

Диаграммы состояний

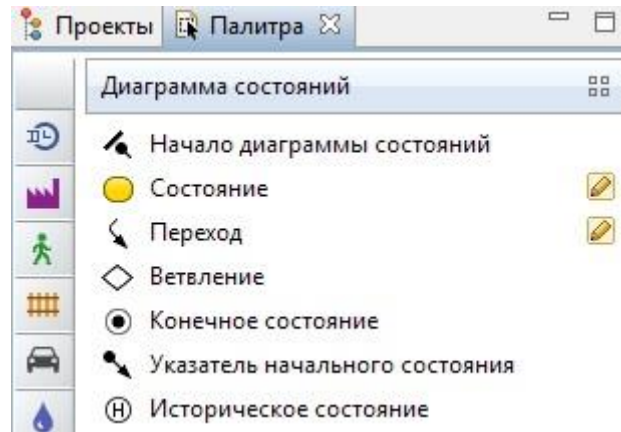


Рис.1

Два основных элемента: **Состояние** и **Переход**. Чтобы создать состояние надо перенести символ **Состояние** на рабочий стол.

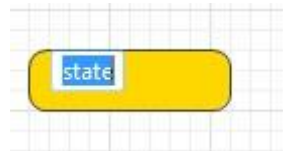


Рис.2

Можно менять размеры фигуры

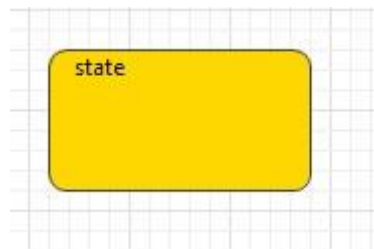


Рис.3

Можно сделать любой цвет заливки. Например, зеленый.

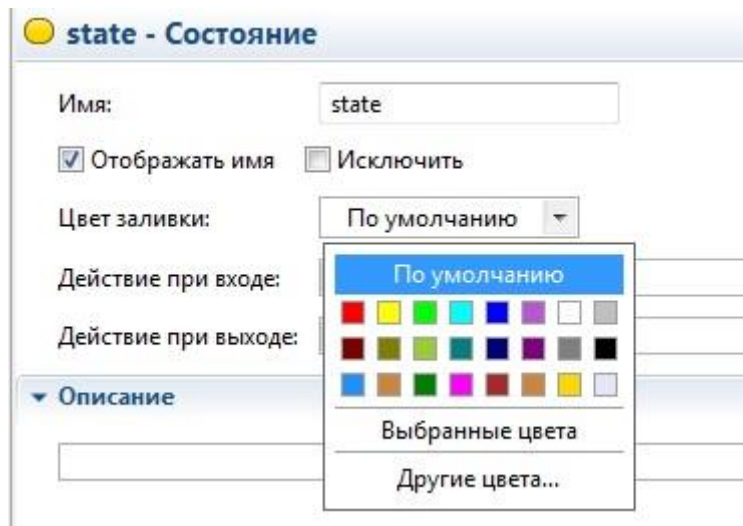


Рис.4

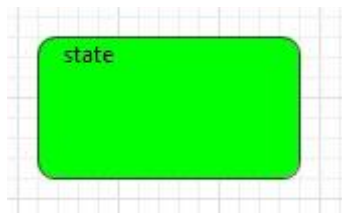


Рис.5

Можно переименовать состояние и написать название на русском языке, например, *состояние1*.

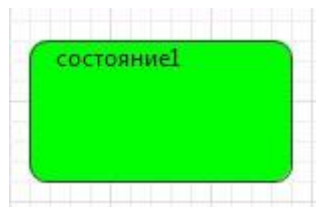


Рис.6

Обратите внимание: как только появилось состояние, у вас появилась ошибка.

Теперь как нарисовать переход. Это можно сделать двумя способами. Перетаскиваем переход. Для этого щелкаем 2 раза по значку **Переход**.

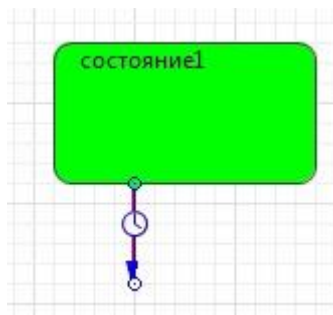


Рис.7

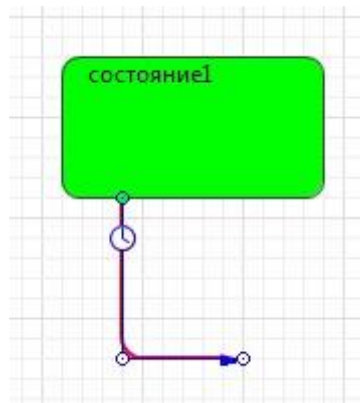


Рис.8

Создадим еще одно Состояние второе.

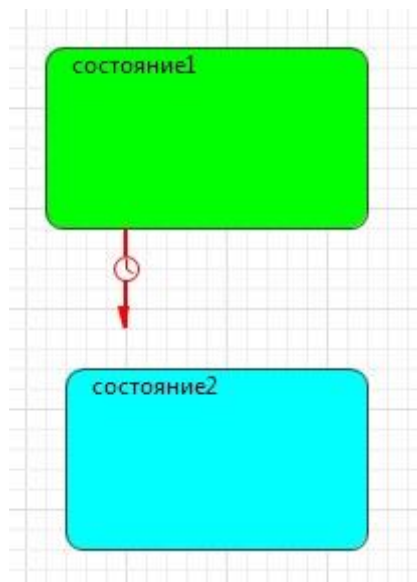


Рис.9

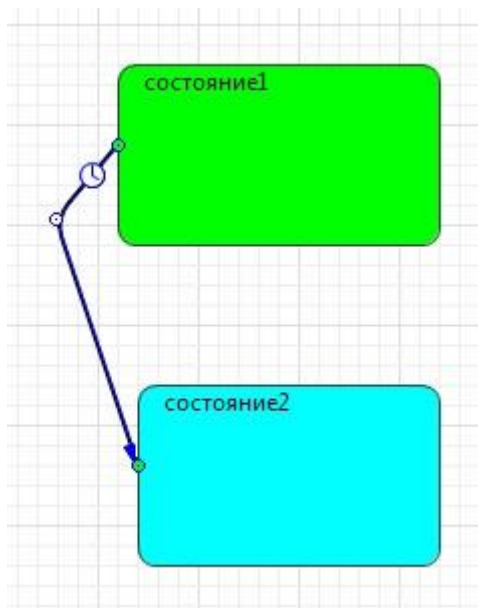


Рис.10

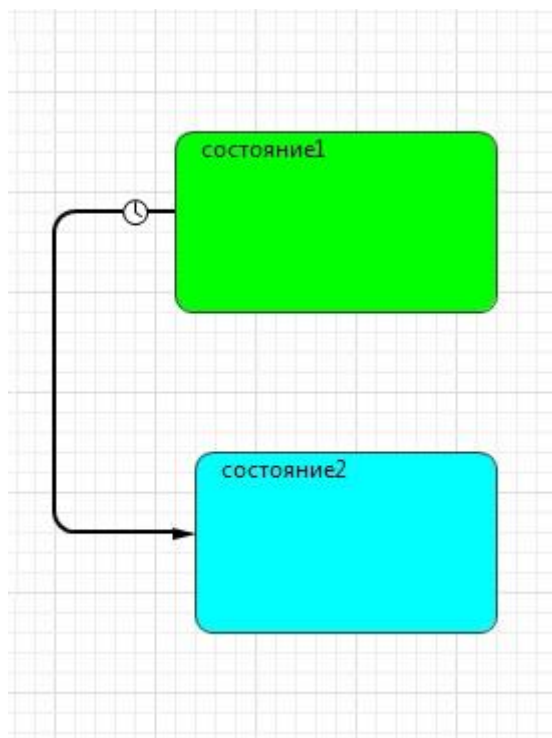


Рис.11

Переход осуществляется по таймауту через 1 сек из состояния1 в состояние2

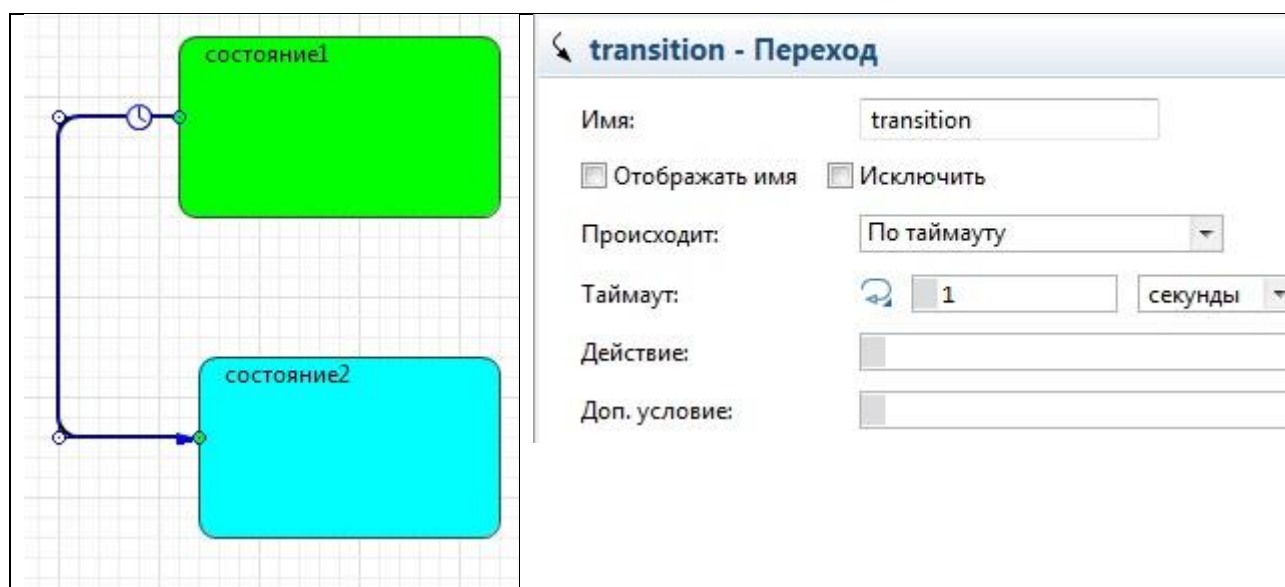


Рис.12

Значок перехода можно перемещать по линии перехода

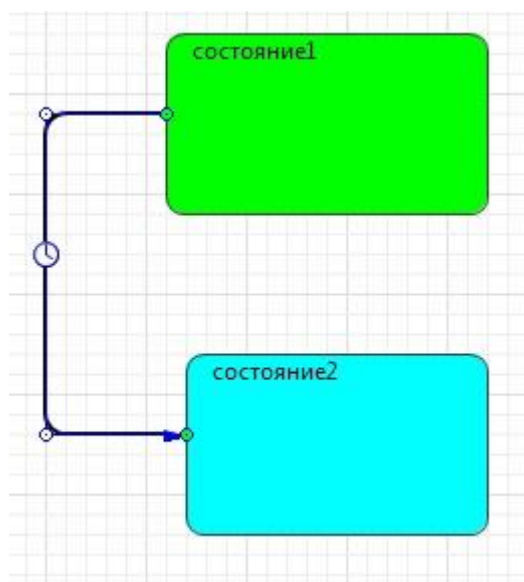


Рис.13

Значок перехода меняется при выборе условий перехода

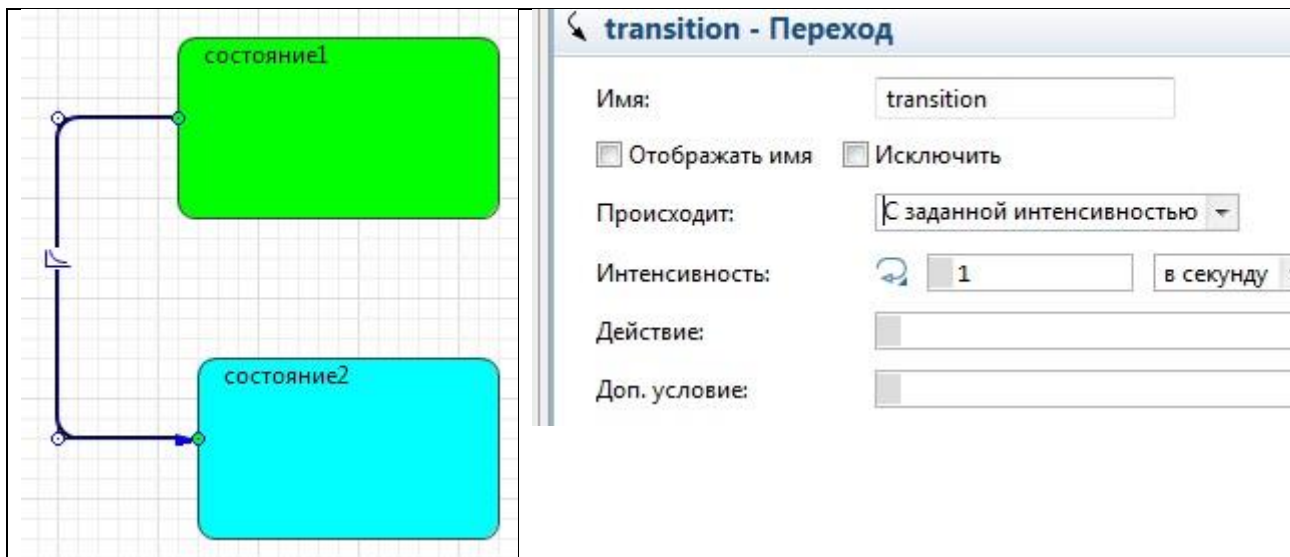


Рис.14

Нарисуем обратный переход. Установим, что переход1 и переход2 выполняются по таймауту с частотой 1 сек

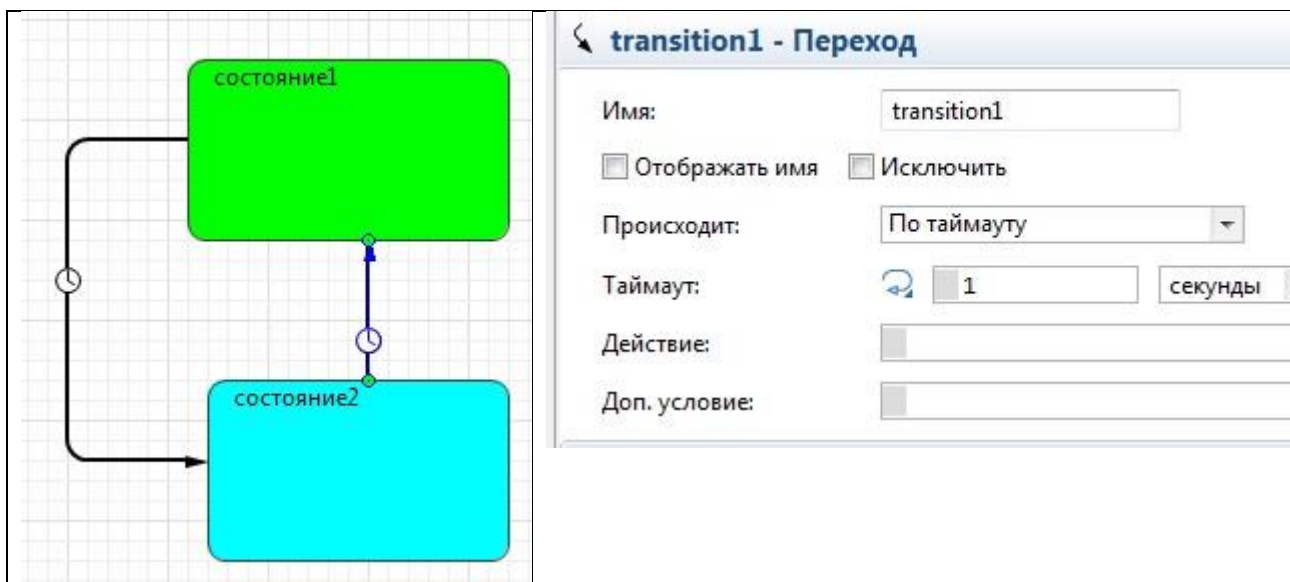


Рис.15

Но у нас 4 ошибки. Почему? Диаграмма состояний всегда должна иметь своё Начало. Т.е. диаграмма состояний всегда должна иметь свое Начало, конца может и не быть, но Начало должно быть всегда.

Поэтому есть специальный элемент, который называется Начало диаграммы состояний. По сути это указатель, который указывает какое состояние запускается первым. Мы перемещаем указатель Начало диаграммы состояний и должны его соединить с каким-либо состоянием.

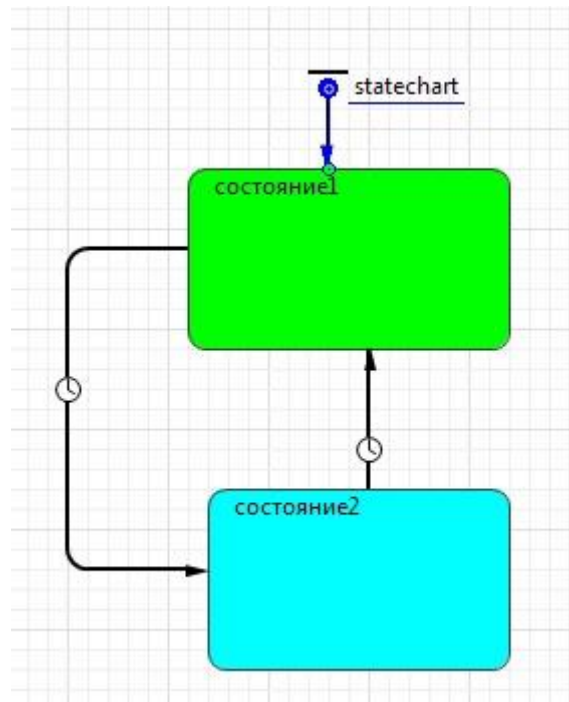


Рис.16

Обратите внимание, что все ошибки исчезли.

