Discrete Intra-Agent Dynamics: Statecharts

Nathaniel Osgood

MIT 15.879

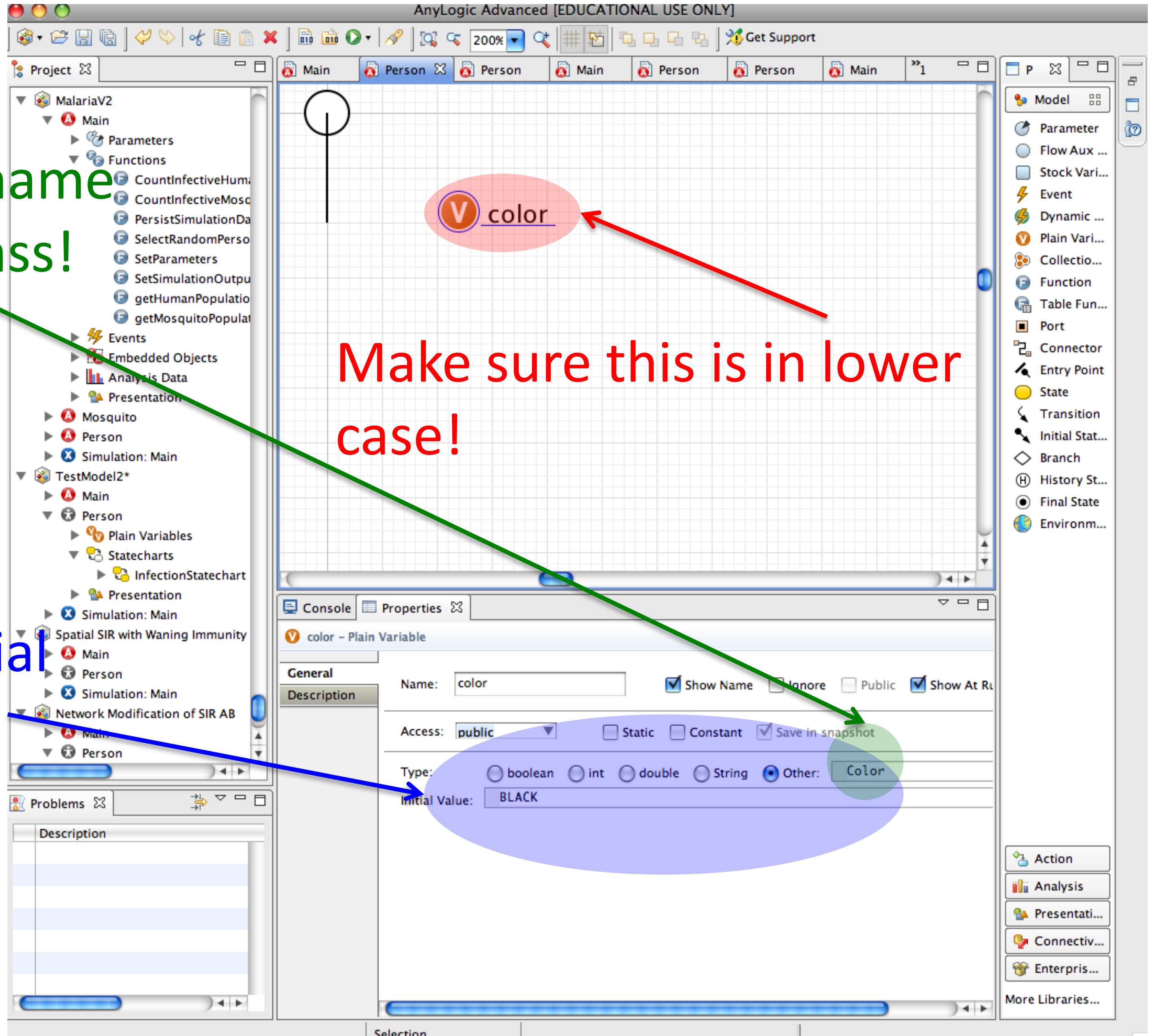
March 7, 2012

Hands on Model Use Ahead

Load Previous Built [& Provided] Model:

MinimalistNetworkABMModel

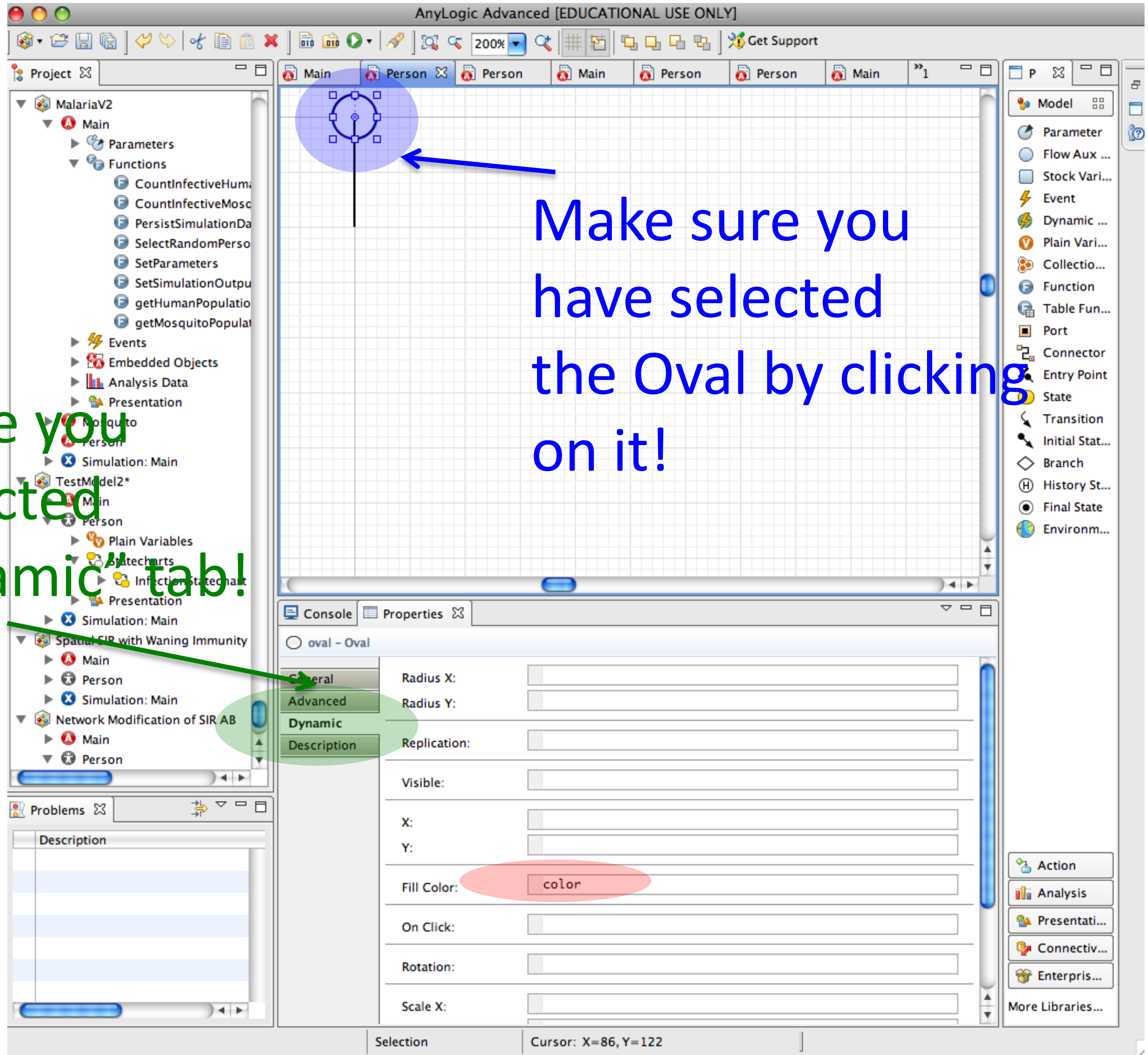
Adding "Color" Variable



This is the name of a Java class!

Fill in the type and Initial Value (watch for correct case!!)

Make Oval "Color" property Use Variable



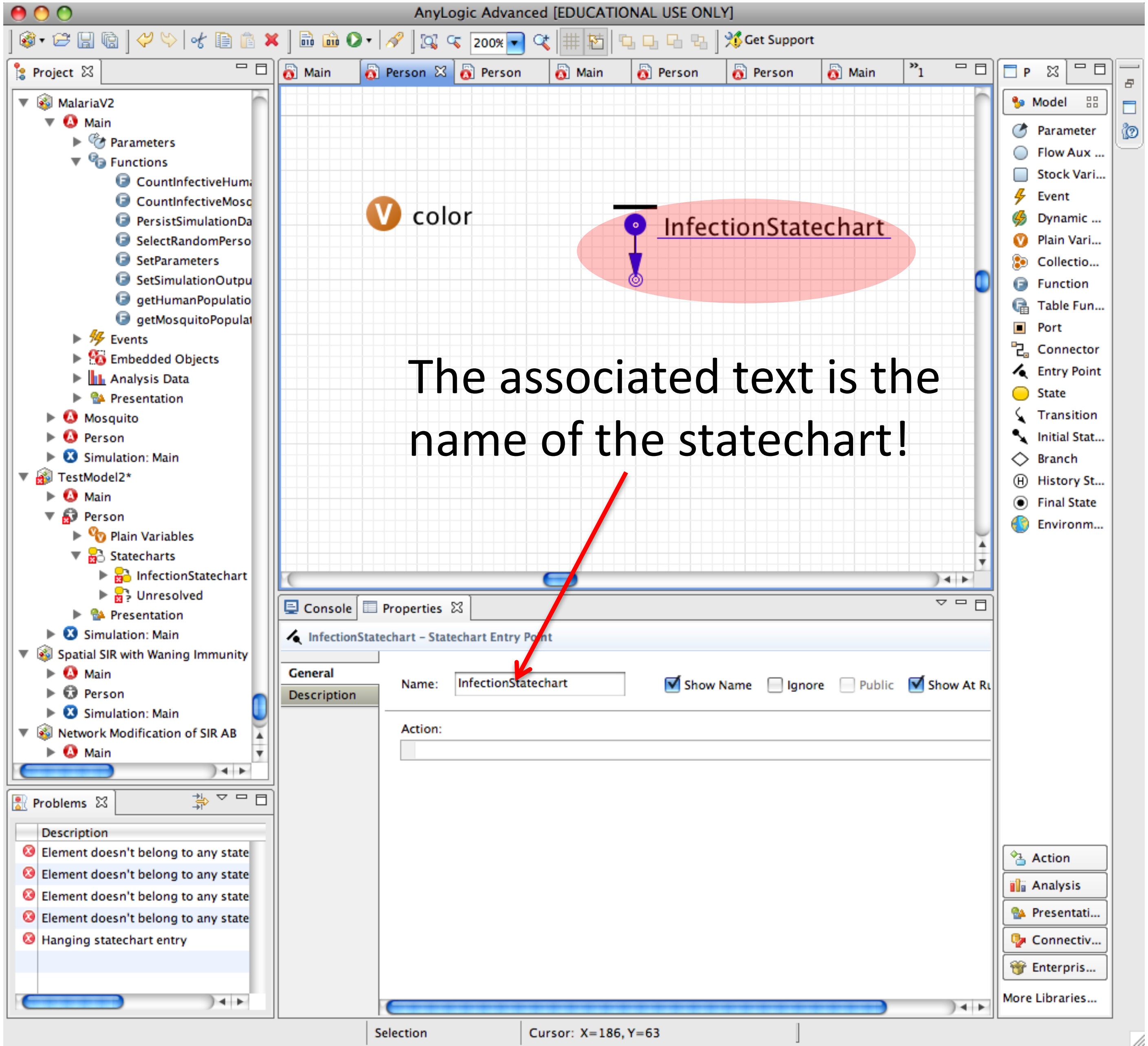
Make sure you have selected the Oval by clicking on it!

Make sure you have selected the "Dynamic" tab!

Discrete Agent Dynamics

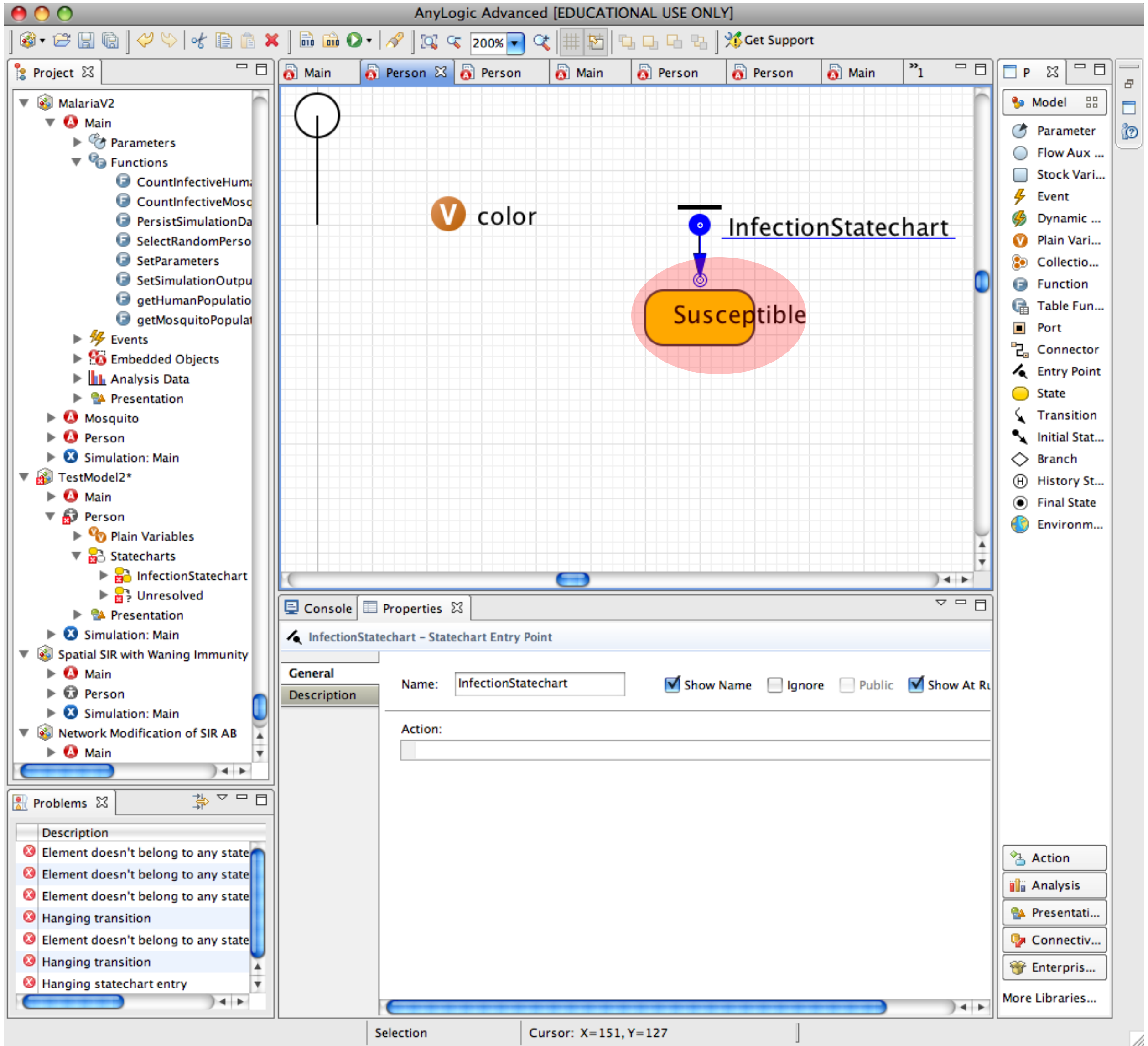
- Frequently we can represent agent behaviour using as transitioning among a set of mutually exclusive and collectively exhaustive states in a “state chart”
- For a given simple statechart, the agent is in exactly one state at a time
- Fixed transitions between states define possible evolution
- The transitions between states occur instantaneously, based on some condition

