Expressing Discrete Inter-Agent Dynamics: Messaging & Events

Nathaniel Osgood

3-1-2011

Discrete Agent Coupling via Messages

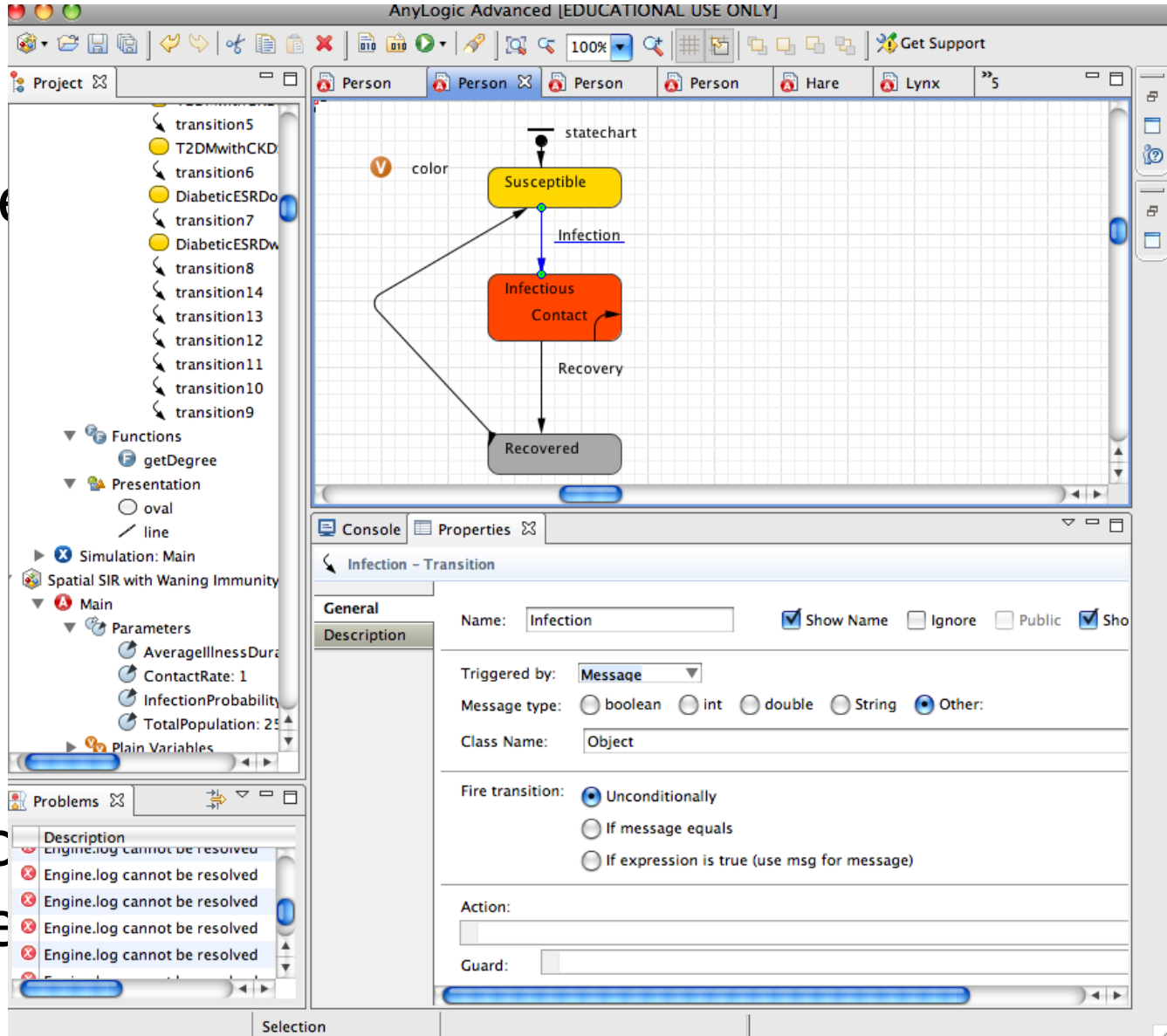
- Within AnyLogic, agents can be coupled by either discrete (instantaneous and individuated) or continuous (ongoing and gradual) coupling
- The preferred mechanism for discrete coupling is *messages* sent between agents
  - Many types of messages payloads are possible
  - An agent can send a given message to one or more agents
  - Frequently messages are sent locally to neighbors within the environment
    - Neighboring nodes on the network
    - Nearby neighbors in space

Messages & Statecharts

- Messages may be handled in many ways
- One of the most common ways in which messages are handled is by statecharts
  - A transition can be triggered ("guarded" or gated) by a message
  - A transition may be associated with an action that fires off a message to other agents (or to other statecharts within the agent)

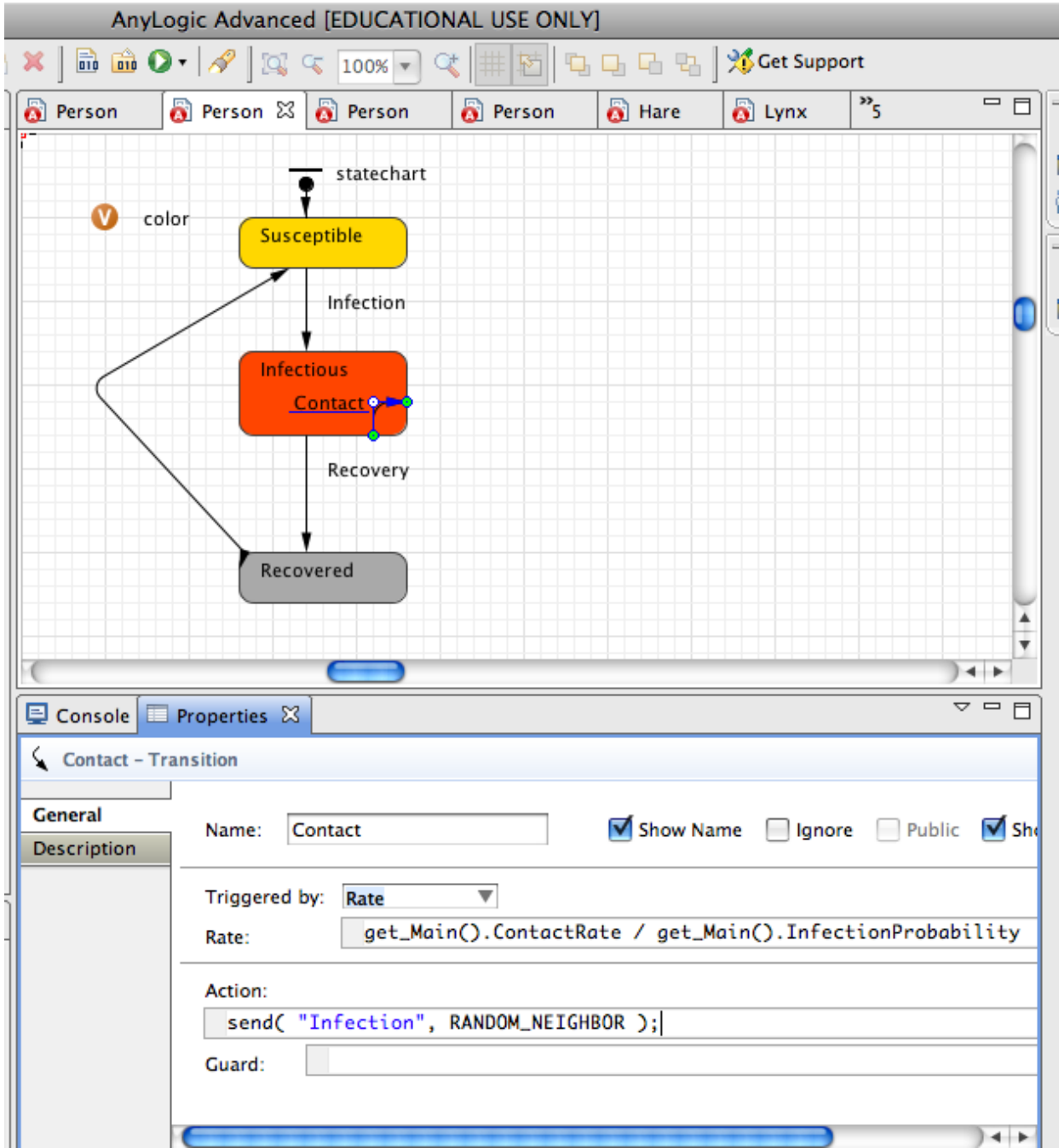
Receiving a Message

- In this case, only 1 message type exists, so the simple fact that a message has been received is sufficient; there is no need to inspect message conte



Sending a Message

- (Self-transition because remains in state)



Message Sending

- Messages may be sent to either
  - A particular, explicitly specified agent
  - An implicitly specified class of agents
    - Neighboring agents in the environment topology
    - Random agents
    - All agents
    - Any connected agents
    - All connected agents
- Mechanism:
  - send(*Message*, *DestinationObject*)
  - send(*Message*, *AgentClassId*)

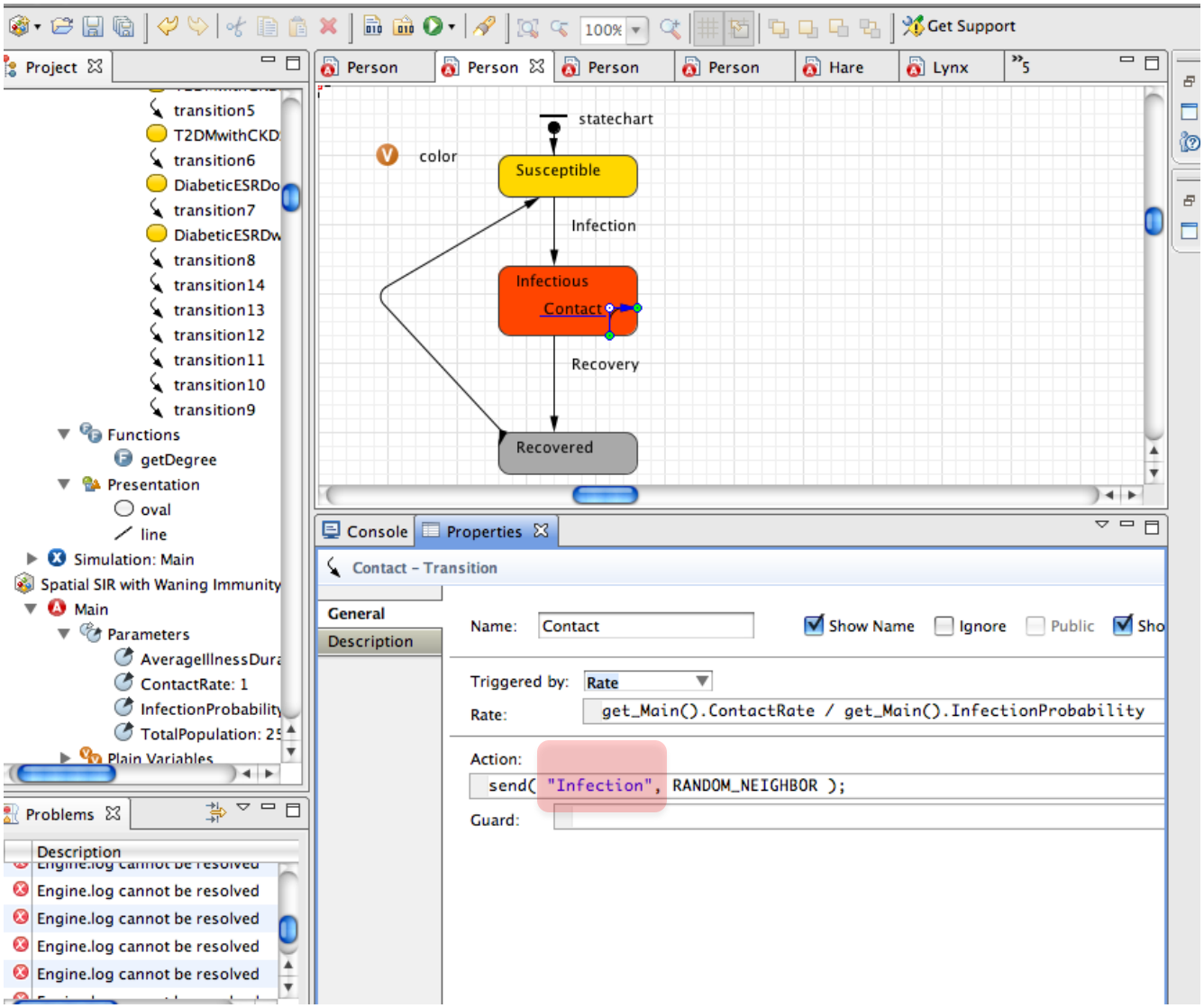
Synchronous vs. Asynchronous Delivery

- Messages may be sent in two ways
  - Via ***send***: Asynchronous
    - Delivery occurs sometime after call to send
  - Via ***deliver***
    - Synchronous
    - Risks infinite loops in delivery (if destination also calls deliver in the reverse direction)

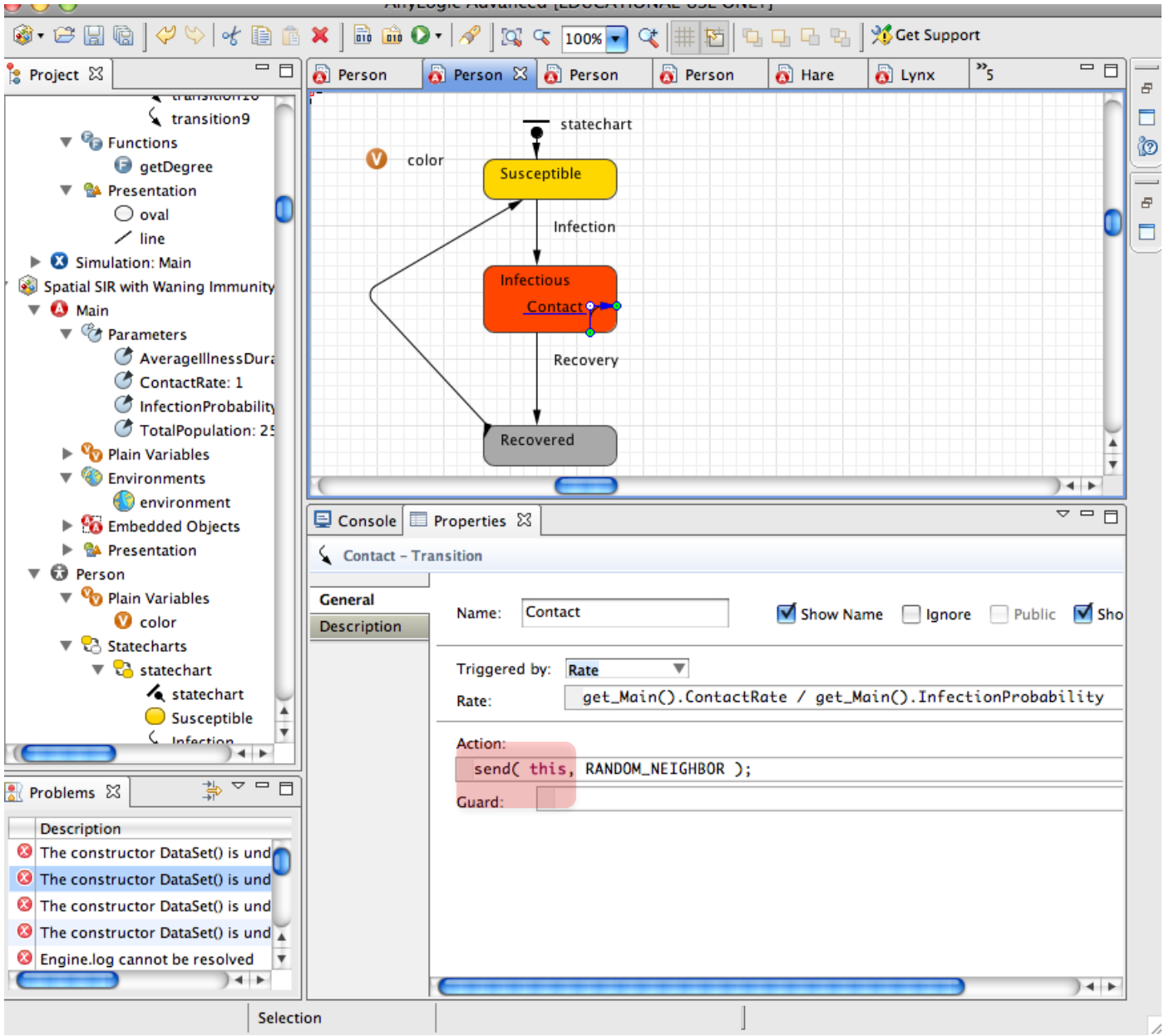
Message Payloads

- Sometimes just the fact that a message has been sent provides us with all of the information we need
- Sometimes just distinguishing different message types is sufficient
- We will sometimes send messages with payloads of data that provide extra information, e.g.
  - The agent that sent the message (eg that is infecting us)
  - Particular parameters
- Can send multiple message types
  - Boolean/int/double/String/Other (can specify class type)