Переименуем название Начало диаграммы состояний

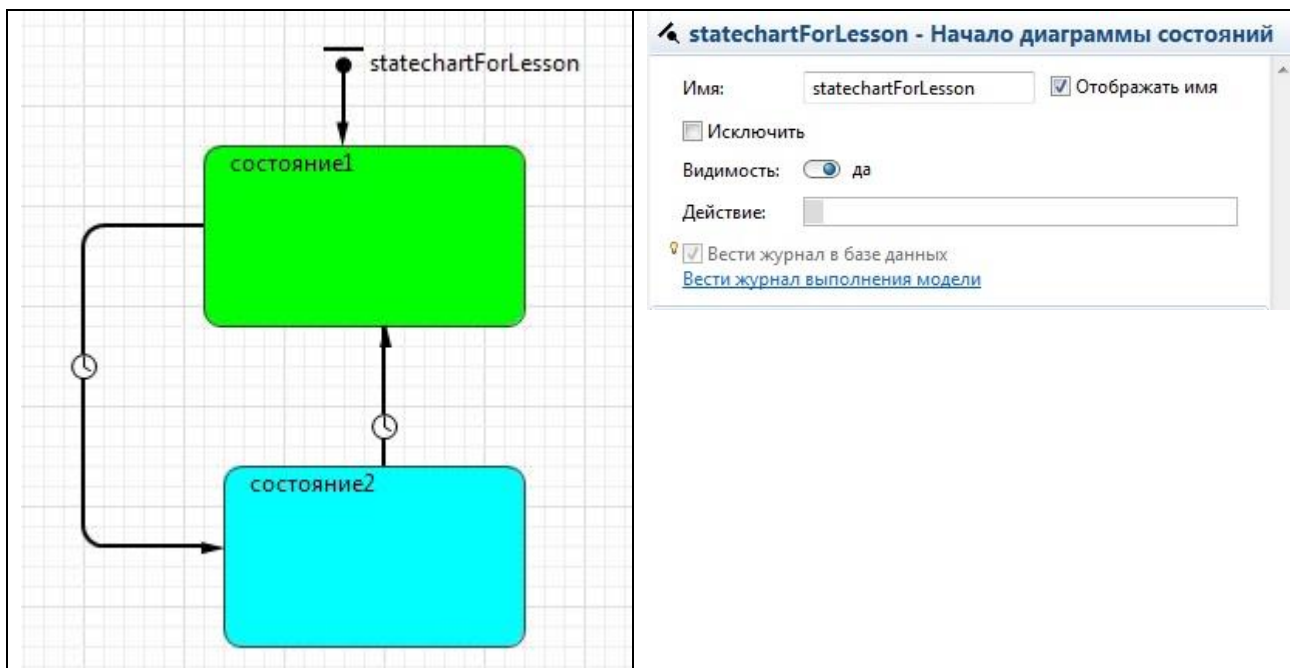


Рис.17

Состояния бывают двух типов: простые и сложные. Простые состояния это *состояние1* и *состояние2*. Потому что в них ничего нет. Есть состояния сложные, которые включают в себя другие состояния. Например, временно отключим Начало диаграммы состояний

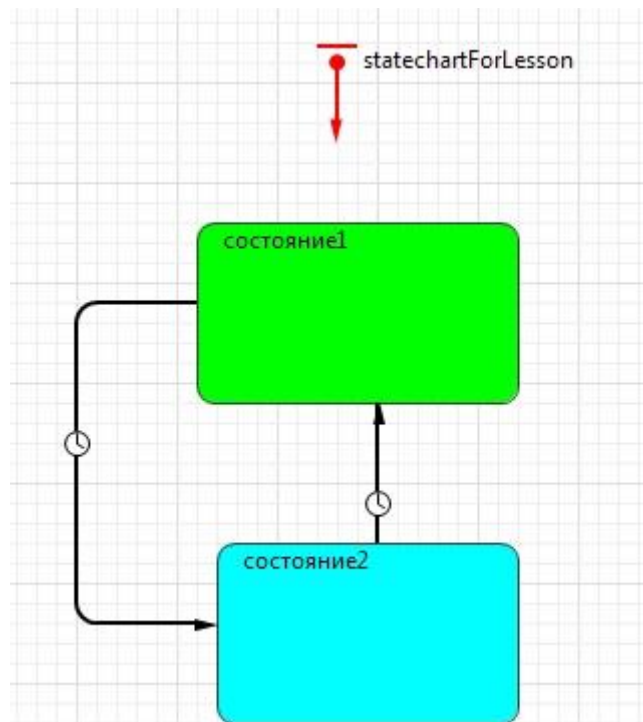


Рис.18

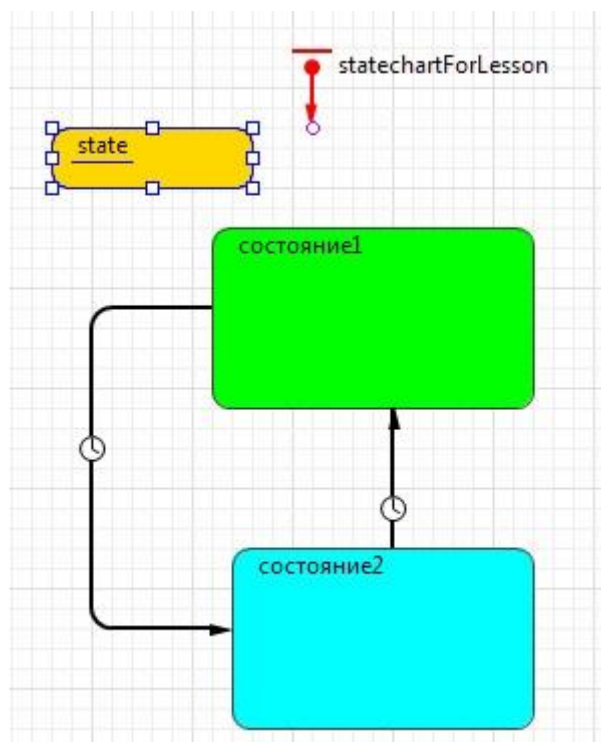


Рис.19

Изменим размеры состояния (state) так, чтобы охватить оба простых состояний.

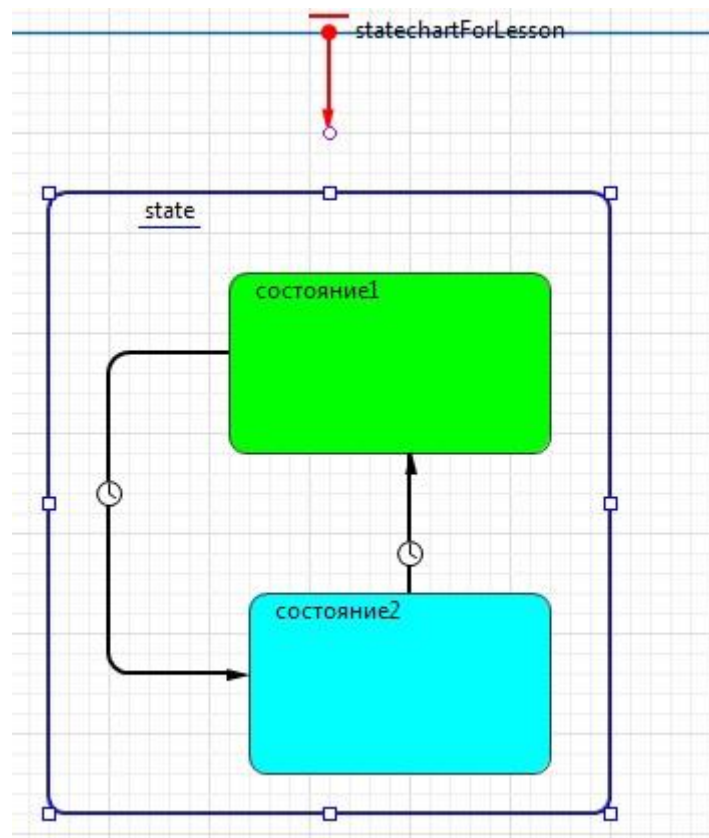


Рис.20

Переименуем состояние

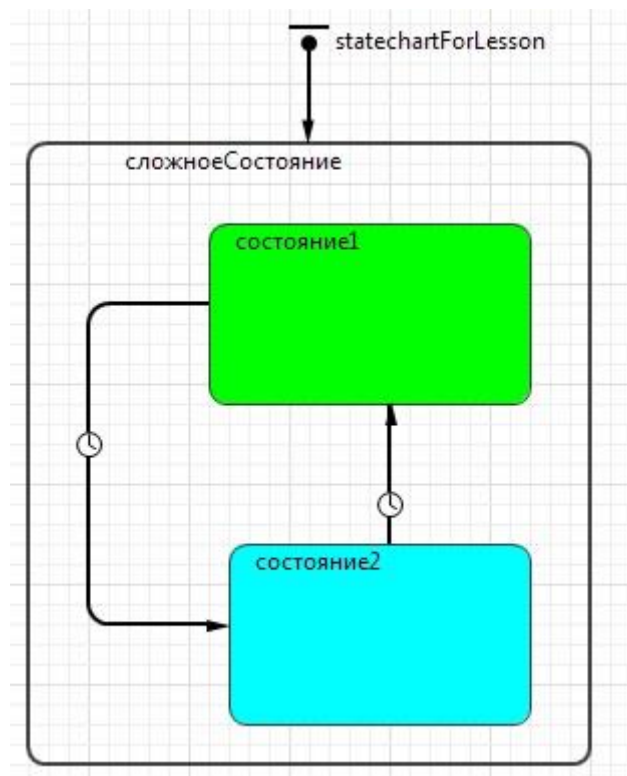


Рис.21

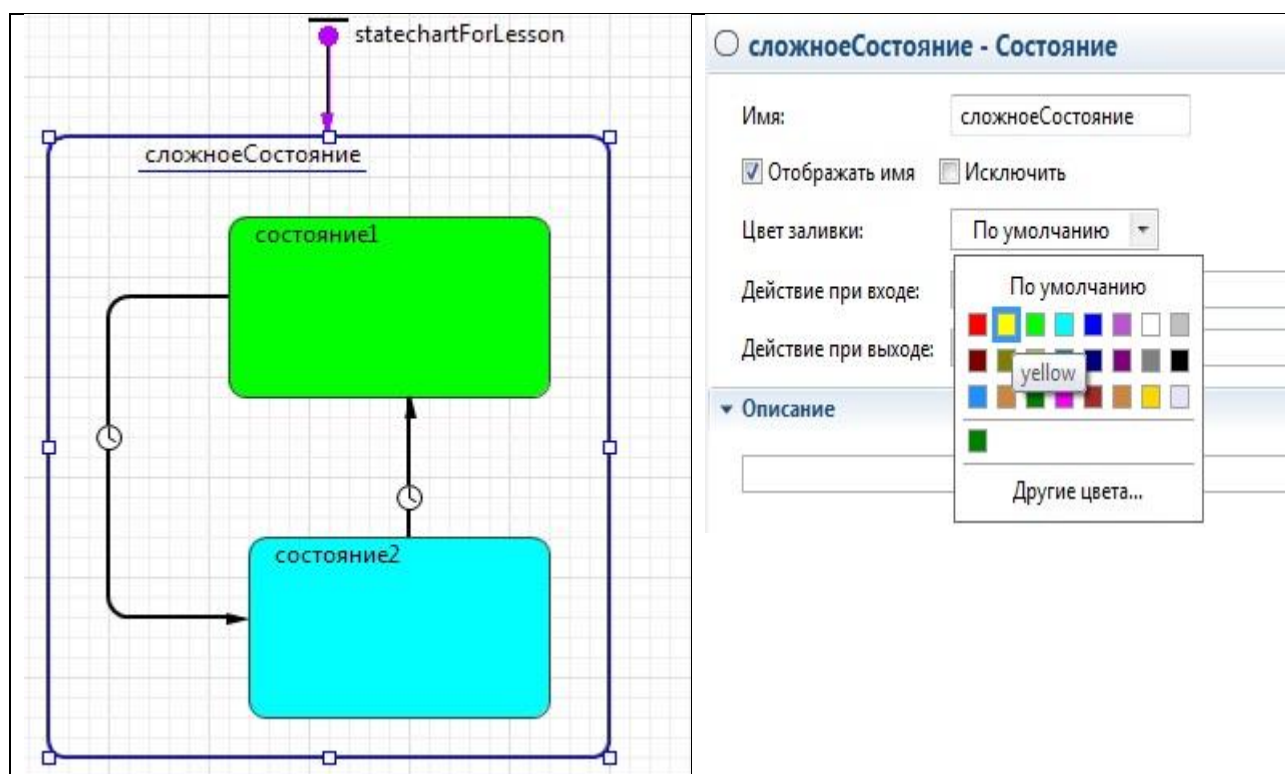


Рис.22

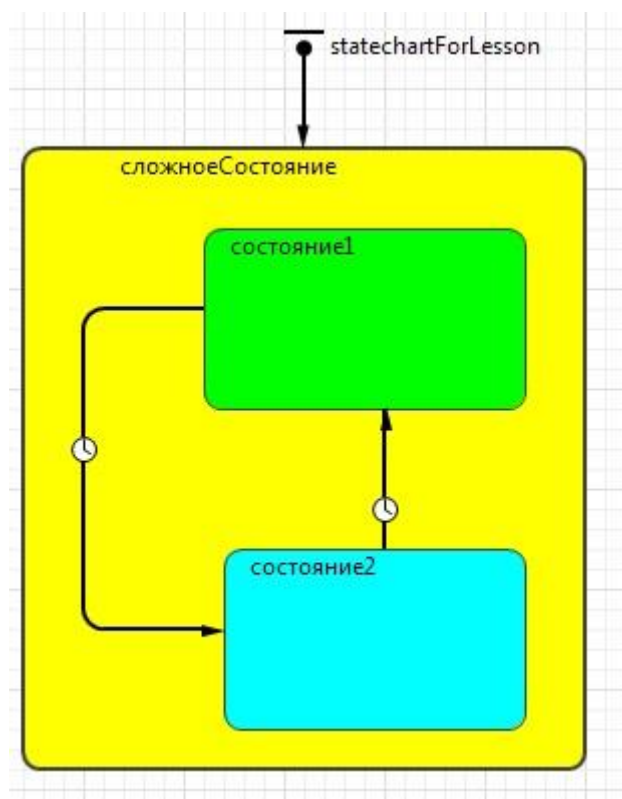


Рис.23

По умолчанию сложное состояние всегда бесцветное

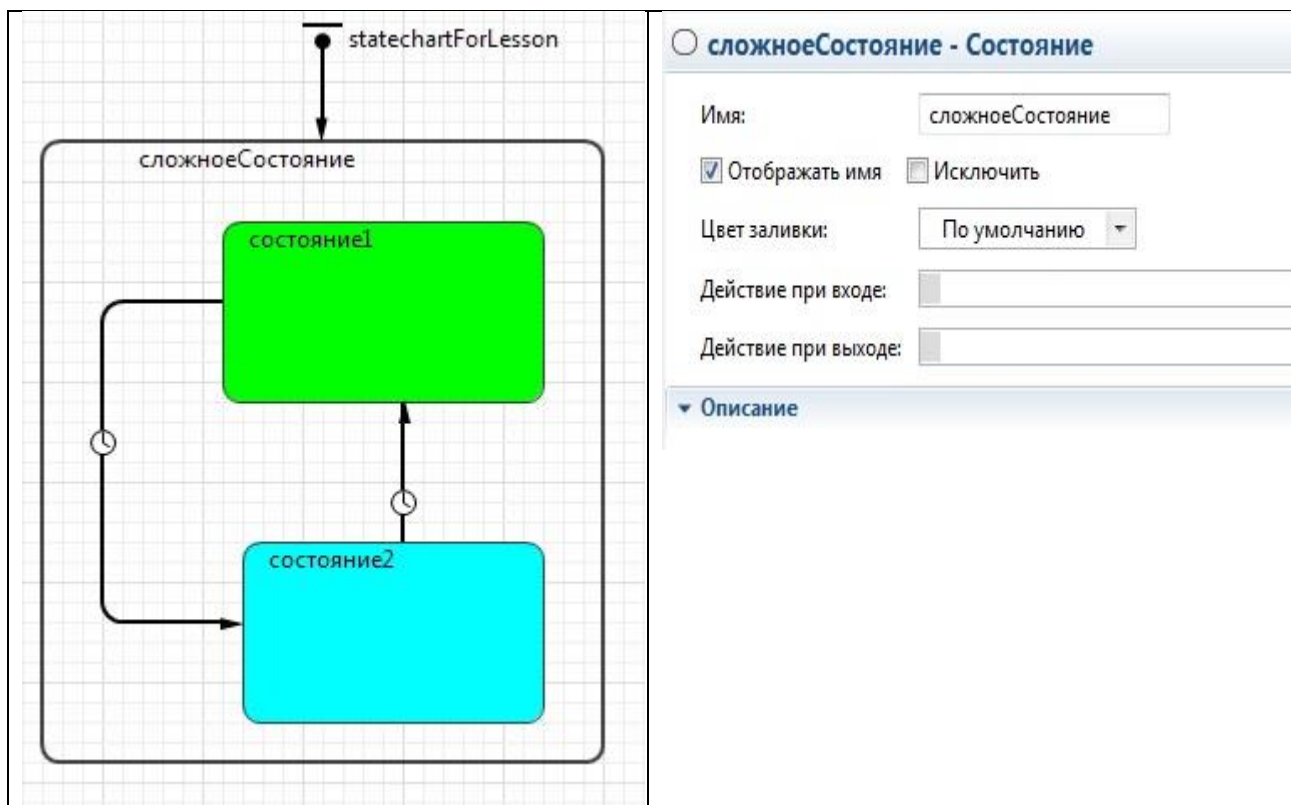


Рис.24

Сложные состояния могут в себя включать только другие состояния.
Переходы не входят в сложное состояние.

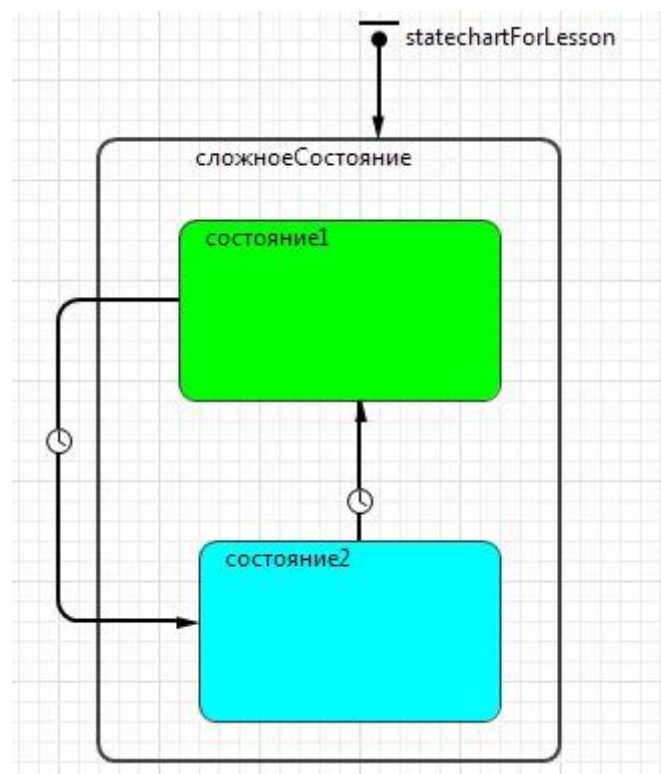


Рис.25

Переходы не зависят от того, где они находятся: внутри состояния, вне состояния. Переходы показывают только направление движения. Нарисуем еще два состояния.