Add Entry Point of State chart

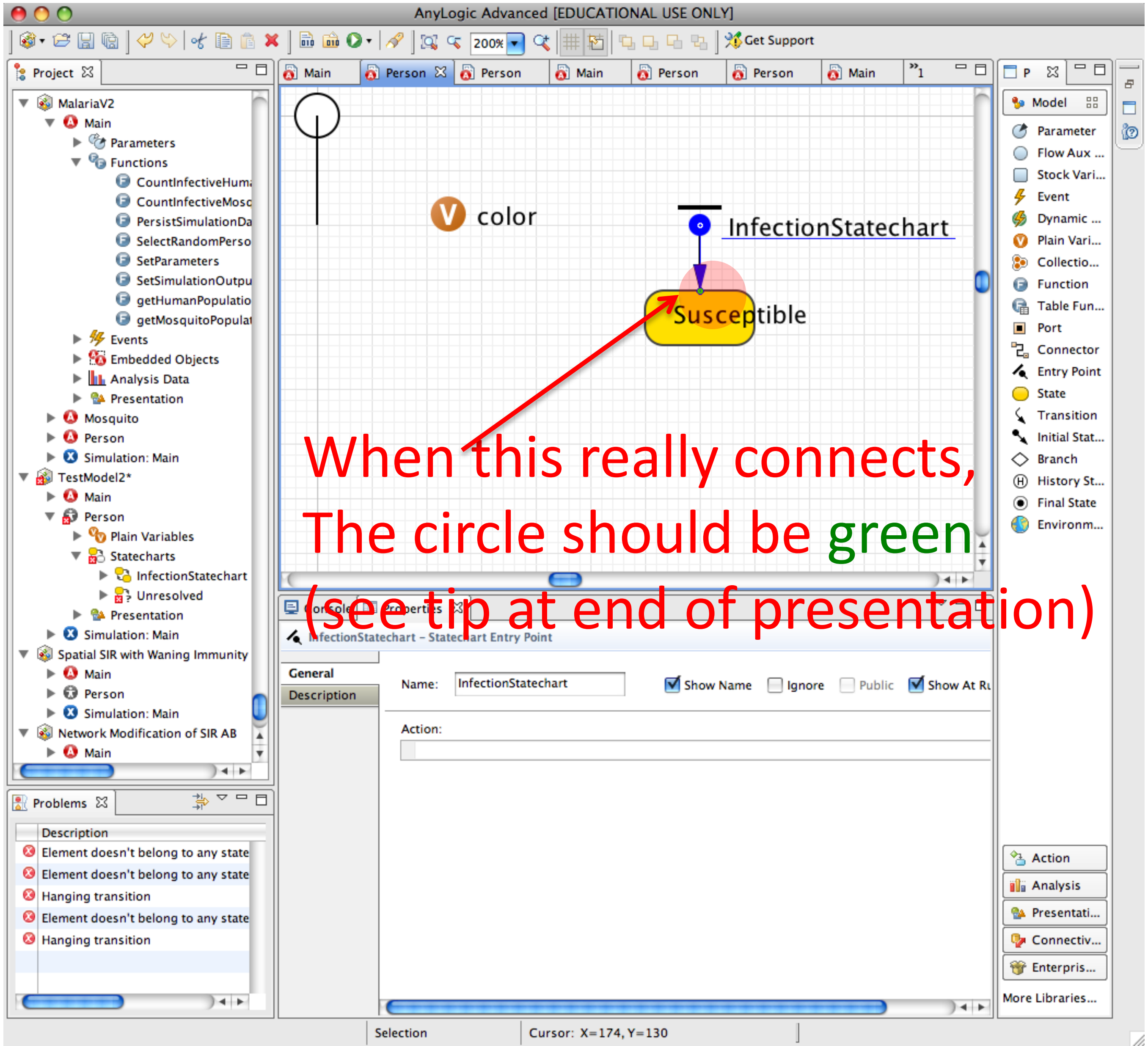


The associated text is the name of the statechart!

Add in “Susceptible” State

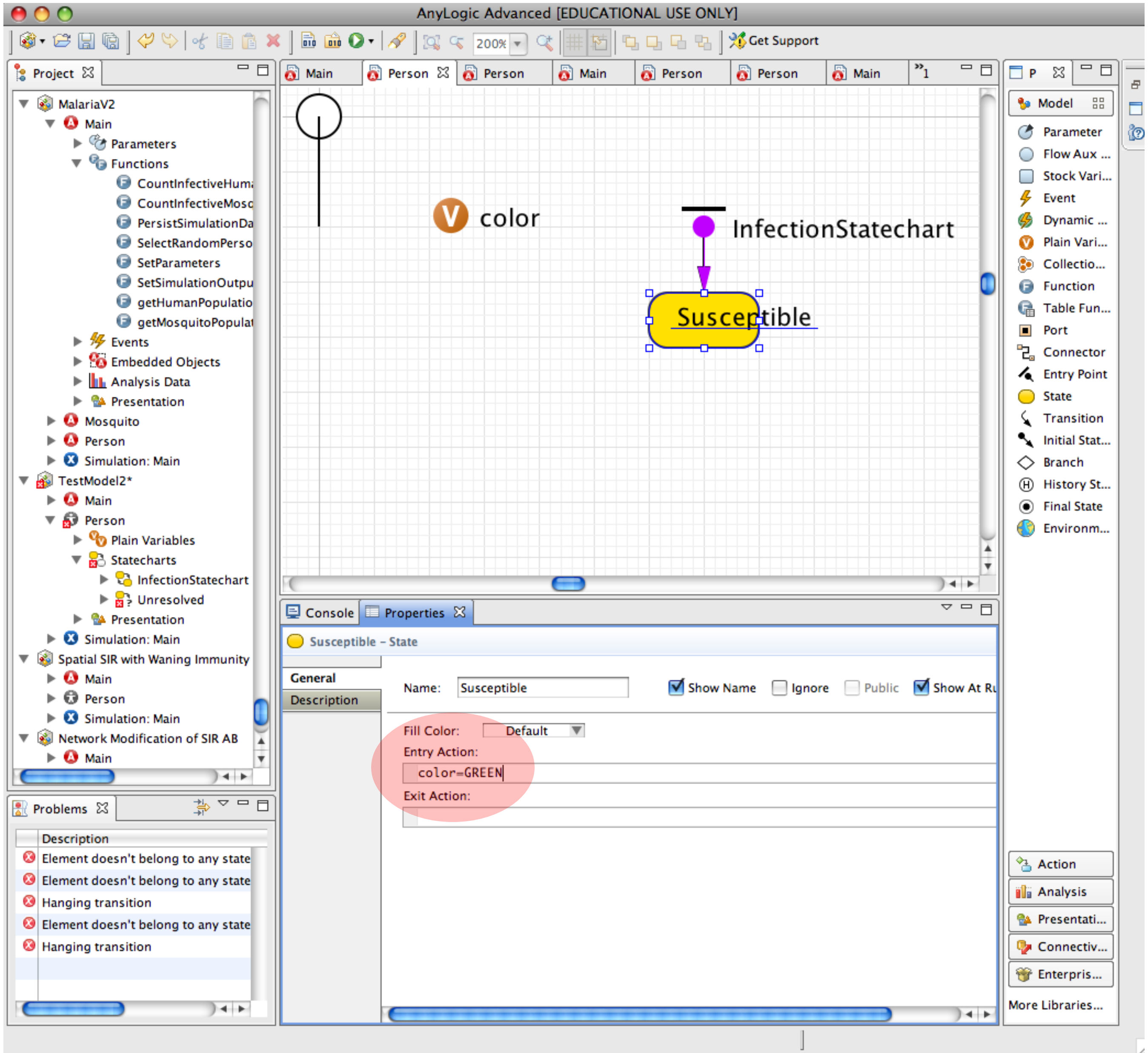


Connect with Entry Point

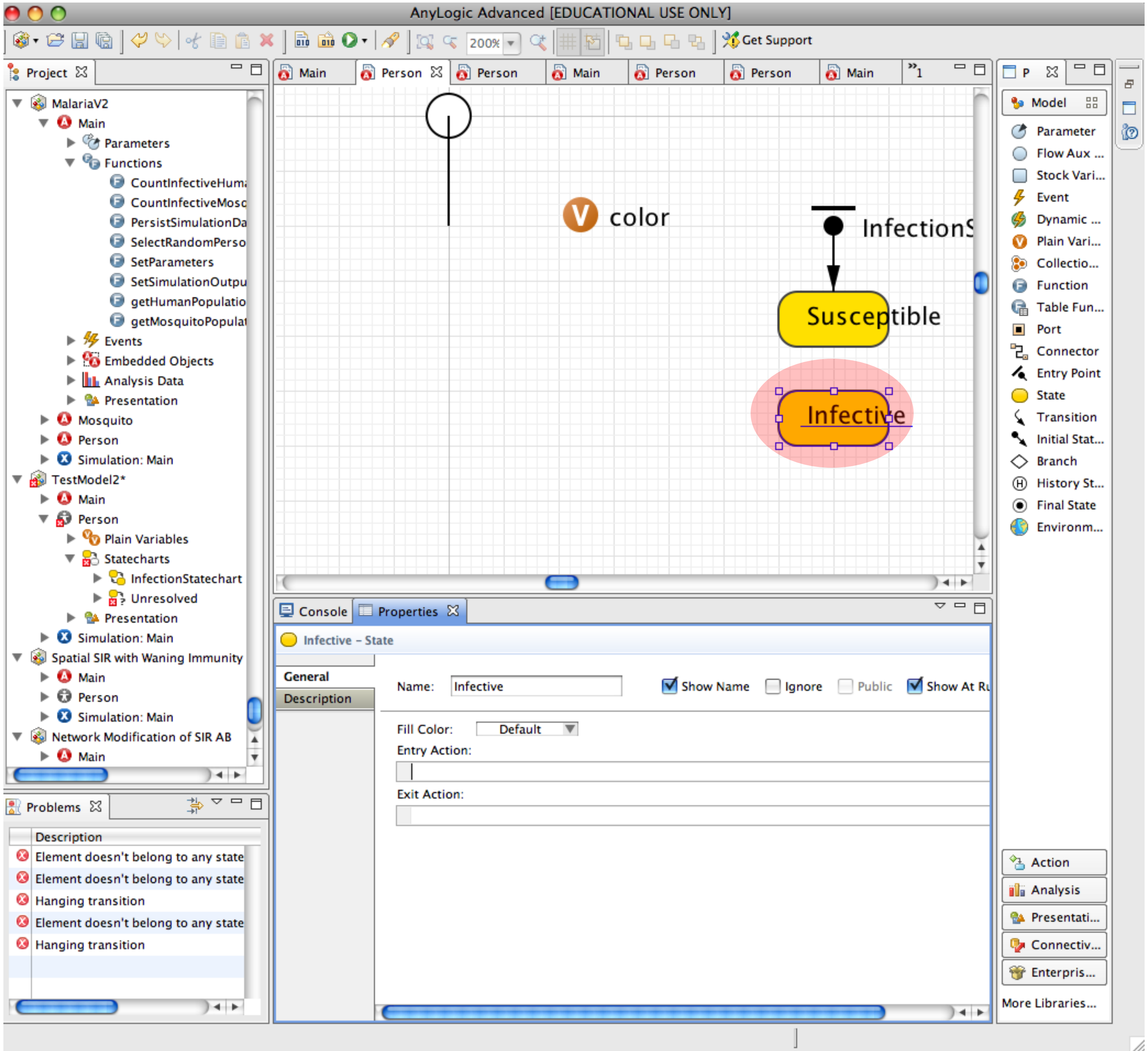


When this really connects,
The circle should be green
(see tip at end of presentation)

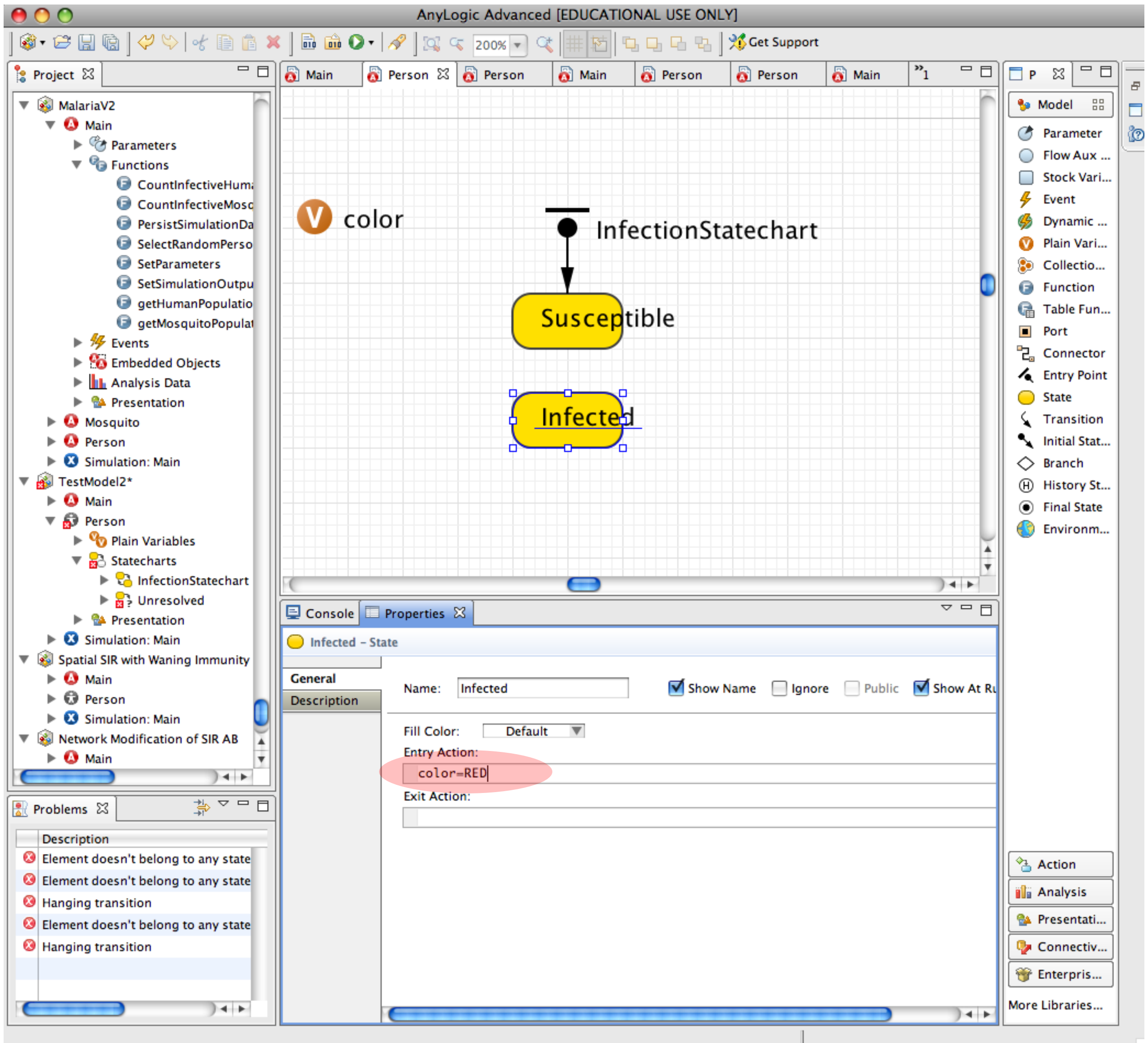
Fill In Code to Color Green when Enter State