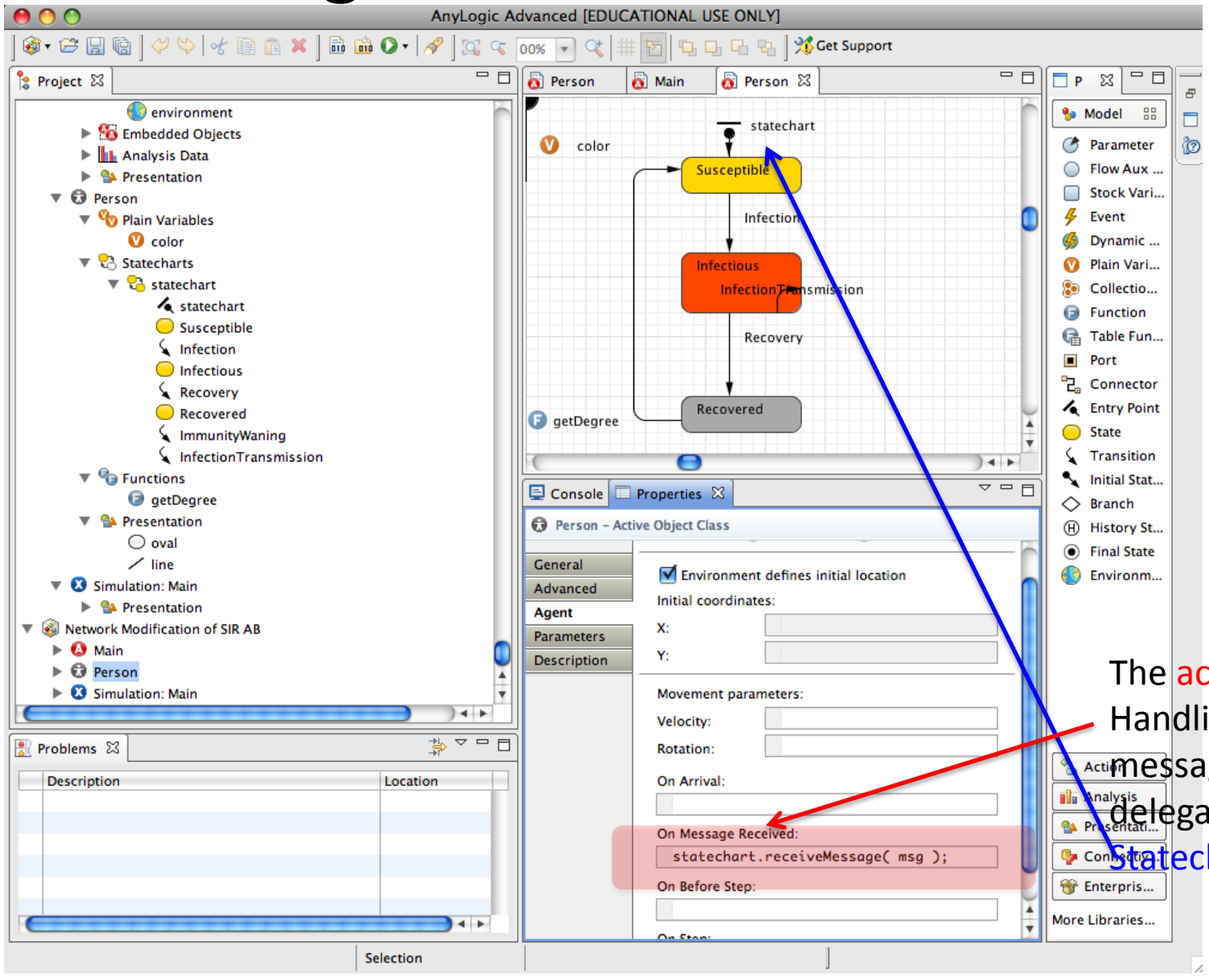
Sending a Message with a String Payload



Sending a Message with Object Payload

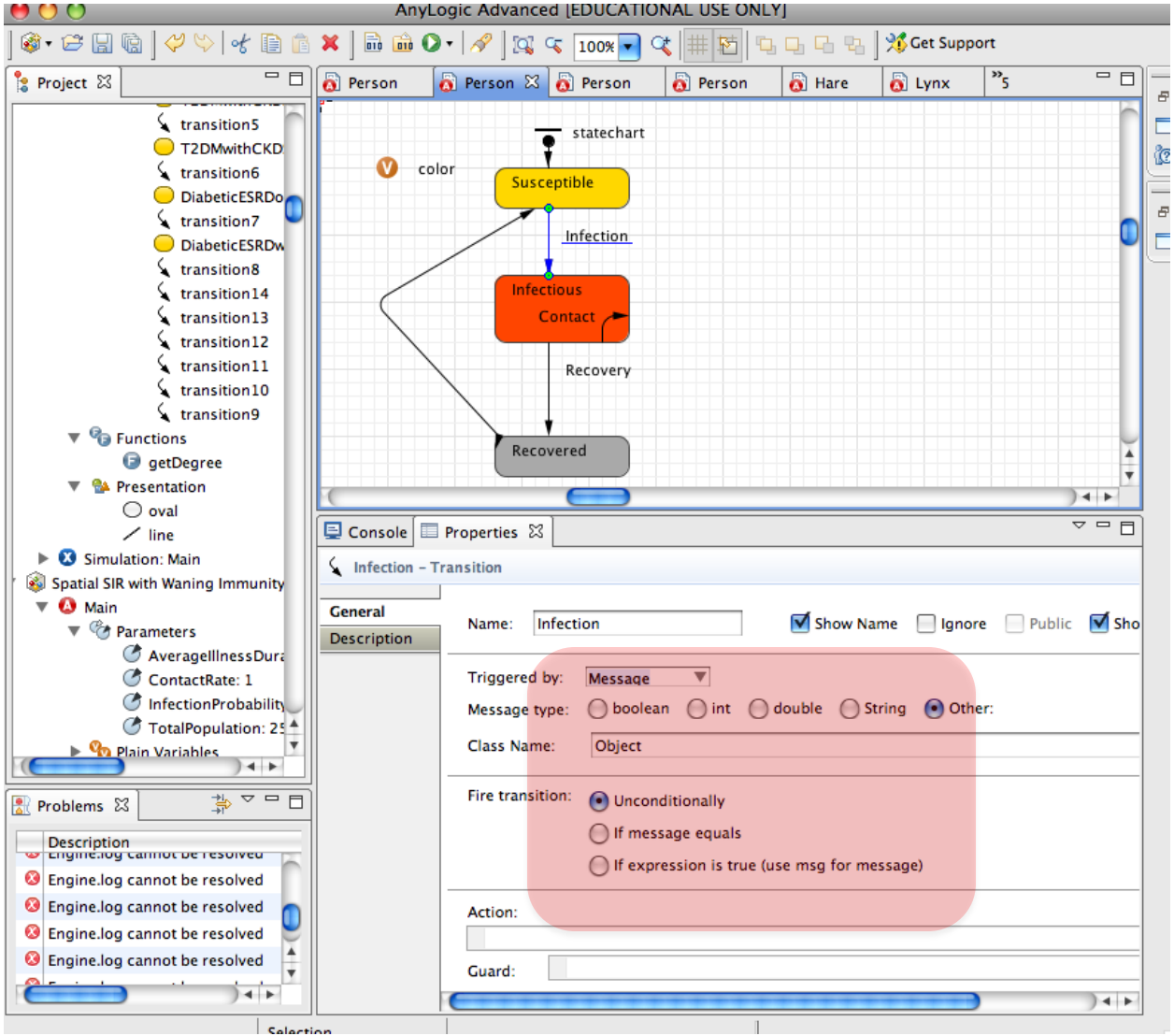


Receiving a Message: Forwarding Messages on to the Statechart



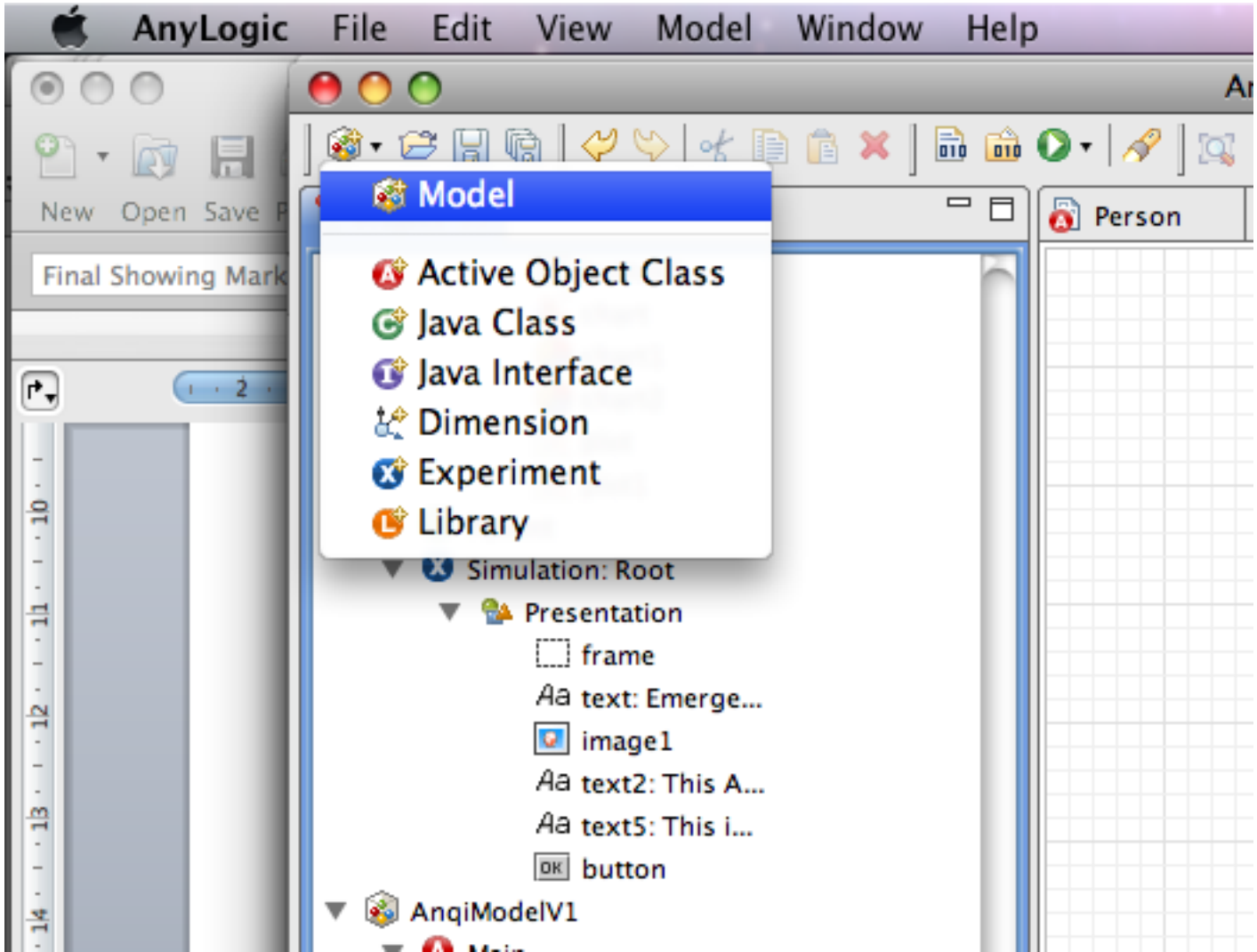
The action for Handling received messages delegates to the Statechart object