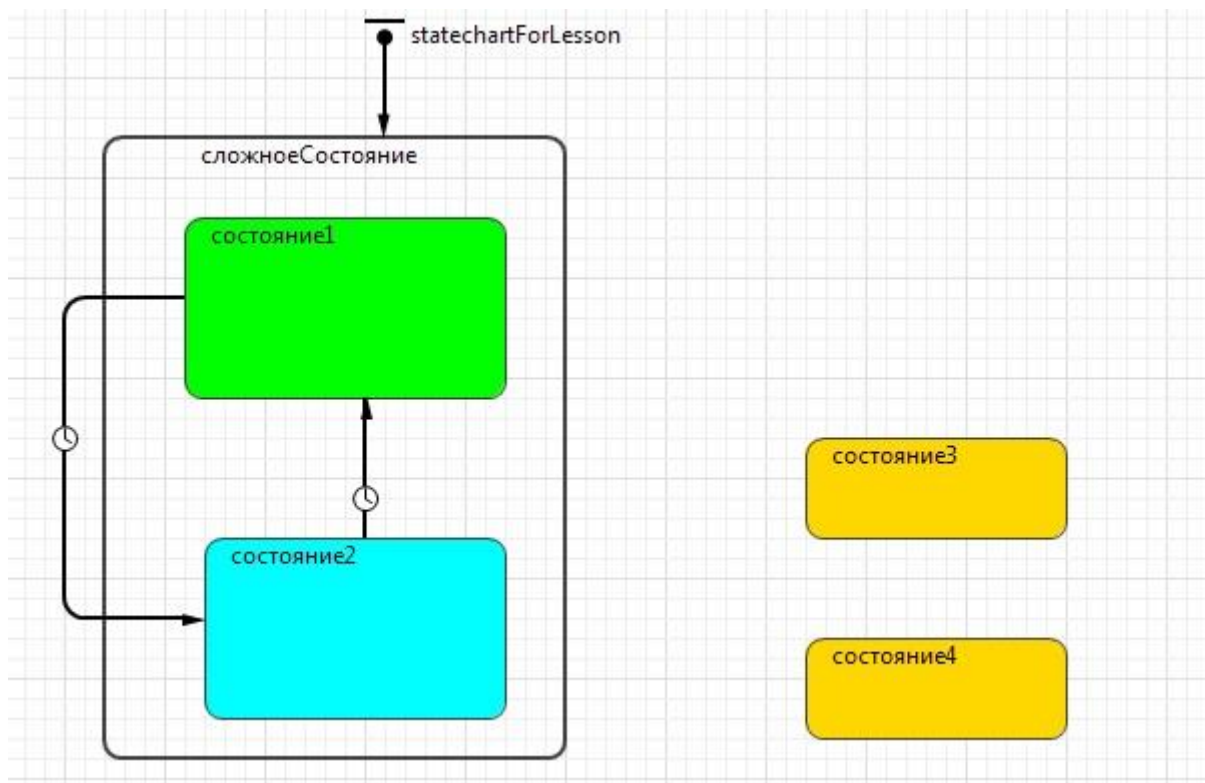


Рис.26

Пусть переход в состояние3 происходит из сложного состояния

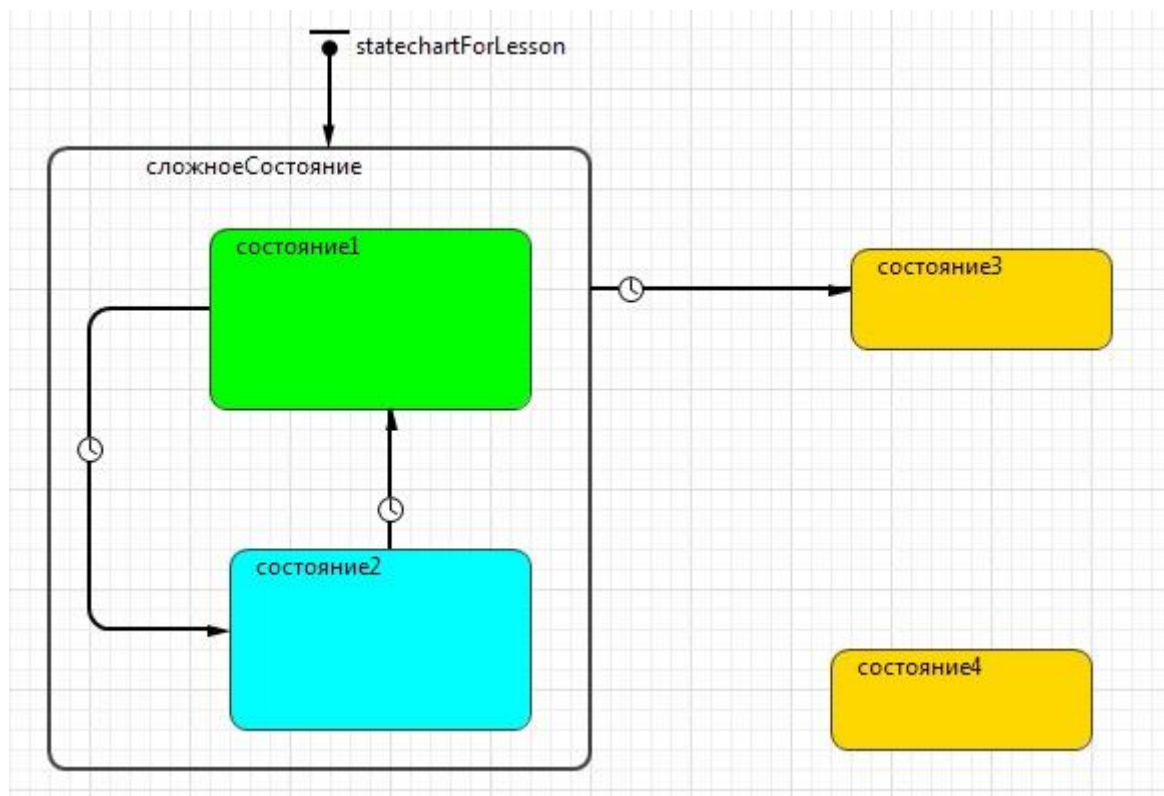


Рис.27

Пусть переход из сложного состояния переход будет выполняться по какому либо условию, а обратный переход из состояния3 будет происходить в состояние1.

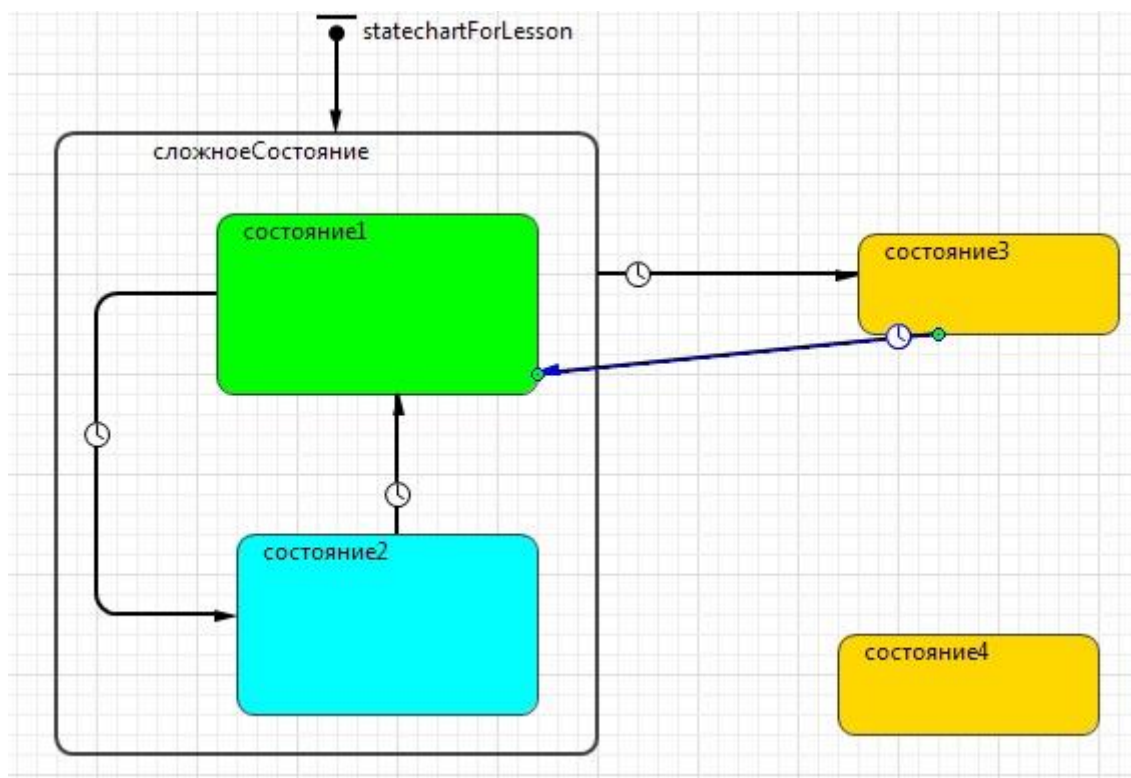


Рис.28

Для красоты будем рисовать ровными прямыми линиями

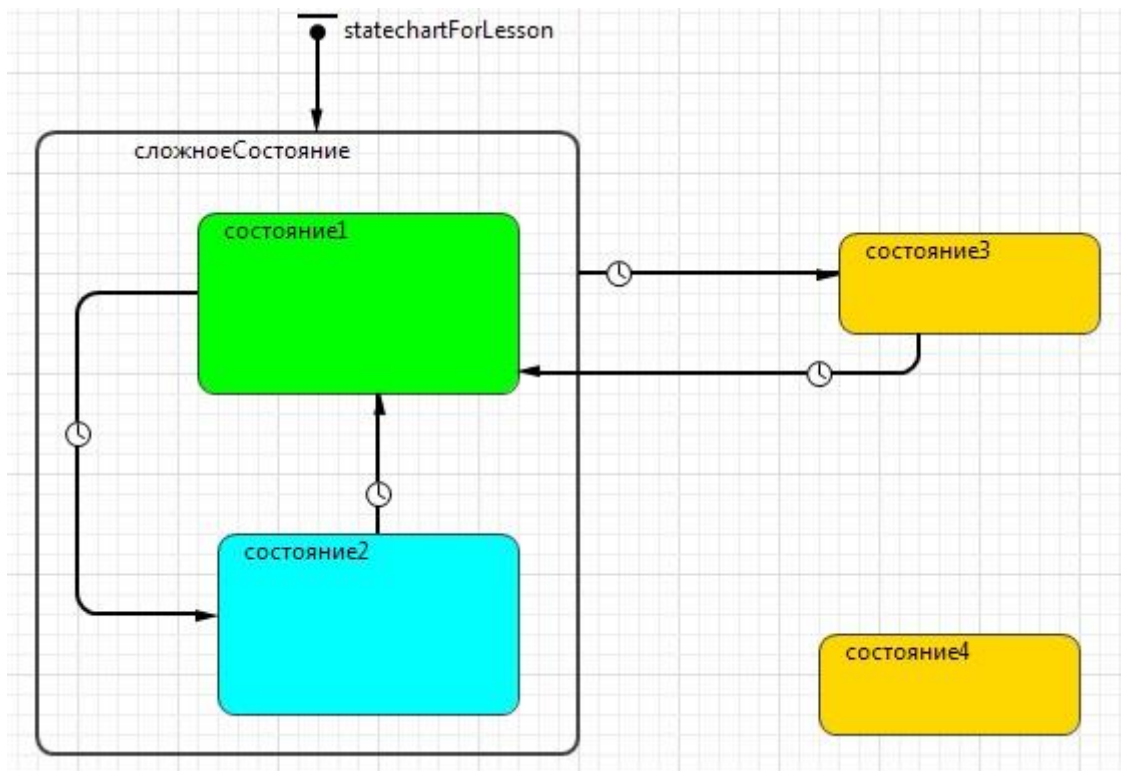


Рис.29

Независимо от того в каком состоянии находится сложная система, т.е. в состоянии1 или состоянии2 , если условие выполняется будет переход в состояние3. А вот обратно переход из состояния3 осуществляется в состояние1.

Ошибок 2. Если состояние является сложным, то мы должны в нем поместить Указатель начального состояния. Указатель Начального состояния показывает в какое состояние переходит сложное состоян при входе в него.

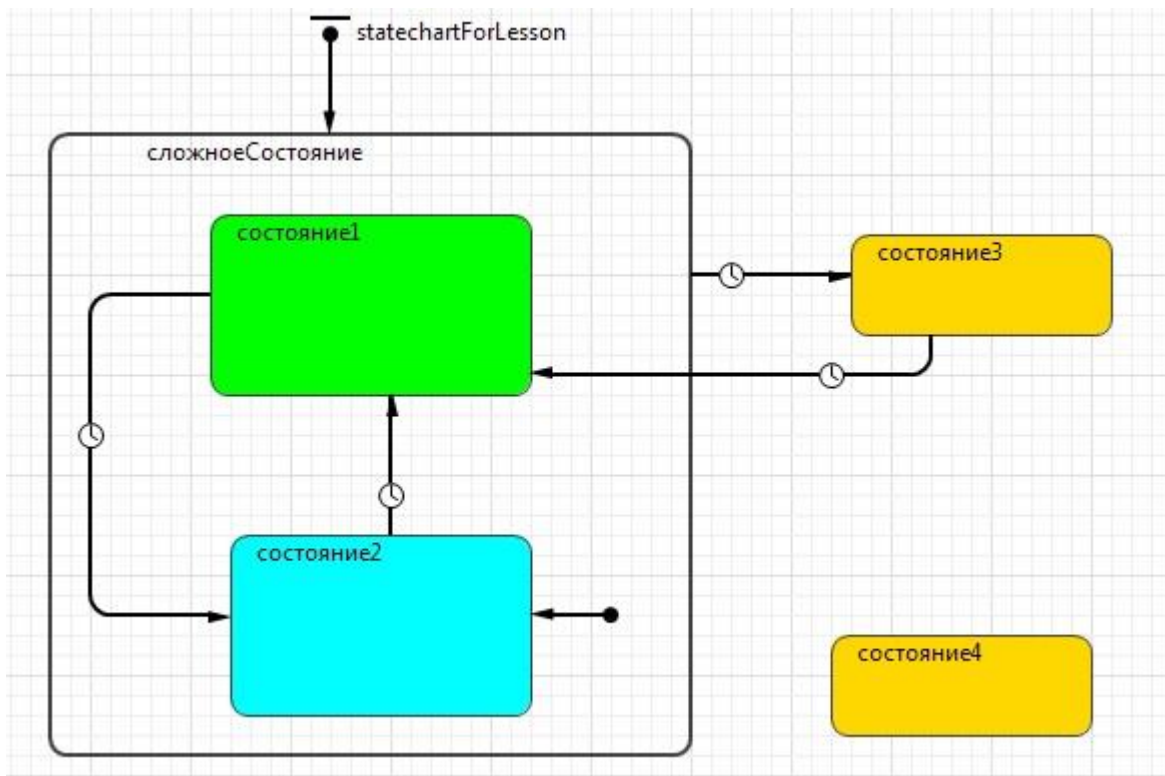


Рис.30

Если переход из состояния3 происходит в сложное состояние, то мы переходим в состояние2

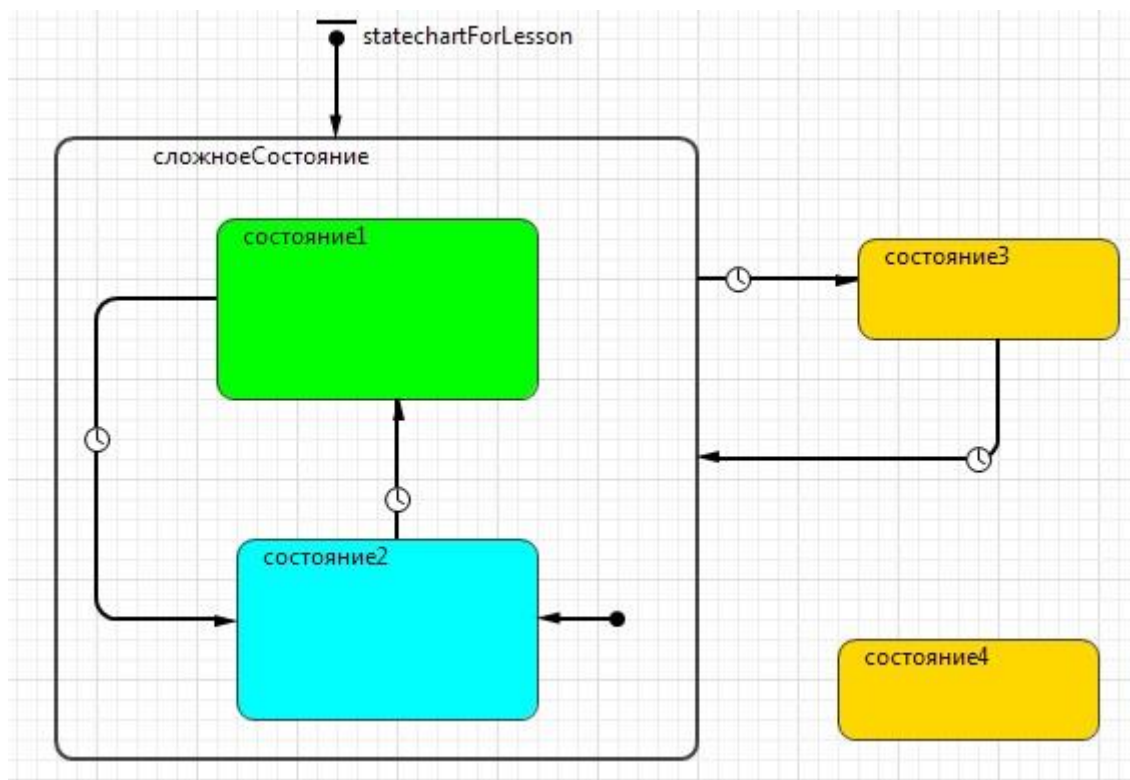


Рис.31

Если переход из состояния3 будет происходить к состоянию1, как показано на рис.32, тогда мы из состояния3 перейдем в состояние1

