

Лабораторная работа 2

Разработка моделей на основе диаграмм состояний (стейтчарты) в AnyLogic

Цель работы: научиться описывать изучаемые процессы и объекты через диаграммы состояний.

Если у активного объекта можно выделить несколько состояний, выполняющих различные действия при происхождении каких-то событий, или если у активного объекта есть несколько качественно различных поведений, последовательно сменяющих друг друга при происхождении определенных событий, то поведение такого объекта может быть описано в терминах диаграммы состояний. Диаграмма состояний позволяет графически задать пространство состояний алгоритма поведения объекта, а также события, которые являются причинами срабатывания переходов из одних состояний в другие, и действия, происходящие при смене состояний.

Задание: построить модель объекта согласно своему варианту по описанию в файле Стейтчарты_индивидуальные_задания.

Рассмотрим пример построения модели в виде стейтчарта.

Модель пешеходного перекрестка: переключение состояний в стейтчартах

Постановка проблемы

В качестве простейшего примера применения стейтчартов рассмотрим модель регулируемого пешеходного перехода со светофором, разрешающим или запрещающим движение транспорта (рис.1).

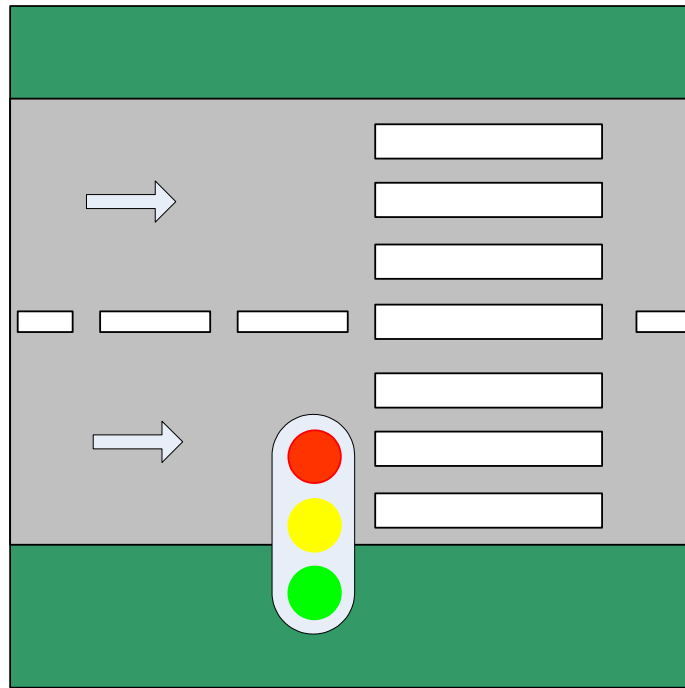


Рис.1. Простейший регулируемый пешеходный переход

Светофор может находиться в следующих состояниях: разрешение движения транспорта (зеленый), подготовка к запрещающему сигналу (мигающий зеленый), приготовление к остановке (желтый), запрет движения (красный) и приготовиться к движению (красный и желтый). Светофор работает в автоматическом режиме, циклически, в каждом состоянии находясь определенный постоянный период времени.


Рассматриваемые вопросы

В результате построения этой модели будут рассмотрены следующие новые вопросы:

Стейтчарты	Построение стейтчартов, действия при входе и при выходе из состояния, иерархические состояния.
Переходы в стейтчартах между состояниями	Переход по исчерпанию таймаута. Переход по событию. Переход по условию.

Построение модели

Создайте новый проект под названием *PedestrianCross*.

Наша модель имеет только один активный объект, представляющий светофор, поэтому корневой объект *Model* будет единственным активным объектом нашей модели. В поле редактора структуры активного объекта *Model* поместите иконку стейтчарта . Справа появится окно свойств этого нового объекта, в котором в поле имени будет стоять предопределенное имя *statechart*. Замените это имя на *traffic_light* (рис.2.)