Receiving a Message

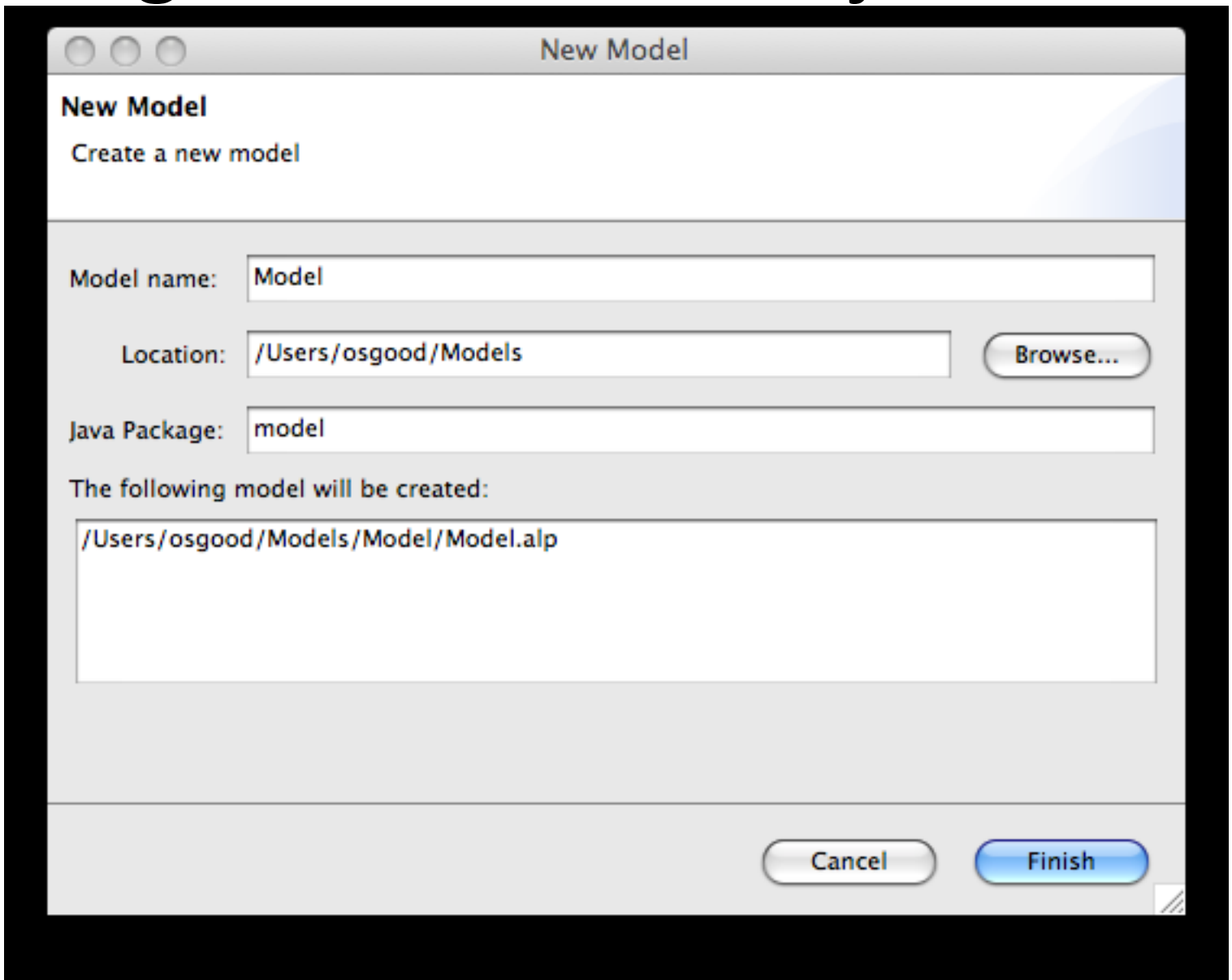


Building Up a Simple Agent-Based Model: The Manual Technique

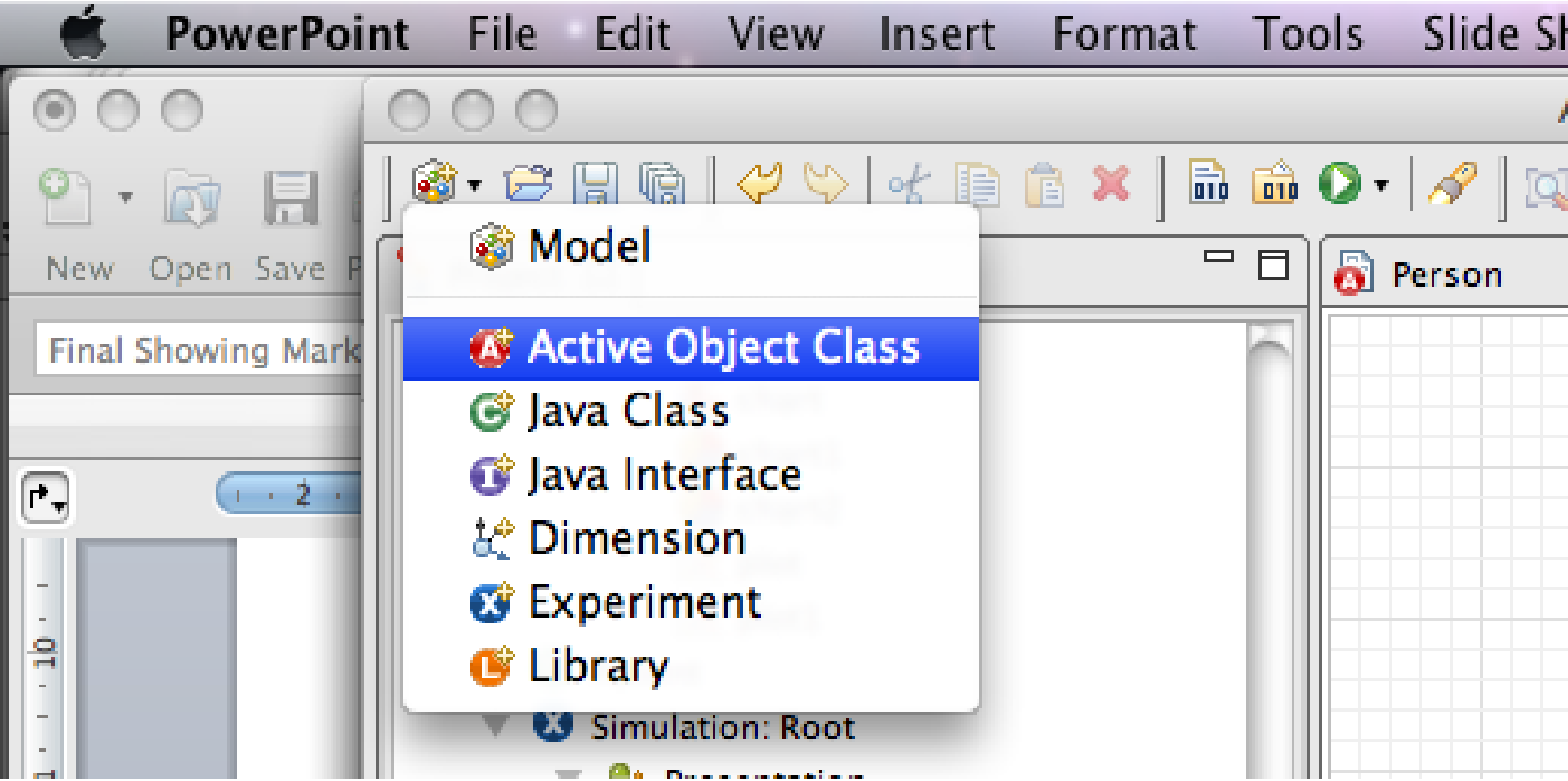
Add a New Model Project



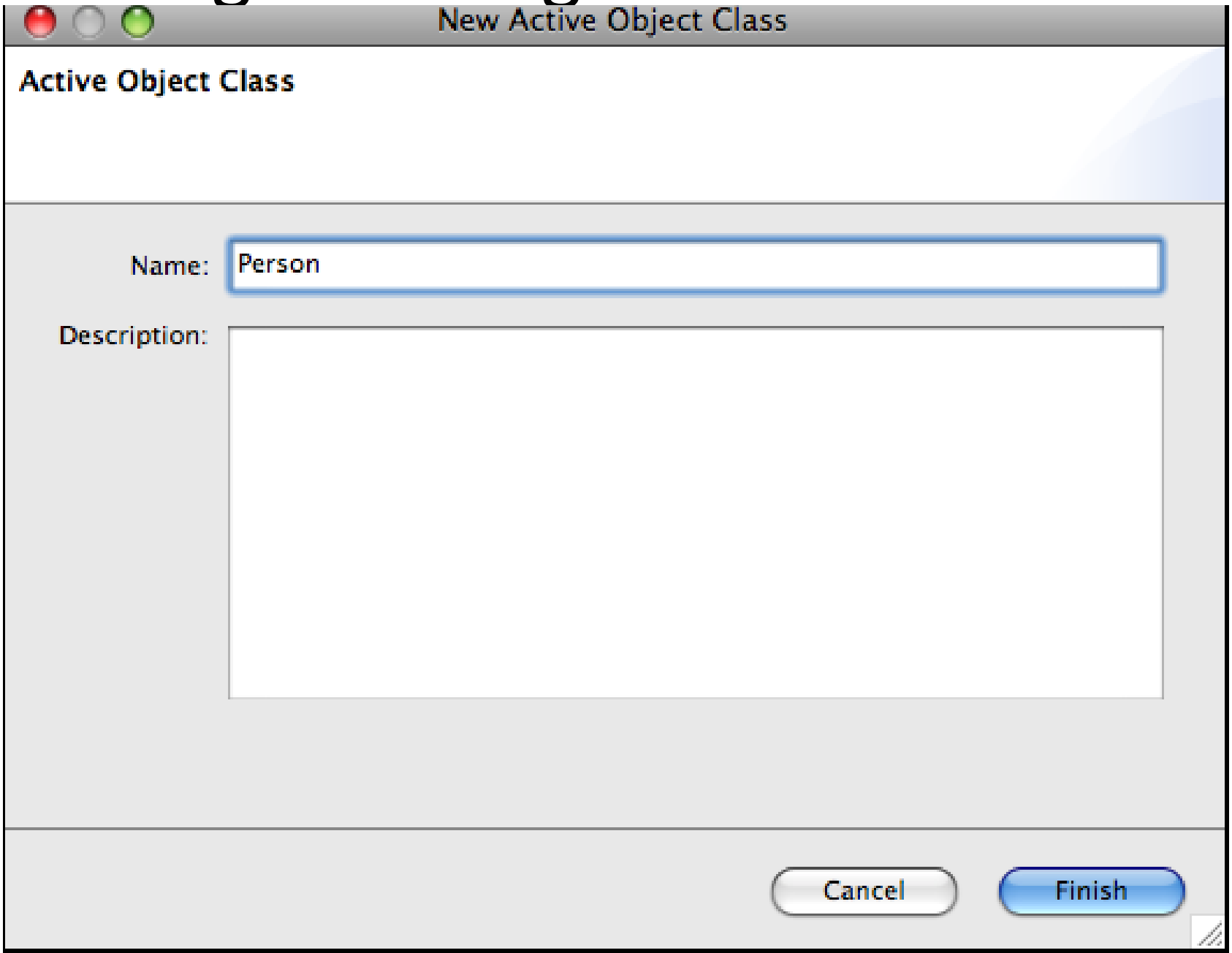
Filling in the Model Project Details



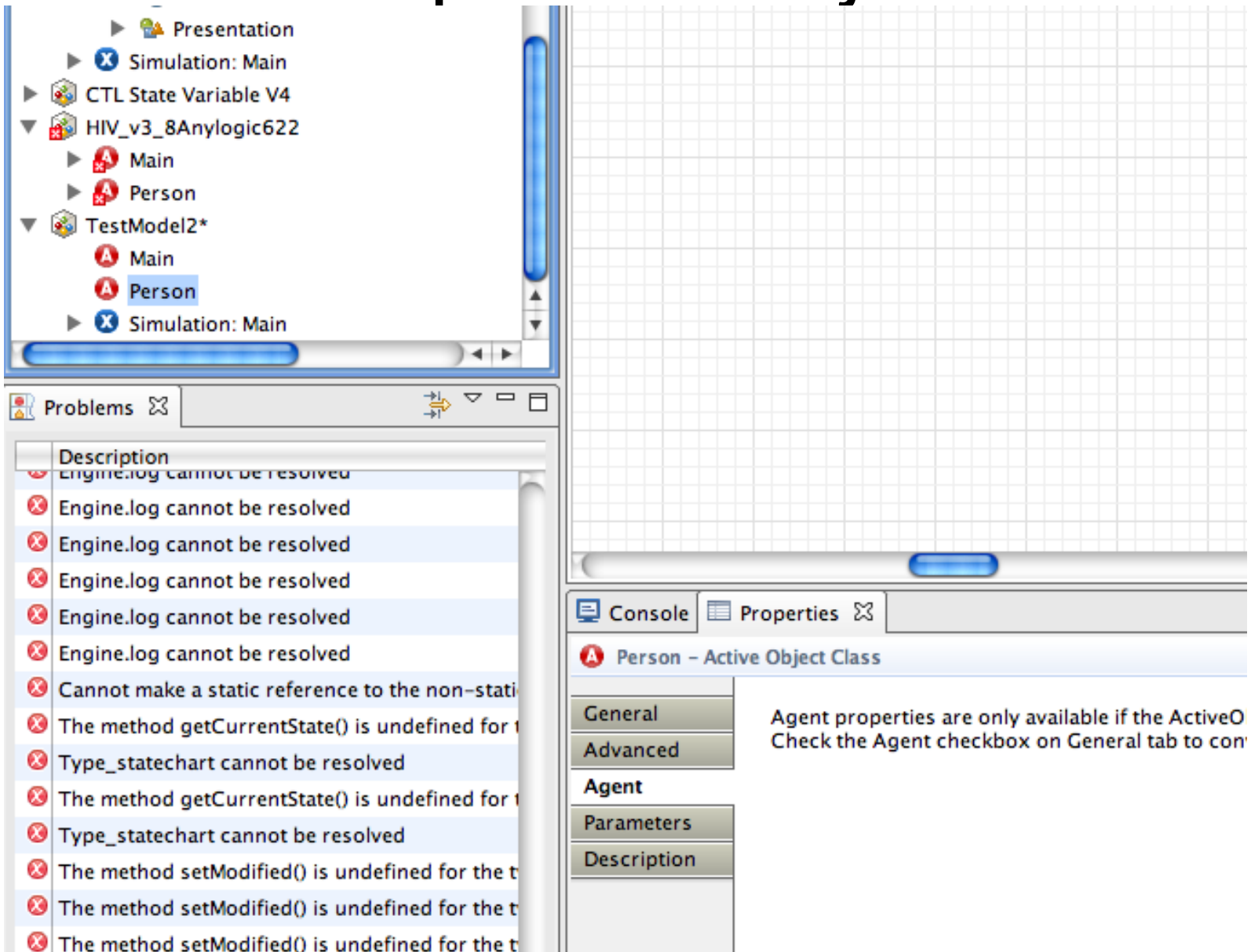
Add an Active Object Class



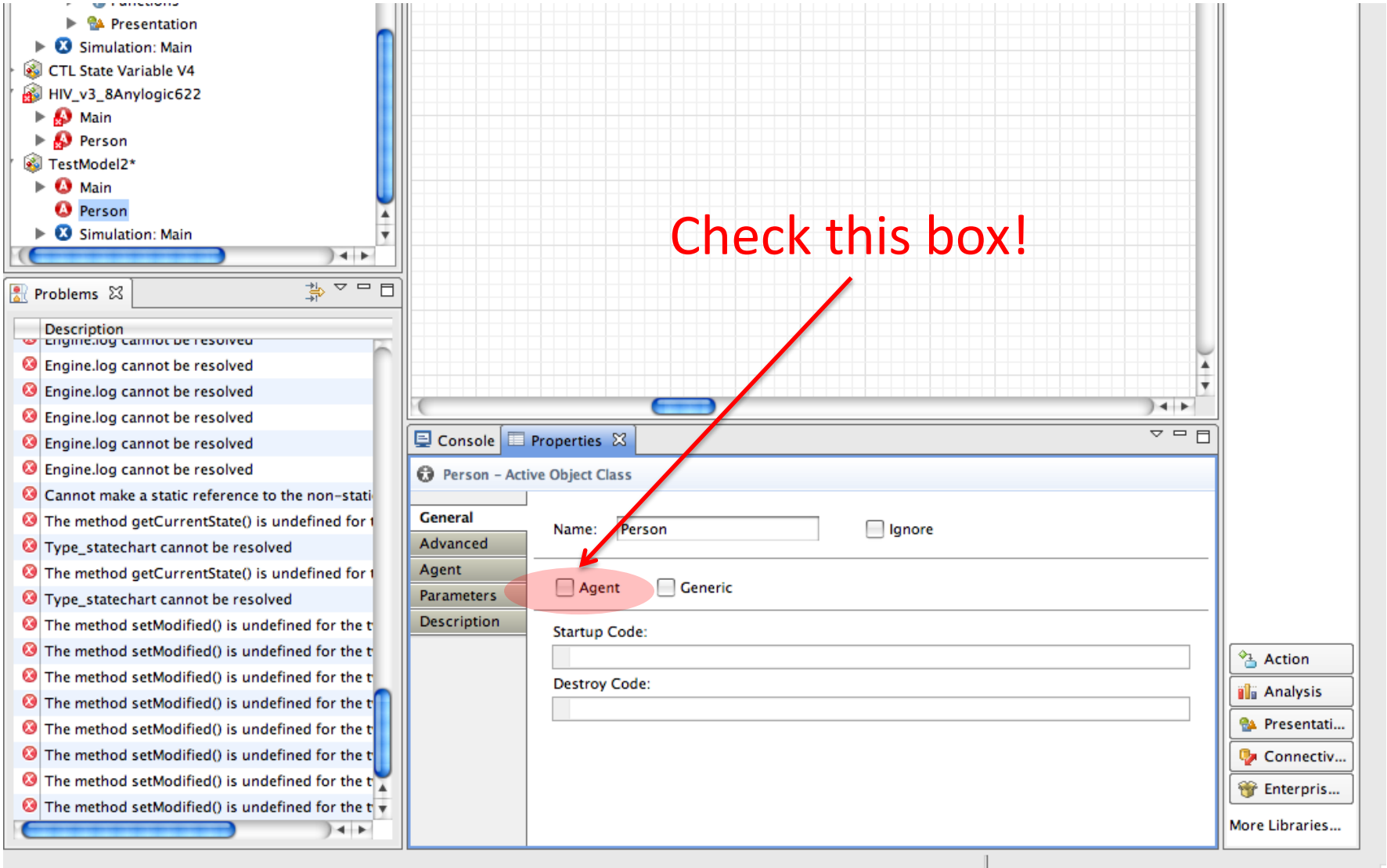
Filling in the Agent Class Details



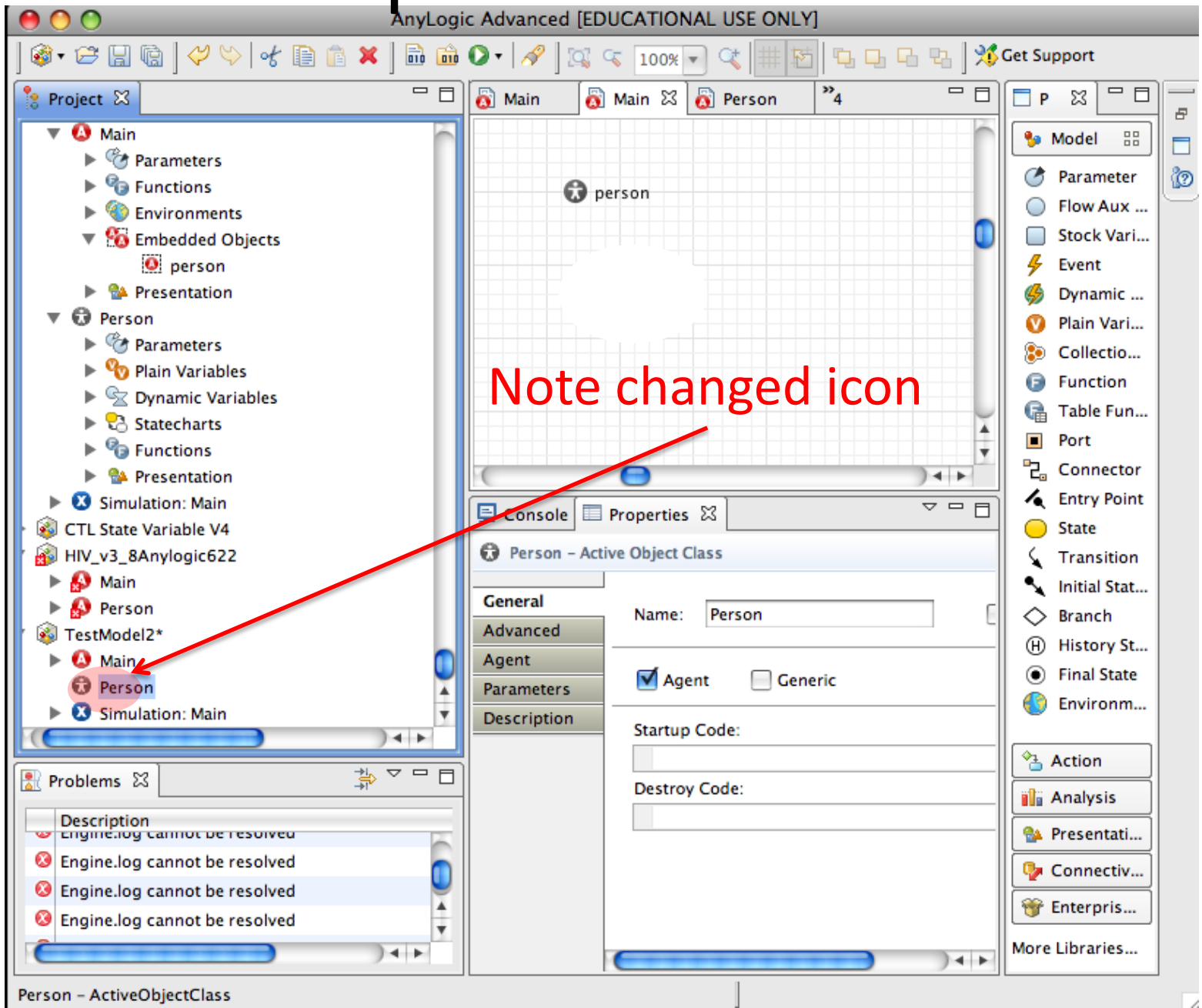