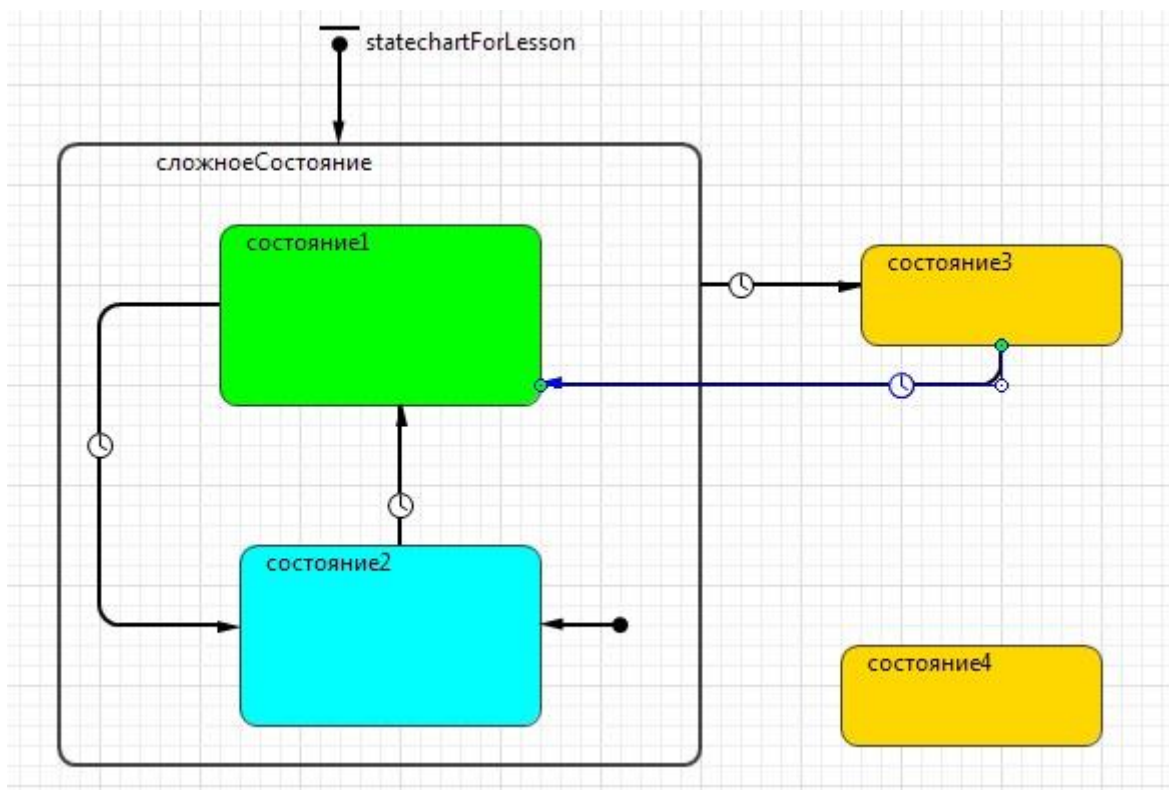


Рис.32

Состояние4 сделаем наоборот. Мы можем нарисовать переход из состояния2 в состояние4 при получении какого либо сообщения.

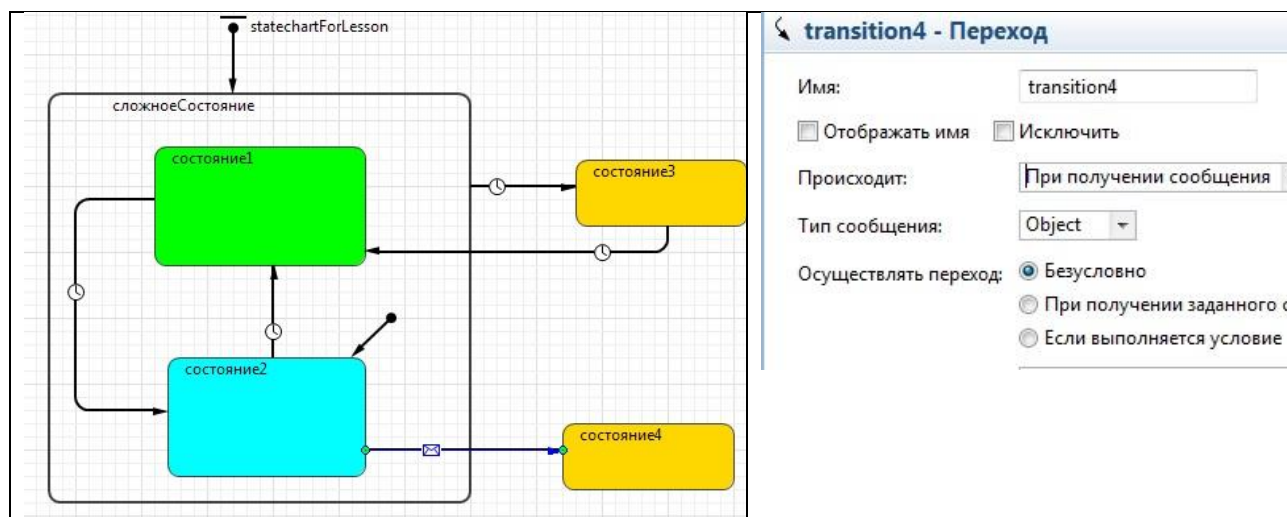


Рис.33

А обратно мы уже можем двигаться вот сюда

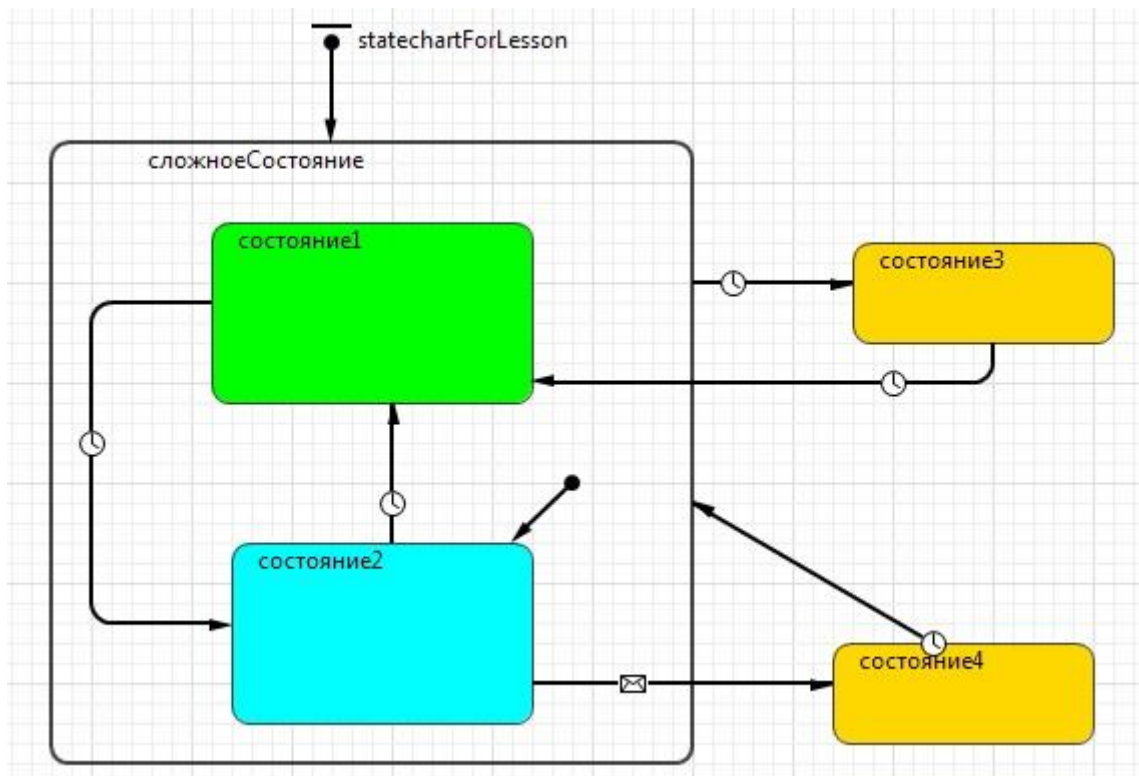


Рис.34

Нарисуем более красиво

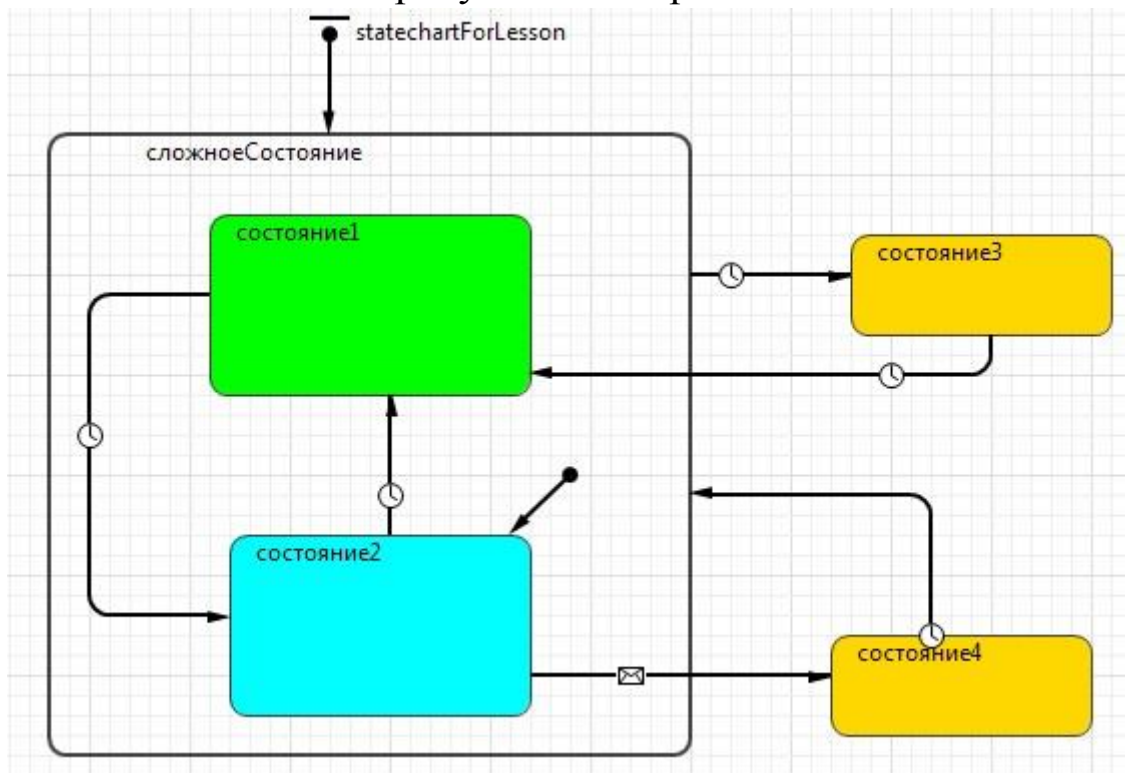


Рис.35

Из состояния2 мы по таймауту переходим в состояние1, но если агент получил в состоянии2 сообщение, то мы переходим в

состояние4, а затем переходим по таймауту в сложное состояние, а исходным в сложном состоянии является состояние2.

Есть еще одно состояние, которое называется Историческое состояние. Суть его в следующем: это по сути псевдо состояние, не настоящее состояние, это ссылка на последнее посещенное состояние внутри составного состояния. Исторические состояния используются для моделирования возврата к ранее прерванной деятельности. Например, видоизменим схему, введя историческое состояние:

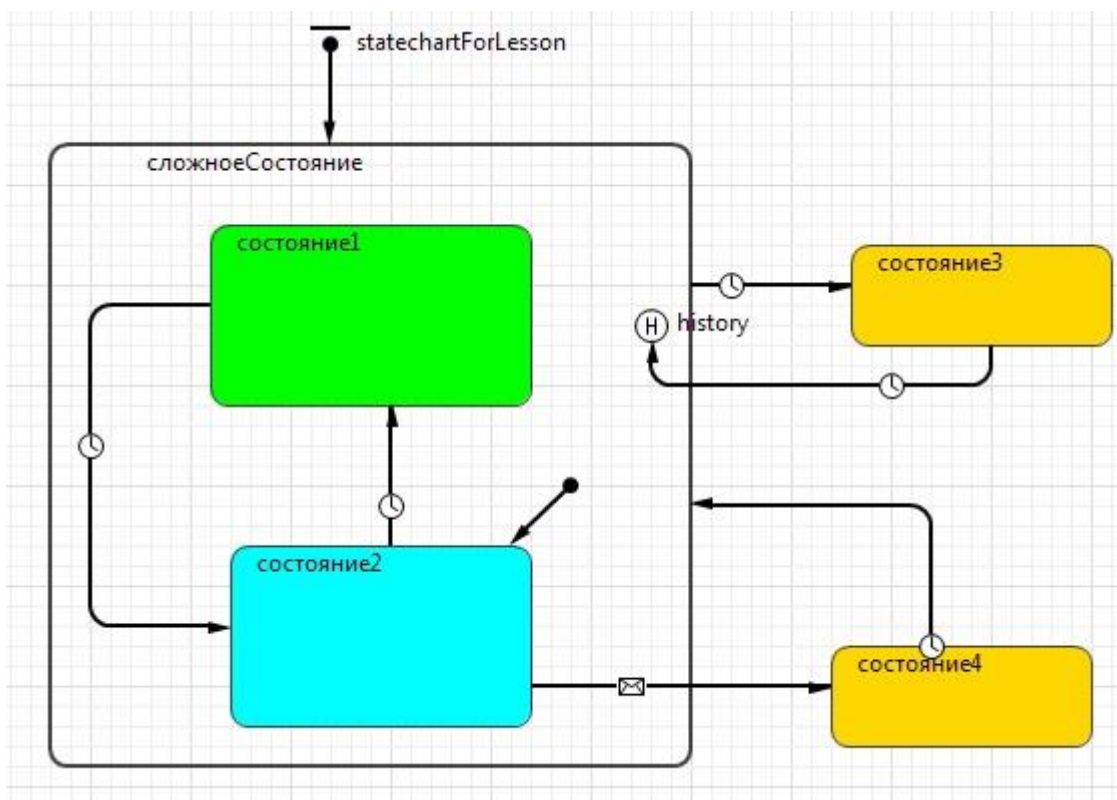


Рис.36

Если из состояния3 мы переходим в сложное состояние, т.к. Историческое состояние находится внутри сложного. Когда мы попадаем в Историческое состояние, диаграмма состояний вспоминает, где она была в последний раз. Допустим она вспоминает, что она была в состоянии1, и возвращает нас в состояние1.

Еще один пример. Расширим состояние2 до сложного

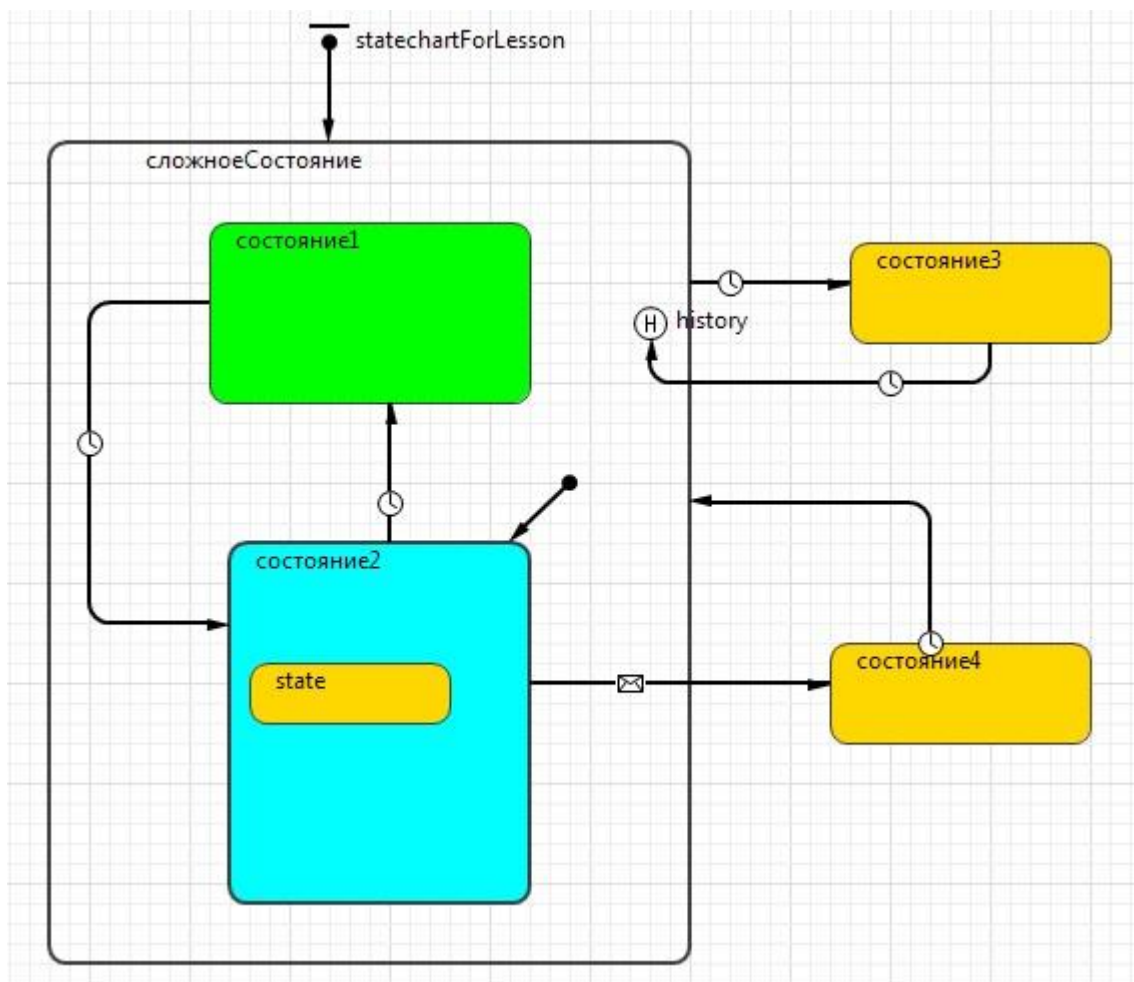


Рис.37

Введем 2 состояния внутри состояния2, назовем их состоянием5 и состоянием6 и сделаем переходы между состоянием5 и состоянием6.