Adding in “Infective” State



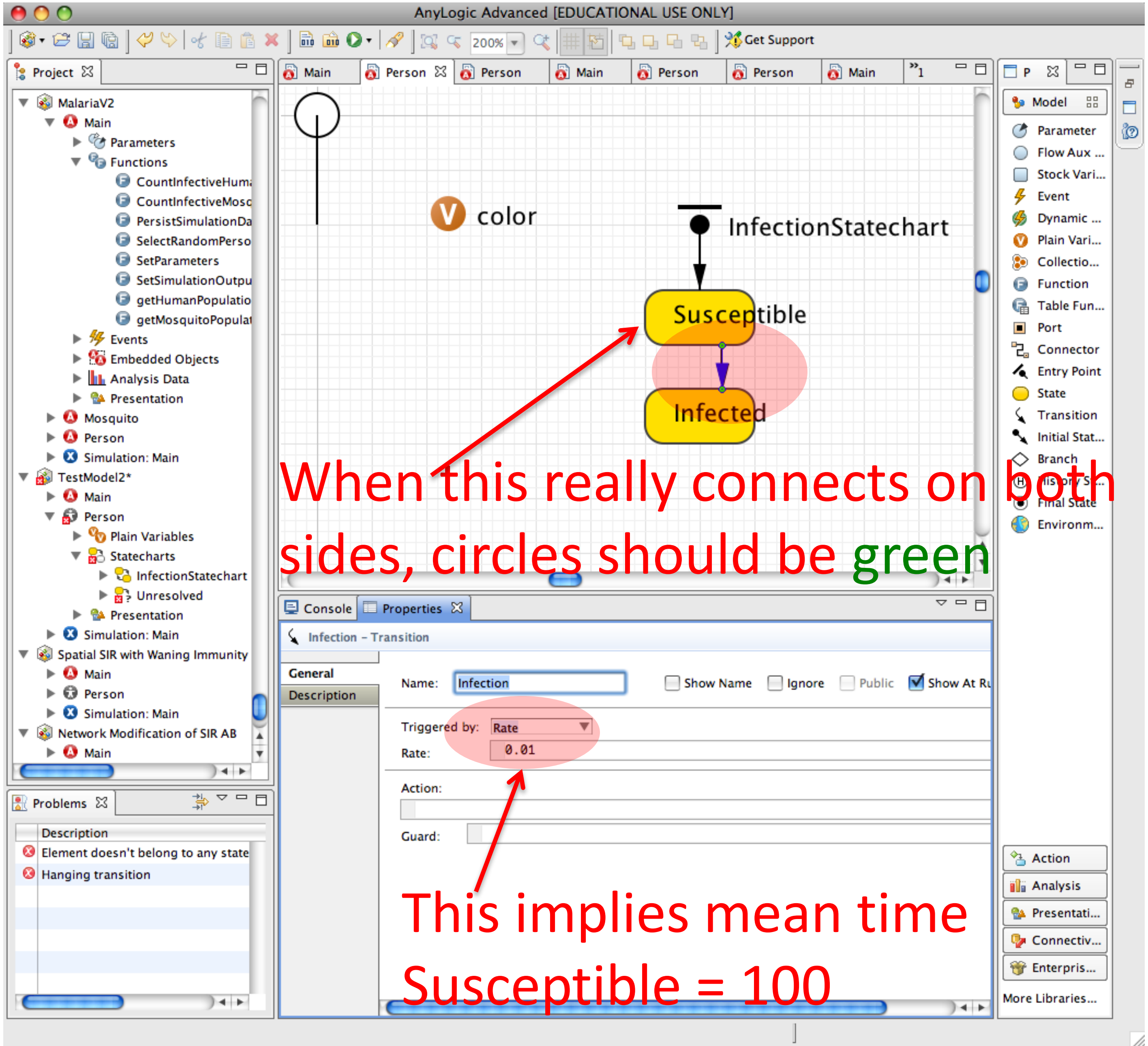
Set to Color Red when Enter State



Discrete Agent Dynamics: Transitions

- Many transition conditions are possible
  - Timeout: Spending some period of time in the state
  - Fixed rate: Leave state with some fixed change per unit time
    - This is similar to "first order interarrival time", and is conceptually linked to the operation of first-order delays in stock & flow diagrams
  - Variable rate: If desired, we can change the rate over time – *but Anylogic only "notices" changes when eg agent re-enters the state*
  - Message received: We can transition when a message (any message or particular type of message) is received
  - Predicate: Only transition when condition becomes true
- These transitions can be conditionally "routed" via branches
  - Conditions can determine to what destination state a particular transition will travel

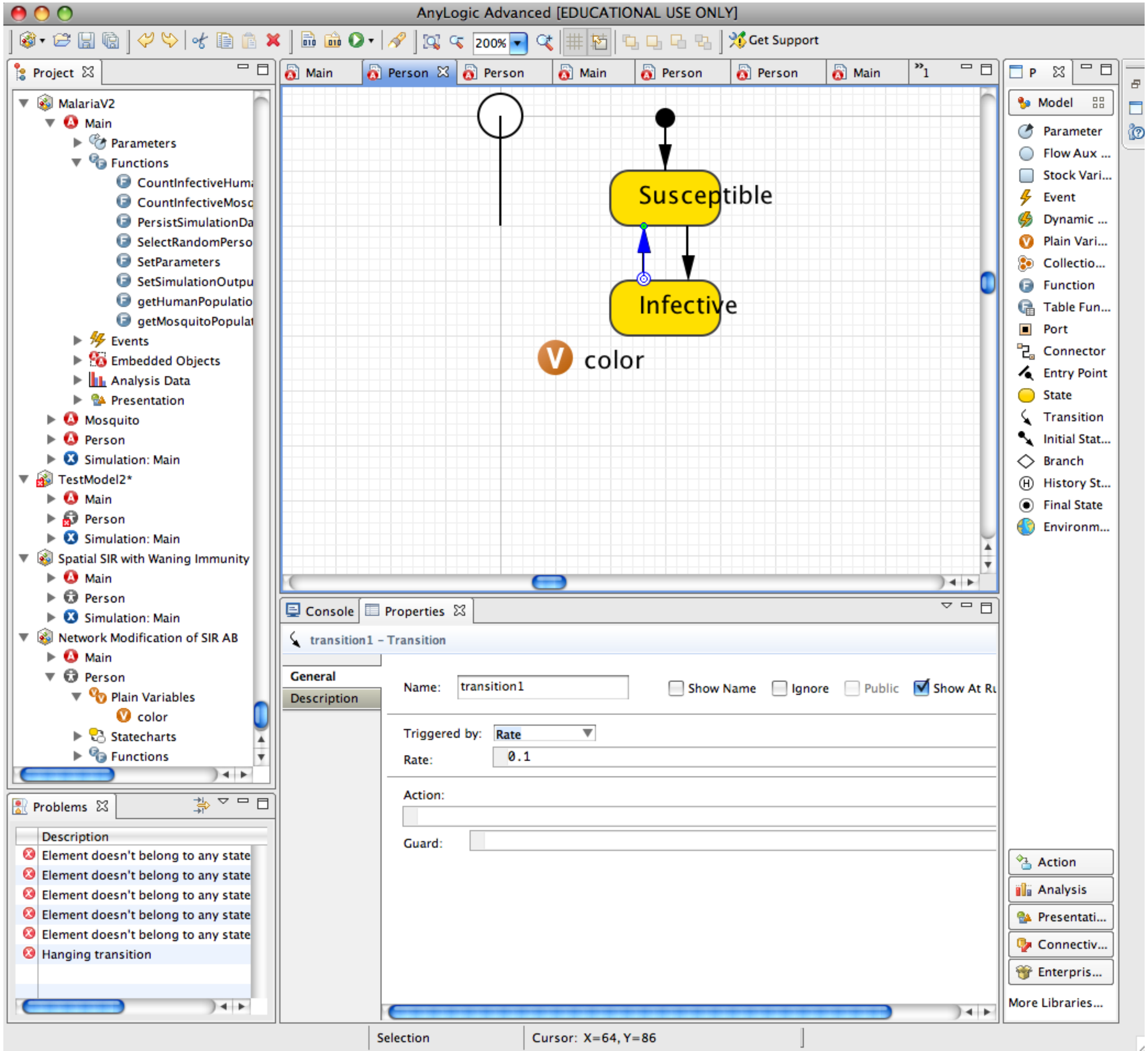
Adding Fixed Rate Transition



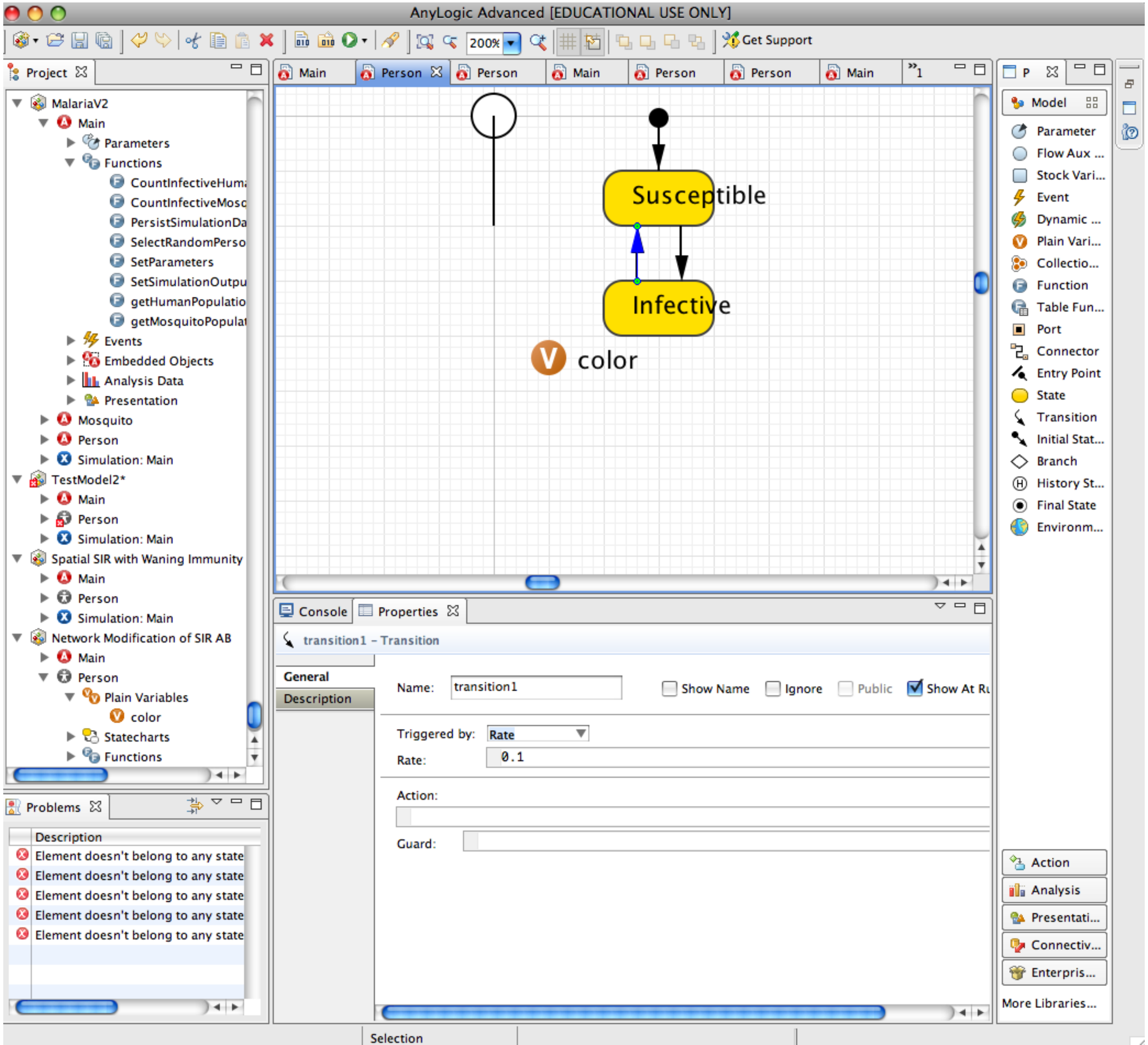
When this really connects on both sides, circles should be green

This implies mean time Susceptible = 100

Tip: Beware Loose Connections

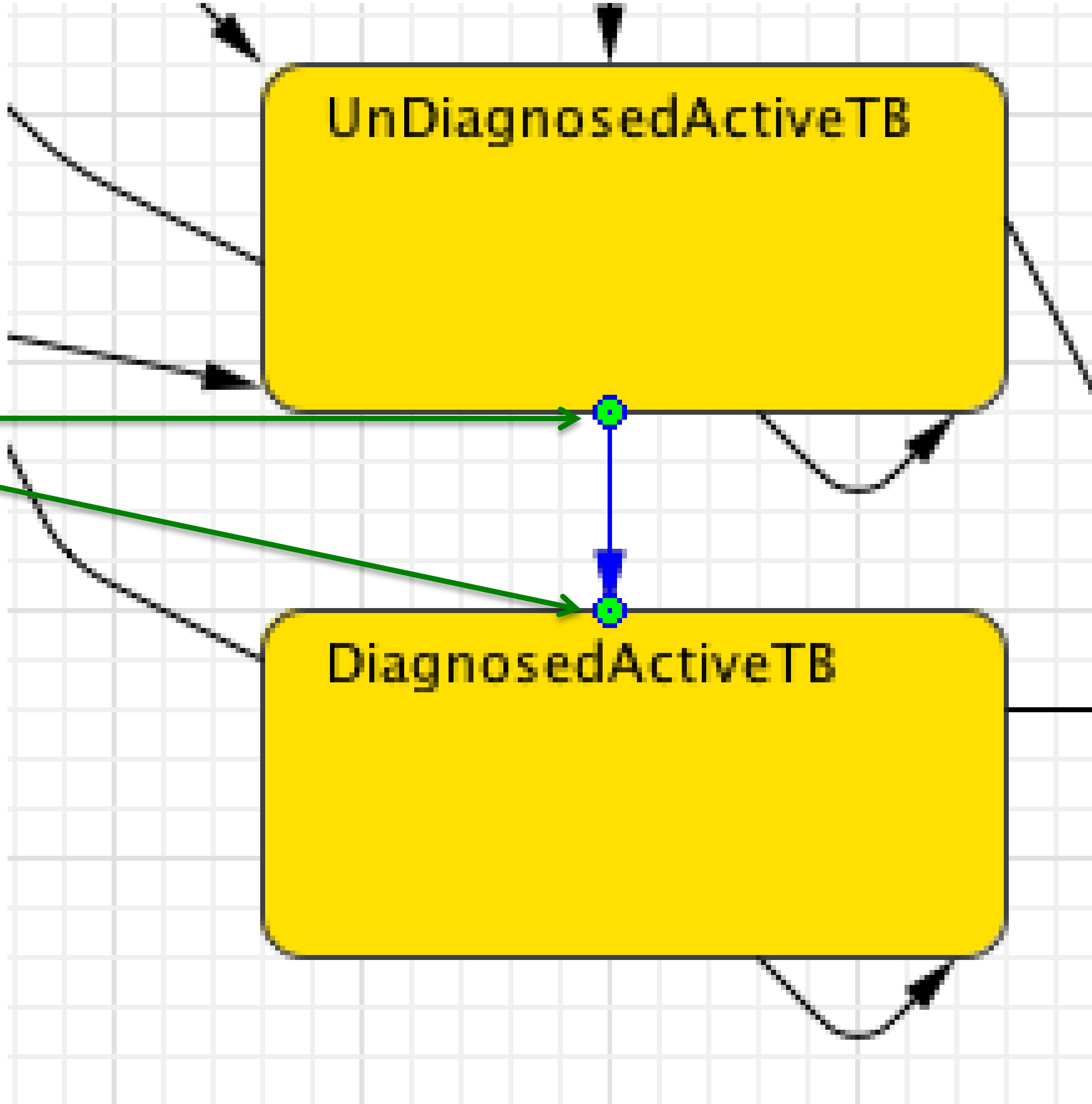


Corrected


AnyLogic Advanced [EDUCATIONAL USE ONLY]
Get Support
200%
Project
Main
Person
Person
Main
Person
Person
Main
MalariaV2
Main
Parameters
Functions
CountInfectiveHum
CountInfectiveMosc
PersistSimulationDa
SelectRandomPerso
SetParameters
SetSimulationOutpu
getHumanPopulatio
getMosquitoPopulat
Events
Embedded Objects
Analysis Data
Presentation
Mosquito
Person
Simulation: Main
TestModel2*
Main
Person
Simulation: Main
Spatial SIR with Waning Immunity
Main
Person
Simulation: Main
Network Modification of SIR AB
Main
Person
Plain Variables
color
Statecharts
Functions
Susceptible
Infective
color
Problems
Description
Element doesn't belong to any state
Element doesn't belong to any state
Element doesn't belong to any state
Element doesn't belong to any state
Element doesn't belong to any state
Console
Properties
transition1 - Transition
General
Description
Name:
transition1
Show Name
Ignore
Public
Show At Ru
Triggered by:
Rate
Rate:
0.1
Action:
Guard:
Model
Parameter
Flow Aux ...
Stock Vari...
Event
Dynamic ...
Plain Vari...
Collectio...
Function
Table Fun...
Port
Connector
Entry Point
State
Transition
Initial Stat...
Branch
History St...
Final State
Environm...
Action
Analysis
Presentati...
Connectiv...
Enterpris...
More Libraries...
Selection