The Updated Project



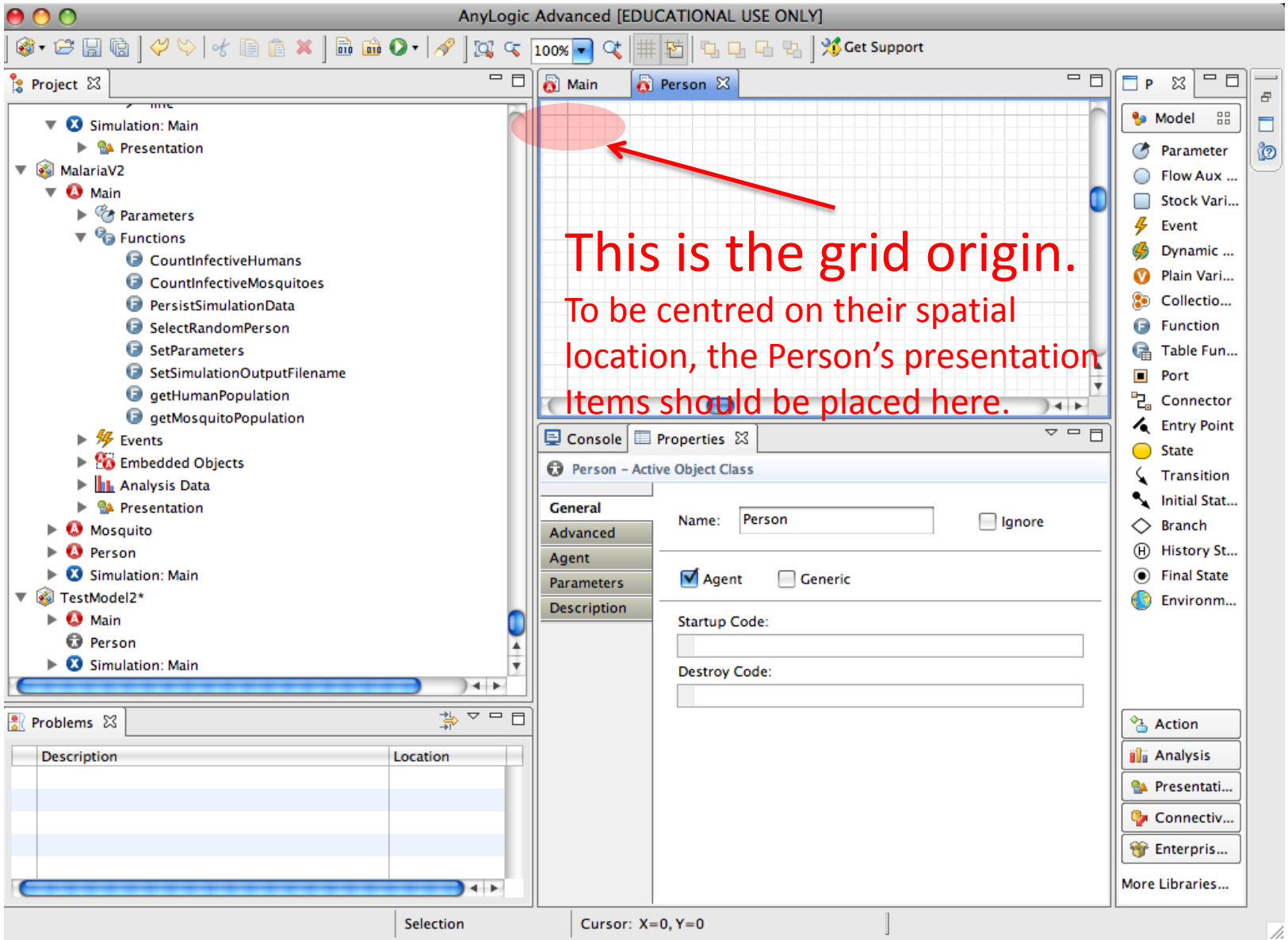
Declaring “Person” as an Agent



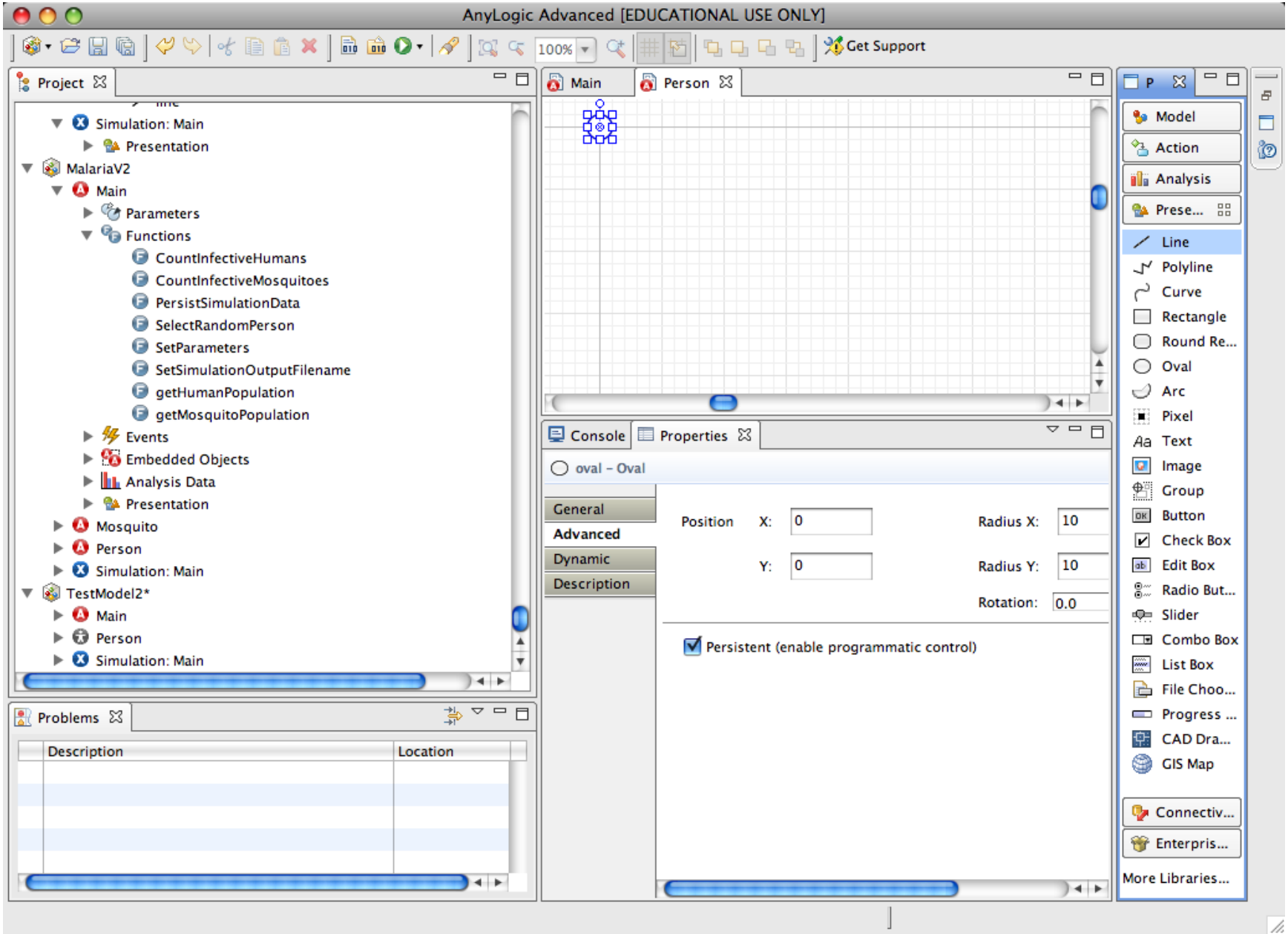
Updated Result



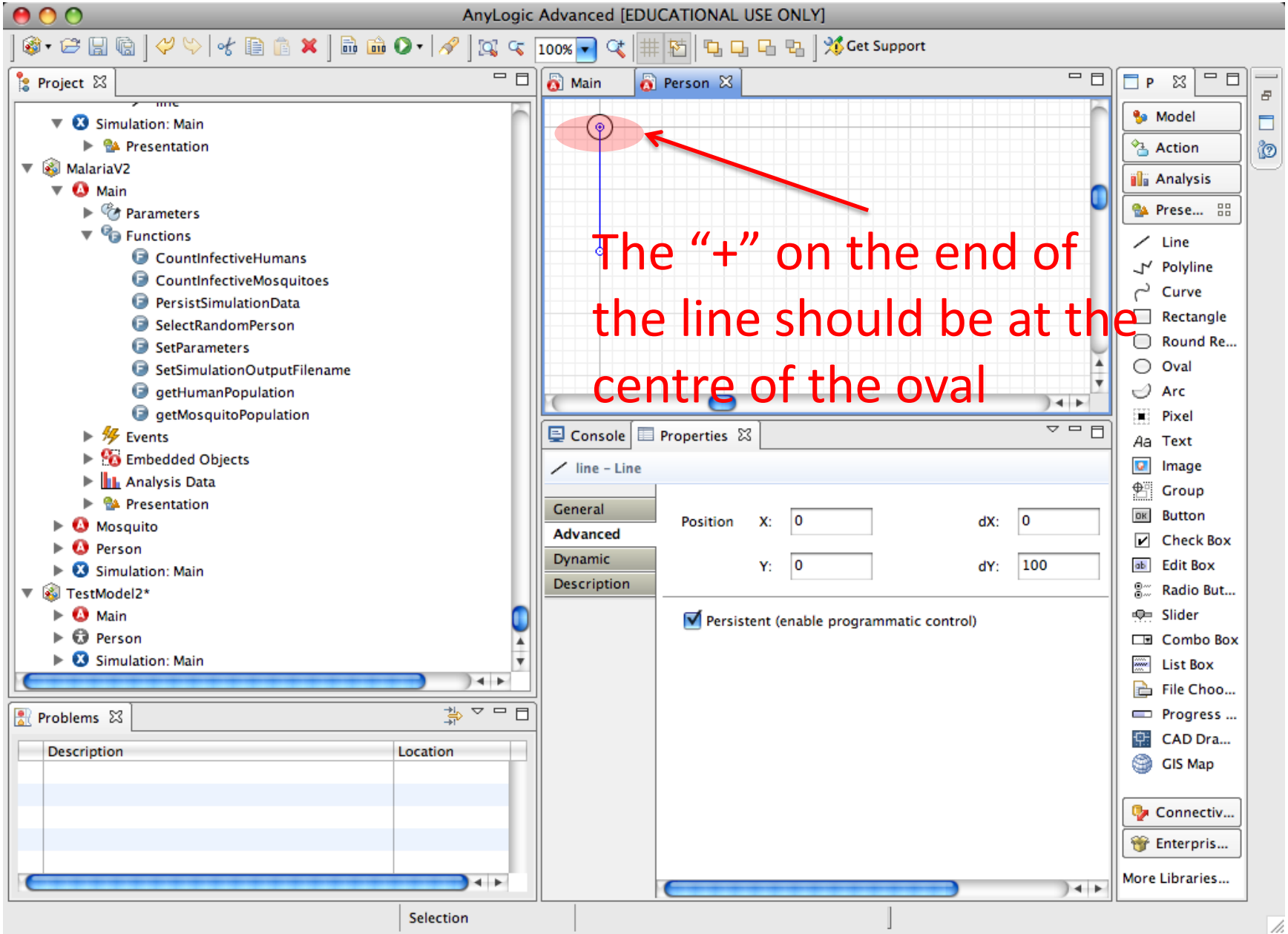
Double-Click on “Person” & Scroll Until you See The Cross-Hairs



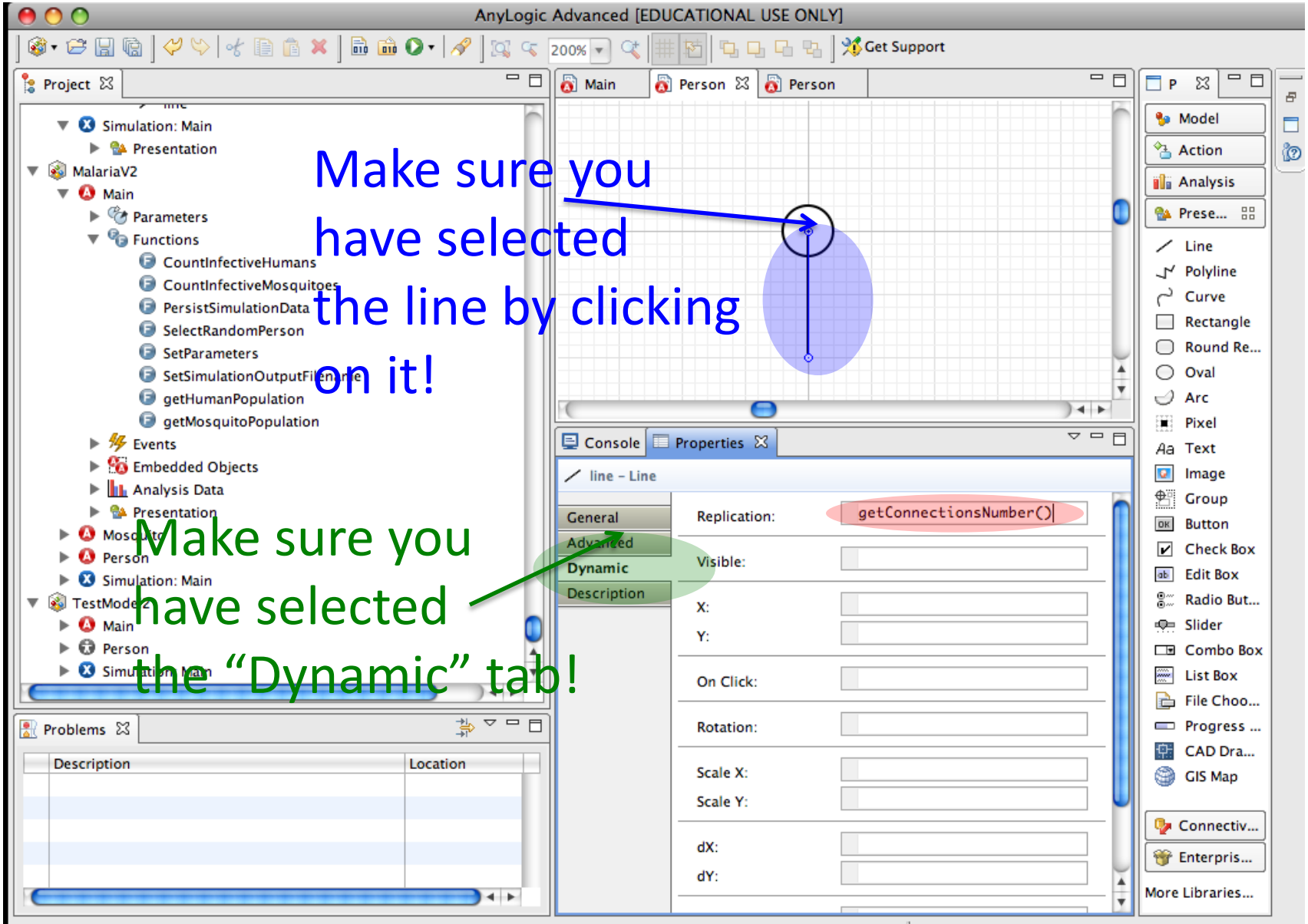
Create an Oval at the Origin (Cross-Hairs)



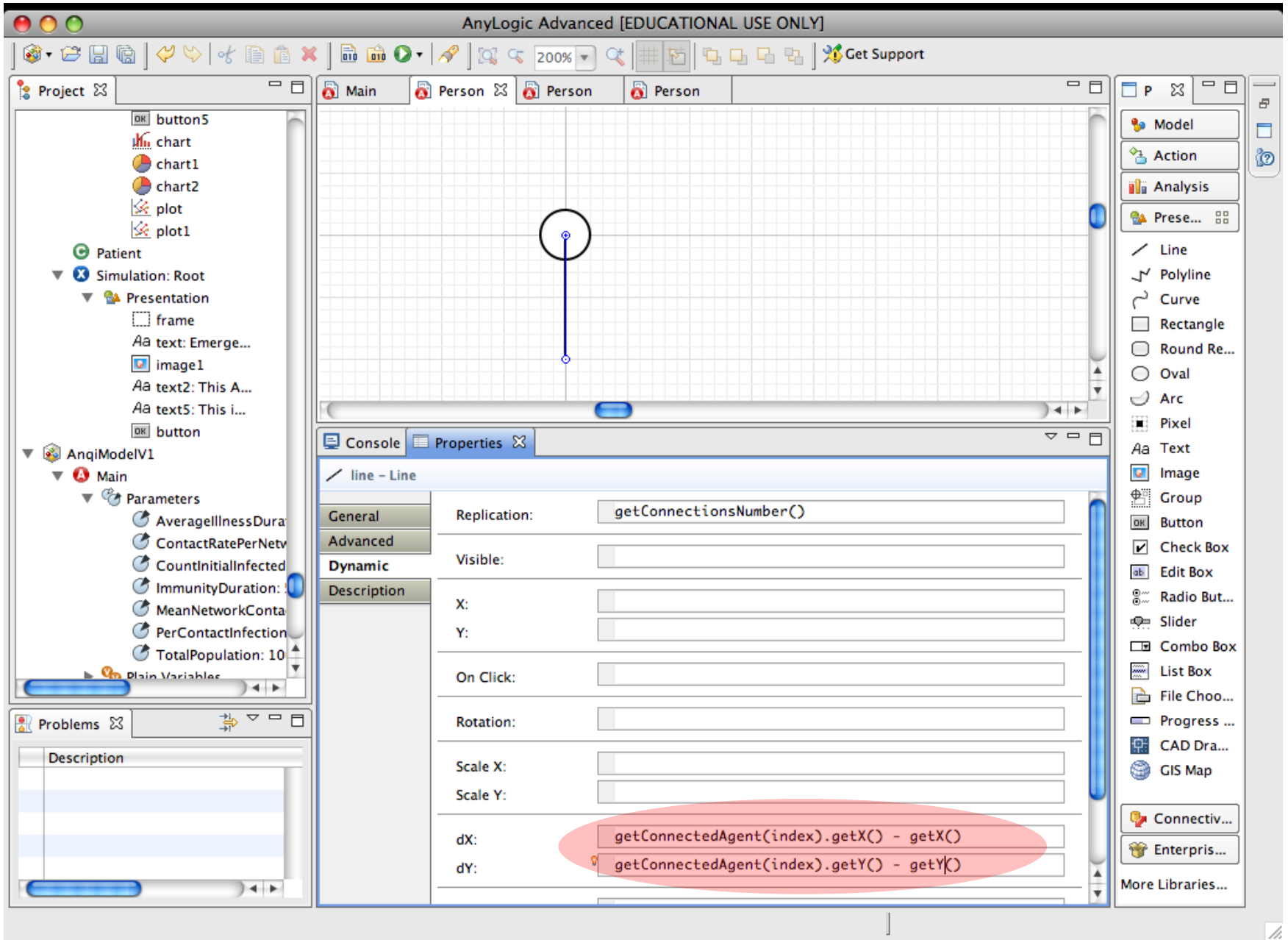
From the Centre of the Oval, Draw a Line