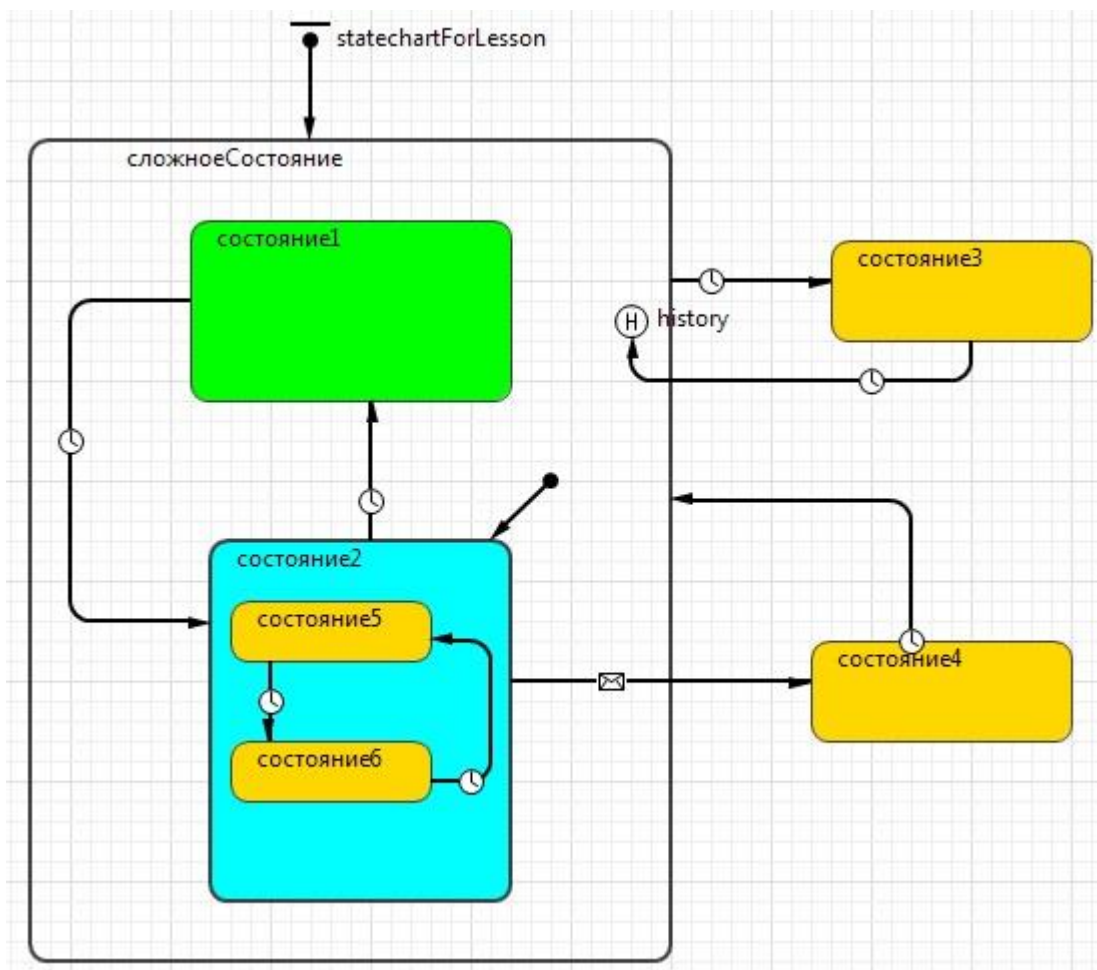


Рис.38

Состояние бывает 2-х видов: глубокое и поверхностное.

Т.к. состояние2 теперь сложное, необходимо добавить указатель начального состояния.

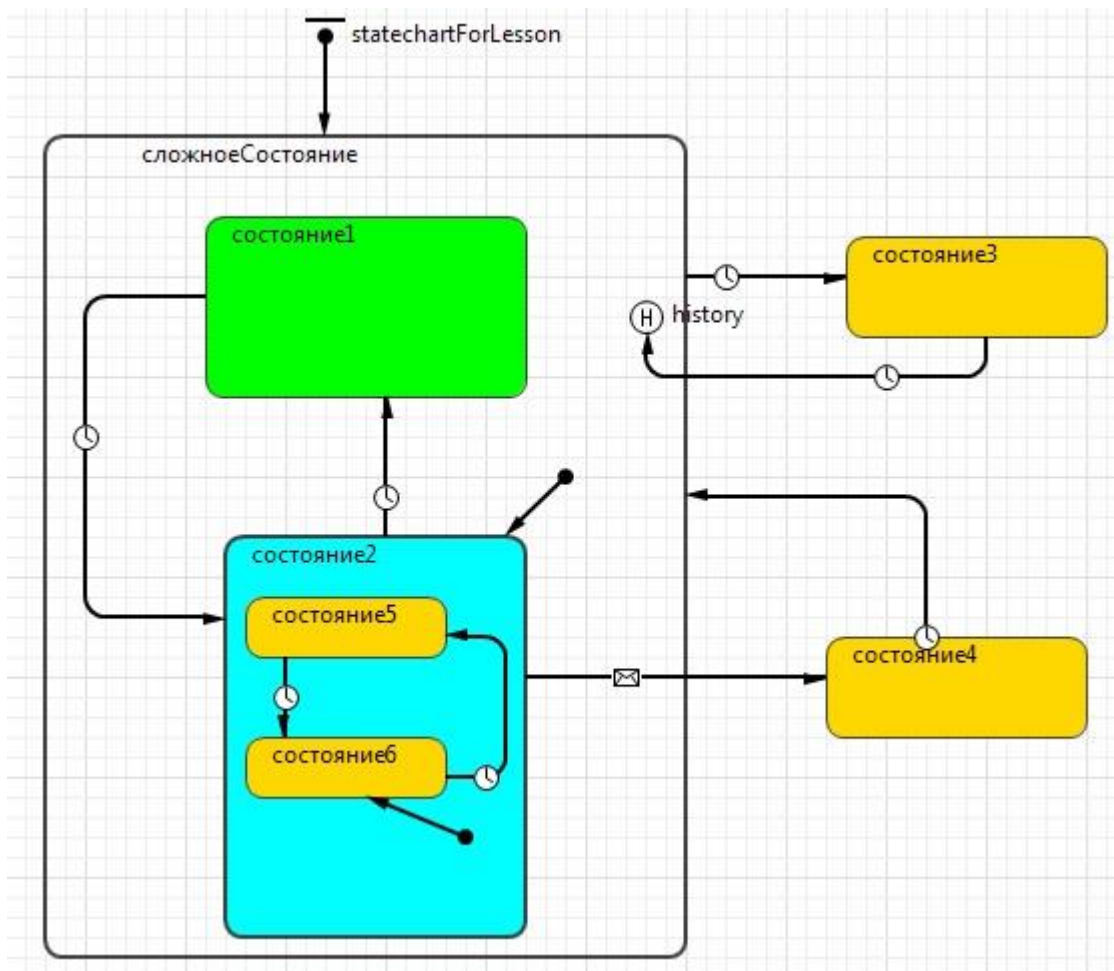


Рис. 39

Поверхностное состояние предполагает, что мы запоминаем состояние только внутри этого состояния. Если мы Историческое состояние делаем поверхностным. Поверхностное состояние предполагает, что мы запоминаем состояние только внутри этого состояния.

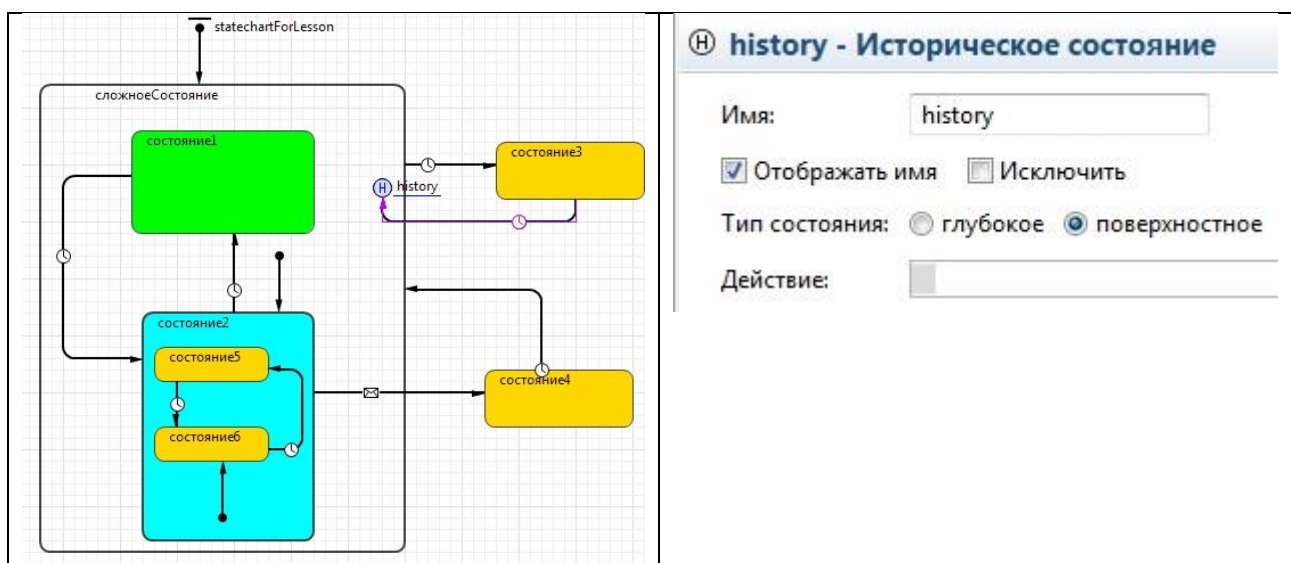


Рис.40

Если мы берем глубокое состояние, то мы запоминаем состояние вплоть до самой глубокой ступени генерации.

Если Историческое состояние мы делаем поверхностным, то мы запоминаем в каком из двух состояний мы находились: состояние1 или состояние2.

Если Историческое состояние мы делаем глубоким, то если мы находились в состоянии2, то мы запоминаем в каком состоянии 5 или 6 мы находились. Это псевдо состояние, это ссылка на последнее посещенное состояние.

Еще один элемент это конечное состояние.

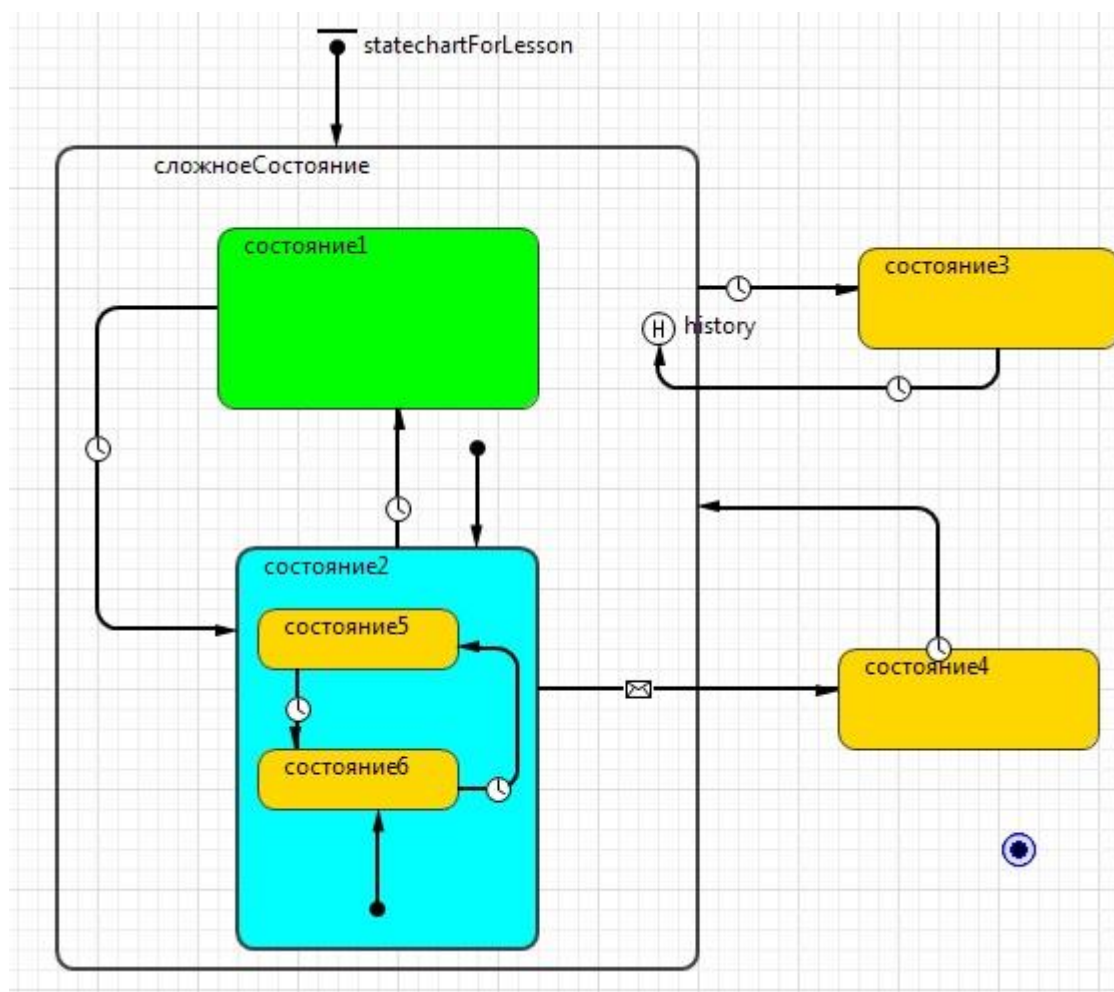


Рис.41

Если Начальное состояние является обязательным, то Конечное состояние не является обязательным. Конечное состояние нам говорит, что моделирование завершено.

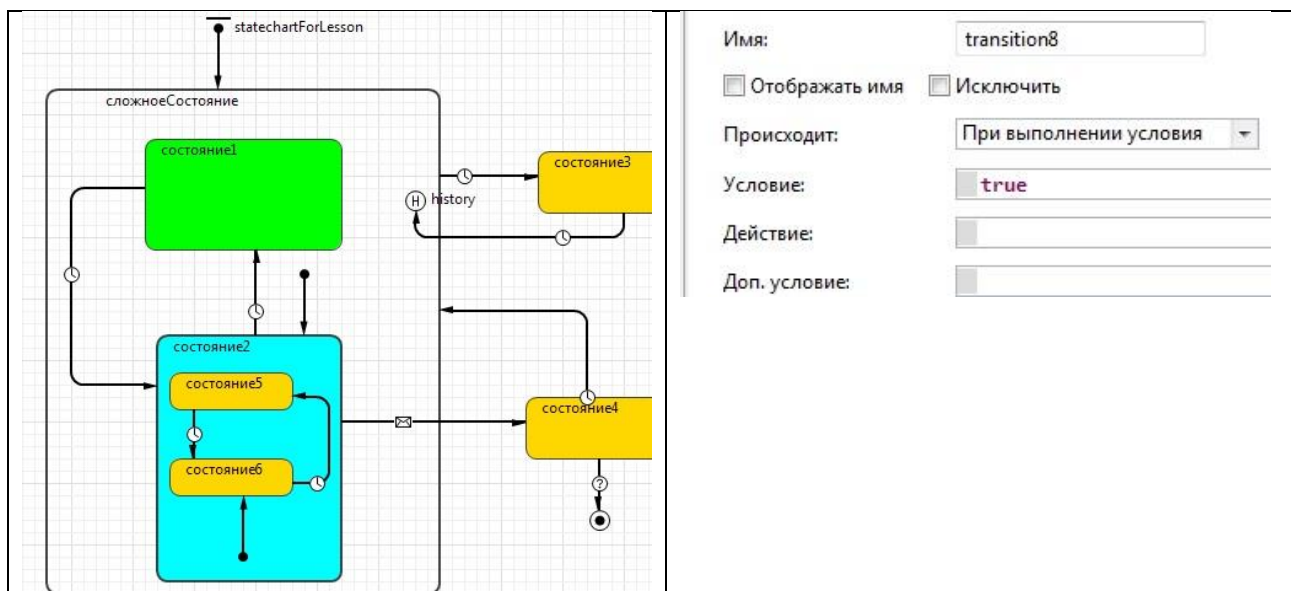


Рис.42

Конечное состояние может располагаться в любом месте
Например,

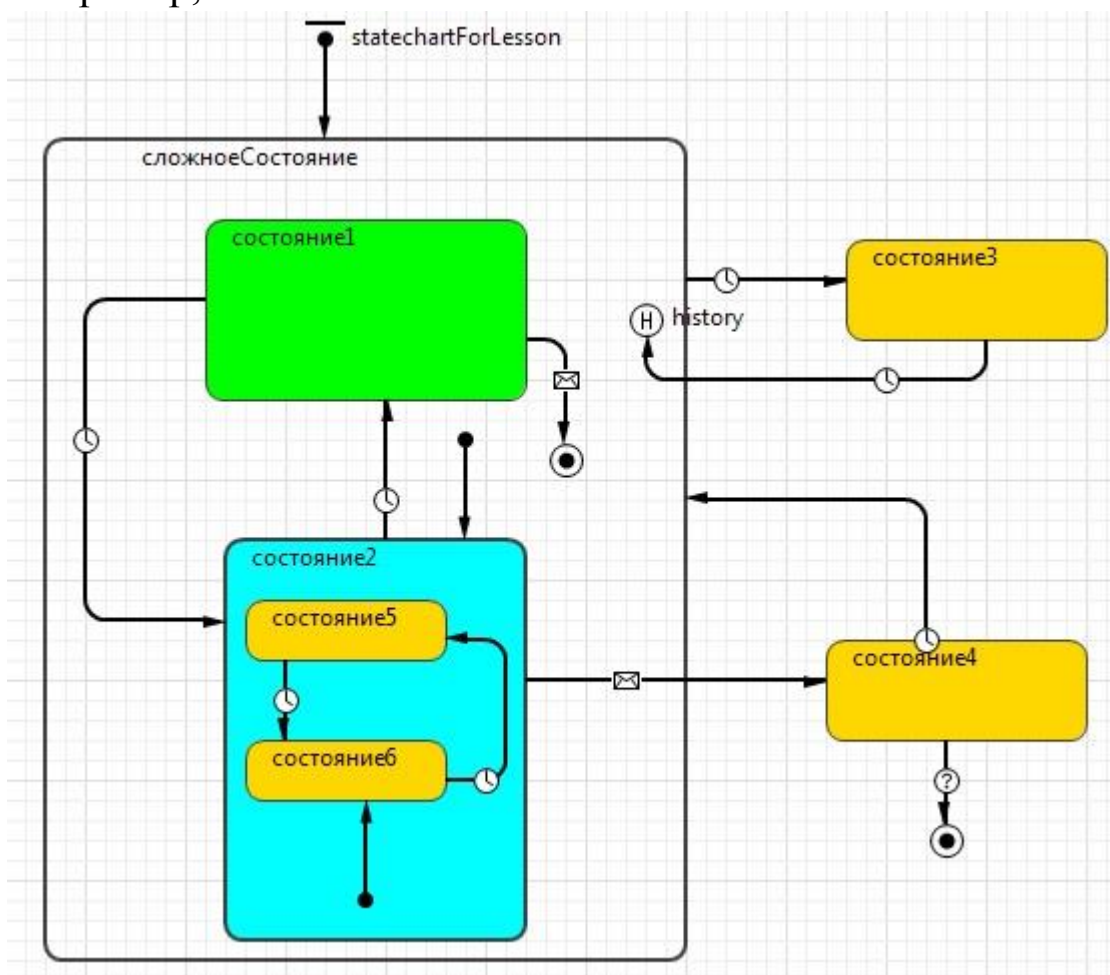


Рис.43

Какие переходы активны?