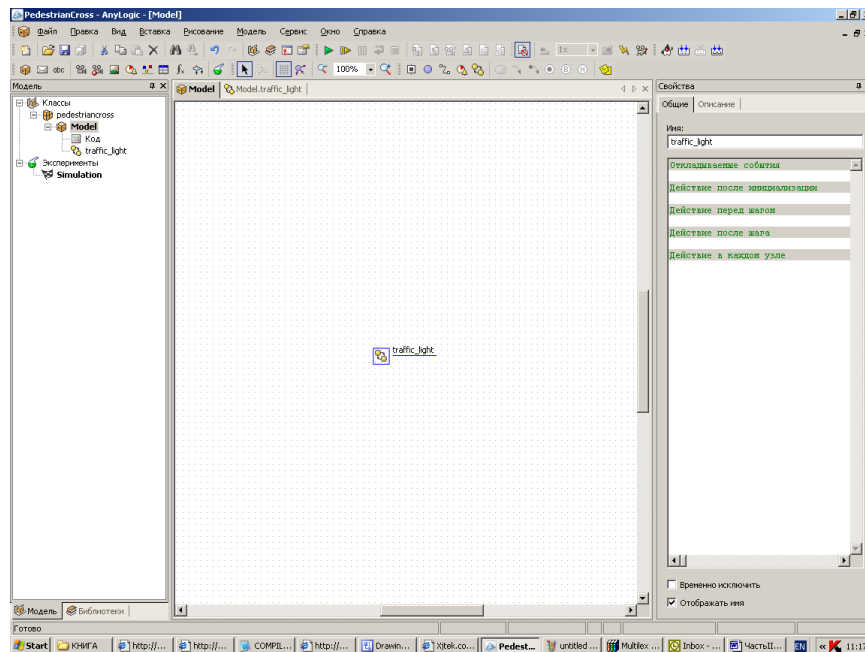


Рис.2. Окно редактора структуры объекта Model

После двойного щелчка мыши на иконке стейтчарта *traffic_light* откроется окно редактора этого стейтчарта с уже введенным одним состоянием с именем *state* с входящей в него стрелкой, что показывает, что это начальное состояние (рис.3). В действительности, это состояние, как и все другие, можно редактировать, изменять его имя, сделать его не начальным (убрать указатель) и т.п.

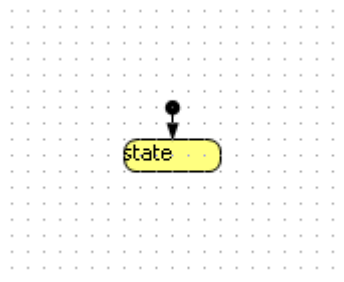


Рис.3. Фрагмент окна редактора стейтчарта

В соответствии с алгоритмом работы светофора введем кроме начального еще дополнительно четыре состояния. Начальное состояние назовем *go* (движение разрешено - зеленый), затем светофор переходит в состояния *attention* (внимание, подготовка к запрещающему сигналу – мигающий зеленый), *slow* (приготовиться к остановке - желтый), остановка, *stop* (запрет движения, красный) и *ready* (приготовиться к движению - красный и желтый). Состояние *attention*, фактически, представляет собой пару состояний, в одном из них зеленый горит, в другом – нет.

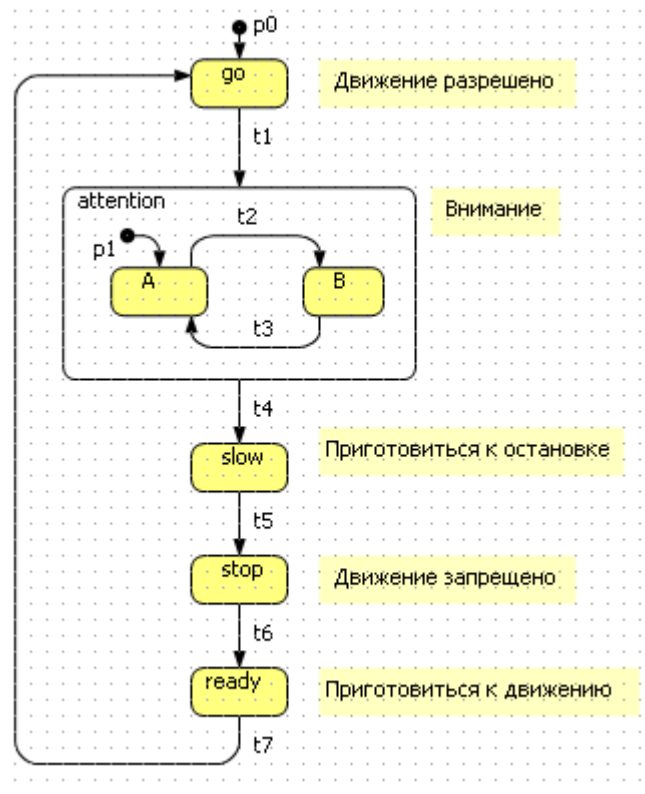










Рис.4. Стейтchart светофора

Для того, чтобы построить стейтchart, следует использовать иконки      на инструментальной панели окна редактора, которые становятся активными, если активизировано окно редактора стейтchart. С помощью иконки  рисуются состояния – как простые, так и гиперсостояния, иконка  используется для рисования переходов между состояниями и т.п. Комментарии помещаются на поле редактора с помощью иконки . Заметьте, что для выделенного объекта справа появляется окно его свойств, в котором можно изменить параметры и, в частности, имя объекта, если это необходимо. Для перемещения имени объекта по полю редактора следует выделить этот объект. Структурные ошибки при рисовании стейтchart - повисшие переходы, дублированные указатели начального состояния и т.п., выделяются красным цветом. Для отображения в поле редактора имени объекта – состояния или перехода – следует выбрать соответствующую опцию в нижней части окна свойств объекта.

Рассмотрим теперь, как задать условия срабатывания перехода из состояния в состояние. Все переходы в нашем автоматическом светофоре выполняются по таймауту, т.е. по истечении интервала времени, который прошел с момента прихода системы в данное состояние. Пусть в состоянии *go* светофор должен находиться 25 секунд, затем 7 сек мигает зеленый сигнал, затем 4 сек горит желтый в состоянии *slow*, в течение 20 сек движение запрещено и 4 сек светофор находится в состоянии *ready*.

Для того, чтобы задать такие условия срабатывания переходов, сделайте активным переход *t1*, в поле *Происходит* выберите вариант *По таймауту*, а в поле *Таймаут* введите 25 (рис.5). Аналогично задайте условия срабатывания других переходов. Между состояниями A и B

пусть переходы срабатывают через 1 единицу времени (в реальном времени 1 с горит зеленый свет, затем 1 с не горит). Заметьте, что переход срабатывает (в модельном времени) мгновенно.