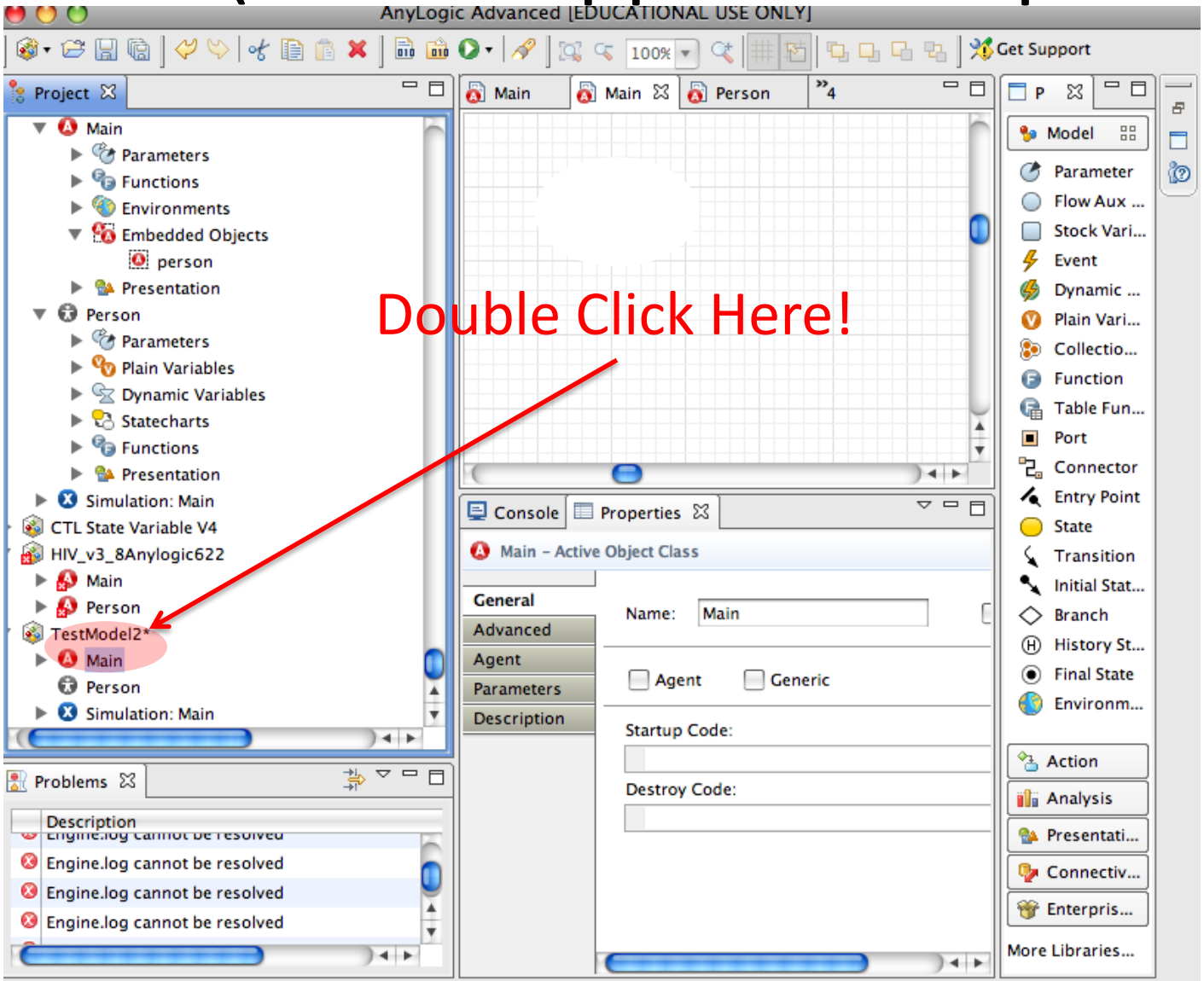
Set the “Replication” Dynamic property of the *Line* so there is 1 for each connection



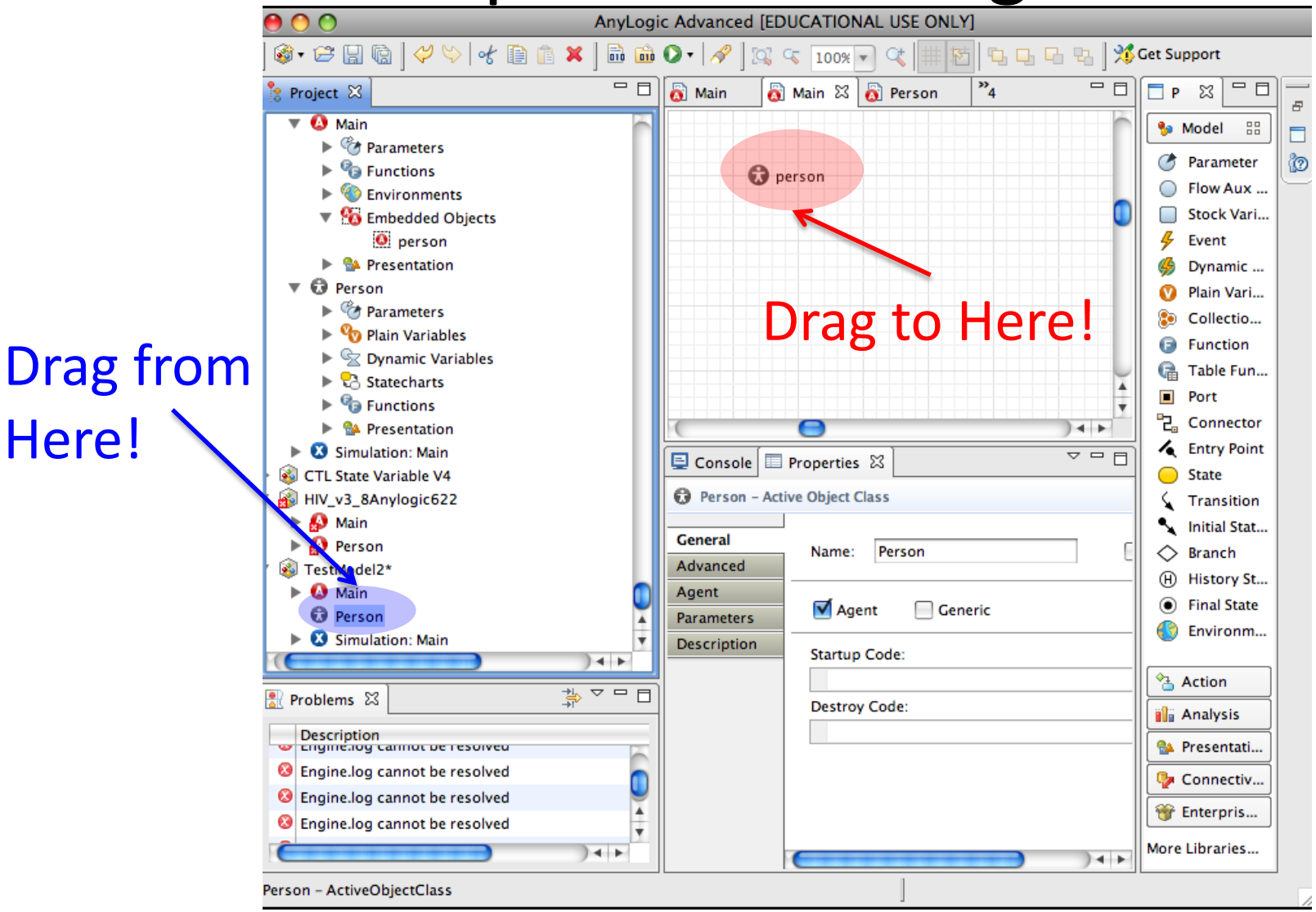
Also set the “dX” and “dY” properties



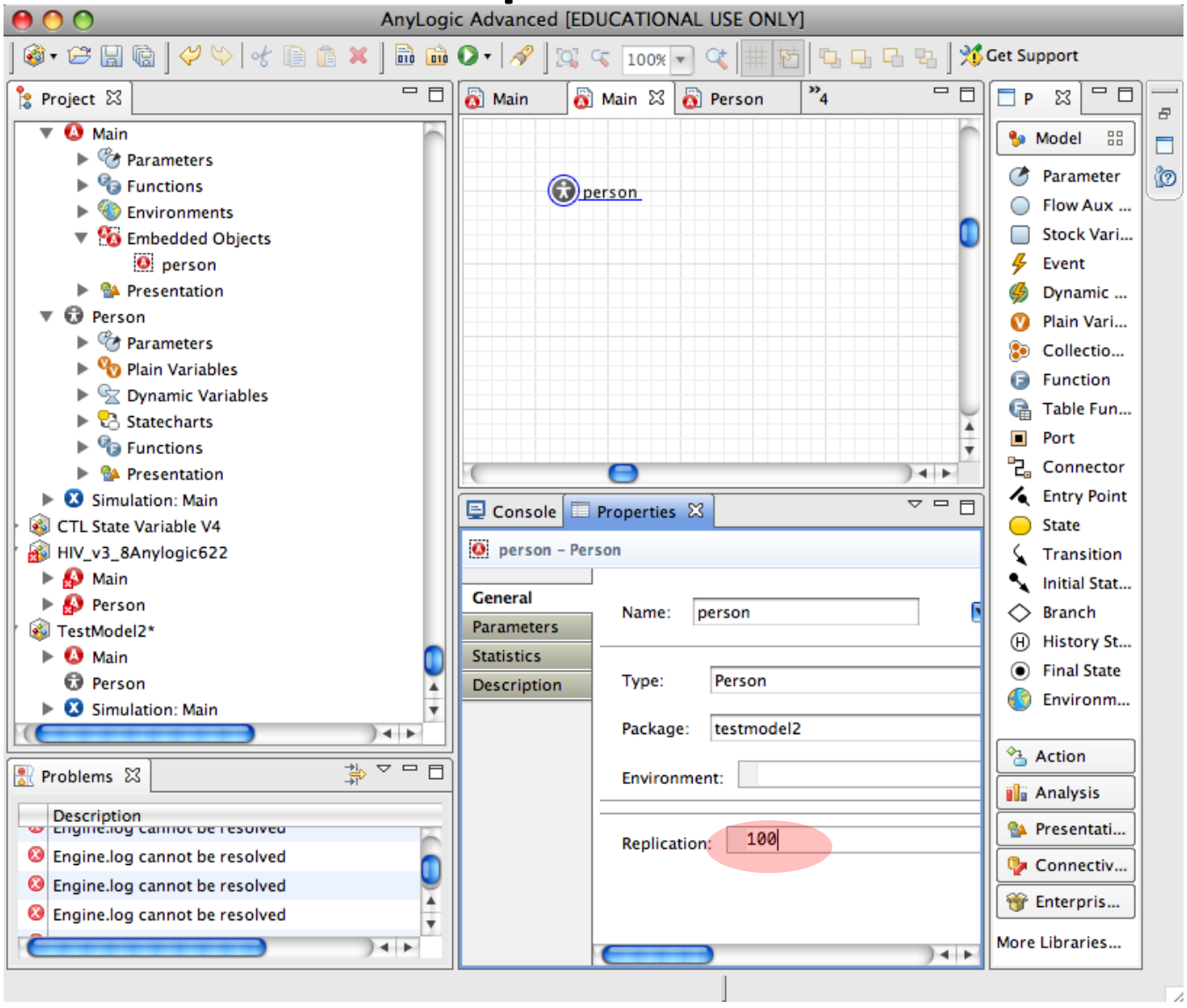
Double Click on “Main” class Name to View this it (Should Appear on Top Tab)



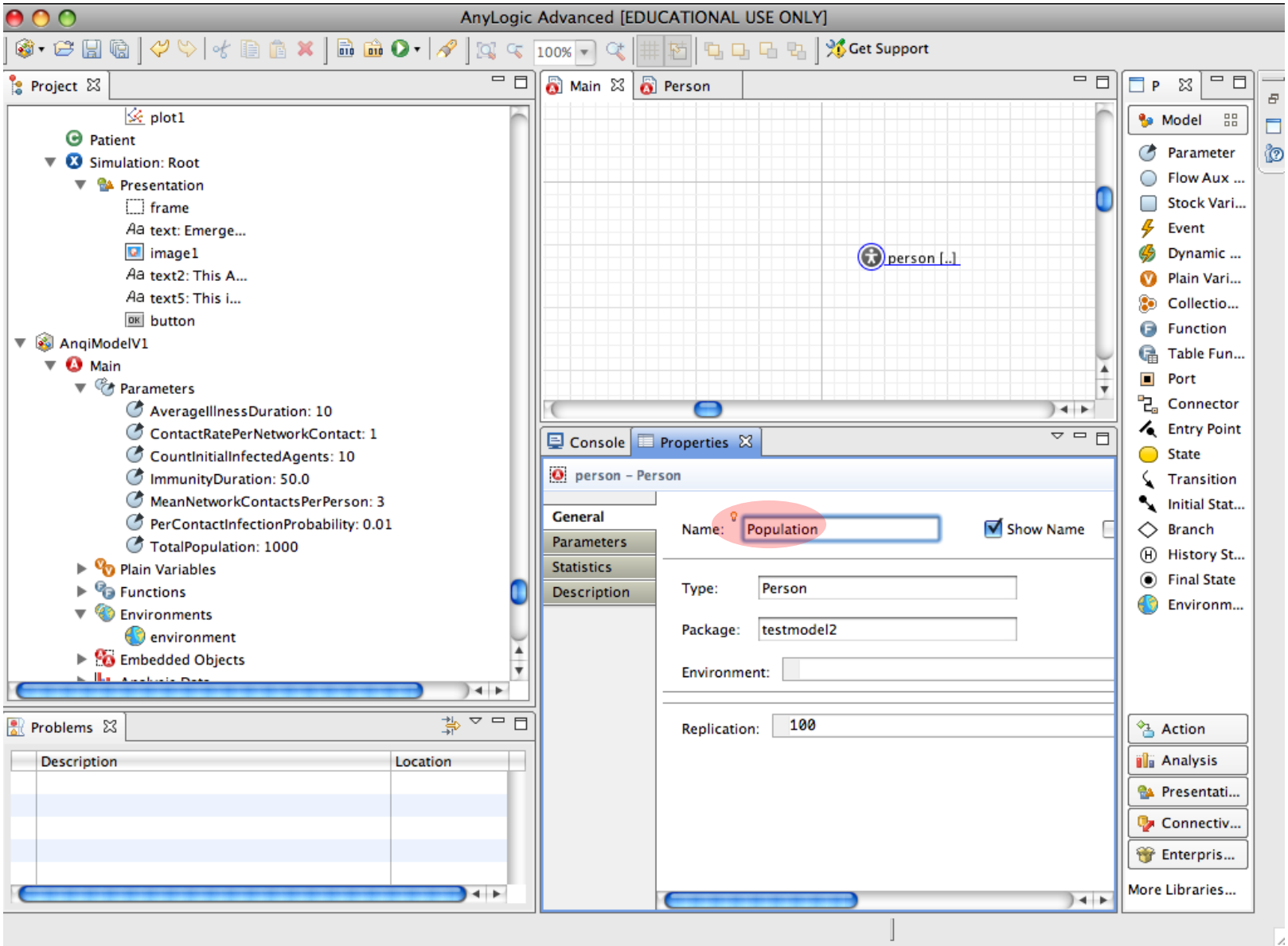
Click and Drag from “Person” into the Space on the Right