Активные состояния и активные переходы

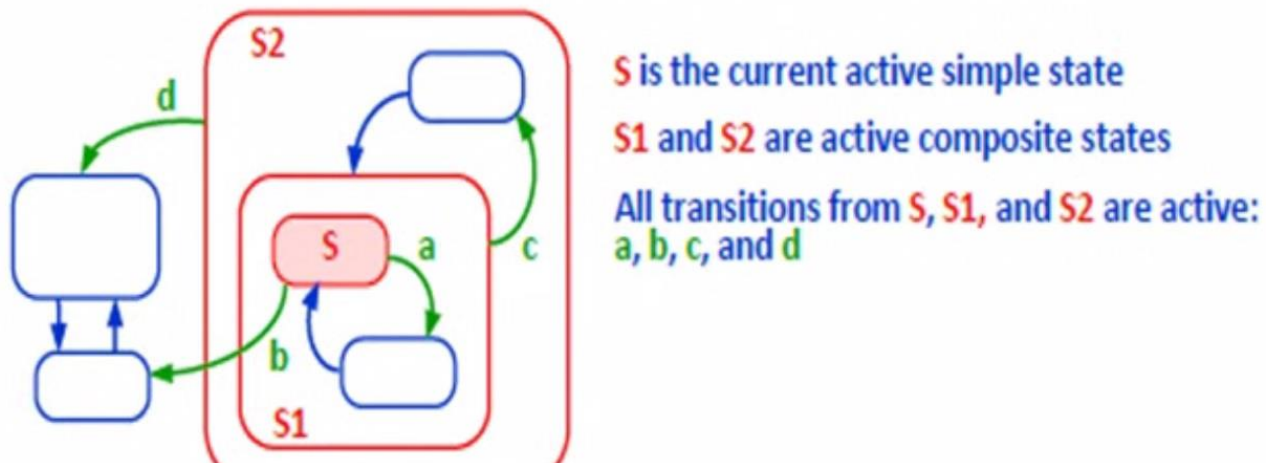


Рис.44

Посмотрим на рис.44. Что мы видим. В каждый момент времени существует одно простое состояние. В данном случае это состояние **S**.

S находится внутри состояния **S1**. А состояние **S1** находится внутри состояния **S2**. Следовательно **S1** и **S2** тоже являются активными состояниями. Соответственно все переходы, выходящие из активного состояния, являются активными. Активные переходы отмечены зеленым цветом. Активный переход, что это значит. Каждый переход имеет 5 вариантов действий (по таймауту, с заданной интенсивностью, при выполнении условия, при получении сообщения, по прибытии агента). Все переходы осуществляются, когда переход является активным.

Подробное описание всех переходов можно посмотреть в справке(рис.45)

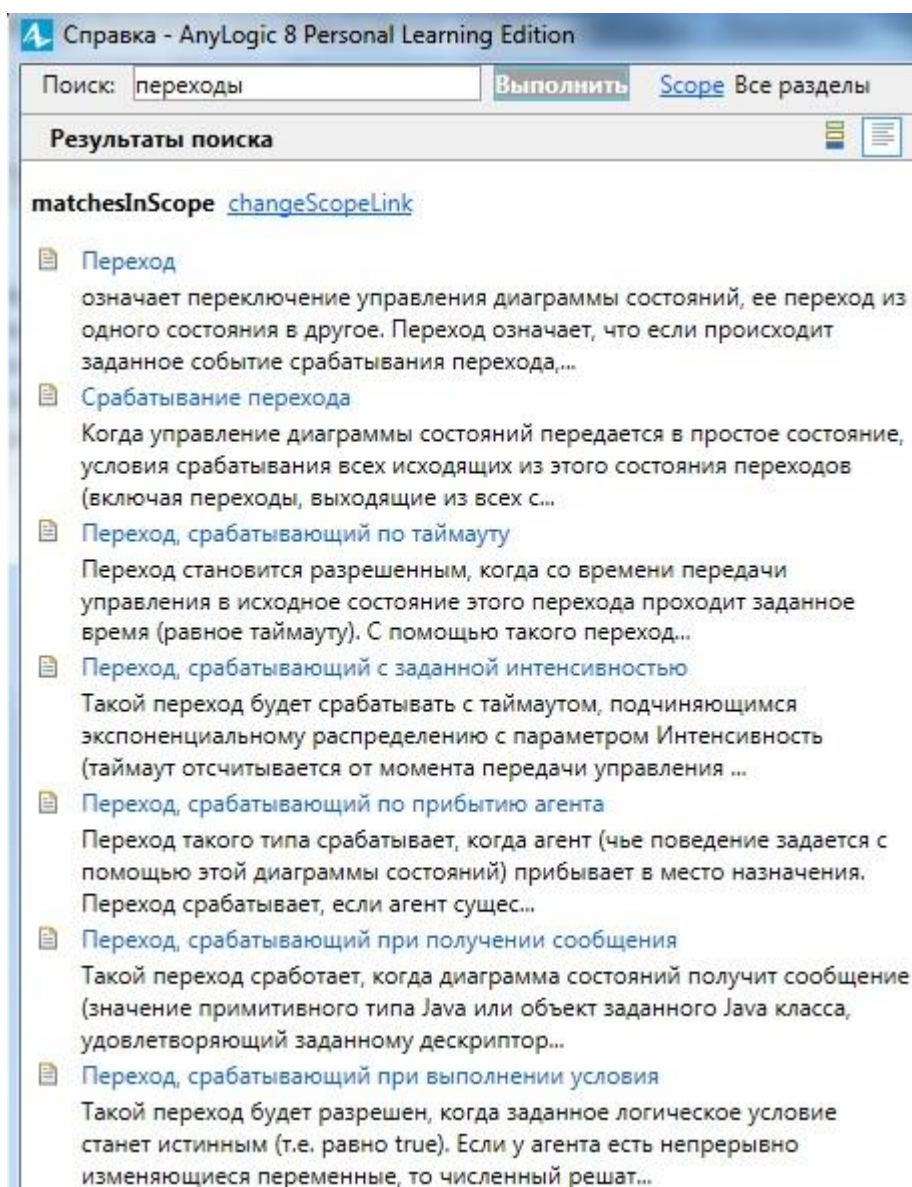


Рис.45

Самый простой переход по таймауту, т.е. каждые, например, 15 сек Из состояния6 мы перейдем в состояние5 через 15 сек. При переходе может что-то произойти. Пример, в поле действия можно задать какую-нибудь операцию. Для примера, заведем некоторую переменную x (рис.47)

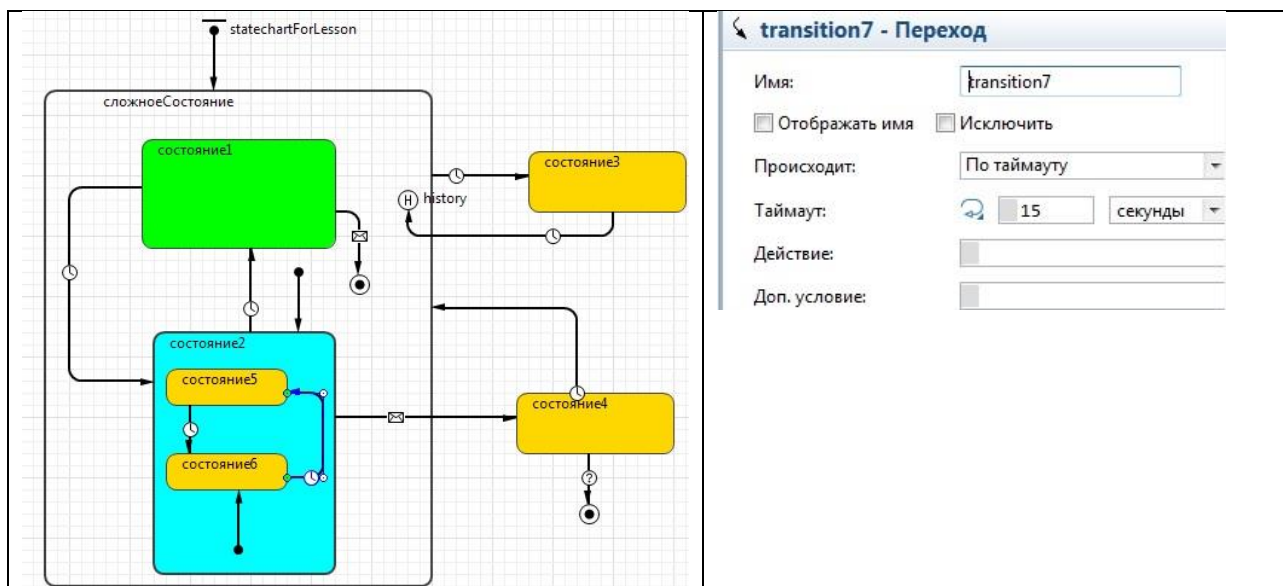


Рис.46

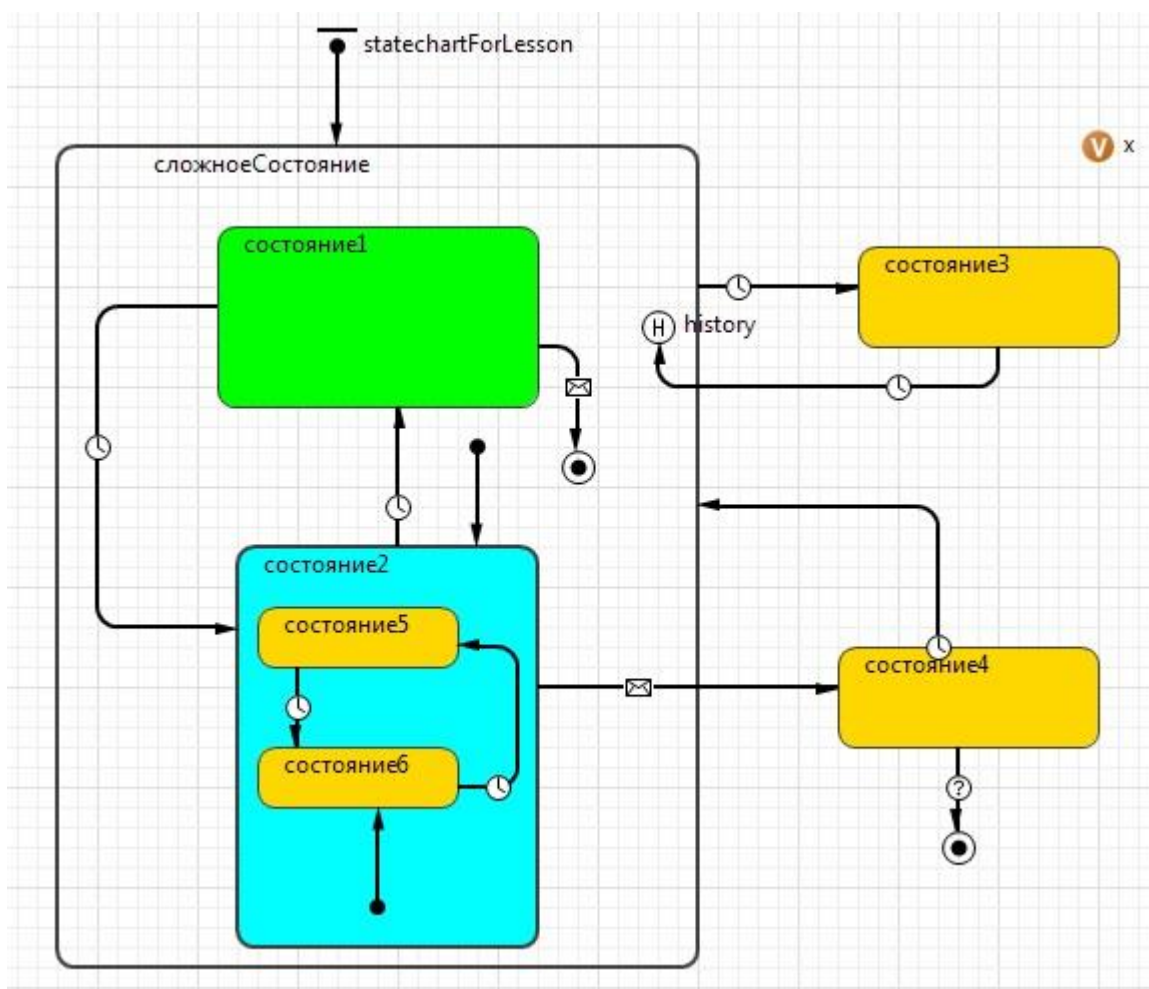


Рис.47

Из состояния6 в состояние5 мы никогда не перейдем. Почему ?
 Второй тип переходов с заданной интенсивностью (рис 48). Здесь задаётся скорость перехода, т.е. переход будет выполняться каждую сек.

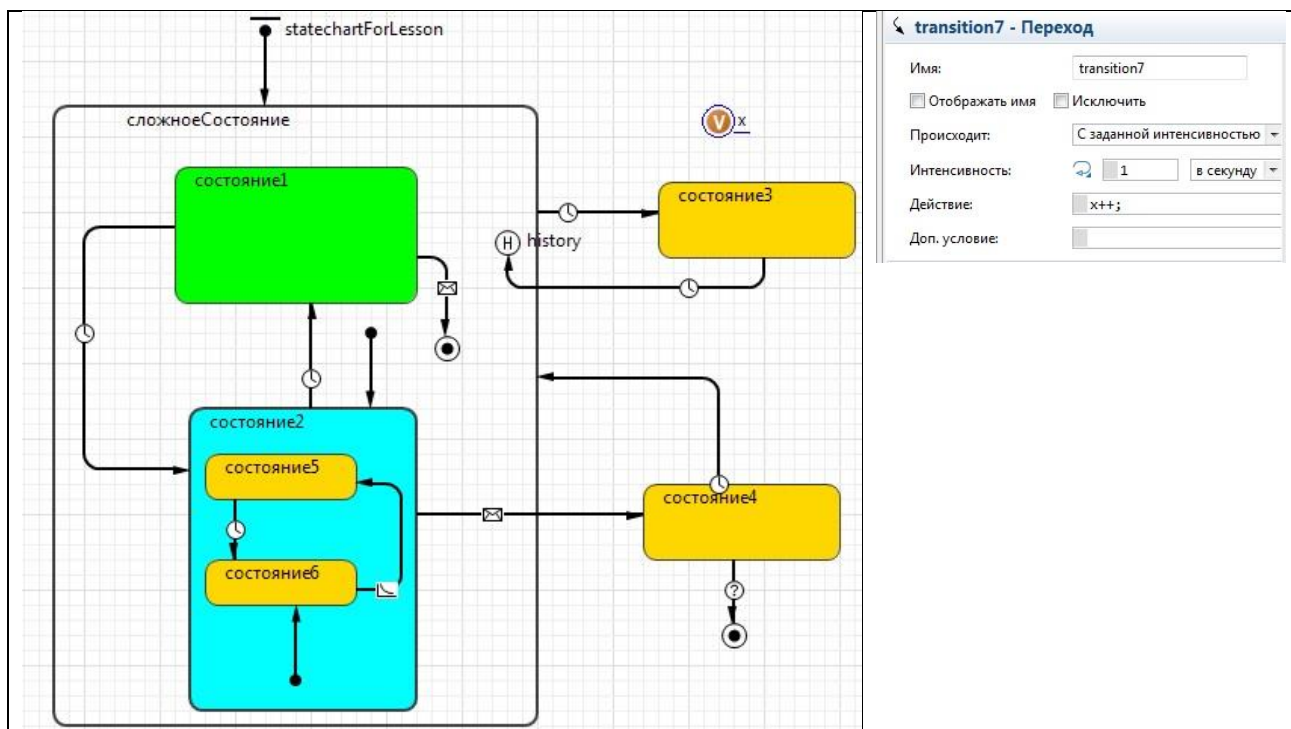


Рис.48

Третий тип переходов это при выполнении некоторого условия. Например, $x=5$

Четвертый тип переходов это при получении сообщения, Мы можем посылать сообщения друг другу.

Пятый тип переходов при получении агента. Этот переход работает, когда мы моделируем агентное поведение. Более того передвижение агентов куда-либо. И как только мы приходим в конкретную точку, этот переход срабатывает.