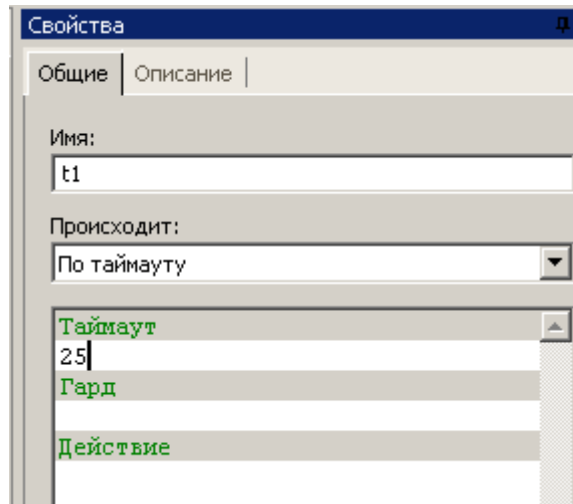



Рис.5. Задание перехода по таймауту

Запустите модель на выполнение. Для того, чтобы можно было наблюдать переходы между состояниями стейтчарта, откройте окно стейтчарта двойным щелчком левой кнопкой мыши на его изображении  в окне корневого объекта модели с именем *root*. Активное в данный момент состояние подсвечивается красным. Проведите эксперименты с моделью при различных масштабах времени. Ваша модель должна быть похожа на модель.

В каждом состоянии светофора должен гореть вполне определенный сигнал: в состоянии *go* должен гореть зеленый, в состоянии *ready* должны гореть красный и желтый и т.п. Откройте окно редактора структуры объекта *Model*. Определите три переменные логического (*boolean*) типа: *red*, *yellow* и *green*, которые будут принимать истинное значение тогда, когда светофор должен гореть соответственно красным, желтым и зеленым светом:



Начальные значения переменных можно не задавать: по умолчанию они будут равны *false*.

Стейтчарт построен именно для управления значениями этих переменных, каждое состояние отвечает за зажигание своего света (или комбинации светов). Например, при входе в состояние *green* должен загореться зеленый свет, а при выходе из этого состояния зеленый должен выключиться, и т.п. Откройте окно свойств состояния *go* и в поле *Действие при входе* запишите "*green = true;*", а в поле *Действие при выходе* запишите "*green = false;*" (рис.6). То же самое нужно определить для состояния В, а у состояния А эти поля нужно оставить без изменения — когда светофор находится в этом состоянии, он вообще не горит. Аналогично, в состоянии *slow* нужно включить желтый сигнал, т.е. при входе в это состояние установить переменную *yellow* в *true*, а при выходе из этого состояния установить ее в *false*. Для состояния

stop то же нужно сделать с переменной *red*, а для состояния *ready* установить в *true* при входе и установить в *false* при выходе из него следует обе переменные – *red* и *yellow*.

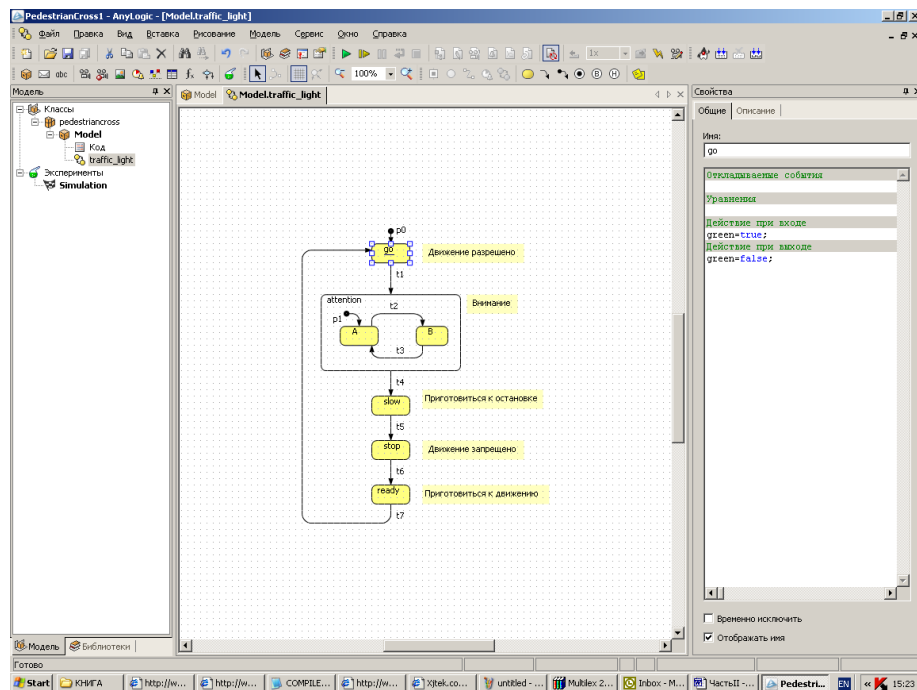


Рис.6. Определение действий при входе и при выходе из состояния

Запустите модель на выполнение при различных масштабах времени. В окне *root* в дереве переменных и параметров модели переменные *green*, *yellow* и *red* будут переключаться между значениями *истина* и *ложь* (представляемыми здесь 1 и 0) в соответствии с алгоритмом переключения светофора. В модели это просто переменные, но их значения можно выводить и управлять в реальном времени переключением светофора.