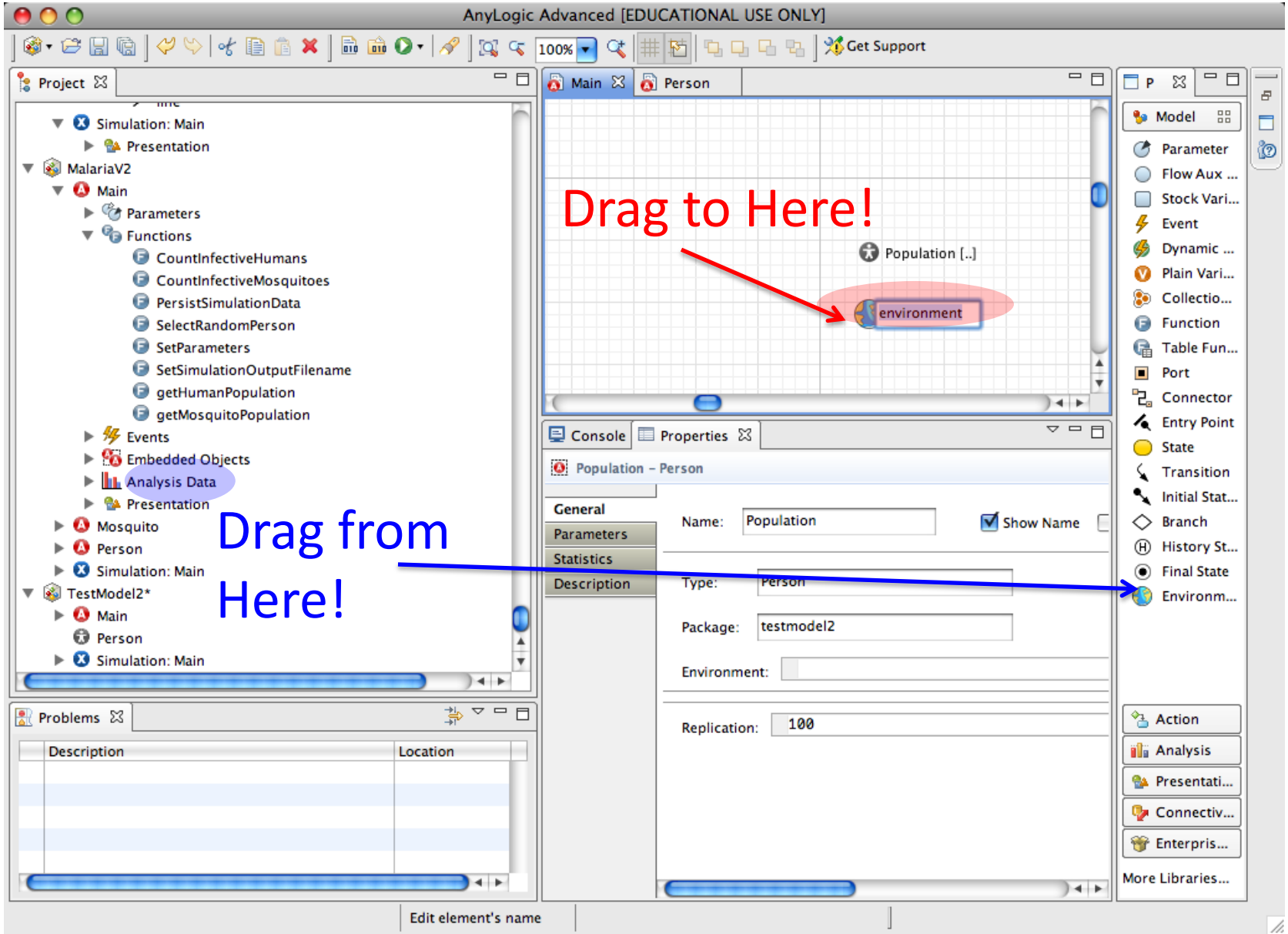
Set the Count of Agents in the Agent Population



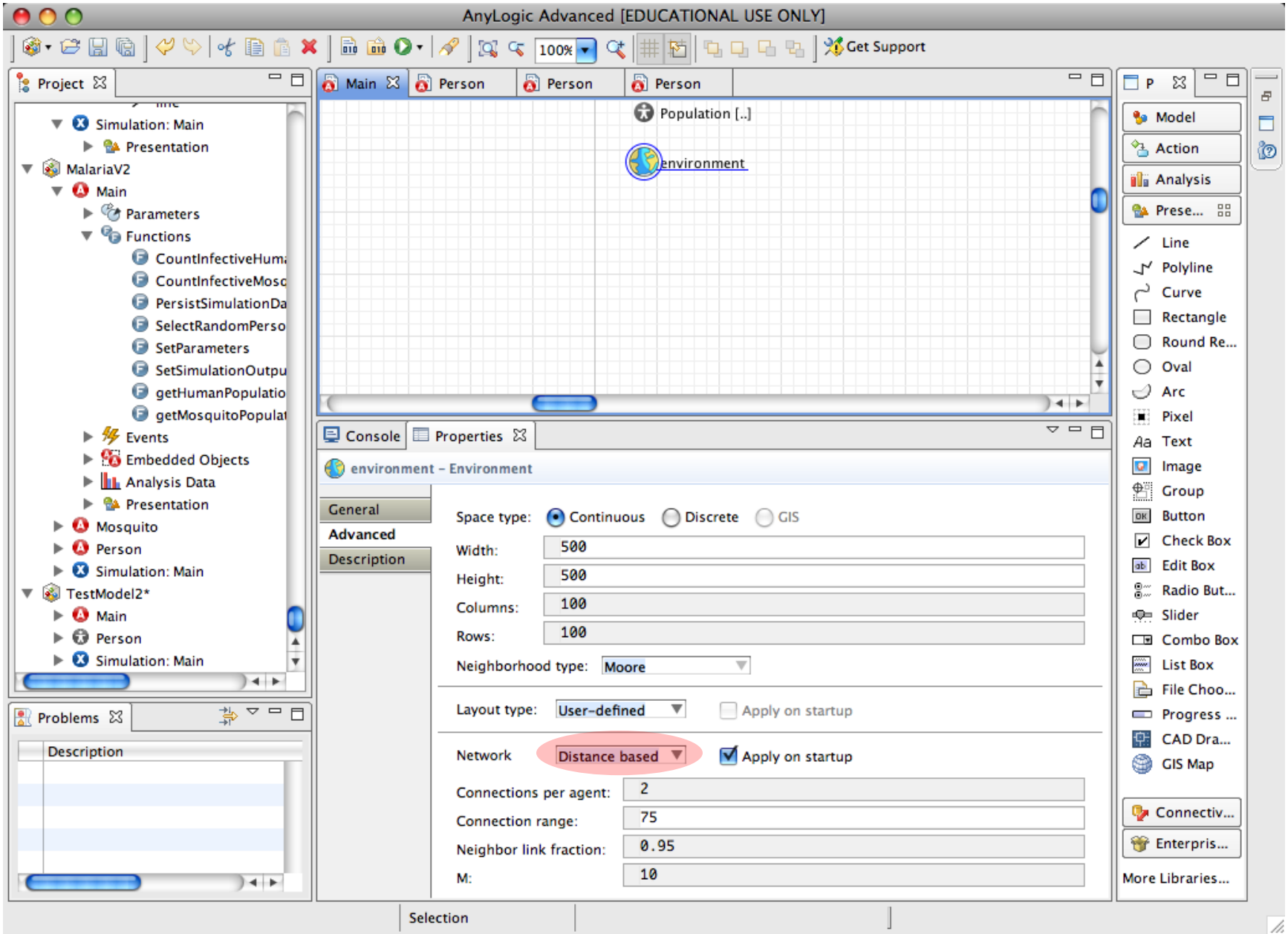
For Clarity, Rename “Person” to “Population”



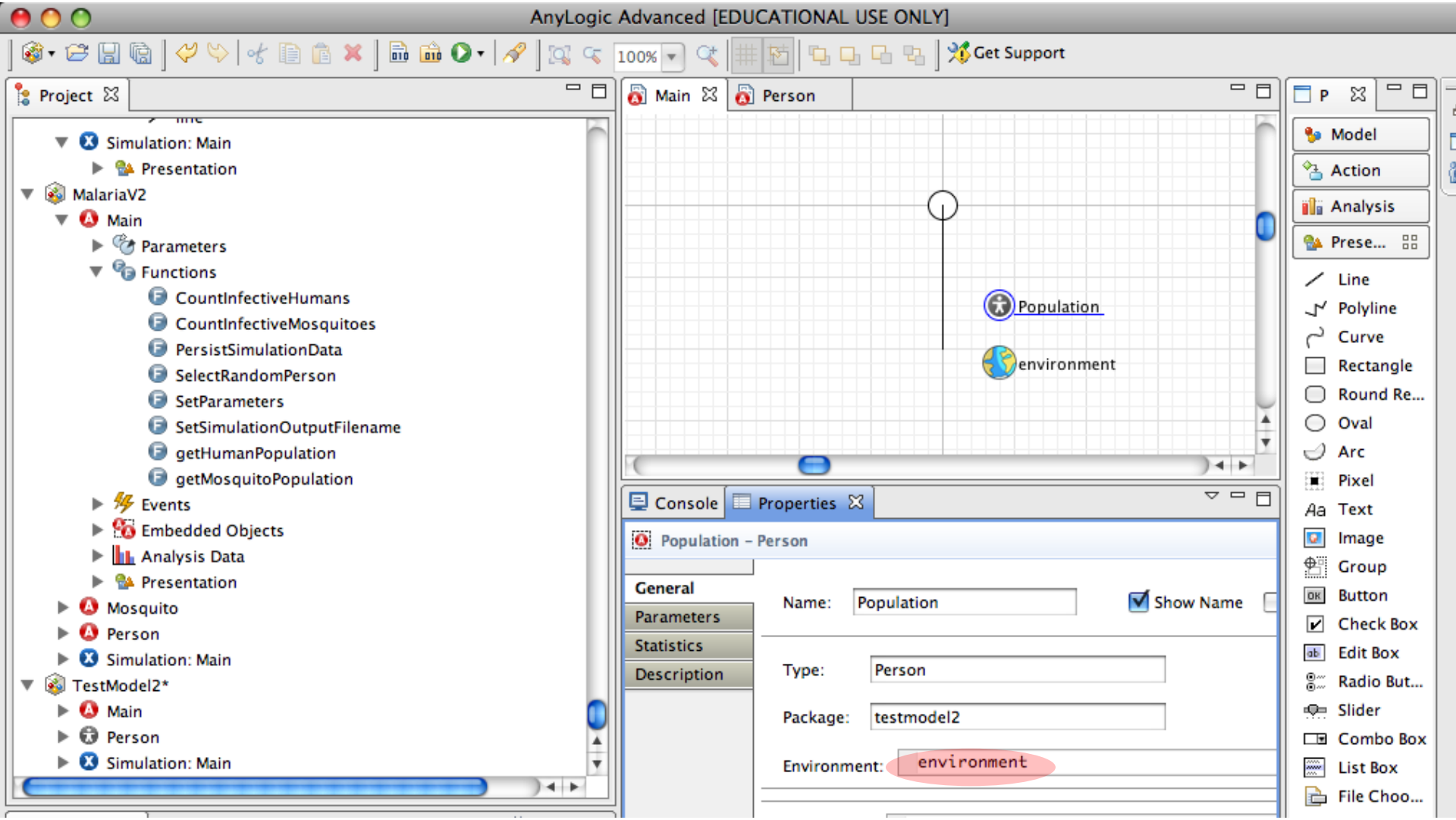
Add an Environment



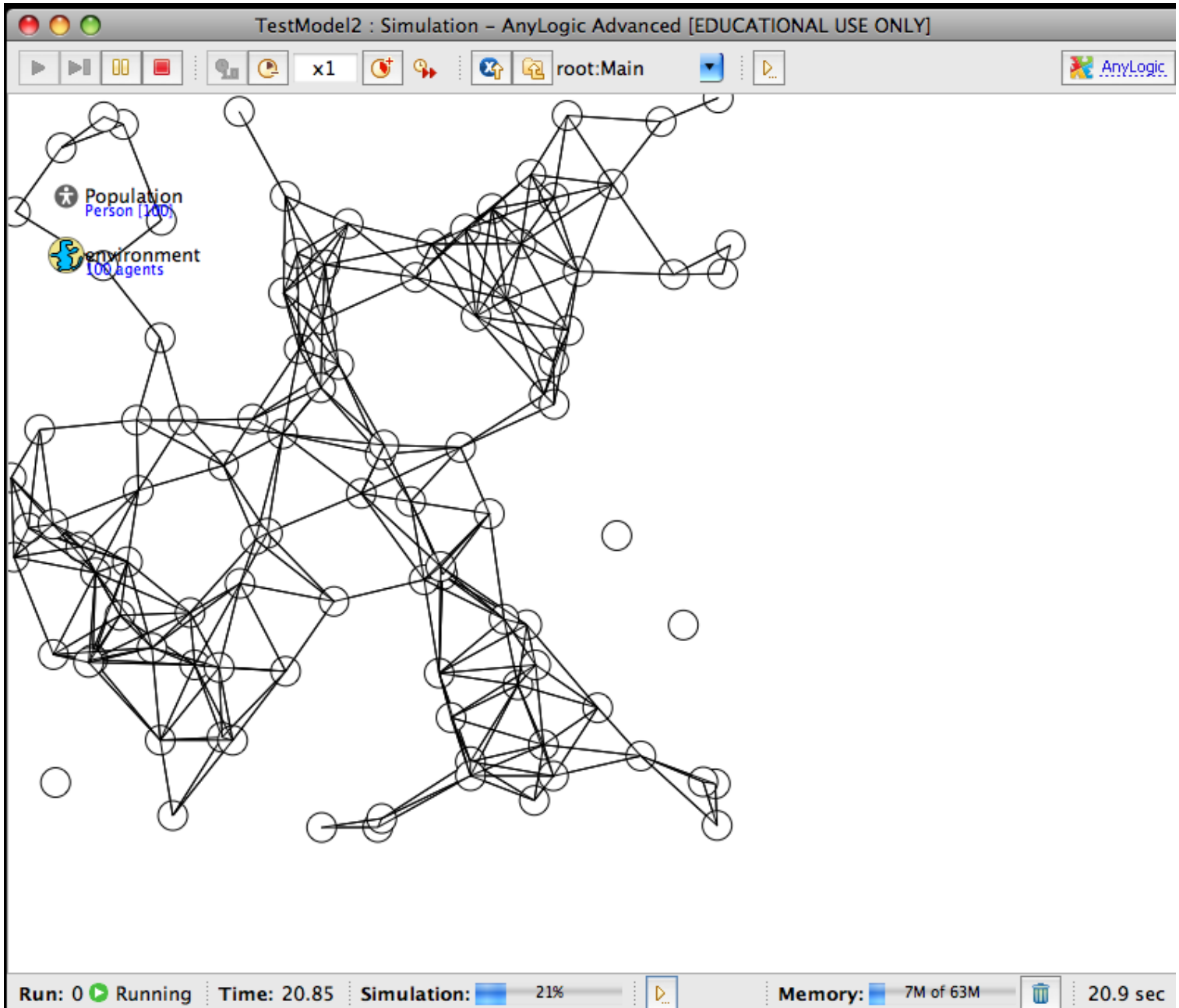
Set the Network Type to Use



Make the Population Depend on the Environment (for placement, connections, etc.)



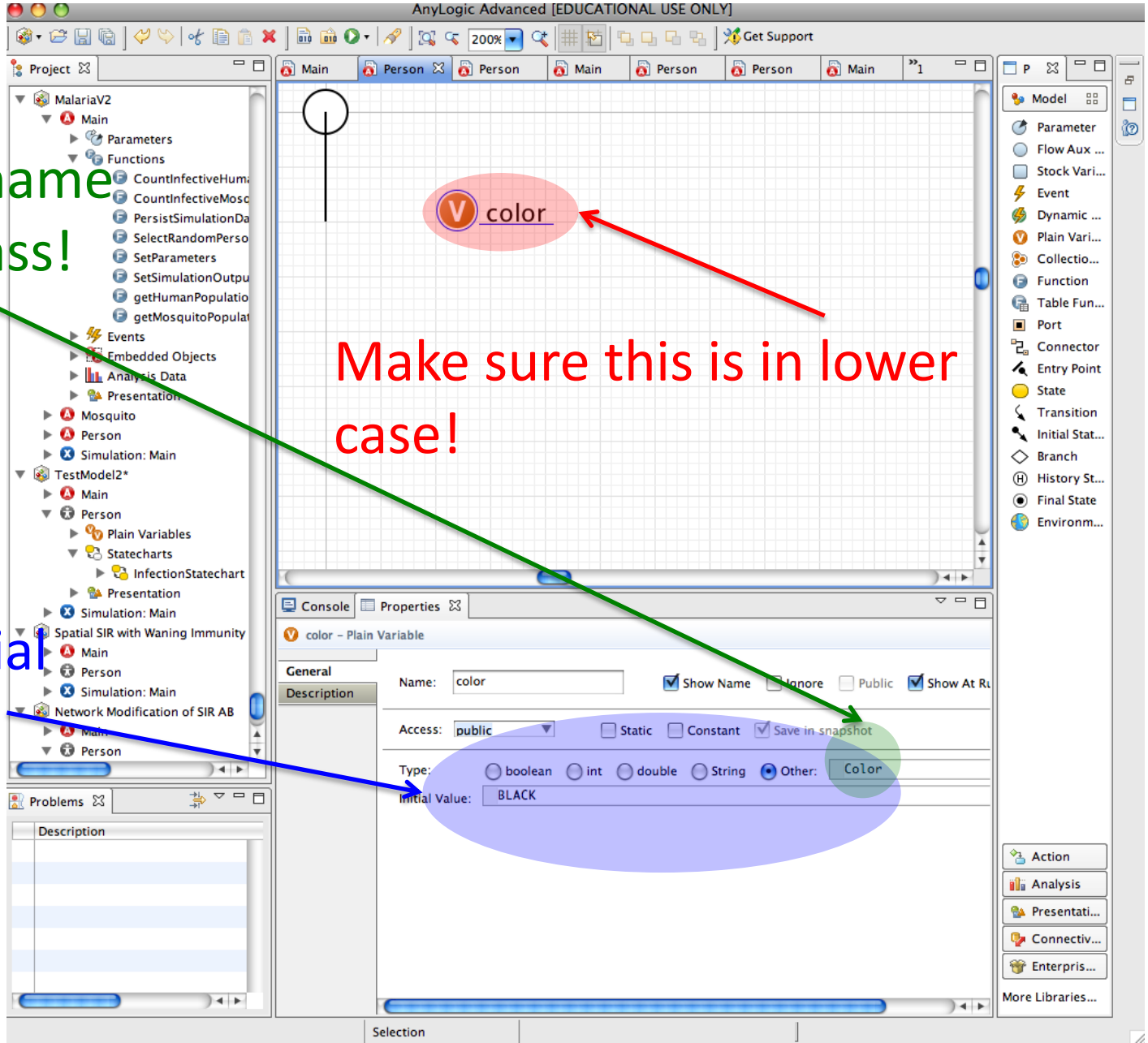
Try Running the Model!



