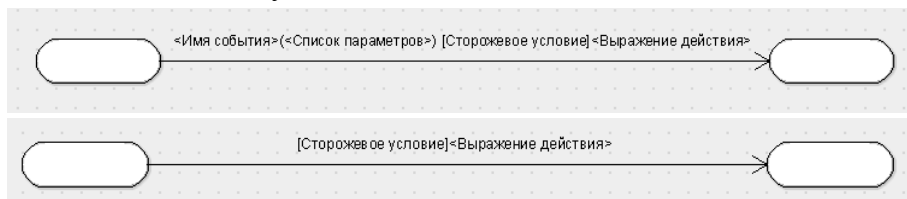
В общем случае из одного состояния может быть несколько переходов с одним и тем же событием-триггером. При этом

никакие два сторожевых условия не должны одновременно принимать значение «истина». Каждое из сторожевых условий необходимо вычислять всякий раз при наступлении соответствующего события-триггера.

Примером события-триггера может служить разрыв телефонного соединения с провайдером интернет-услуг после окончания загрузки электронной почты клиентской почтовой программой (при удаленном доступе к интернет). В этом случае сторожевое условие есть не что иное, как ответ на вопрос: «Пуст ли почтовый ящик клиента на сервере провайдера?». В случае положительного ответа – «истина», следует отключить соединение с провайдером, что и делает автоматически почтовая программа-клиент. В случае отрицательного ответа - «ложь», следует оставаться в состоянии загрузки почты и не разрывать телефонное соединение.

Выражение действия

Выражение действия (action expression) выполняется только при срабатывании перехода. Оно представляет собой атомарную операцию, выполняемую сразу после срабатывания соответствующего перехода до начала каких бы то ни было действий в целевом состоянии. Атомарность действия означает, что оно не может быть прервано никаким другим действием до тех пор, пока не закончится его выполнение. Данное действие может влиять как на сам объект, так и на его окружение, если это с очевидностью следует из контекста модели.



В общем случае выражение действия может содержать целый список отдельных действий, разделенных символом «;». Все действия списка должны различаться между собой и следовать в порядке записи. На синтаксис записи выражений действия не накладывается никаких ограничений. Основное условие, чтобы запись была понятна разработчикам модели и программистам. Как правило, выражение действия записывают на одном из языков программирования, который предполагается использовать для реализации модели.

Ветвление

Графически ветвление на диаграмме деятельности обозначается небольшим ромбом, внутри которого нет никакого текста. В этот ромб может входить только одна стрелка от того состояния действия, после выполнения которого поток управления должен быть продолжен по одной из взаимно исключающих ветвей.

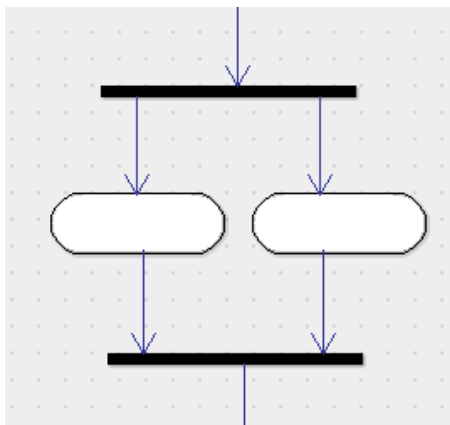


Принято входящую стрелку присоединять к верхней или левой вершине символа ветвления. Выходящих стрелок может быть две или более, но для каждой из них явно указывается соответствующее сторожевое условие в форме булевского выражения.

Параллельные процессы

Один из недостатков обычных блок-схем алгоритмов связан с проблемой изображения параллельных ветвей отдельных вычислений. Поскольку распараллеливание вычислений существенно повышает общее быстродействие программных систем, необходимы графические примитивы для представления парал-

тельных процессов. В языке UML для этой цели используется специальный символ для разделения и слияния параллельных вычислений или потоков управления. Таким символом является прямая черточка.