Создание анимации

Анимация для этой модели весьма просто строится средствами AnyLogic (рис.7). Все объекты в анимации статические кроме цвета сигналов светофора. Светофор строится из трех эллипсов, статически повернутых на 45 градусов. Динамическое значение цвета верхнего сигнала светофора необходимо установить *Color.red*, если переменная *red* истинна, в противном случае цвет его нужно установить *Color.gray* (серый). Это записывается следующим выражением:

red? Color.red : Color.gray

Цвет среднего и нижнего сигналов светофора следует установить в поле динамических их значений соответственно так:

yellow? Color. yellow : Color.gray

green? Color. green : Color.gray

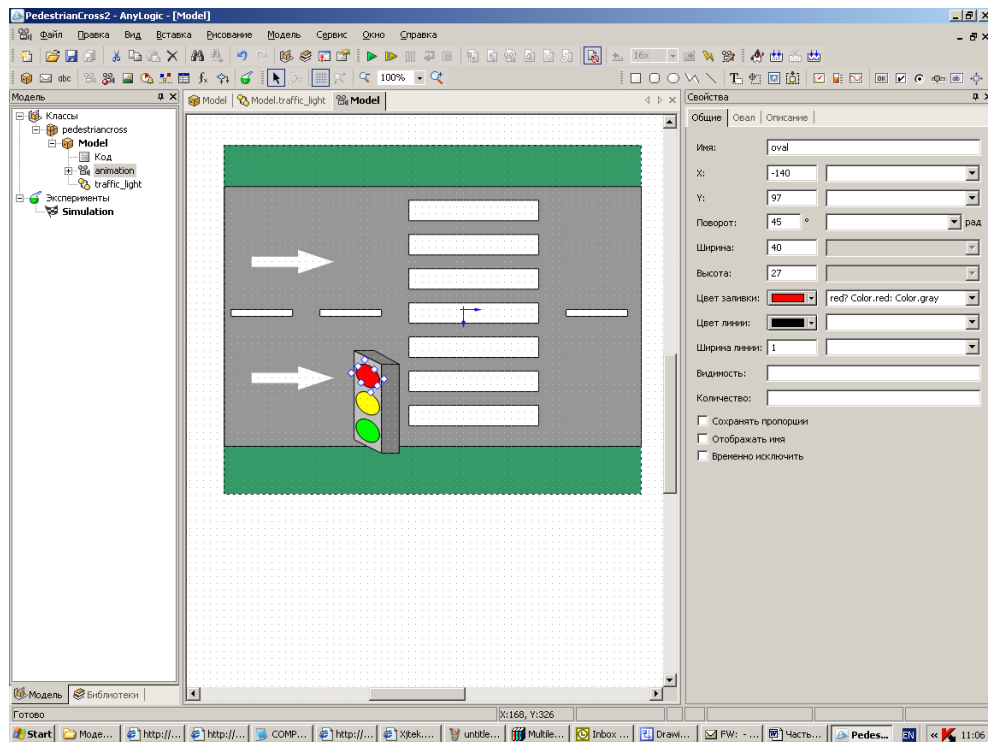


Рис.7. Задание динамического значения цвета для светофора

Запустите модель. На рис.8 зафиксирован момент выполнения модели в состоянии *ready*.

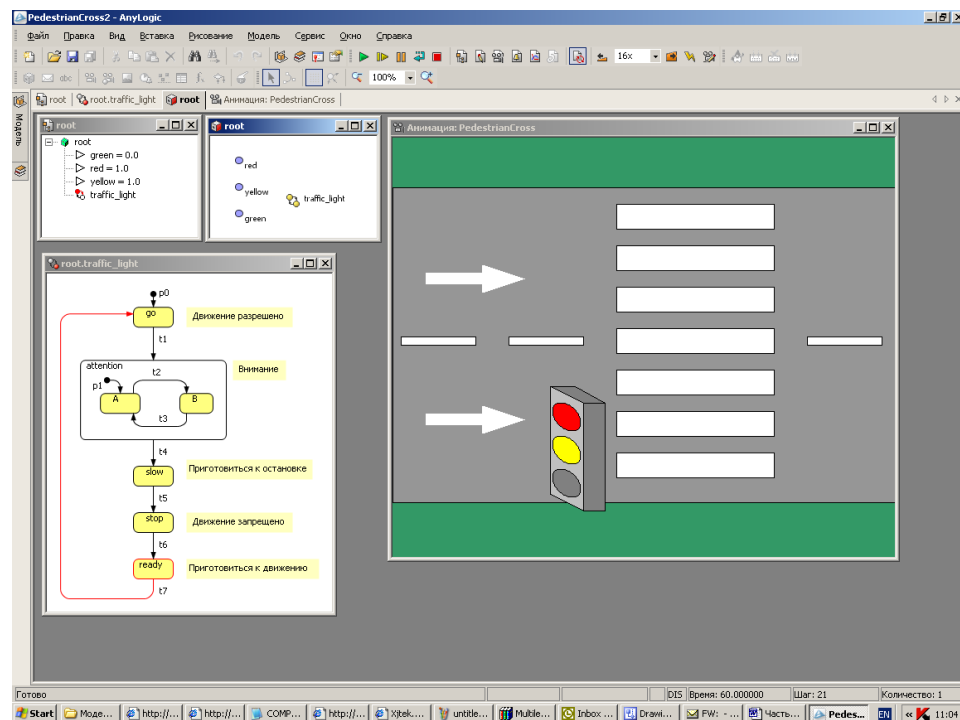
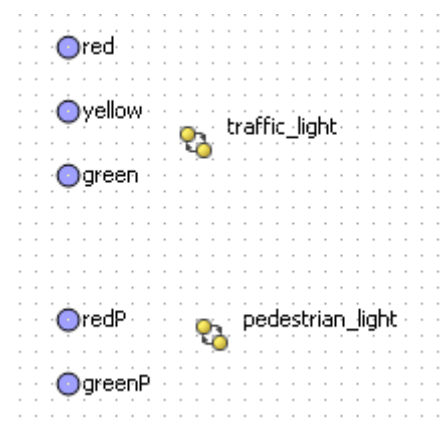