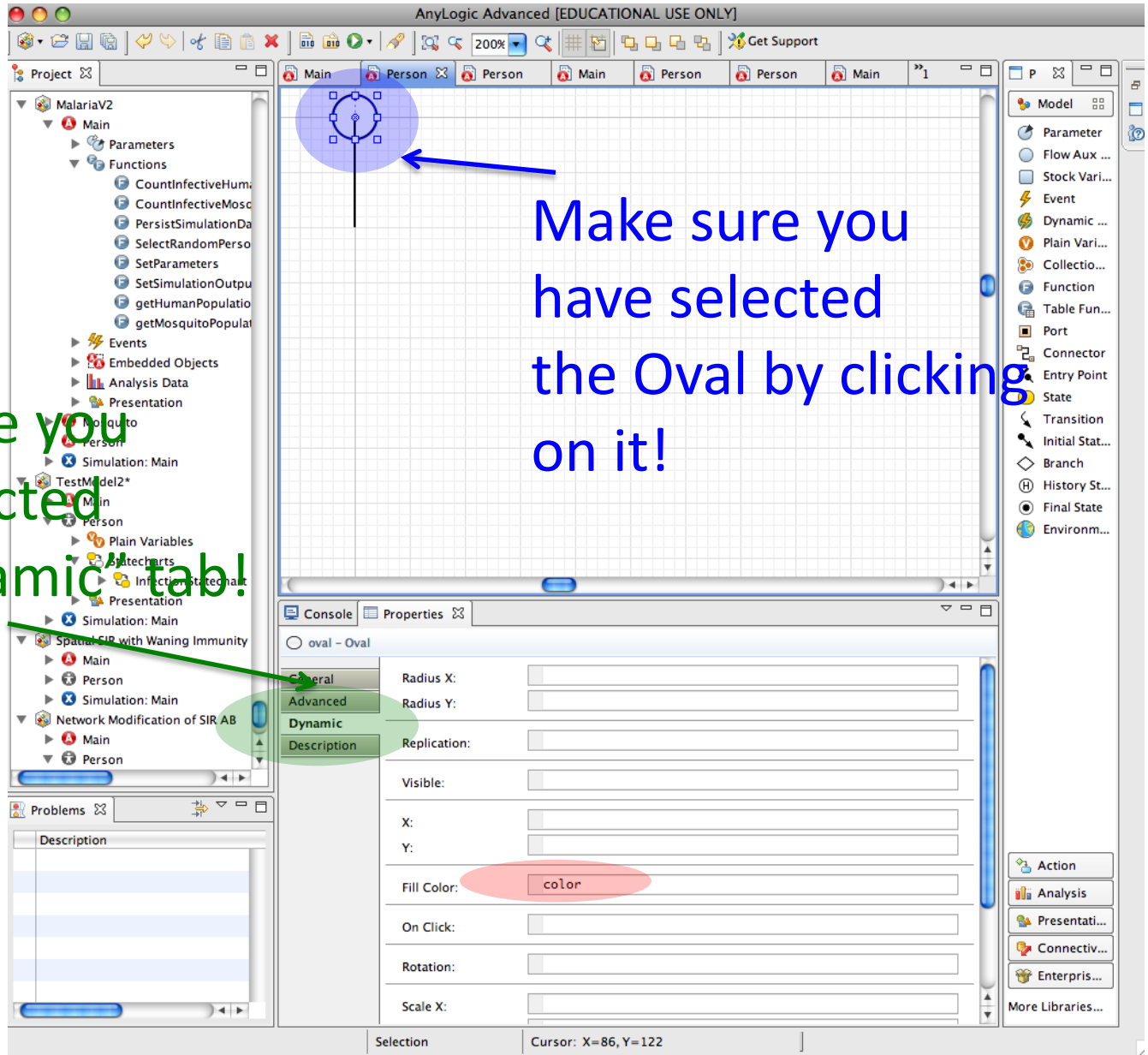
Adding “Color” Variable

This is the name of a Java class!

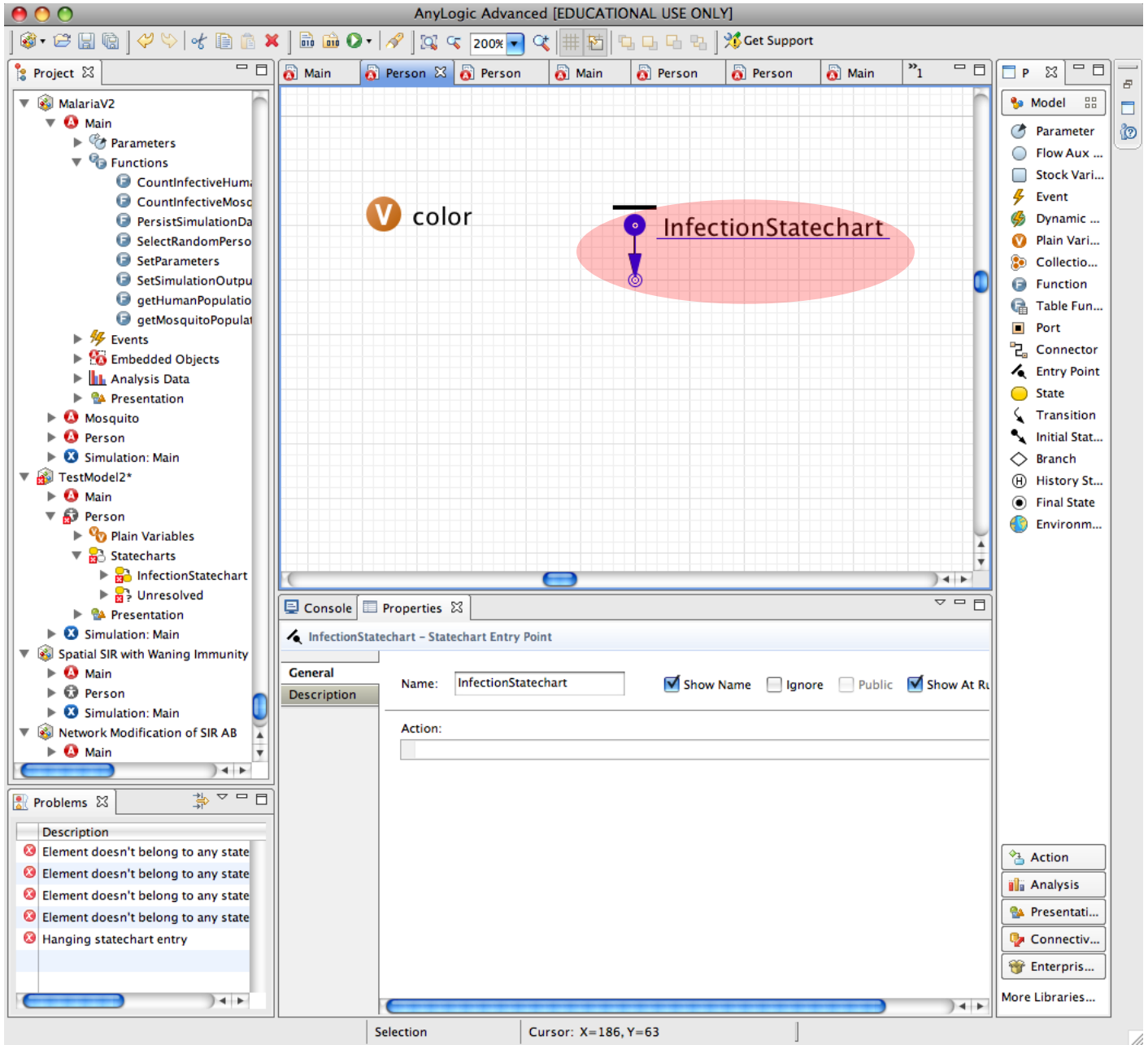
Fill in the type and Initial Value (watch for correct case!!)