Рассмотрим пример пересылки сообщений. Для этого модифицируем схему как показано на рис.49-54

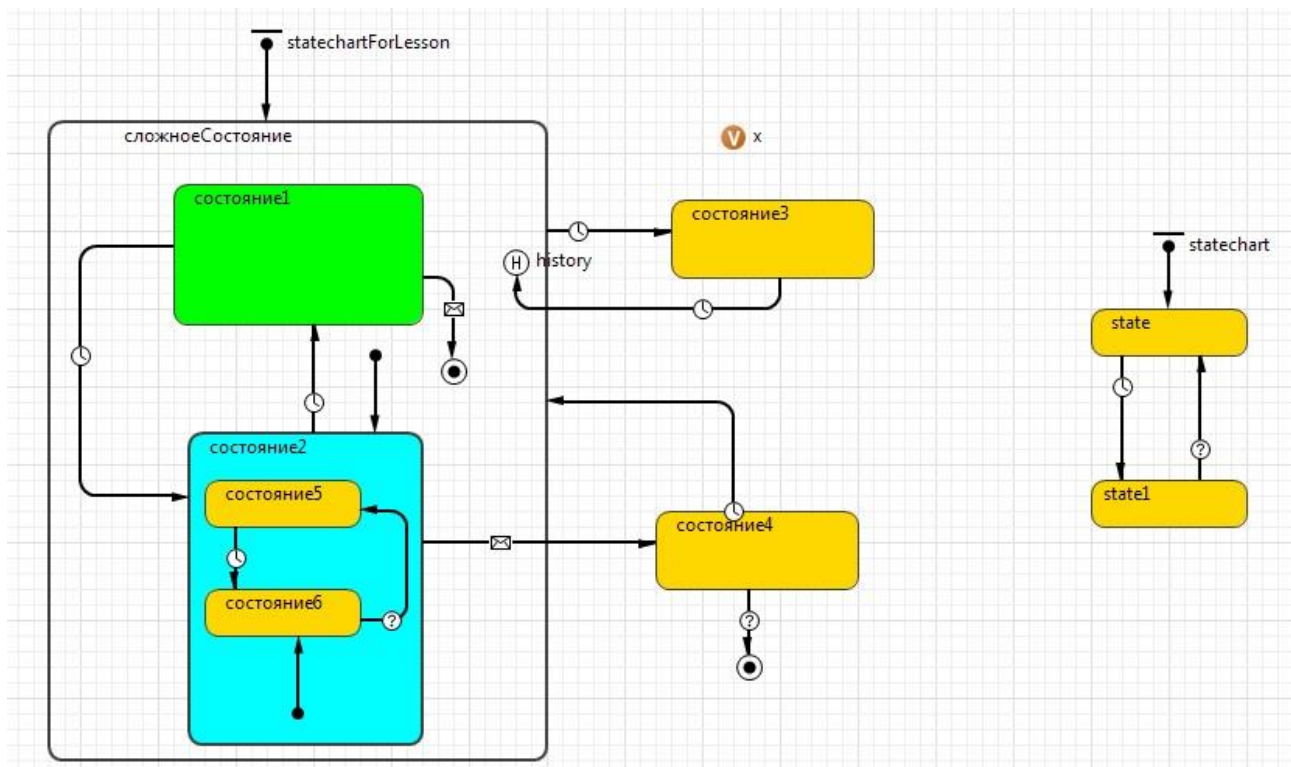


Рис.49

state - Состояние

Имя: ☒ Отображать имя ☐

Цвет заливки:

Действие при входе:

Действие при выходе:

Описание

Рис.50

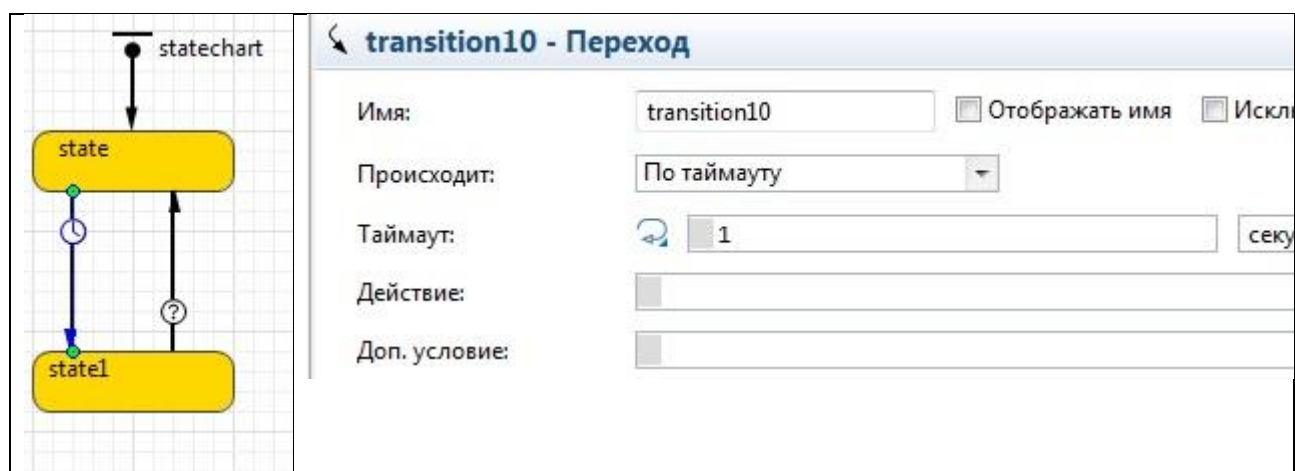


Рис.51

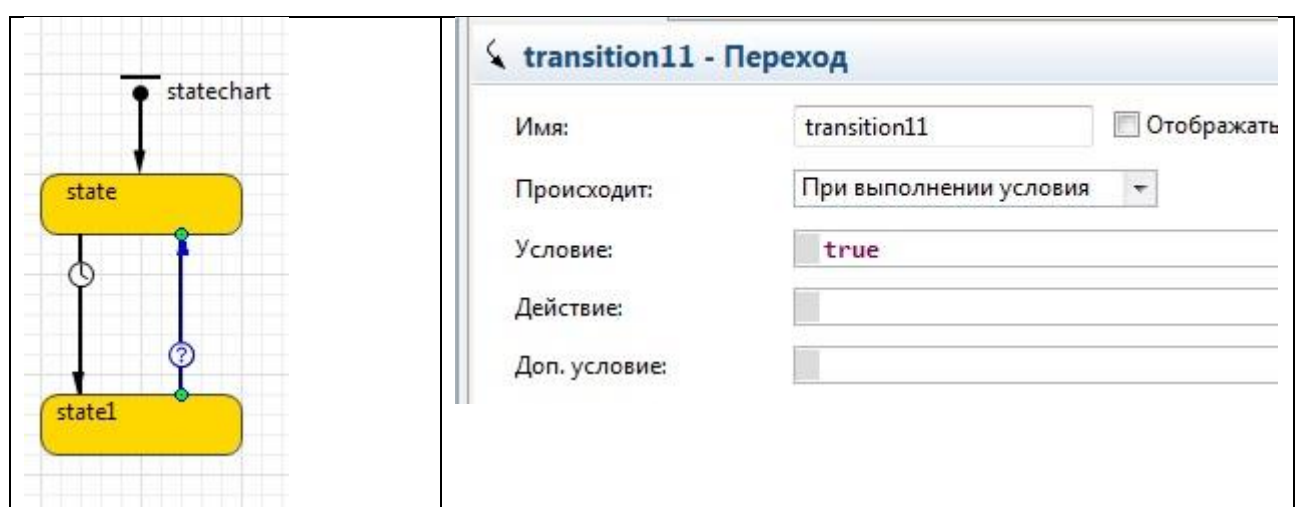


Рис.52

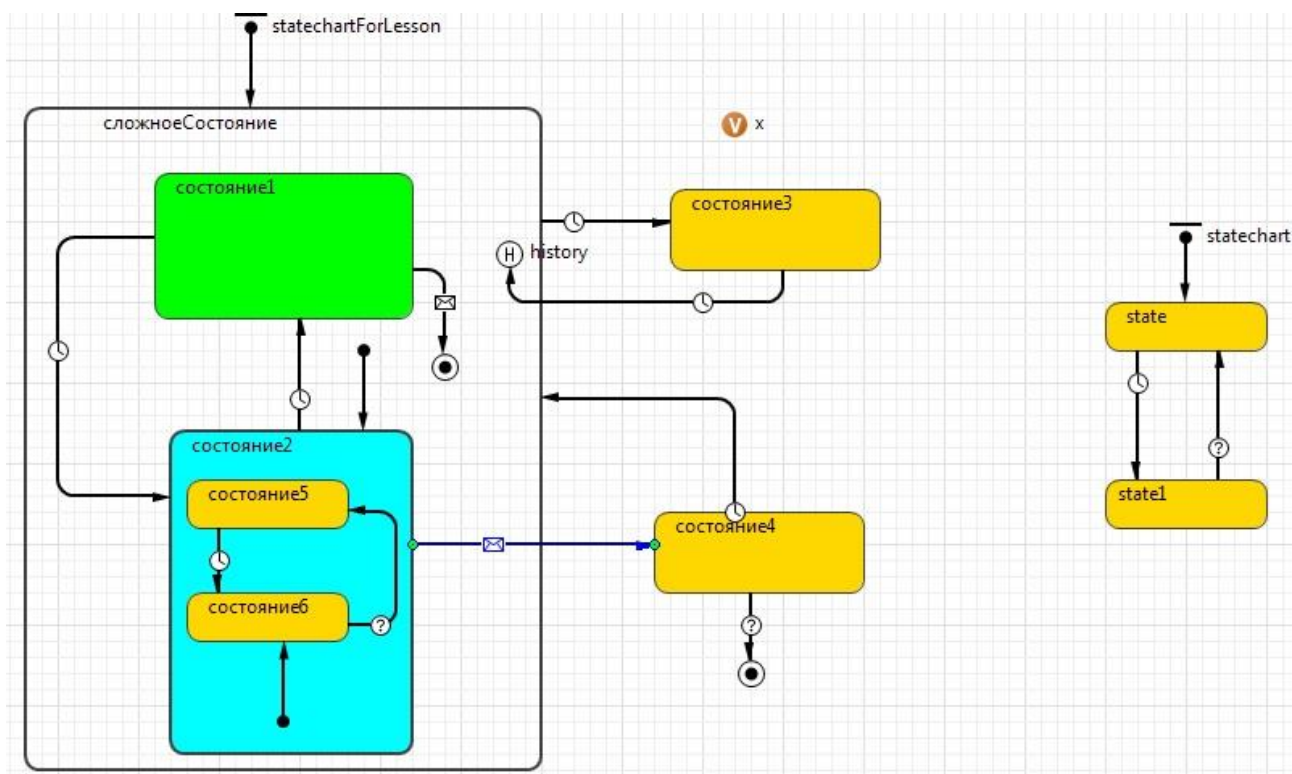


Рис.53

transition4 - Переход

Имя: ☐ Отображать имя ☐ Искл

Происходит:

Тип сообщения:

Осуществлять переход: ☐ Безусловно ☒ При получении заданного сообщения: ☐ Если выполняется условие

Сообщение:

Действие:

Доп. условие:

Рис.54

Можно не анализировать тип сообщения. Для этого (рис.55)
В свойствах пропишем осуществлять переход безусловно (рис.56)

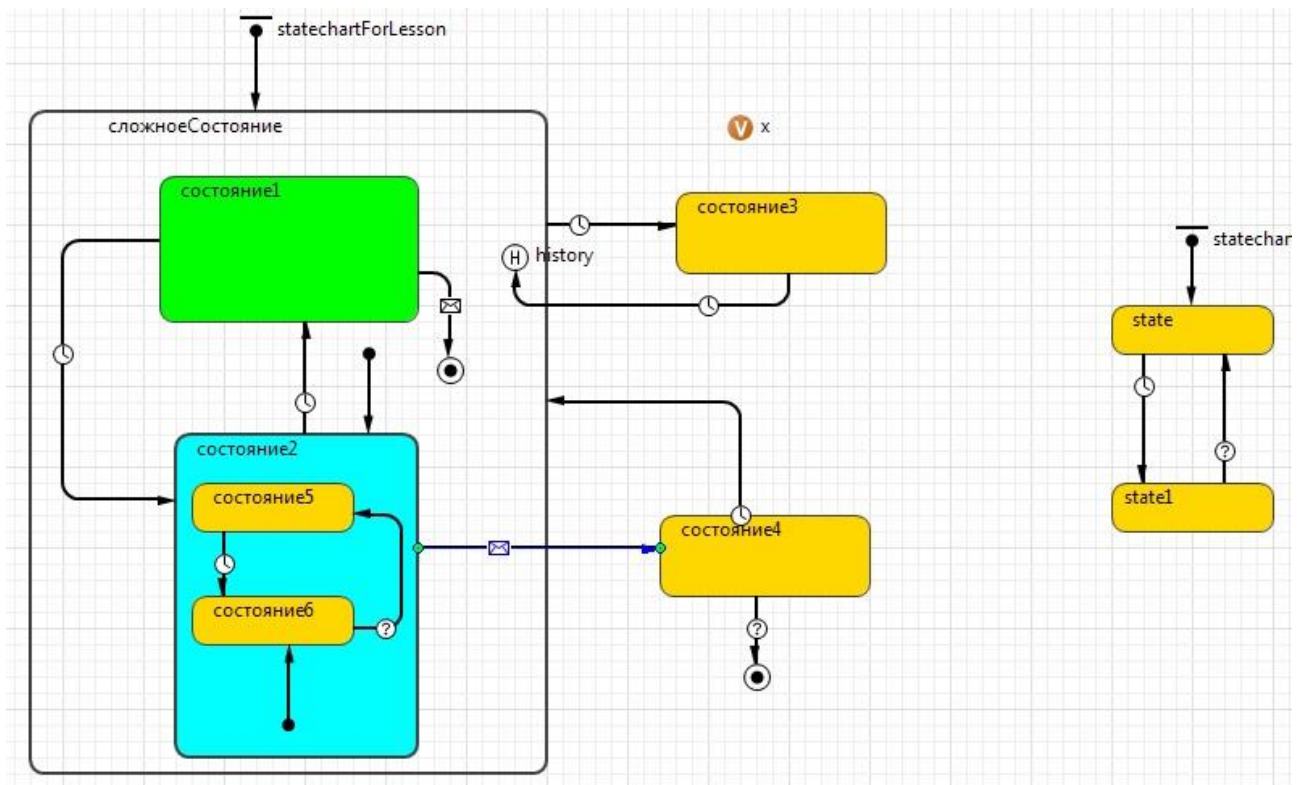


Рис.55

transition4 - Переход

Имя: ☐ Отображать имя

Происходит:

Тип сообщения:

Осуществлять переход: ☒ Безусловно
☐ При получении заданного сообщения:
☐ Если выполняется условие

Действие:

Доп. условие:

Рис.56

Обратите внимание: до сих пор мы не обращали внимание на дополнительное условие. Это так называемый защитник. Это дополнительное условие, которое проверяется когда переход готов к осуществлению. Например, мы получили сообщение и готовы перейти, но мы проверяем дополнительное условие и ждем пока дополнительное условие выполнится.

Еще один элемент диаграммы состояний, который мы не рассмотрели это Ветвление. Аналогии с блок схемами здесь нет, но символ похожий. Ветвление это по сути блок выбора.