Рис.8. Выполнение модели пешеходного перехода

Срабатывание перехода по сигналу

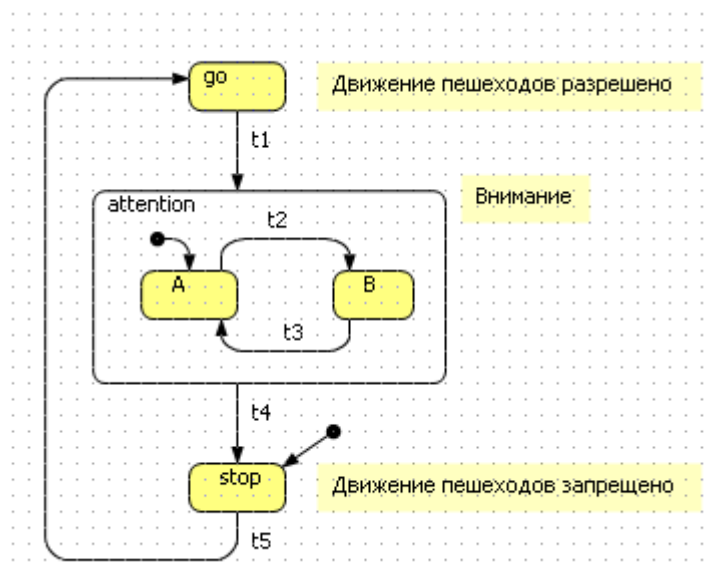
Добавим к модели пешеходного перехода второй светофор – для пешеходов. Он имеет только два света – зеленый и красный, и три состояния - разрешающее



переход (зеленый), внимание (мигающий зеленый) и запрещающее переход (красный). В модель добавим еще две булевские переменные *greenP* и *redP*, их значения буду устанавливаться в состояниях стейтчарта, который будет управлять светофором пешеходов. Добавим новый стейтчарт в окне редактора объекта Model, назвав его *pedestrian_light*.

Поскольку этот стейтчарт похож на стейтчарт светофора траффика (рис.4), его можно получить изменением уже построенного стейтчарта *traffic_light*. Откройте окно редактора объекта Model, и нажав клавишу Ctrl на клавиатуре, перетащите иконку *traffic_light* в другое место. Этот объект скопируется и ему будет дано другое имя – *traffic_light1*. Измените имя стейтчарта на *pedestrian_light* и отредактируйте сам стейтчарт так, чтобы в нем были только состояния *go*, *attention* и *stop*. Поскольку эти имена принадлежат элементам объекта *pedestrian_light*, конфликта имен с так же названными состояниями другого стейтчарта *traffic_light* здесь не будет.

Изменения, которые должны быть проведены с этим стейтчартом, такие. Во-первых, начальное состояние для светофора пешеходов должно быть *stop* - противоположным начальному состоянию светофора траффика. Кроме того, установка переменных при входе и выходе из состояний должна относиться к переменным *redP* и *greenP*, управляющих зажиганием света именно пешеходного светофора.



Рассмотрим условия срабатывания переходов стейтчарта между состояниями. Очевидно, что необходимо заботливо синхронизировать срабатывания переходов двух стейтчартов так, чтобы всегда, когда светофор пешеходов находится в состояниях *go* или *attention*, светофор траффика находился бы только в состоянии *stop*. Этого можно добиться подбором таймаутов срабатывания переходов. Однако более разумно сделать это, посылая разрешающие сигналы из одного стейтчарта в другой.