Tip: Confirming Transition Connectivity

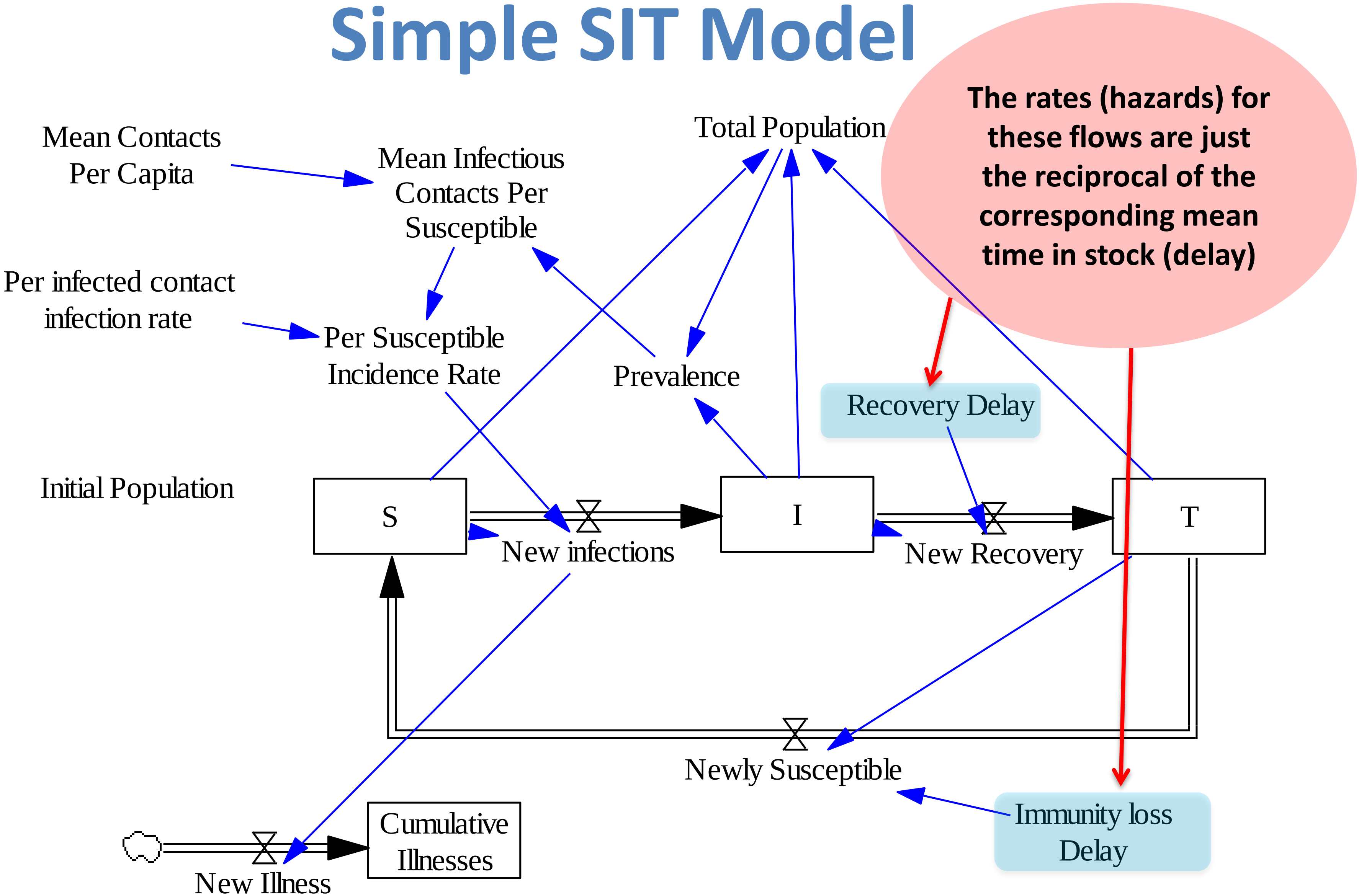
- Ensure that both sides of the transition show green circles when connected
  - Otherwise, may appear connected but will actually be disconnected!



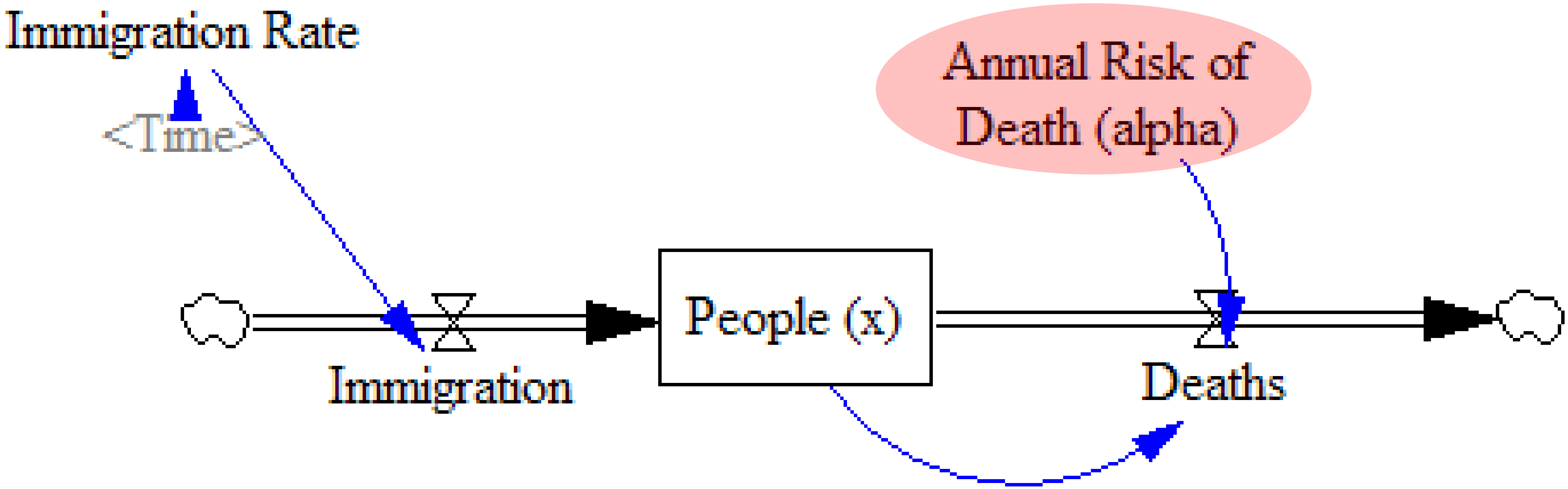
Rates & Flows

- Some may have seen fixed rates before – in the form of "transition rates" in Compartment models
- Within a Compartment/SD model, a flow out of a stock was commonly set by the multiplication of the
  - State variable (Stock)
  - Some rate of transition
- We use different names for these rates
  - "Transition rates"
  - "Likelihood of transition per *Unit Time*"
  - Transition (e.g. "infection", "mortality") "hazard"

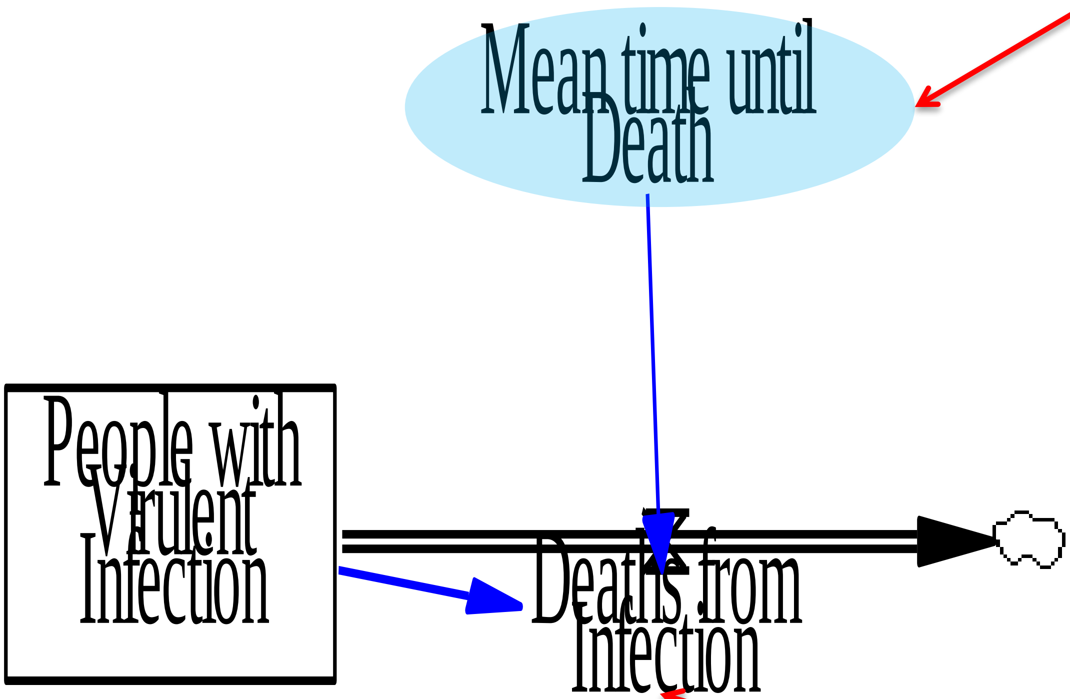
First Order Delays in Action: Simple SIT Model



Example Fixed Transition Rate/Hazard



Example Fixed Transition Rate/Hazard



The transition rate is the *reciprocal* of this number i.e.

$$\frac{1}{Mean\ time\ until\ Death}$$

People with Virulent Infection/Mean time until Death

=

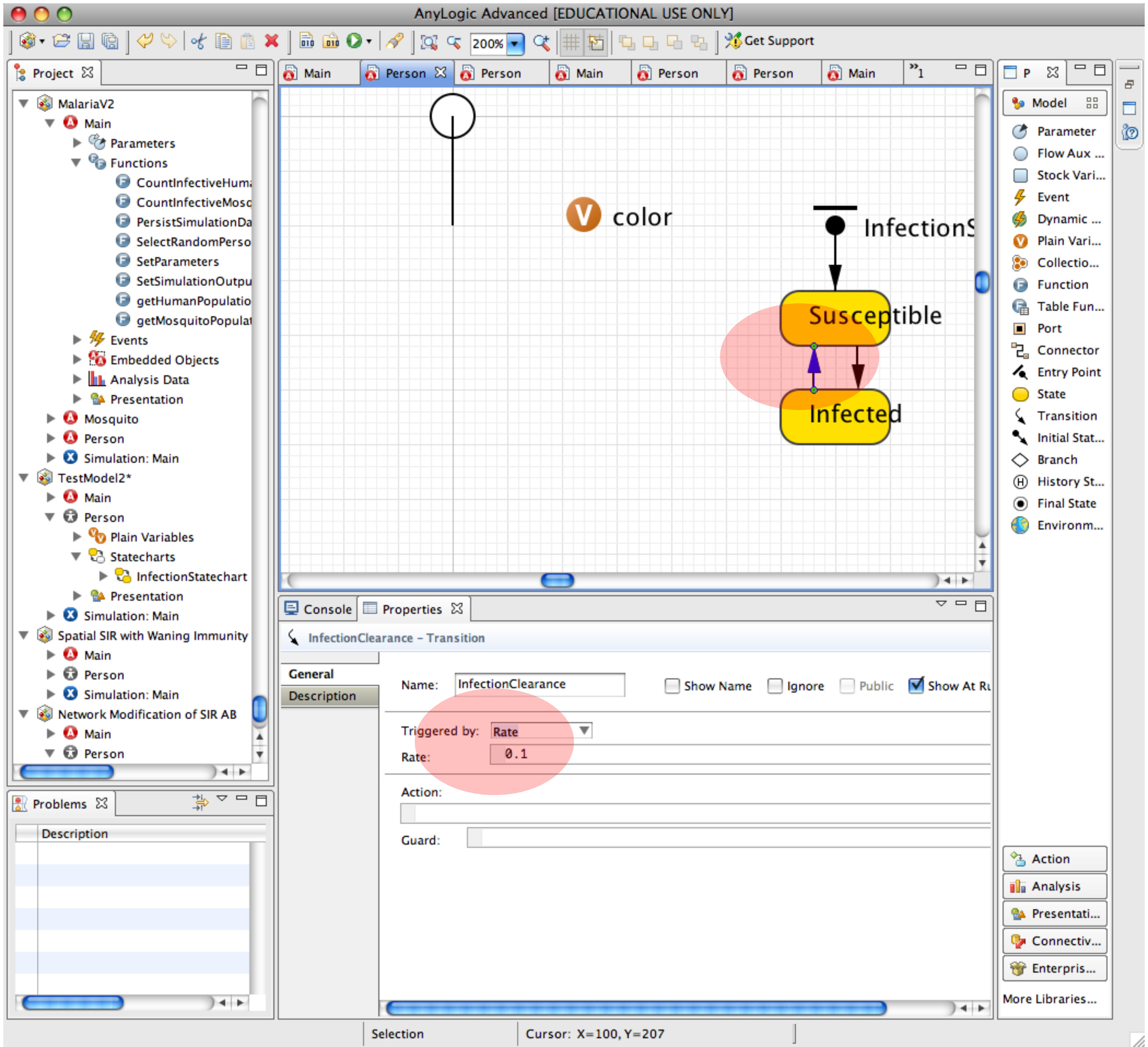
People with Virulent Infection*(1/Mean time until Death)

i.e. People with Virulent Infection*Rate

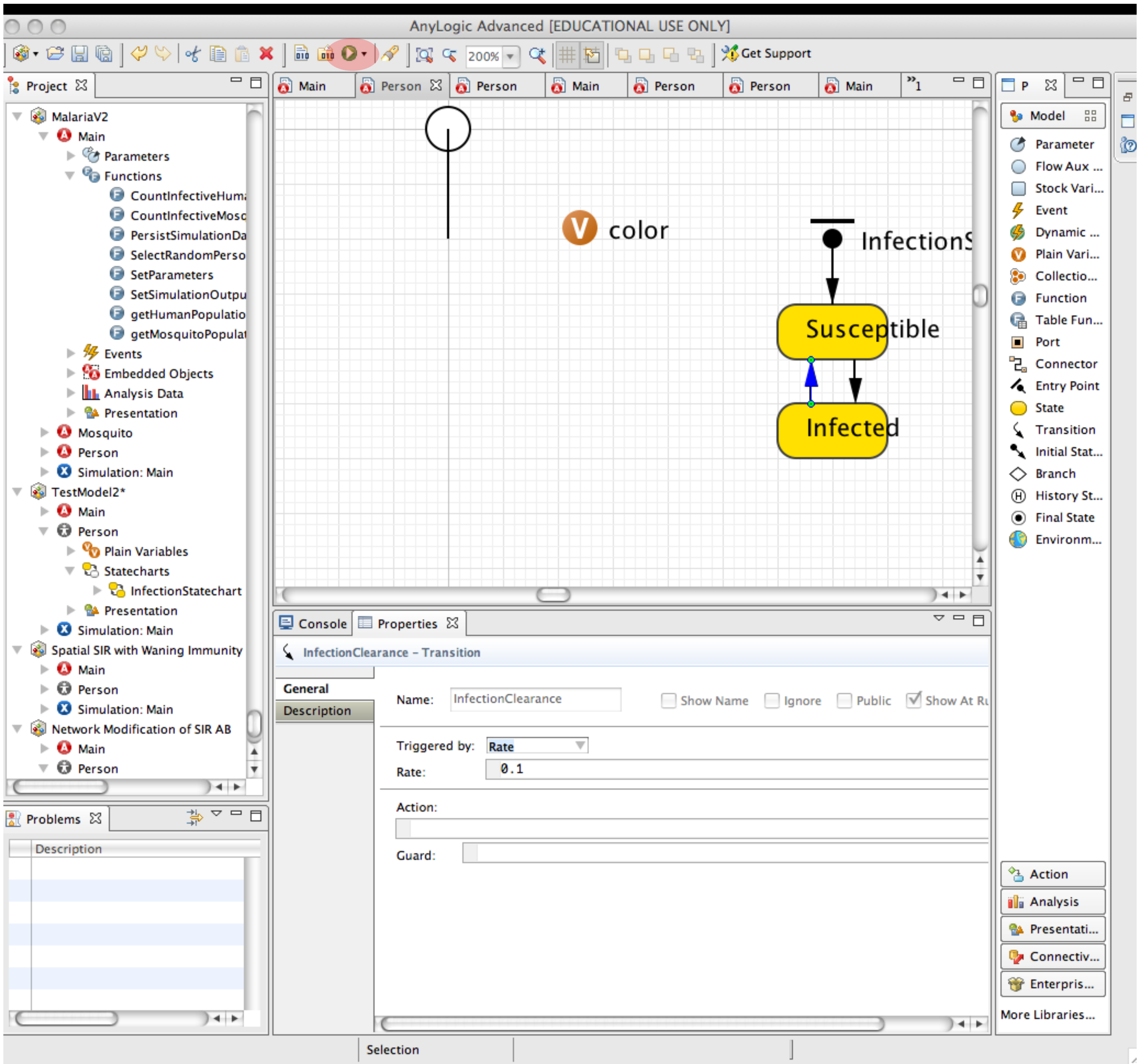
Fixed Rates: Transition “Hazards”