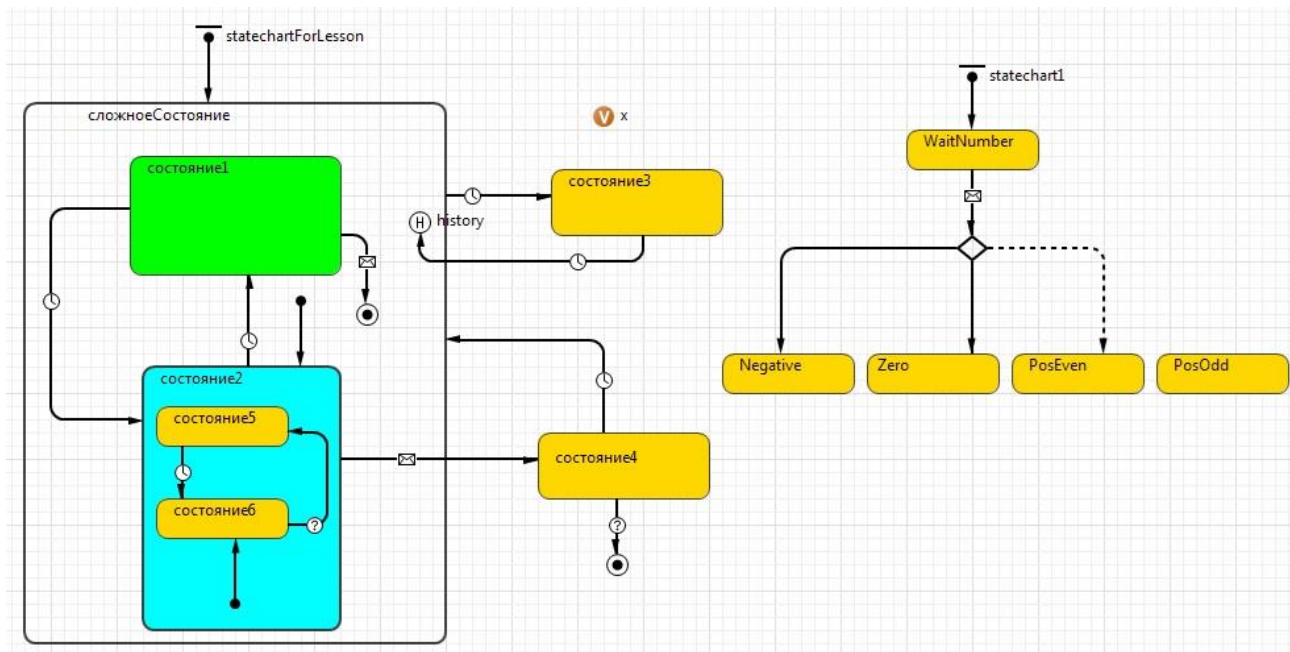


Рис. 57

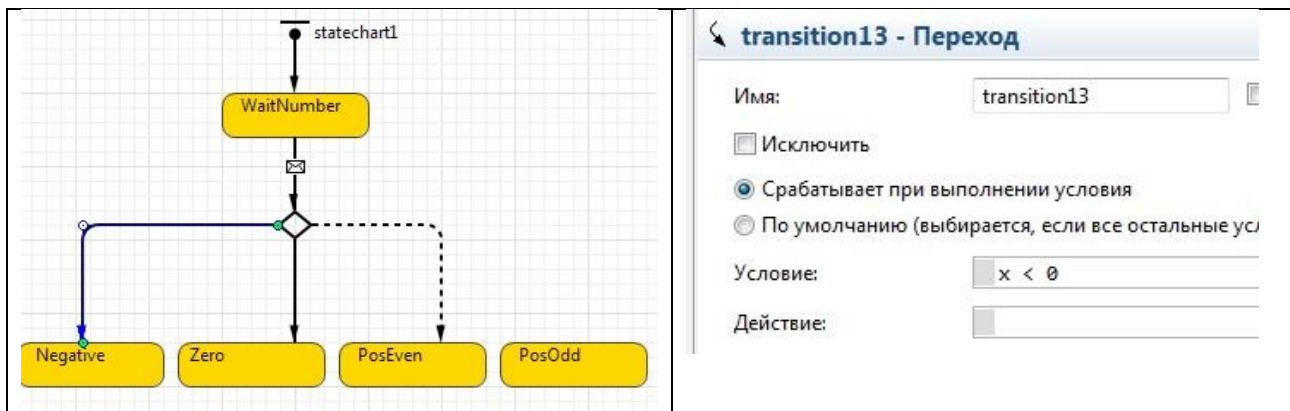


Рис.58

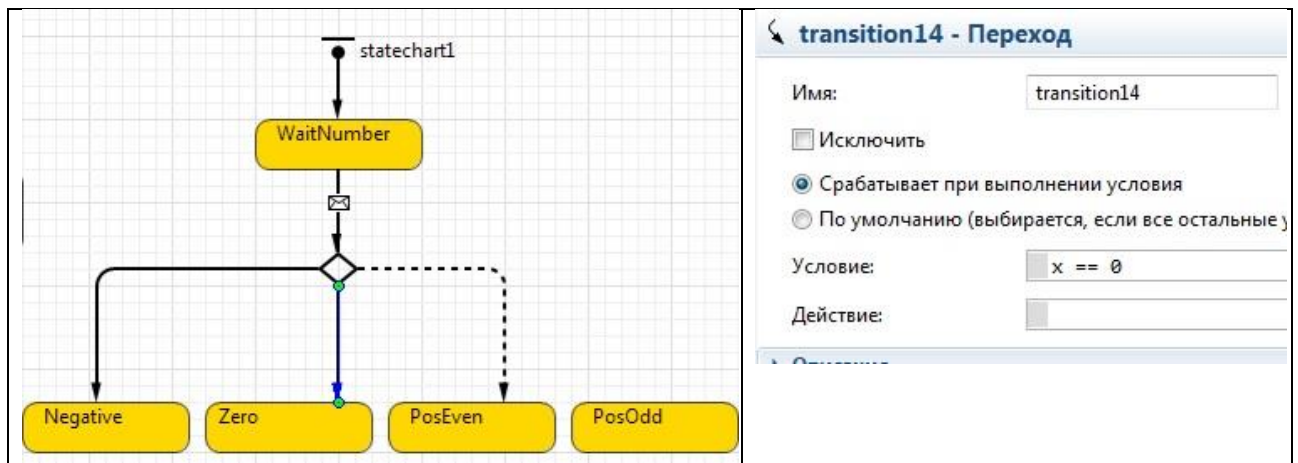


Рис.59

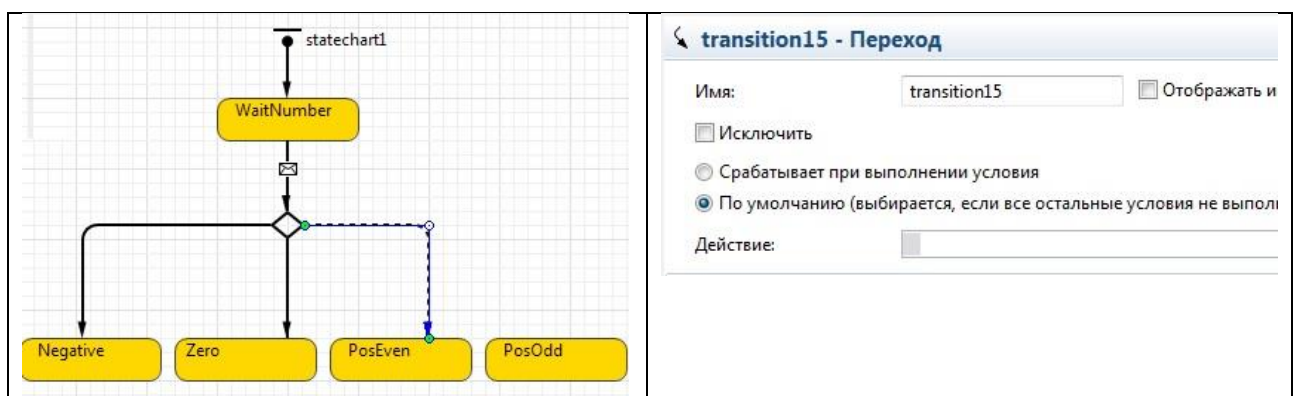


Рис.60

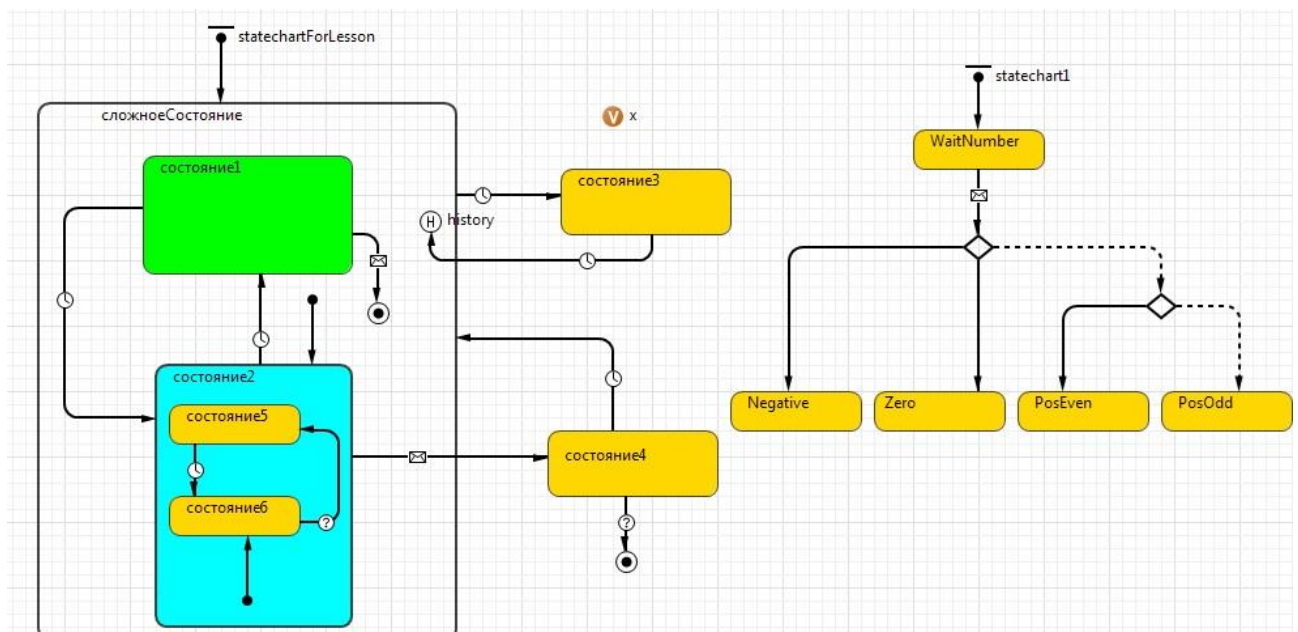


Рис.61

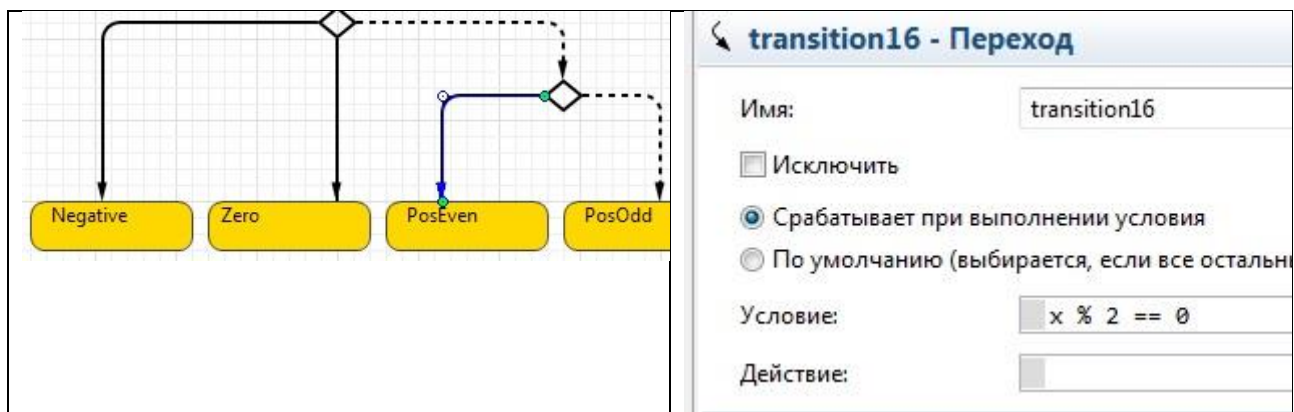


Рис.62

Внутренний переход. Осуществляется внутри какого состояния, не выходя за его пределы.

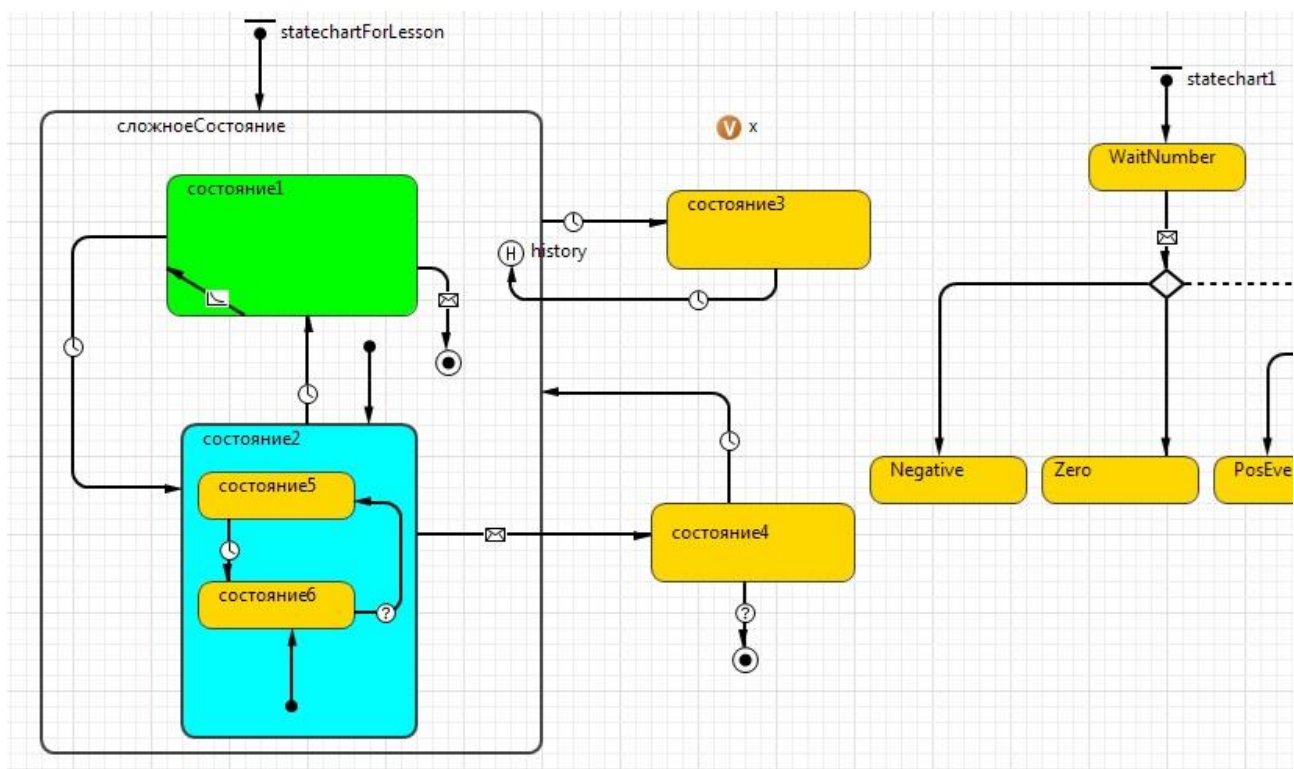


Рис.63

Например, свойства внутреннего перехода могут быть такими, как показано на рис.64

transition18 - Переход

Имя:

☐ Отображать имя ☐ Исключить

Происходит:

Интенсивность: в секунд

Действие:

Доп. условие:

Рис. 64

Пример: модель грузовика доставки

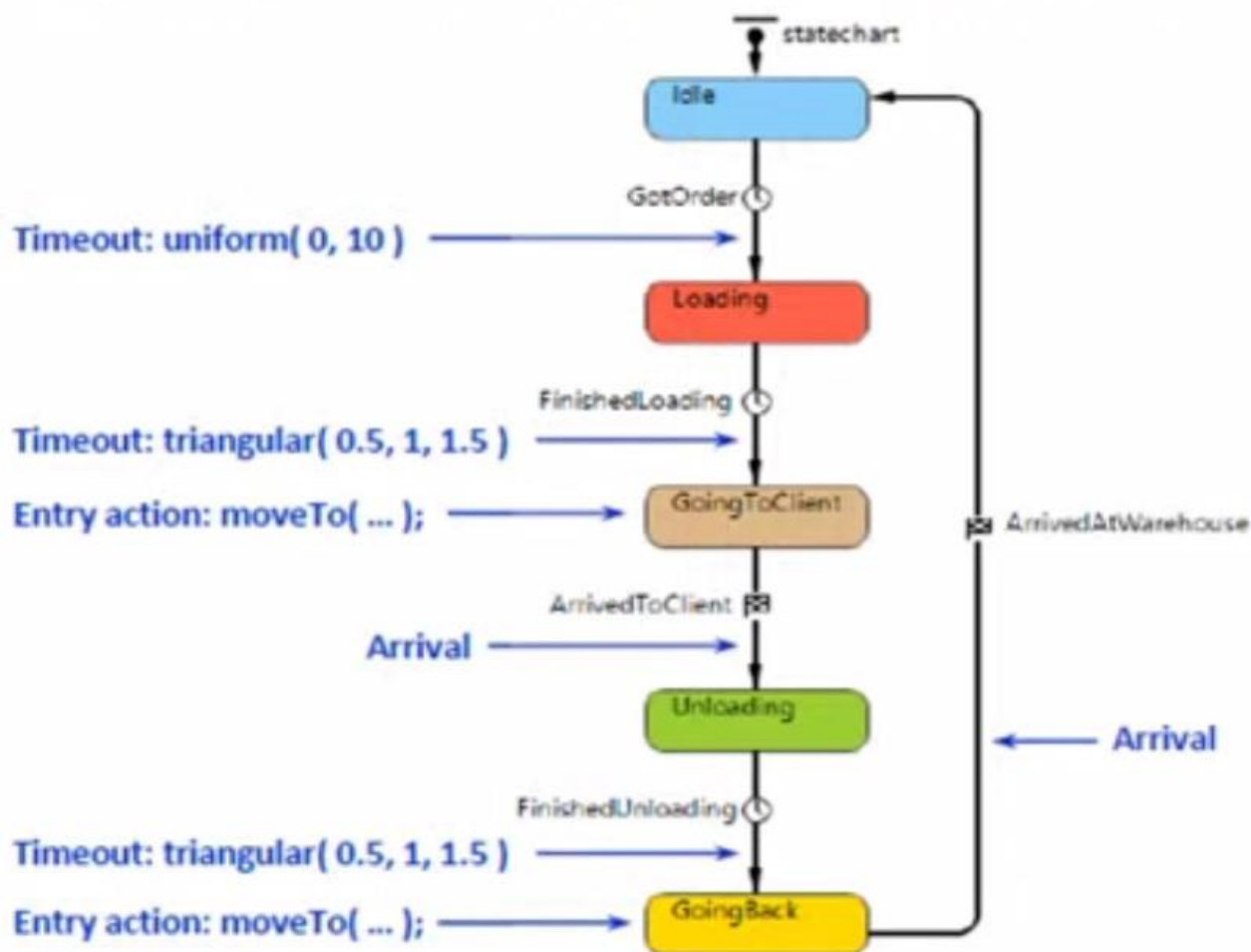


Рис.65

Пример: модель жизненного цикла продукта

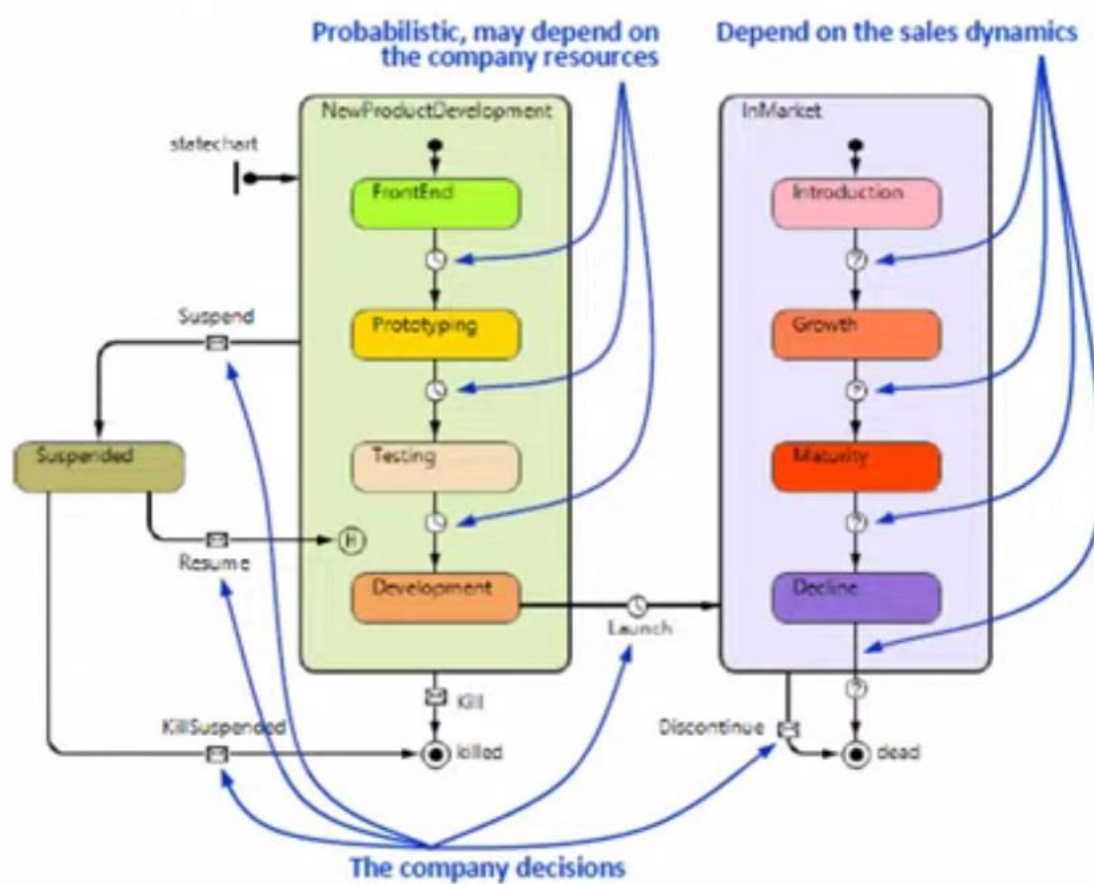


Рис. 66