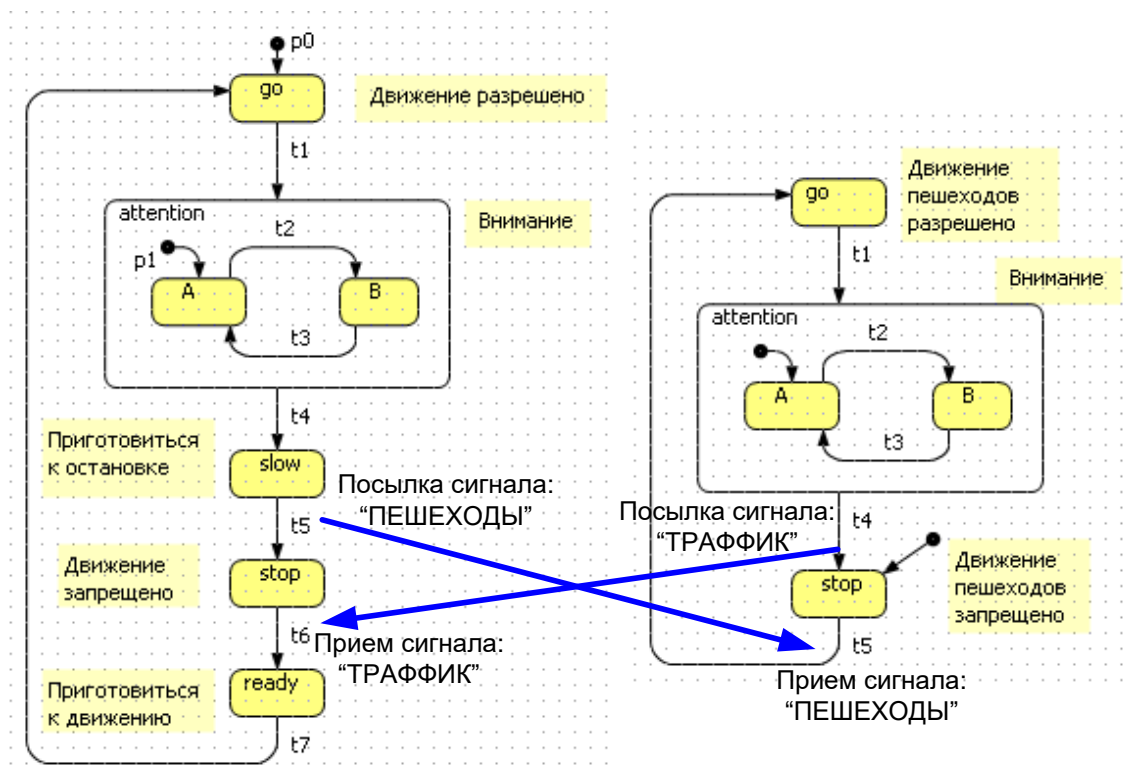


Рис.9 Синхронизация поведений при помощи сигналов

А именно, введем два сигнала, “ТРАФИК” и “ПЕШЕХОДЫ”. Пусть в стейтчарте управления светофором пешеходов переход $t5$ может сработать только если получен сигнал “ПЕШЕХОДЫ”, который будет генерироваться в стейтчарте управления траффиком при переходе $t5$ в состояние, в котором запрещено движение транспорта. В стейтчарте *traffic_light* переход $t6$ пусть может сработать только если получен сигнал “ТРАФИК”, который генерируется в стейтчарте управлением движением пешеходов при переходе $t4$ в состояние, запрещающее движение пешеходов.

В AnyLogic для генерации сигналов существует функция *fireEvent(<сигнал>)*, которая должна вызываться в том стейтчарте, которому предназначен сигнал. Если мы хотим послать (произвольный) сигнал “AAA” стейтчарту st , то необходимо выполнить действие *st.fireEvent(“AAA”)*. Если стейтчарт st находится в состоянии, в котором он ожидает этот сигнал, то соответствующий переход сработает. Стейтчарт ожидает сигнала “AAA”, если в поле *Происходит* окна свойств перехода выбран вариант *По сигналу*, а в поле *Сигнал* этого окна указан этот сигнал “AAA”.