- With “fixed rates”, we are specifying rates of transitions
- Because we are dealing with the chance that each individual transitions, we don’t need to multiply by the number of people at risk
  - Here, there is just 1 person at risk!
- As in Compartment models, these rates can change over time, but the statechart needs to be “made aware” of these changes (see later)
  - Leave & go back into current state (circular transition)
  - Trigger “change” event in Agent

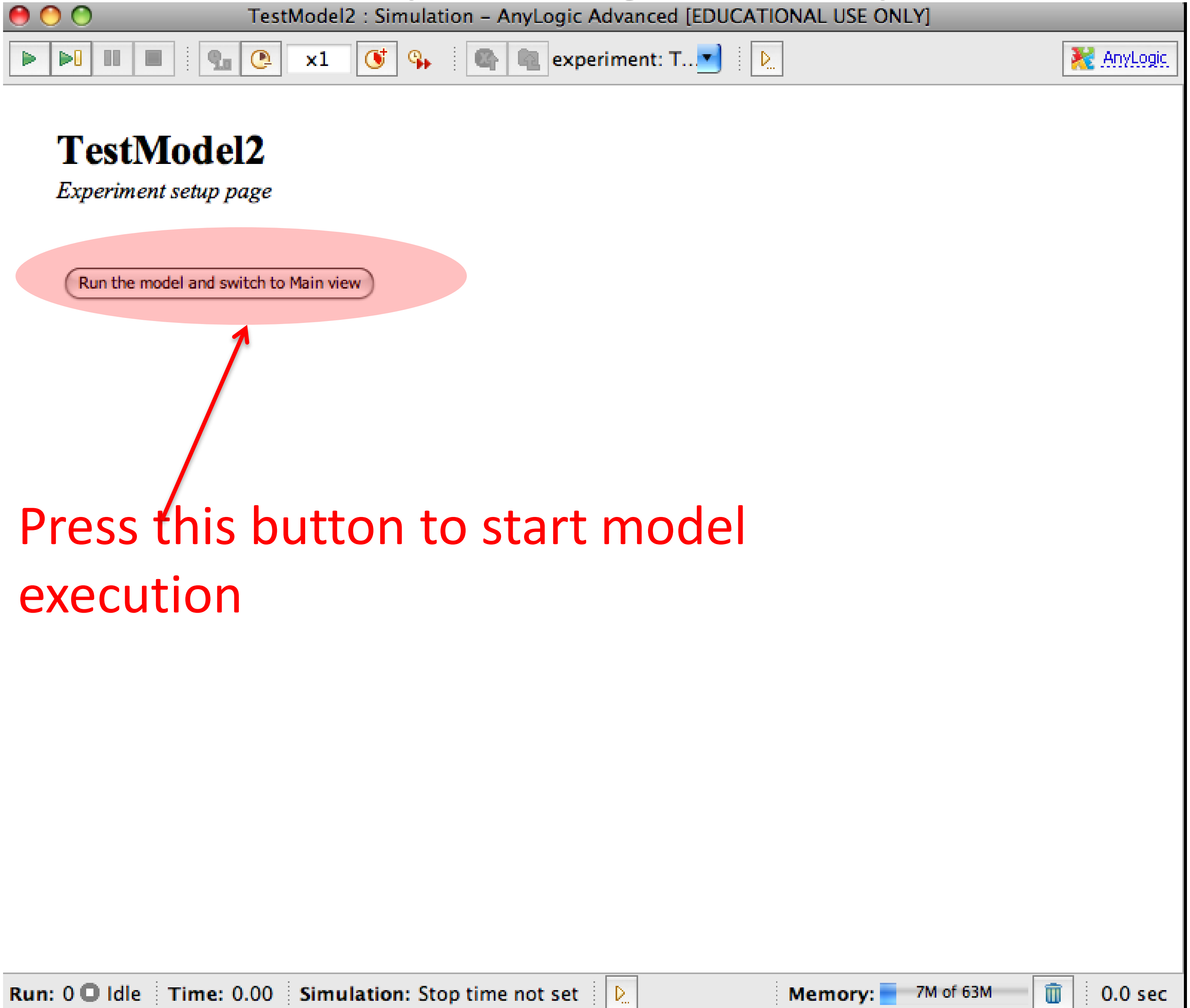
Adding Infection Clearance Transition



Run the Model!

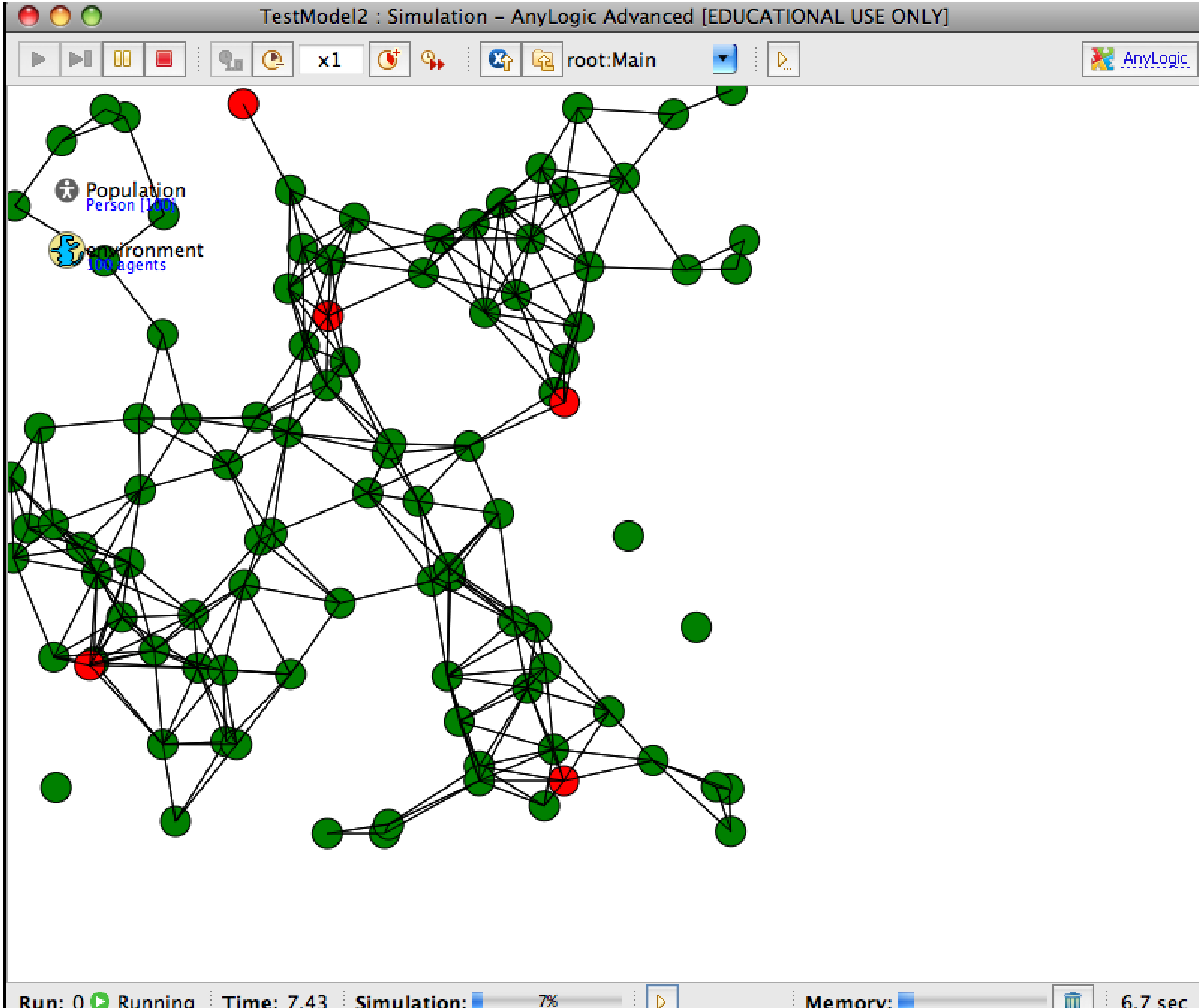


Completing Set-Up

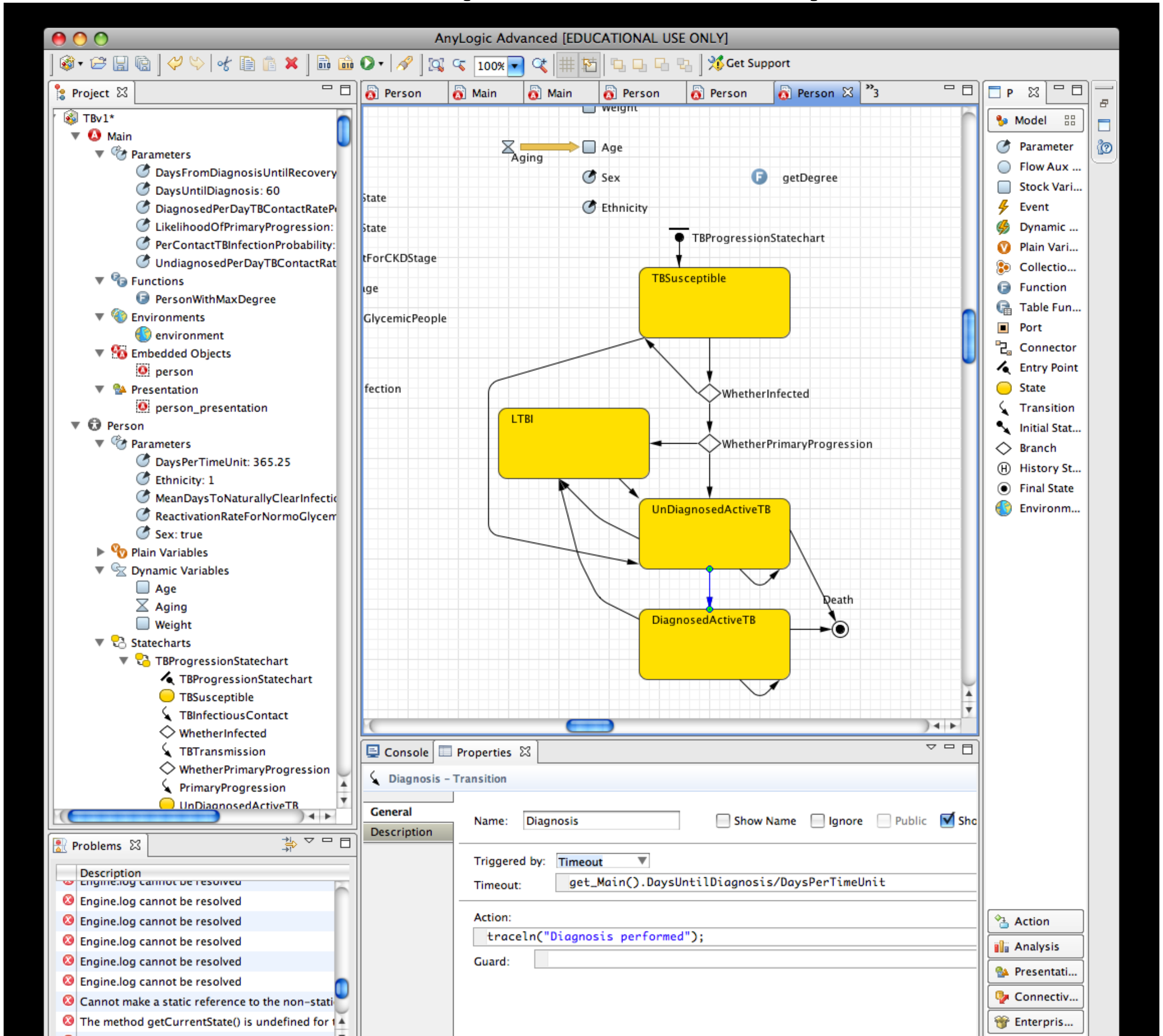


Press this button to start model execution

Model Presentation



Transition Type: Fixed Residence Time (Timeout)

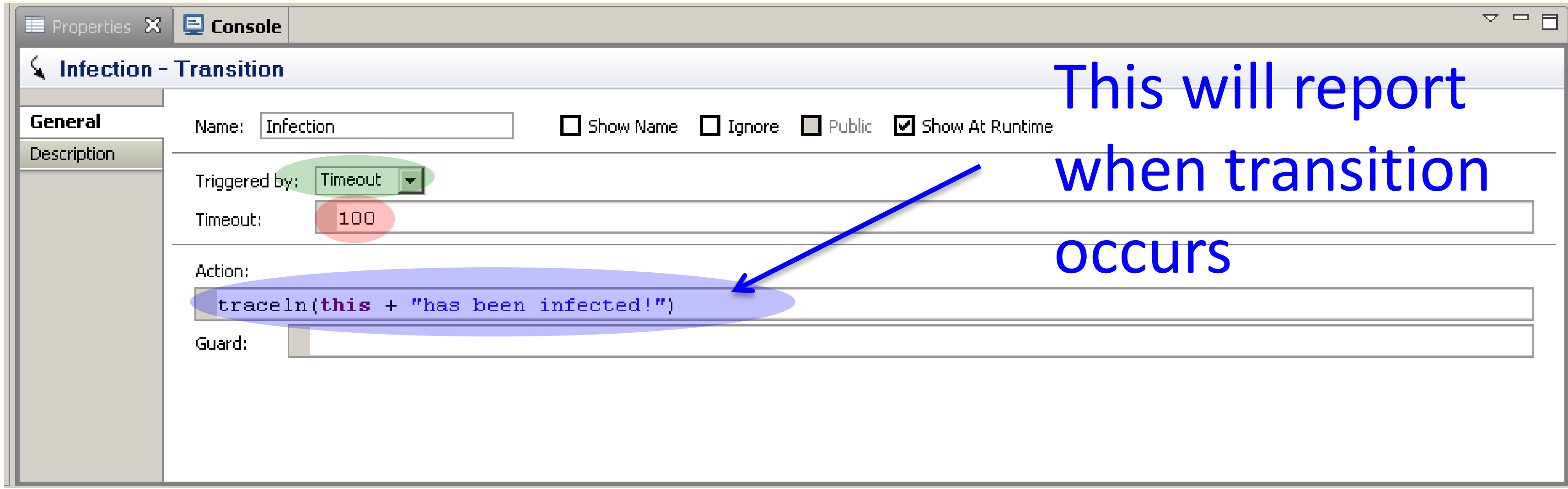


Example of Processes Associated with Fixed Timeouts

- Aging
- Tightly defined time constants associated with natural history
  - While these may be described as associated with a broad distribution (e.g. with a 1^{st} or 2^{nd} order delay), much of that variability may be due to heterogeneity
  - *For a given person, these may be quite specific in duration* $\Rightarrow$ *Can capture through a timeout*

What Happens if this Depends on a Timeout?