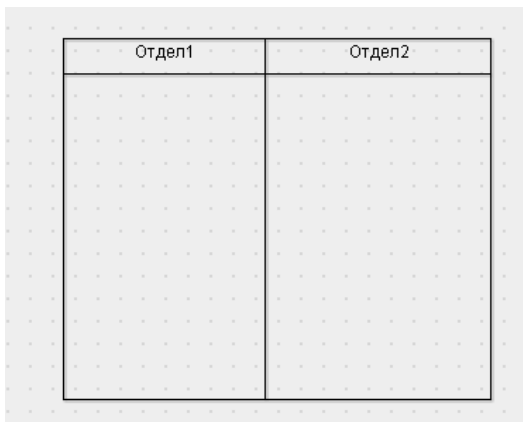


Как правило, такая черточка изображается отрезком горизонтальной линии, толщина которой несколько шире основных сплошных линий диаграммы деятельности. При этом разделение (concurrent fork) имеет один входящий переход и несколько выходящих, а слияние (concurrent join) имеет несколько входящих переходов и один выходящий.

Дорожки

Диаграммы деятельности могут быть использованы не только для спецификации алгоритмов вычислений или потоков управления в программных системах. Не менее важная область их применения связана с моделированием бизнес процессов. Действительно, деятельность любой организации также представляет собой совокупность отдельных действий, направленных на достижение требуемого результата. Однако, применительно к бизнес процессам, желательно выполнение каждого действия ассоциировать с конкретным подразделением компа-

нии. В этом случае подразделение несет ответственность за реализацию отдельных действий, а сам бизнес процесс представляется в виде переходов действий из одного подразделения к другому.



Для моделирования этих особенностей в языке UML используется специальная конструкция, получившее название дорожки (swimlanes). Имеется в виду визуальная аналогия с плавательными дорожками в бассейне, если смотреть на соответствующую диаграмму. Все состояния действия на диаграмме деятельности делятся на отдельные группы, которые отделяются друг от друга вертикальными линиями. Две соседние линии образуют дорожку, а группа состояний между этими линиями выполняется отдельным подразделением (отделом, группой, отделением, филиалом) организации.

Названия подразделений явно указываются в верхней части дорожки. Пересекать линию дорожки могут только переходы, которые, в этом случае, обозначают выход или вход потока управления в соответствующее подразделение. Порядок следования дорожек не несет какой-либо семантической информации и определяется соображениями удобства.

Объекты

В общем случае действия на диаграмме деятельности выполняются над теми или иными объектами. Эти объекты либо инициируют выполнение действий, либо определяют некоторый их результат. Действия специфицируют вызовы, которые передаются от одного объекта графа деятельности к другому. Поскольку в таком ракурсе объекты играют определенную роль в понимании процесса деятельности, иногда возникает необходимость явно указать их на диаграмме.



Для графического представления объектов используется прямоугольник класса, с тем отличием, что имя объекта подчеркивается. Далее после имени может указываться характеристика состояния объекта в прямых скобках. Такие прямоугольники объектов присоединяются к состояниям действия отношением зависимости пунктирной линией со стрелкой. Соответствующая зависимость определяет состояние конкретного объекта после выполнения предшествующего действия.

На диаграмме деятельности с дорожками расположение объекта может иметь некоторый дополнительный смысл. А именно, если объект расположен на границе двух дорожек, то это может означать, что переход к следующему состоянию действия в соседней дорожке ассоциирован с готовностью некоторого документа (объект в некотором состоянии). Если же объект целиком расположен внутри дорожки, то и состояние этого объекта целиком определяется действиями данной дорожки.

Для синхронизации отдельных действий на диаграмме деятельности никаких дополнительных обозначений не используется, поскольку синхронизация параллельных процессов может быть реализована с помощью переходов «разделение-слияние».

Содержание лабораторной работы

В лабораторной работе требуется разработать диаграмму деятельности в соответствии с Вашим вариантом. Отчет о проделанной работе (пример оформления отчета по лабораторной работе смотри приложение 1) должен содержать: название и цель работы; номер варианта и описание задания; диаграмму соответствующую варианту; необходимые комментарии к диаграмме и выводы по проделанной лабораторной работе.

Варианты заданий

Для выполнения работы варианты задания берутся из лабораторной работы №1.