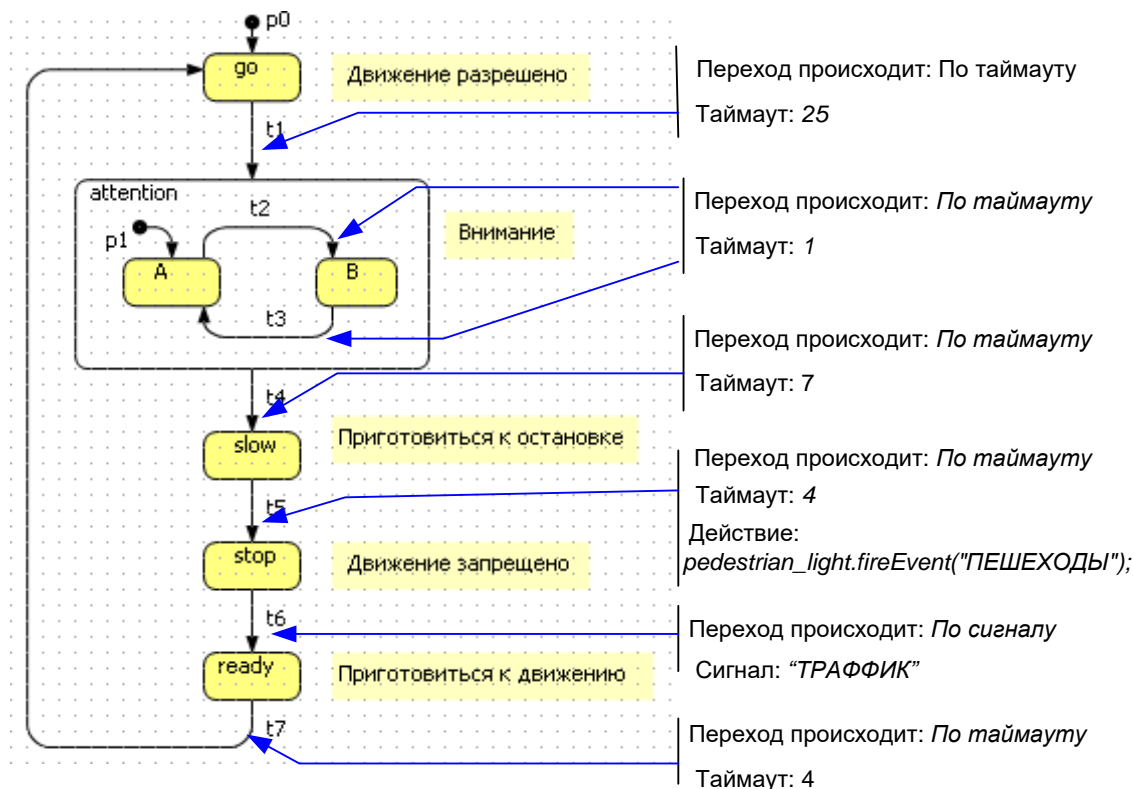



Рис.10. Переходы стейтchartа управления движением

В нашей модели в поле *Действие* перехода *t5* стейтchartа *traffic_light* вставьте команду `pedestrian_light.fireEvent("ПЕШЕХОДЫ");`; а в это же поле перехода *t4* стейтchartа *pedestrian_light* вставьте команду `traffic_light.fireEvent("ТРАФИК");`. Таким образом, каждый из светофоров будет информировать другого о переходе в состояние запрещения соответствующего движения. Для перехода по разрешающему сигналу в стейтchartе *traffic_light*, в поле *Происходит* окна свойств перехода *t6* выберите вариант *По сигналу*, а в поле *Сигнал* этого же окна наберите сигнал "ТРАФИК". Для перехода по разрешающему сигналу в стейтchartе *pedestrian_light*, в поле *Происходит* окна свойств перехода *t5* выберите вариант *По сигналу*, а в поле *Сигнал* этого же окна наберите сигнал "ПЕШЕХОДЫ". Остальные переходы этих стейтchartов пусть срабатывают по таймаутам. Рис.10 показывает, как задать параметры переходов стейтchartа *traffic_light*. Запустите модель на выполнение.

Аналогию модели легко дополнить светофором пешеходов с двумя сигналами – красным и зеленым, значением цвета которых управляется логическими переменными *redP* и *greenP*. Ваша модель должна быть похожа на модель PedestrianCross3.

Срабатывание перехода по условию

Введем вместо автоматического переключения светофора пешеходов кнопку, при нажатии которой пешеходами им разрешается переход. Для этого, активизировав окно редактора анимации, кликнем на иконку "кнопка" () панели инструментов и поместим кнопку куда-нибудь перед пешеходным переходом. Назовем кнопку "ЖДУ", при ее нажатии которой будет устанавливаться в истину новая булева переменная *waiting* (т.е. установим в поле "Реакция на событие" окна свойств этой кнопки команду `"waiting = true"`). Логическую

переменную *waiting* нужно ввести в поле редактора активного объекта Model с начальным значением *false*. Эта переменная фиксирует, ждет ли пешеход возможности перейти дорогу. Значение этой переменной сбрасывается в *false* как только светофор пешеходов переходит в состояние “мигающий зеленый”, т.е. команда “*waiting = false;*” должна быть добавлена в поле Действие окна свойств перехода t1 стейтchartа *pedestrian_light*. Наличие ожидающих на переходе пешеходов является условием переключения светофора *traffic_light* из состояния *go* в состояние *attention* и далее в состояние запрещения движения транспорта. При этом важно, чтобы постоянное нажатие этой кнопки не парализовало движение транспорта. Для этого в стейтchartе *traffic_light* изменим состояние *go*, сделав его иерархическим с двумя простыми состояниями *go1* и *go2* так, как показано на рис.11.