- Set the “Infection” transition to Trigger based on a “Timeout”



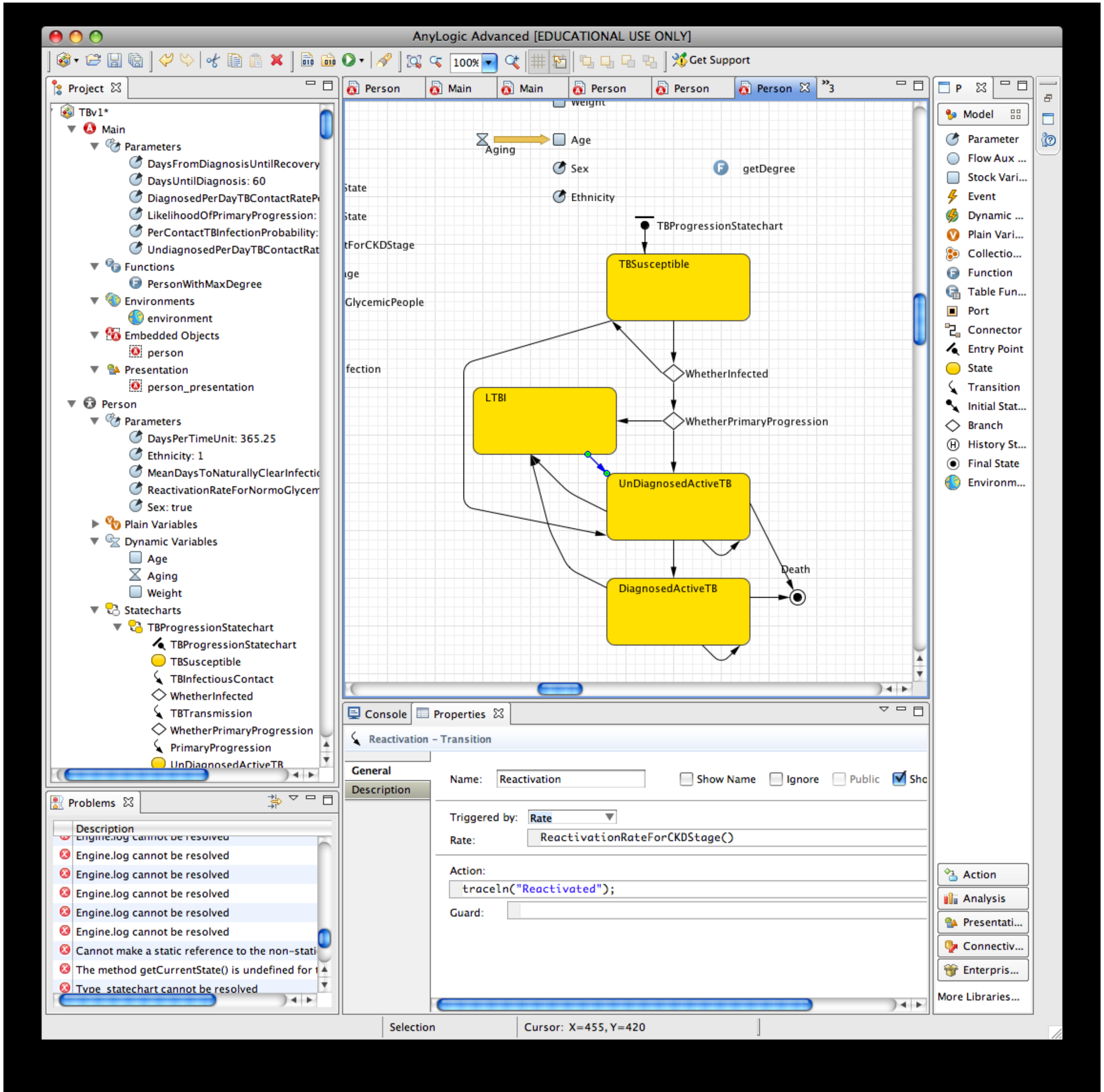
- Make the “Timeout” 100

Now run the model, and observe the difference

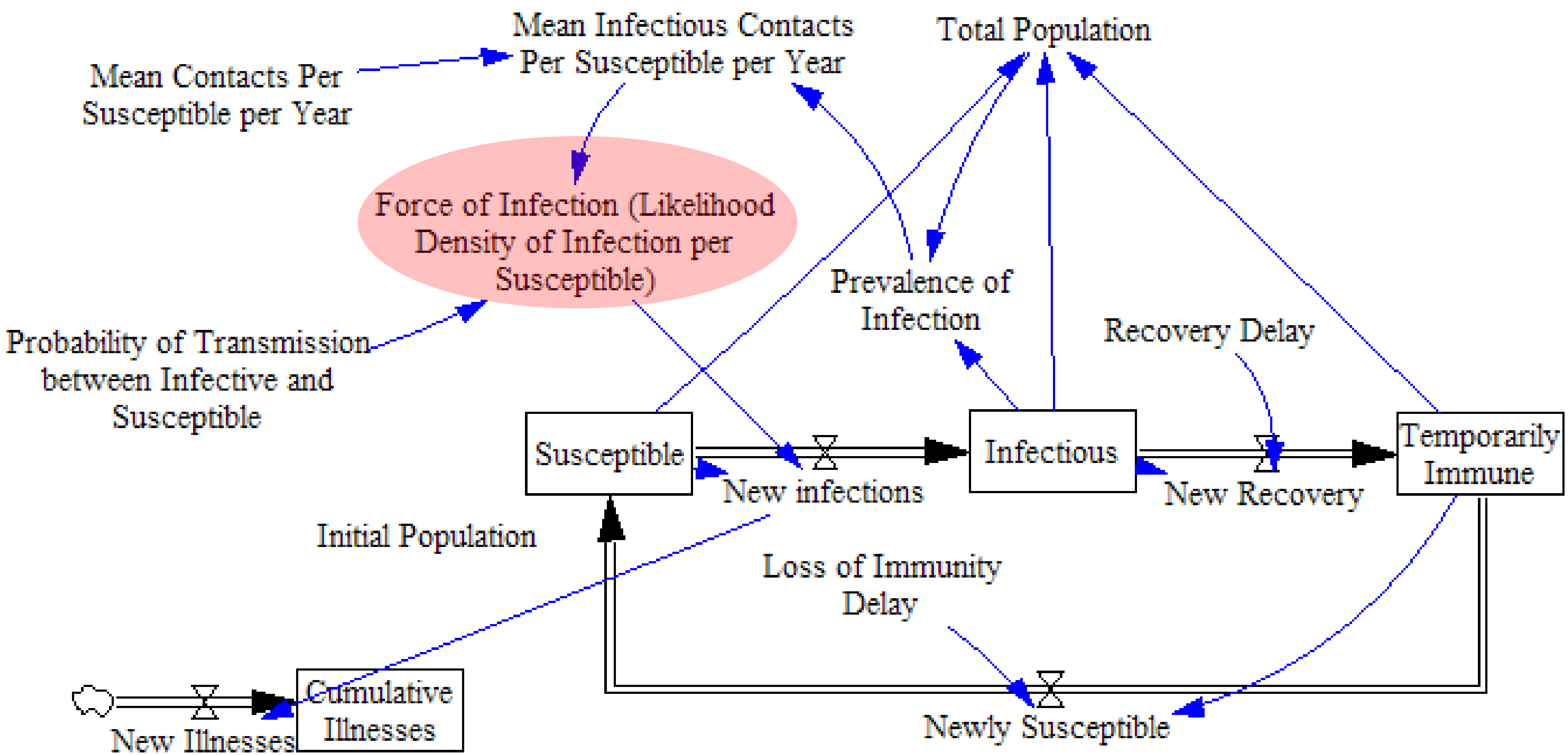
Hands on Model Use Ahead

Load model: TBv1.alp

Transition Type: Variable Rate

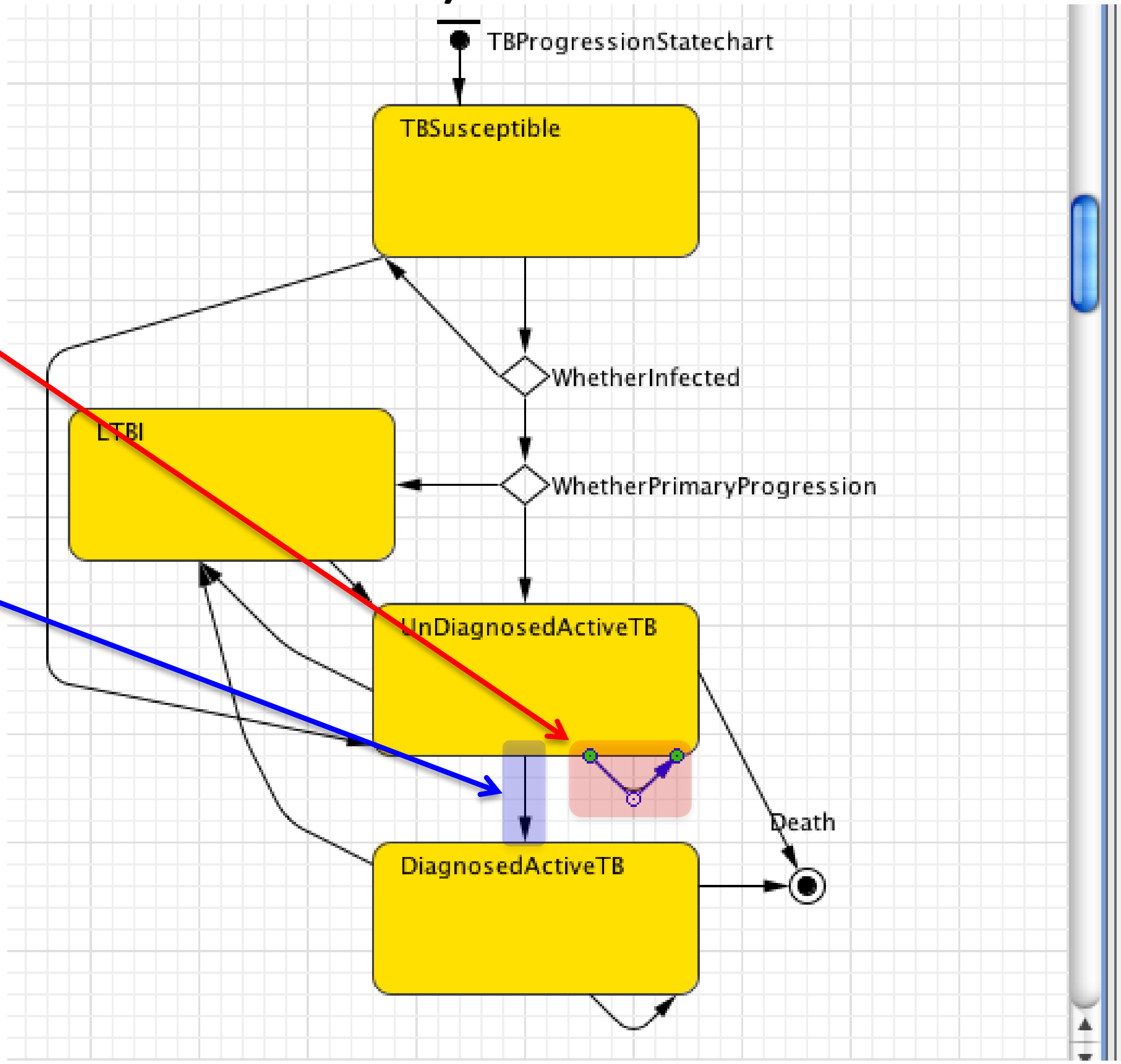


Example Transition Rate/Hazard



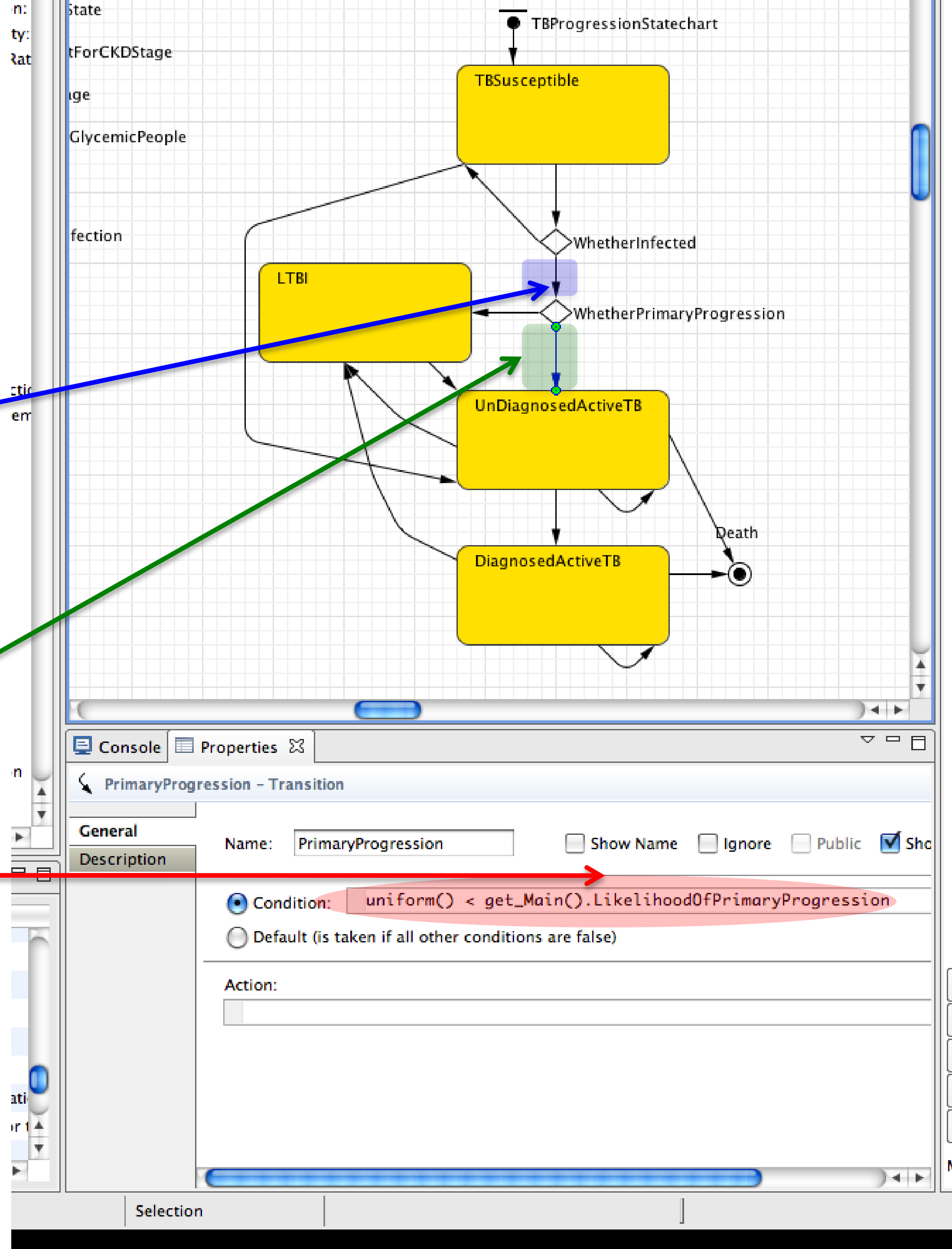
Special Elements: Self-Transition
(Use if Wish To Have State Register Changing Out-transition rates)

The self-transition will "make the state realize" that the rate associated with any out transition (e.g. this one) has changed

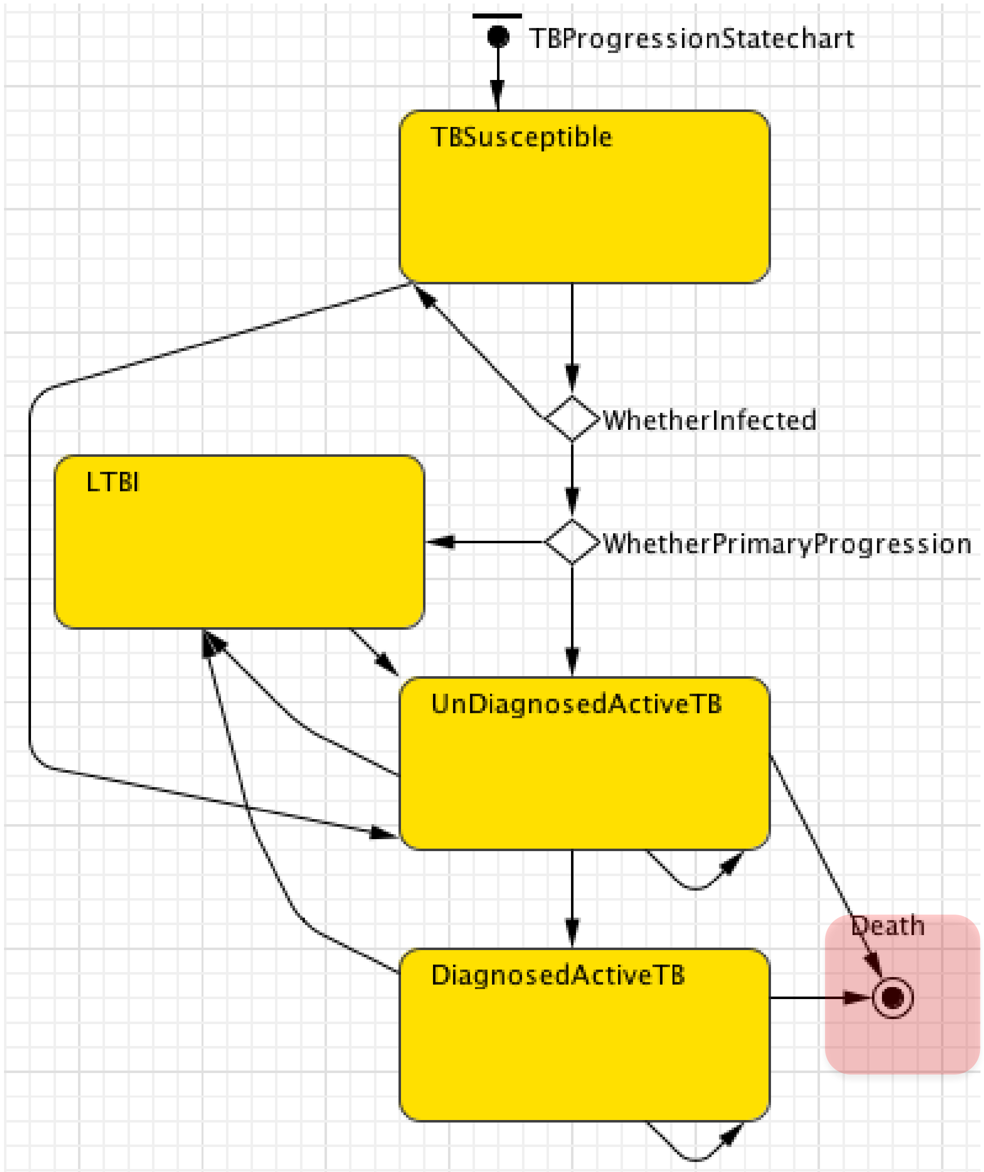


Example Conditional Transition

The incoming transition into “WhetherPrimaryProgression” will be routed to thisoutgoing transitionif this condition is true

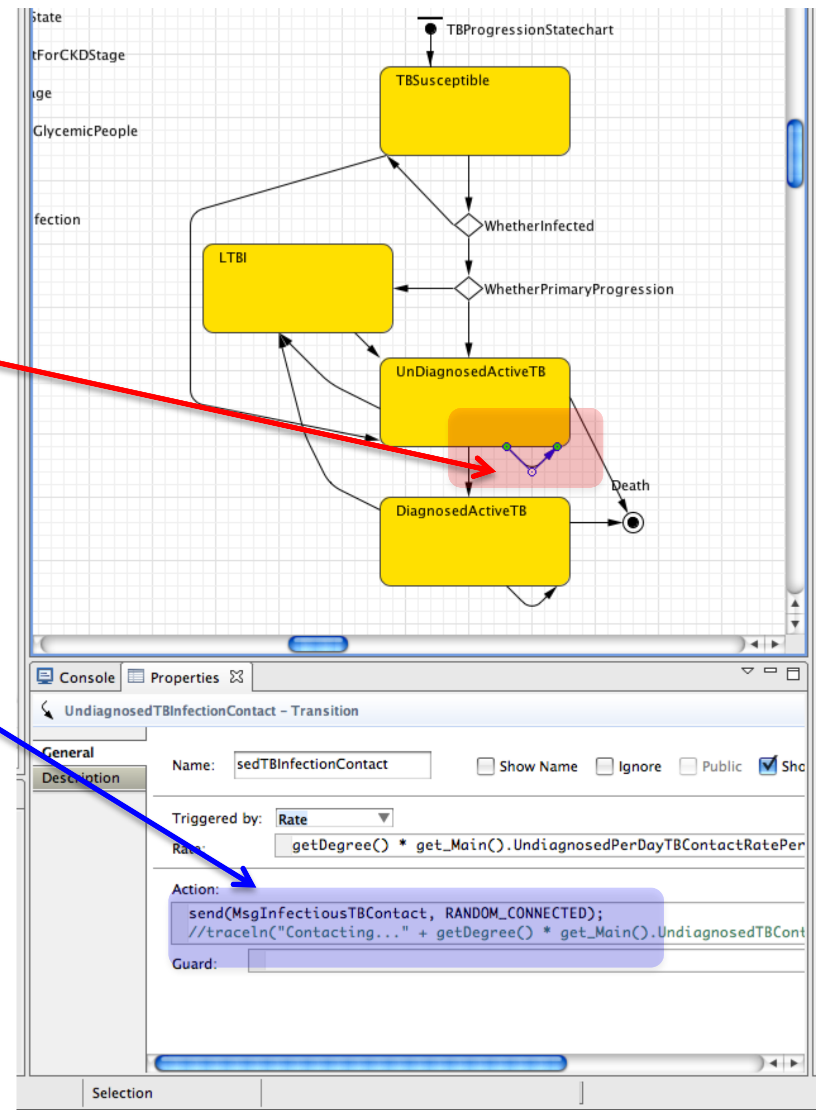


Special Elements: Exit Point



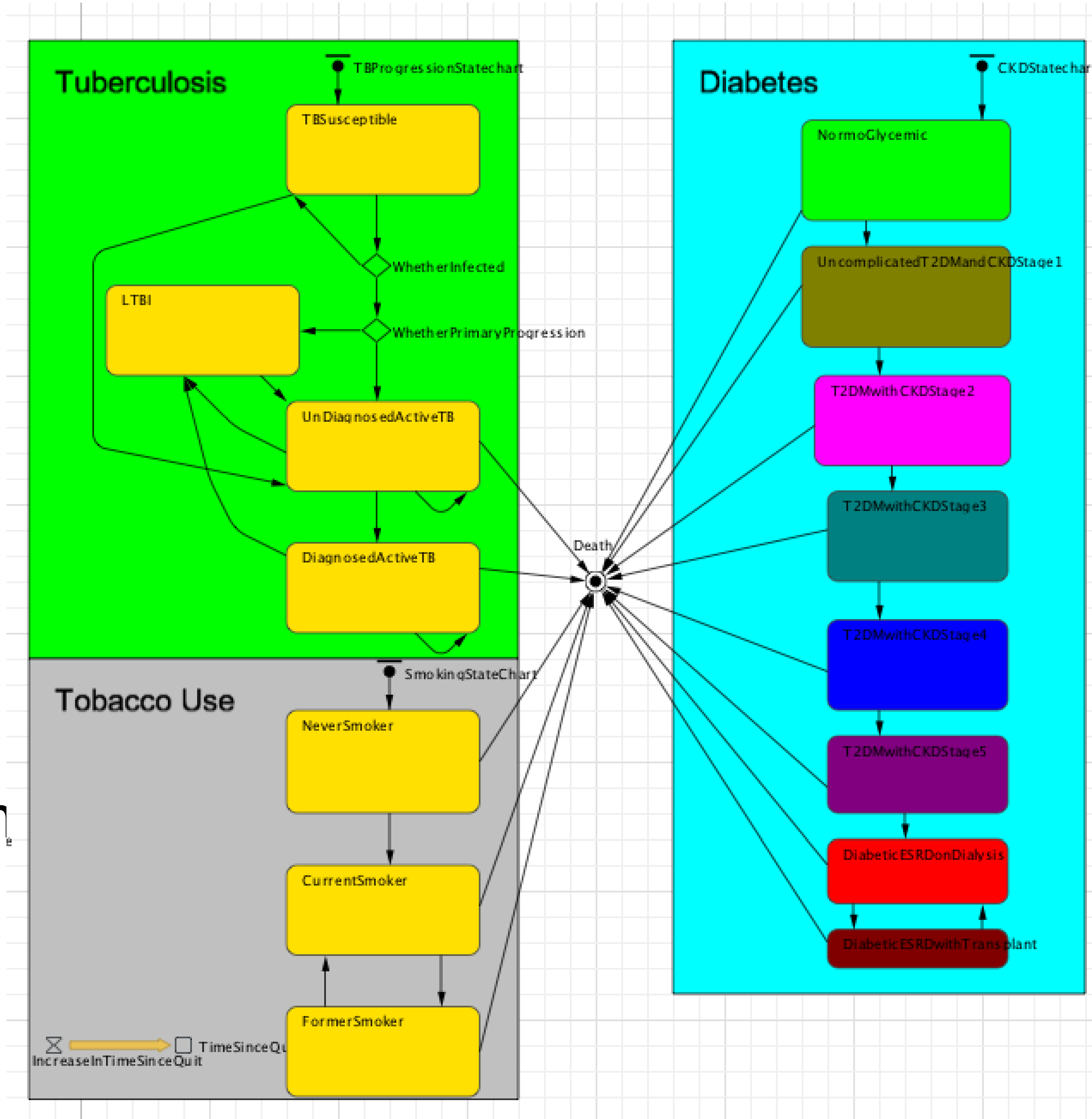
Special Elements: Self-Transition
(Use if Wish To Trigger an Action w/o Leaving State)

The self-transition will invoke this action when it occurs



Parallel Statecharts

- By default, each statechart evolves independently.
- If coupling is desired, can make transitions/actions dependent on state of other statecharts

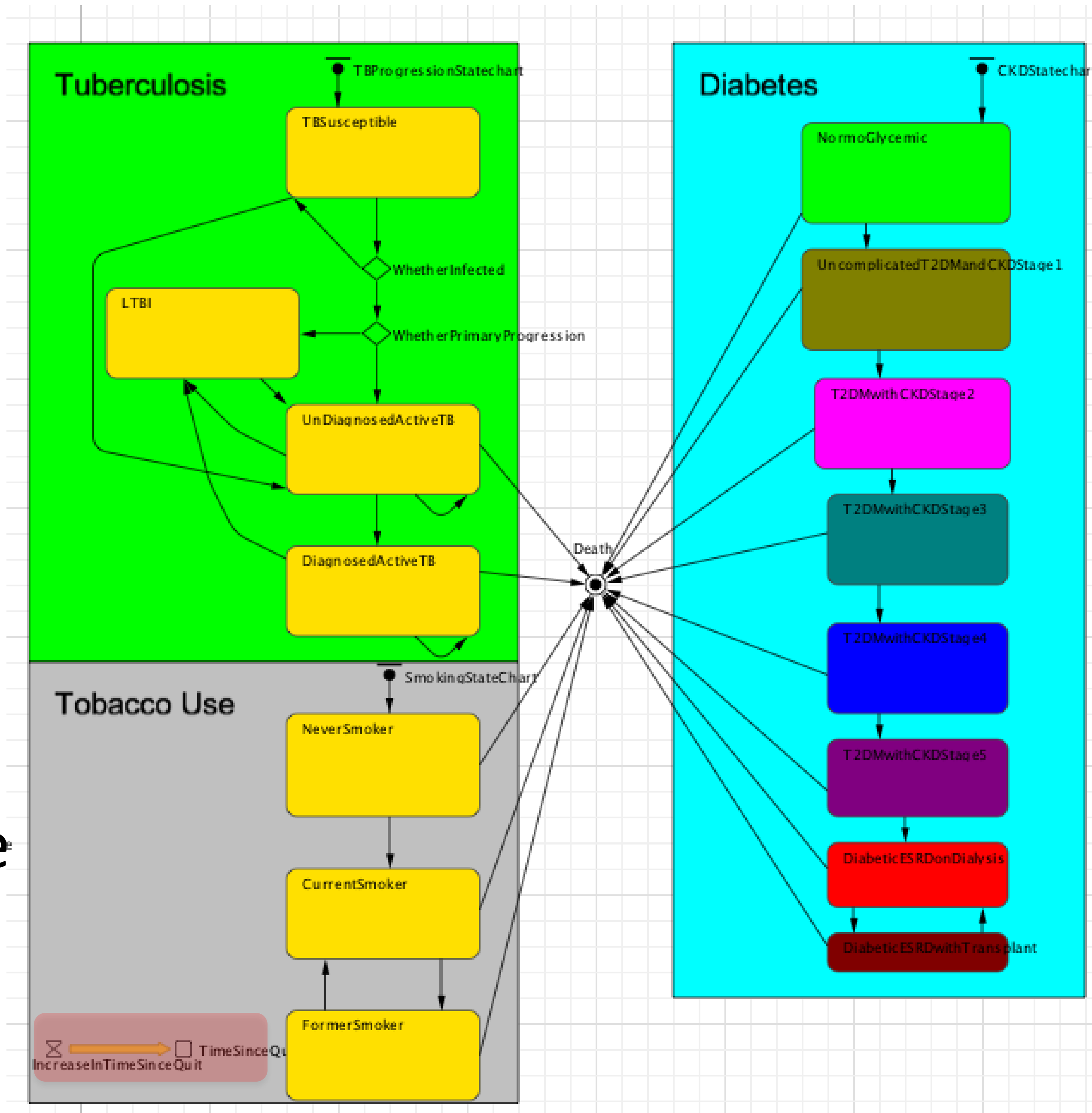


Comparison with Aggregate Stock & Flows

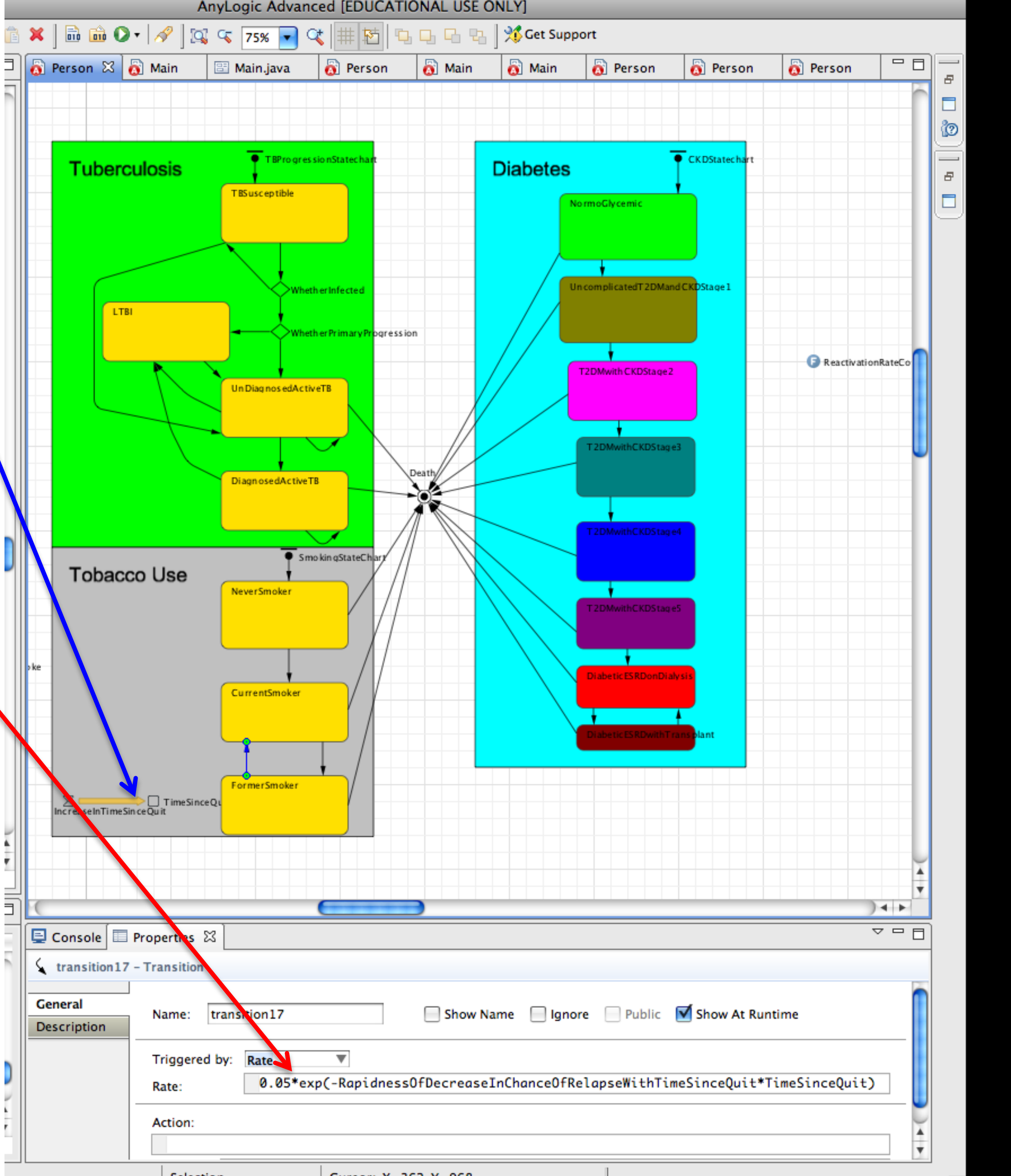
- As for aggregate stocks & flow, individuals' states are discrete
- Unlike aggregate stocks & flows
  - One state within a given statechart is active at a time
  - For parallel flows (e.g. comorbidities), there is no need for considering all combinations of the possible states
  - We can keep track of how long an individual is in a given state & adjust the transition rate accordingly

Parallel Transitions

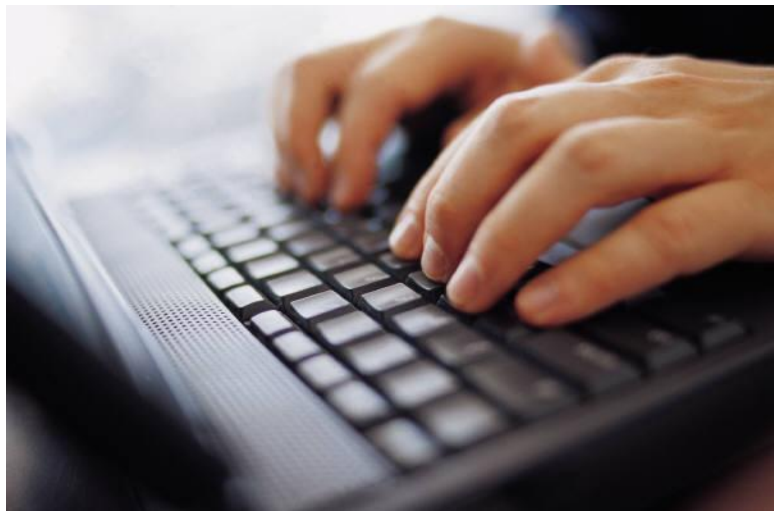
- Example recording the residence time in a state (via a stock with unit inflow -- i.e. just accumulates the time present in that state)