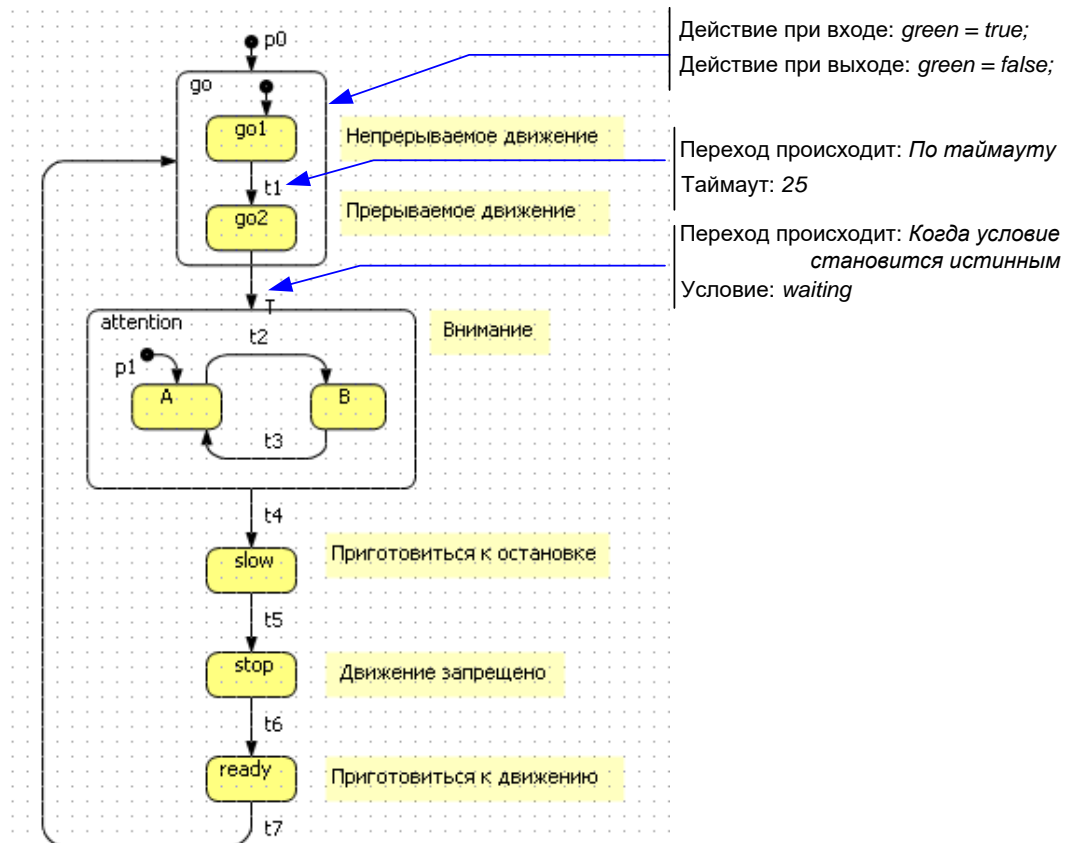


Рис.11. Изменения в стейтchartе *traffic_light*

В состоянии *go1* осуществляется непрерываемое движение транспорта, выход из этого состояния происходит точно когда исчерпается выделенный таймаут (здесь 25 сек), после чего светофор переходит в состояние прерываемого движения *go2*, переход *T* из которого срабатывает, когда будет выполнено условие – наличие пешеходов, ожидающих переход. Если кнопка ЖДУ не была нажата до этого времени с момента последнего разрешения перехода (т.е. переменная *waiting* все еще имеет значение *false*), то транспорту будет разрешено продолжать двигаться до нажатия этой кнопки. Если кнопка уже была нажата (произвольное число раз), то переменная *waiting* приняла значение *true*, и светофор *traffic_light* мгновенно перейдет в состояние *attention* и затем остановит движение. На рис.12 представлен момент после окончания интервала разрешения перехода пешеходов. Сравните вашу модель с моделью PedestrianCross4.

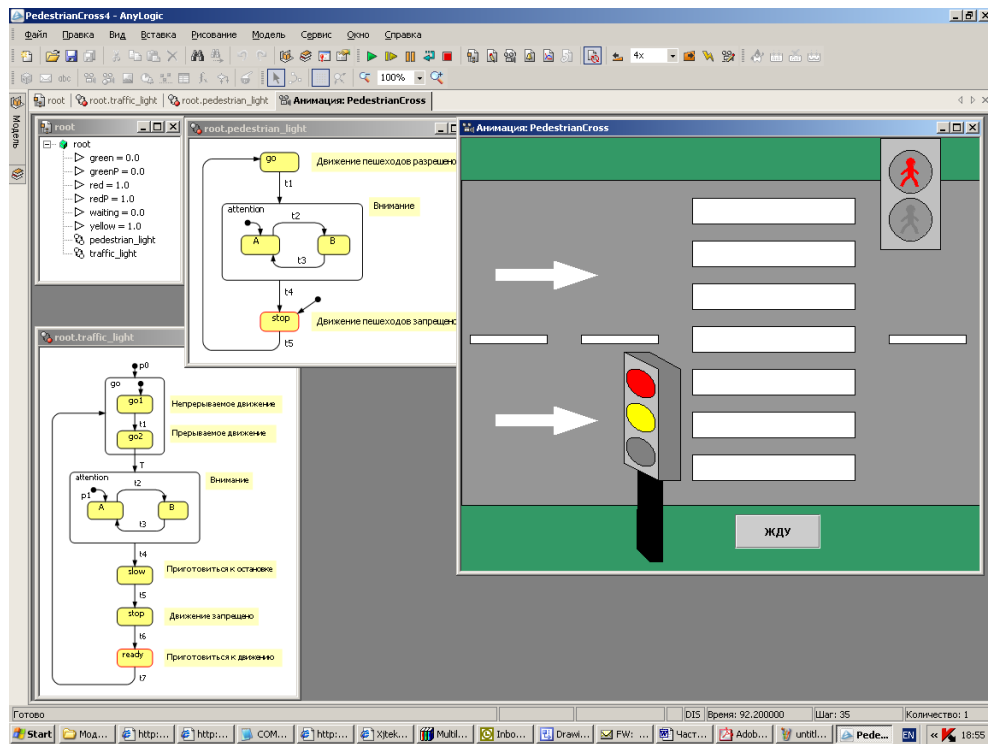


Рис.12. Работа модели пешеходного перекрестка