- The residence time in the state determines the transition rate out of that state.
- Transition rates depending on residence time are generally not possible with aggregate models



Hands on Model Use Ahead



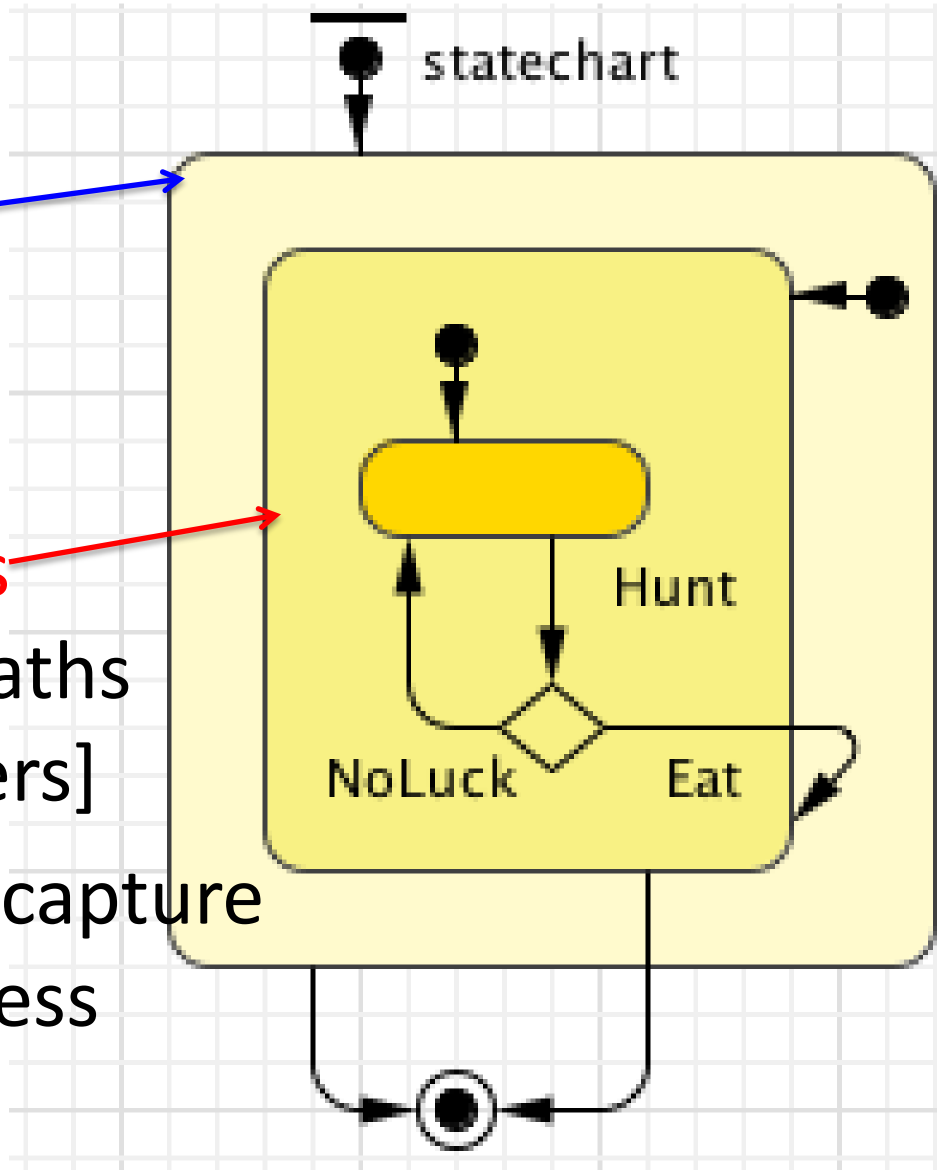
Load Sample Model:

Predator-Prey Agent Based

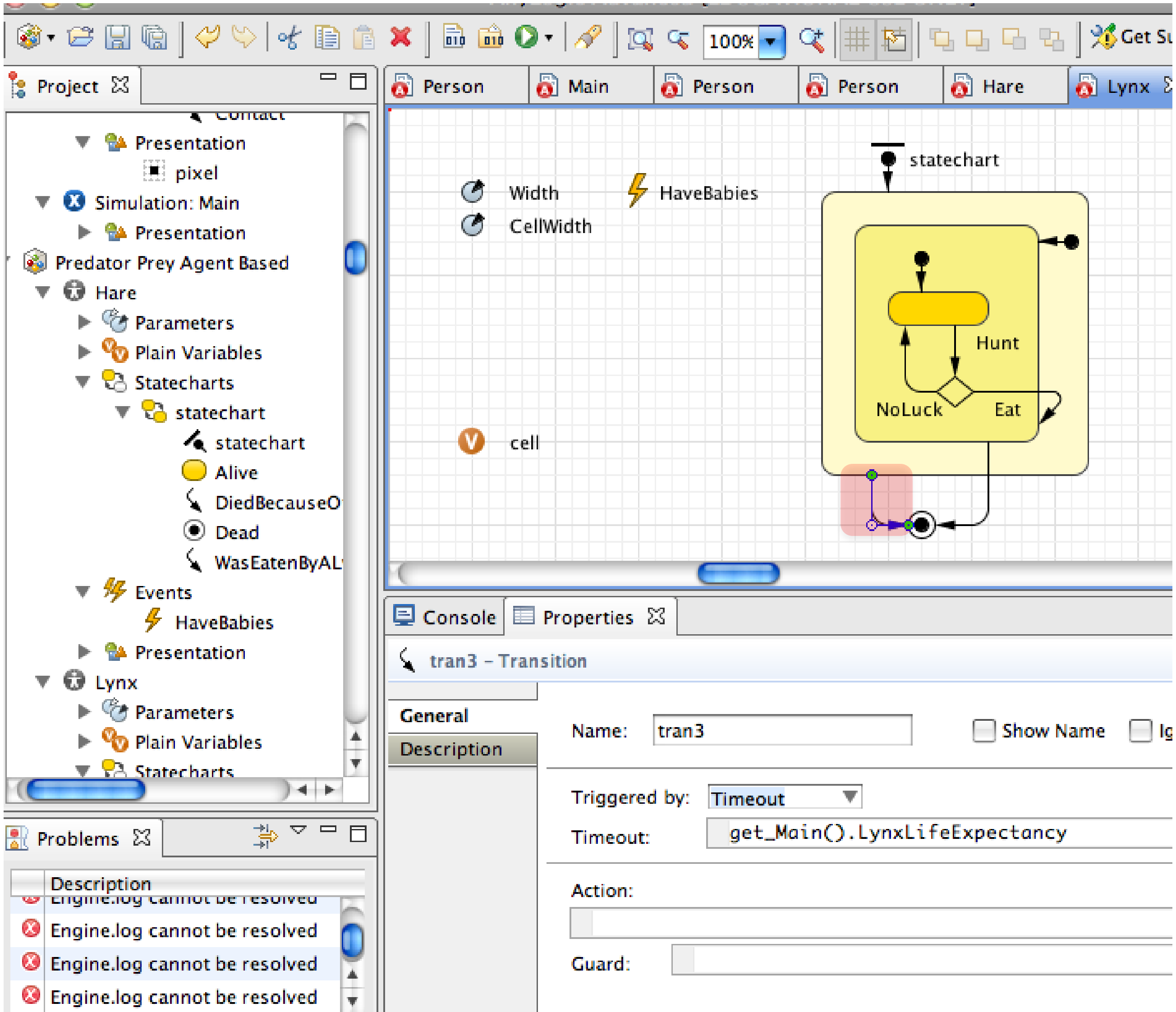
(Via “Sample Models” under “Help” Menu)

Advanced Element: Hierarchical States

- The outermost state captures time since born (for natural deaths)
- The middle-state captures time since last ate (for deaths by hunger). [Eating reenters]
- The inner state transition capture hunting frequency & success

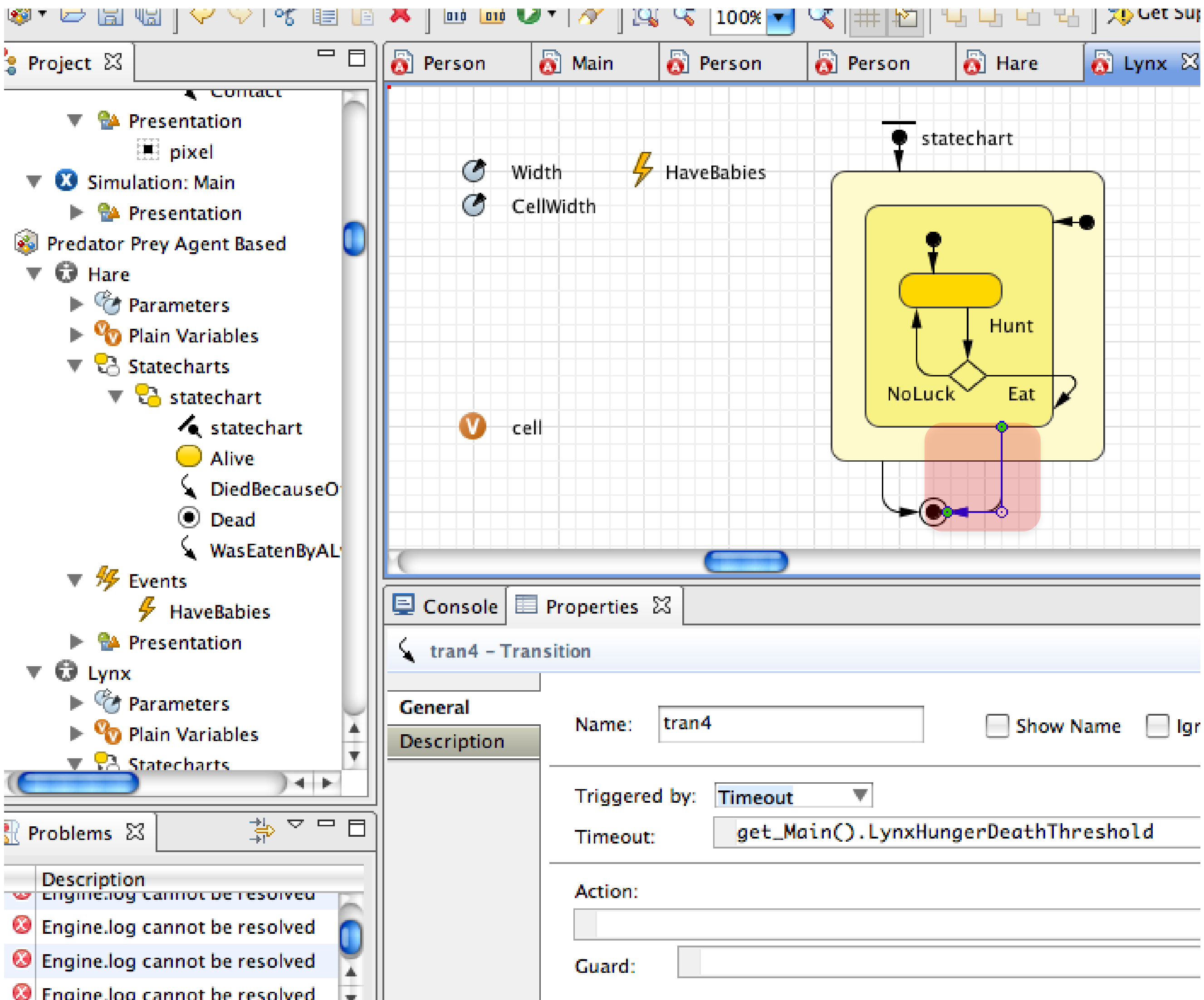


Natural Death Transition

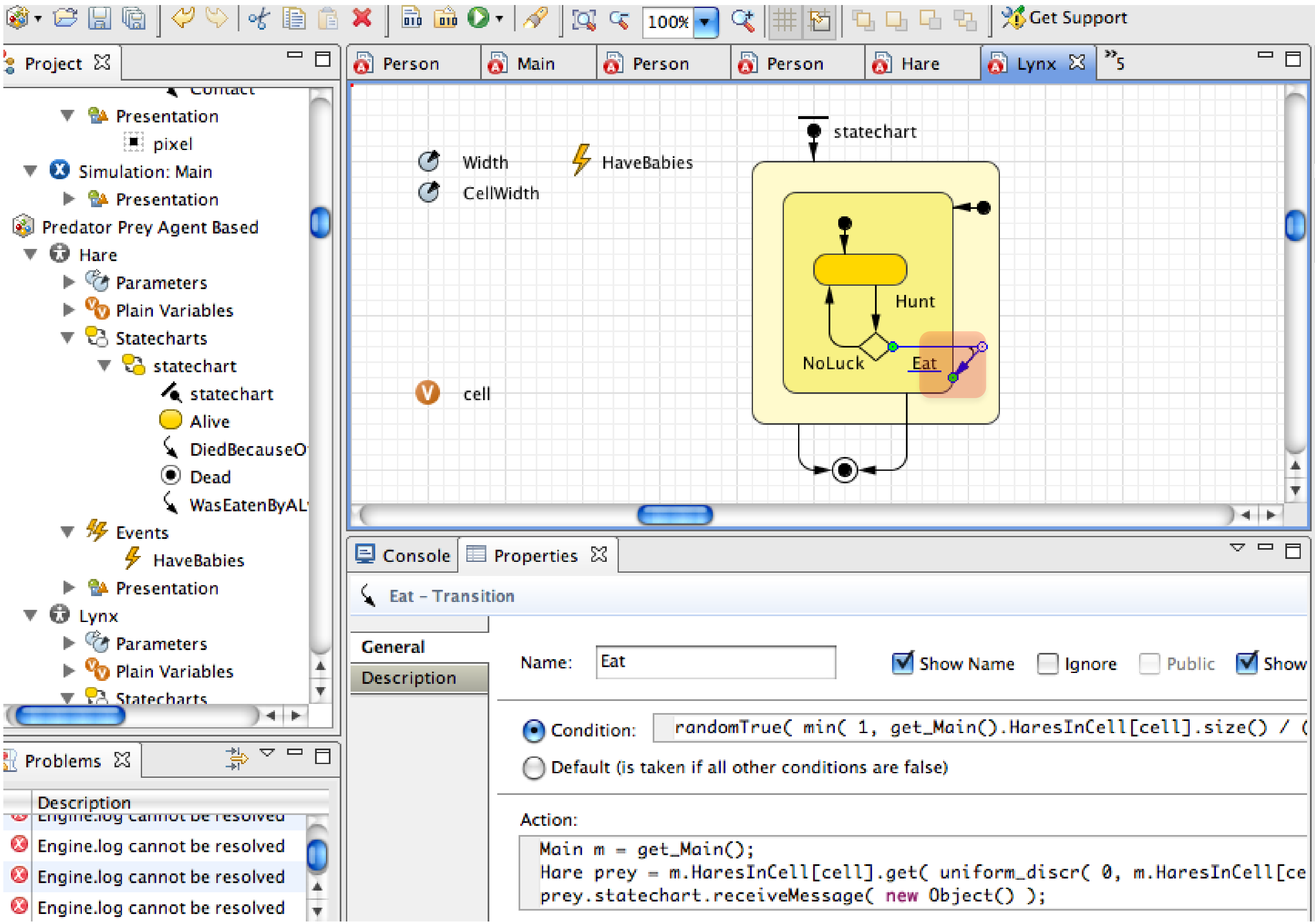


Death By Hunger

(Note that Depends on Time in State – i.e. time Since last ate)



Eating Transition Leaves & Reenters Middle State



Tips on Statechart Code

- Each State & Transition has an integer index
  - This by accessed via a (static) constant holding the name of state within the statechart class (*statechart*.*StateName*)
- To determine length of time spent in state
  - *Statename*.getLocalTime(*StateIndex*)
- To determine current state
  - *statechart*.getActiveSimpleState()
- To find out if a state (either simple or composite) is currently active
  - *statechart*.isStateActive(*StateIndex*)