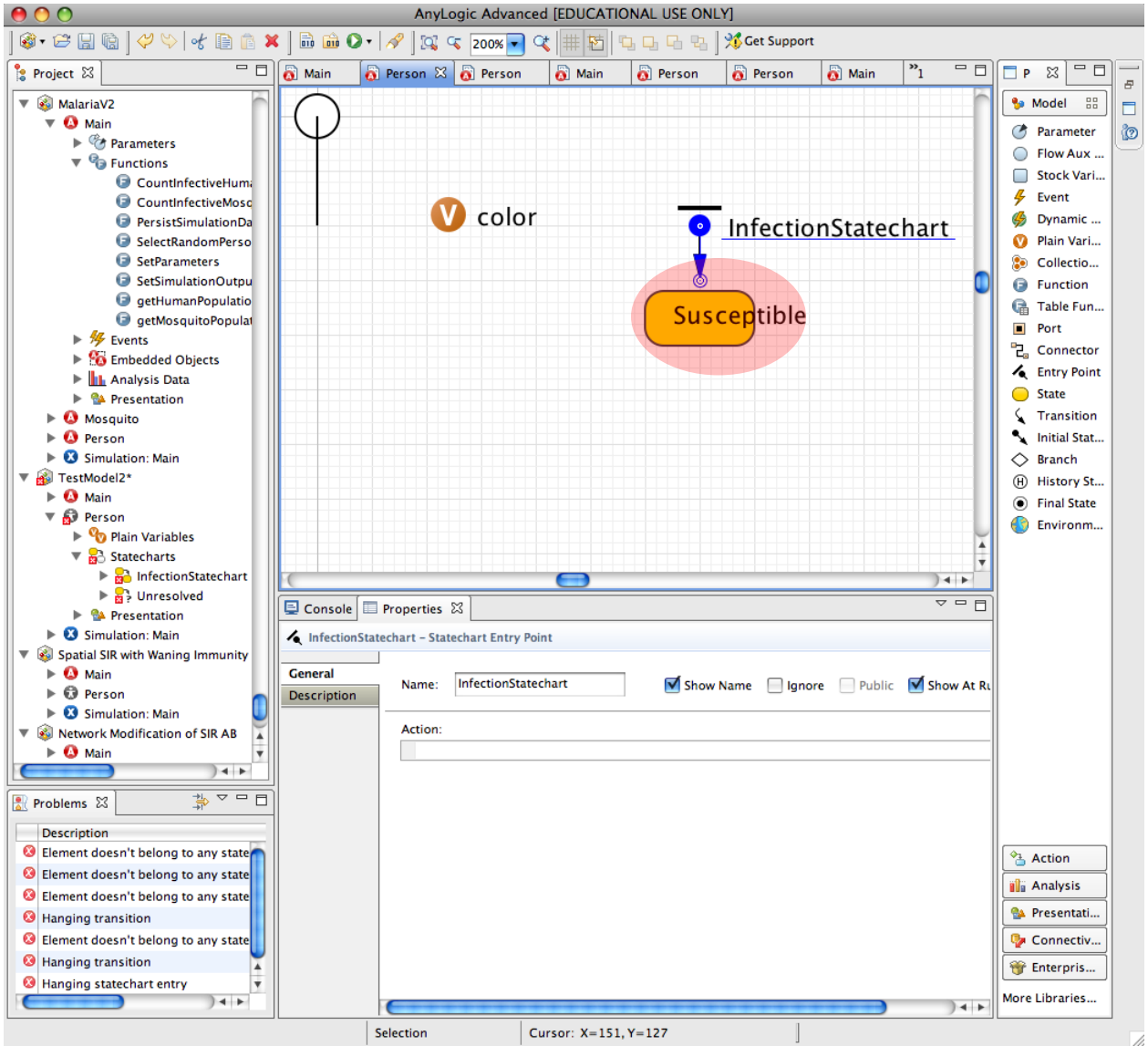
Make Oval “Color” property Use Variable



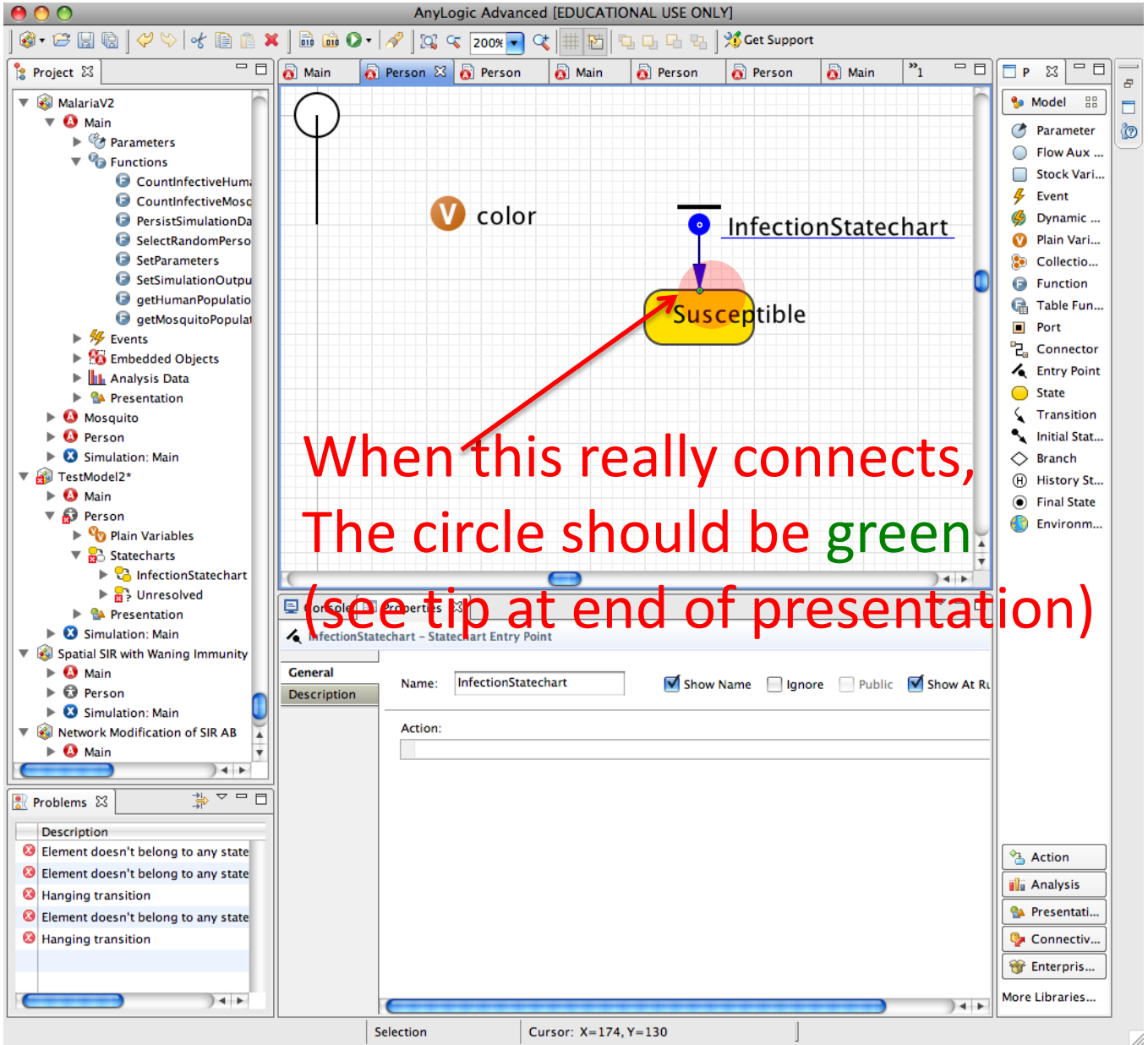
Add Entry Point of State chart



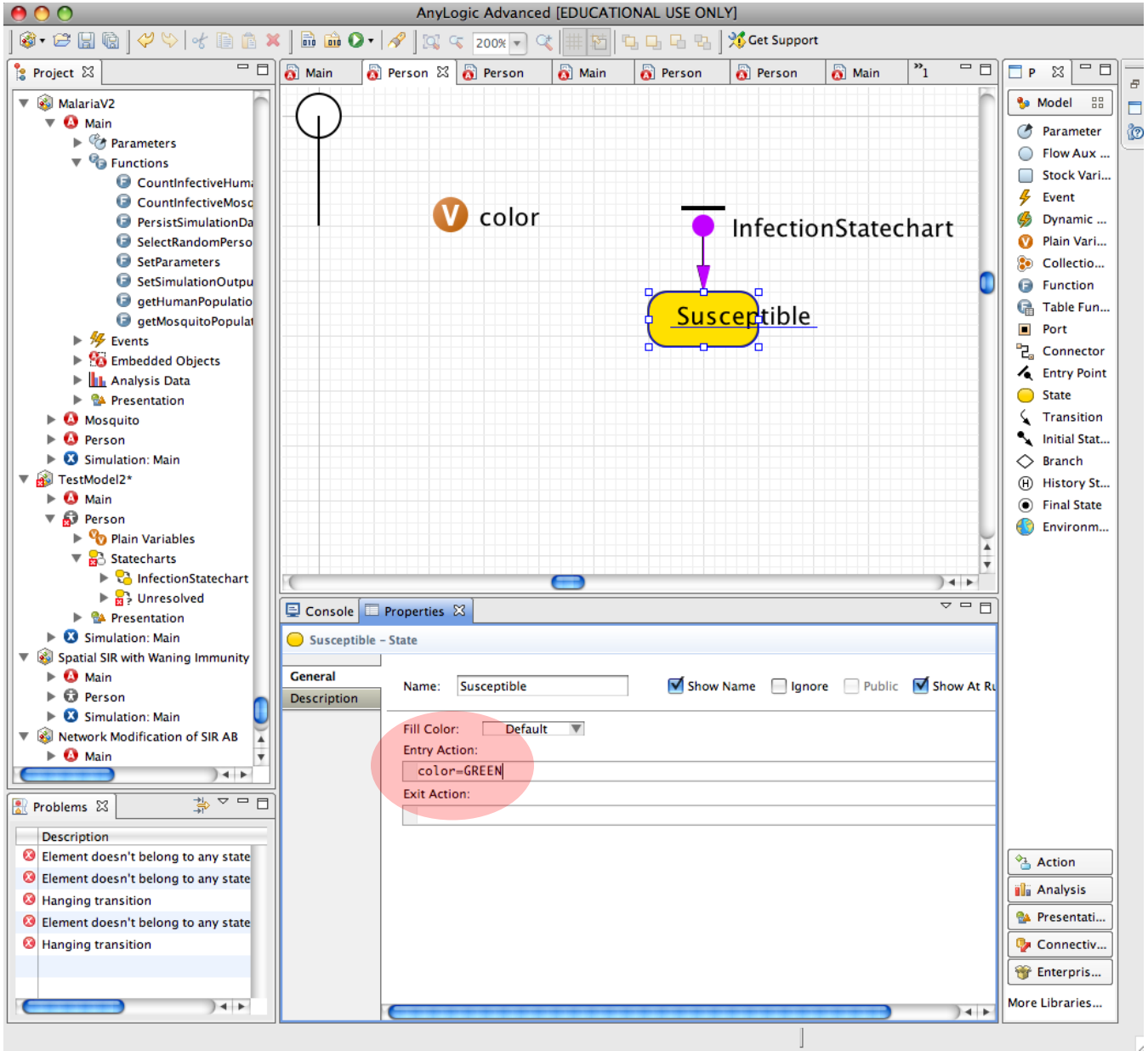
Add in “Susceptible” State



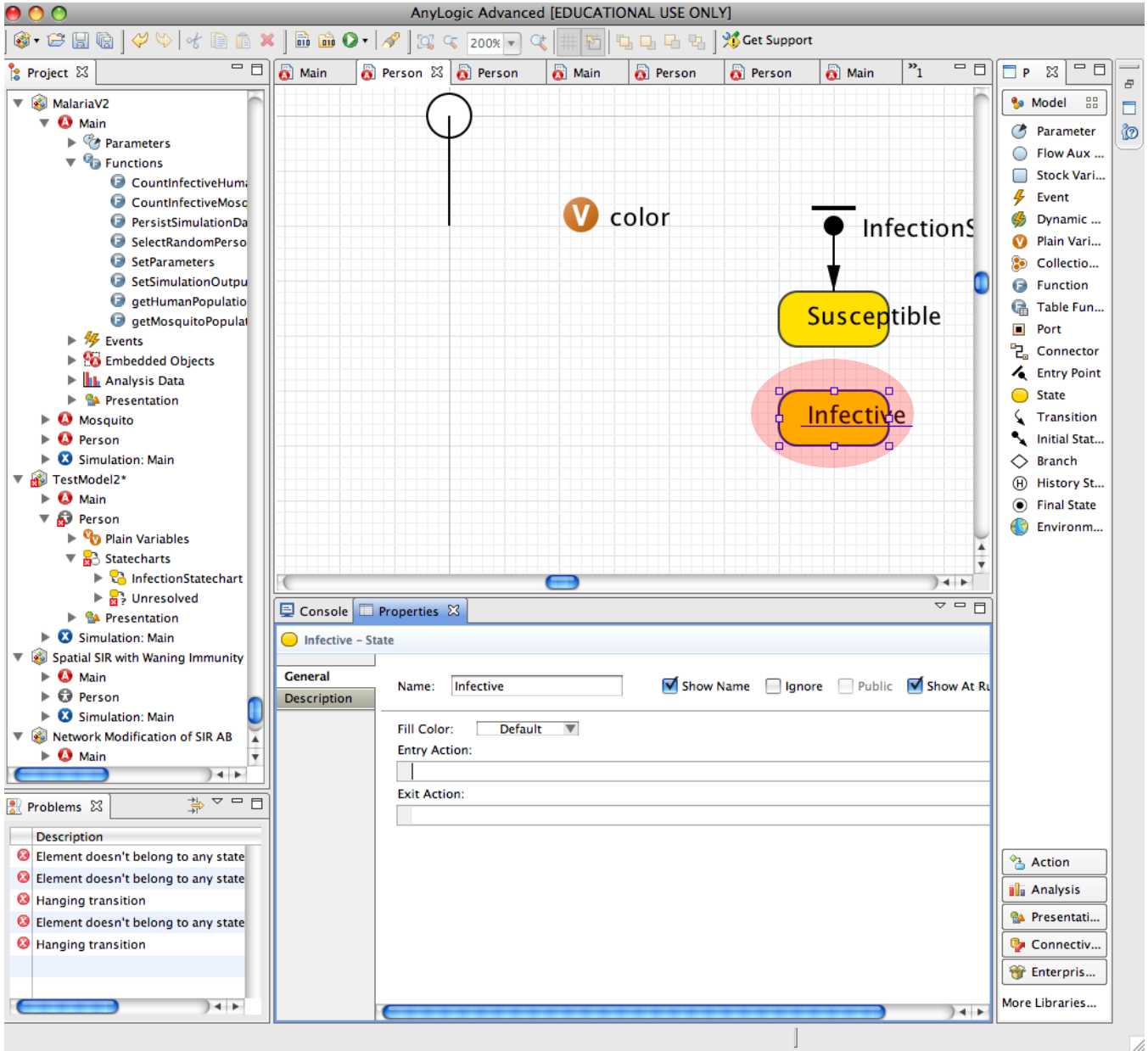
Connect with Entry Point



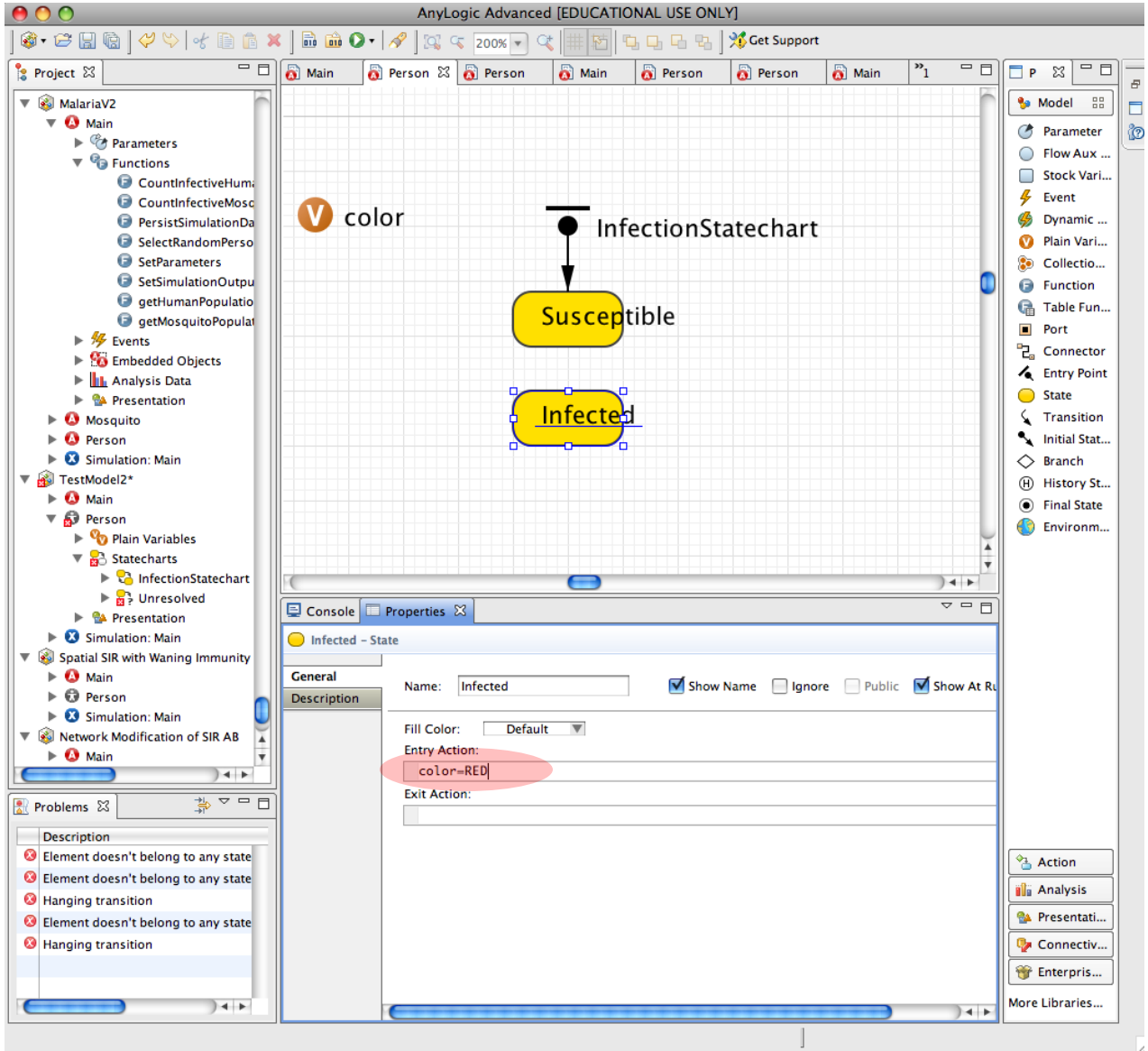
Fill In Code to Color Green when Enter State



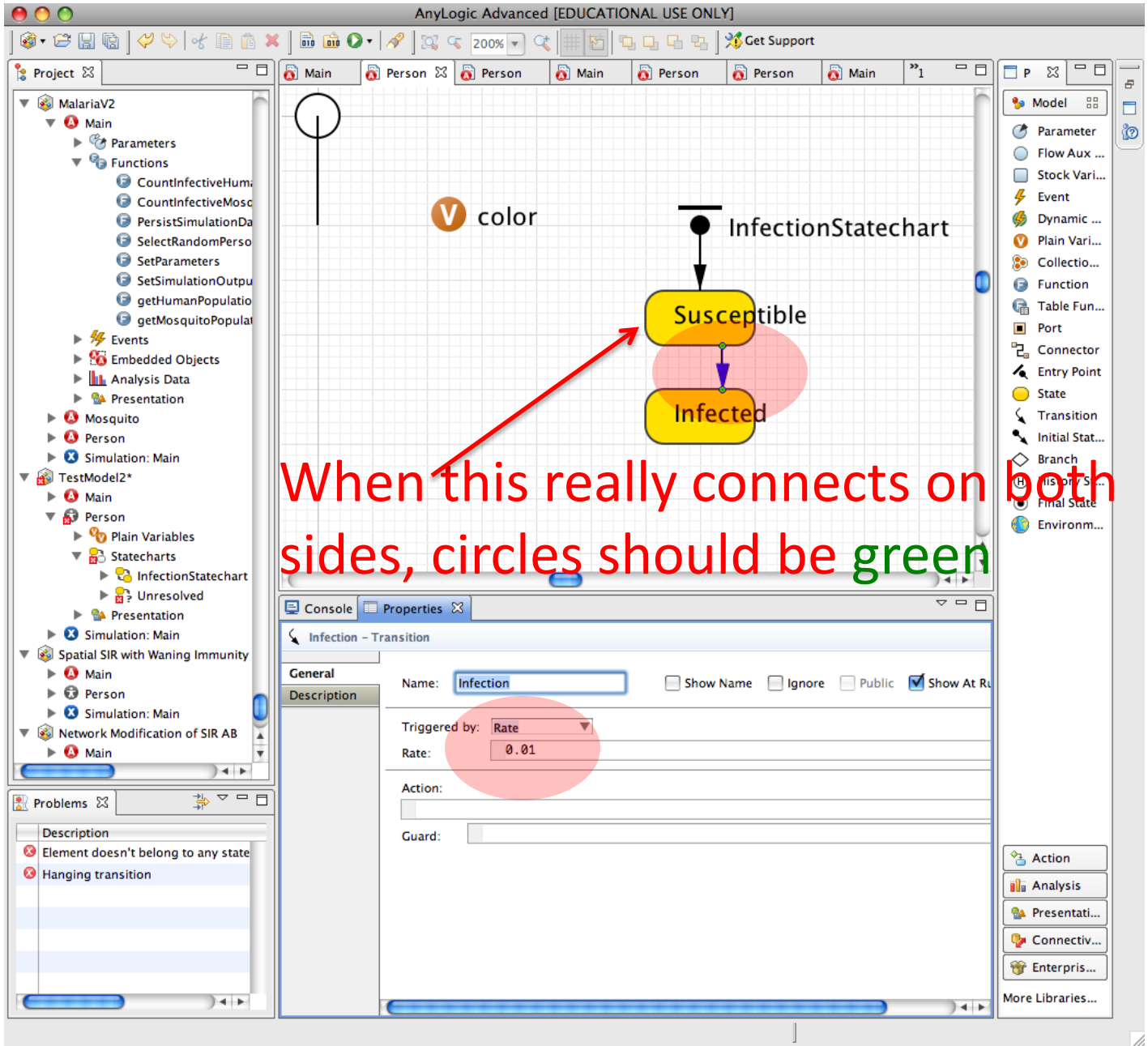
Adding in “Infective” State



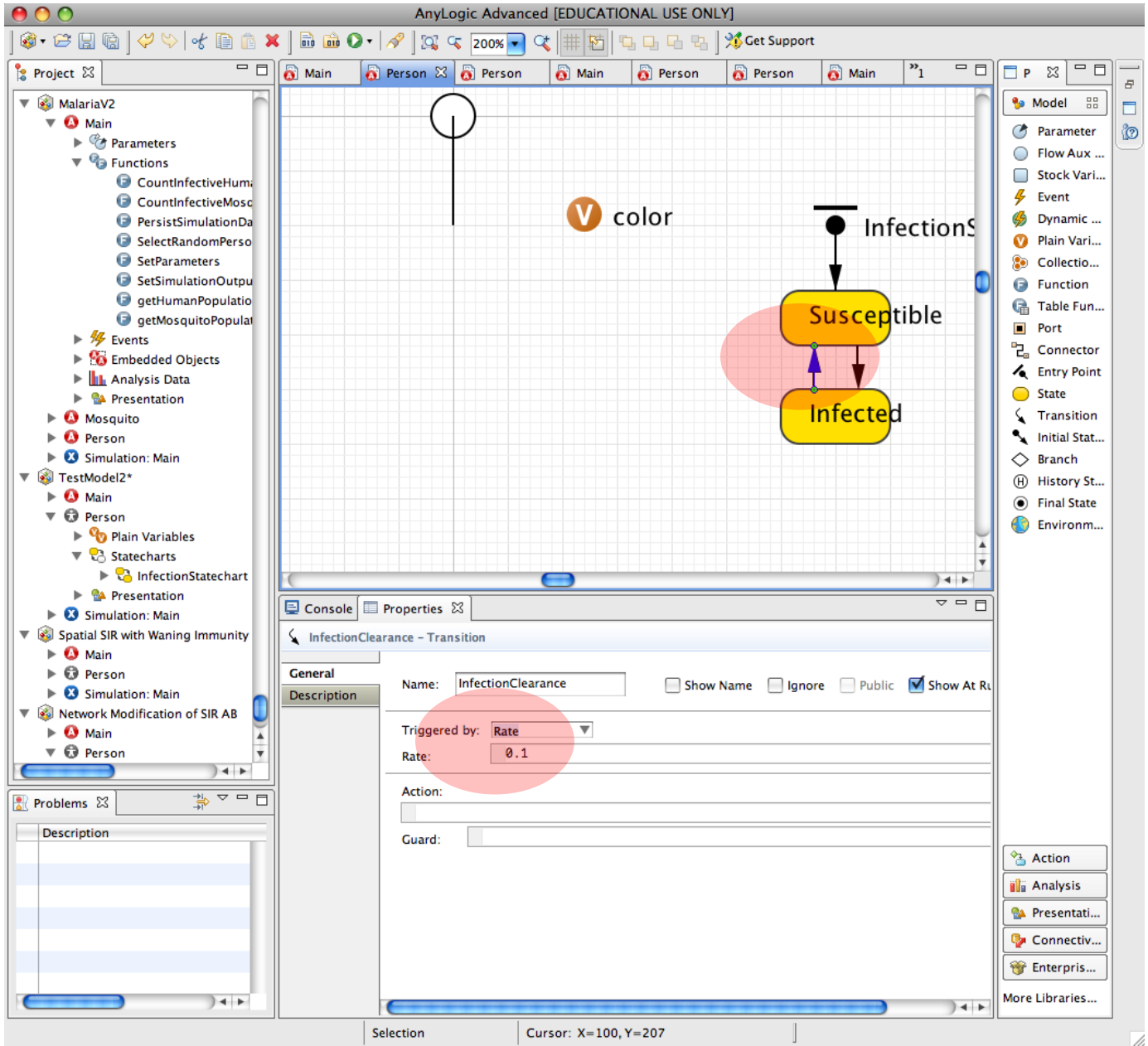
Set to Color Red when Enter State



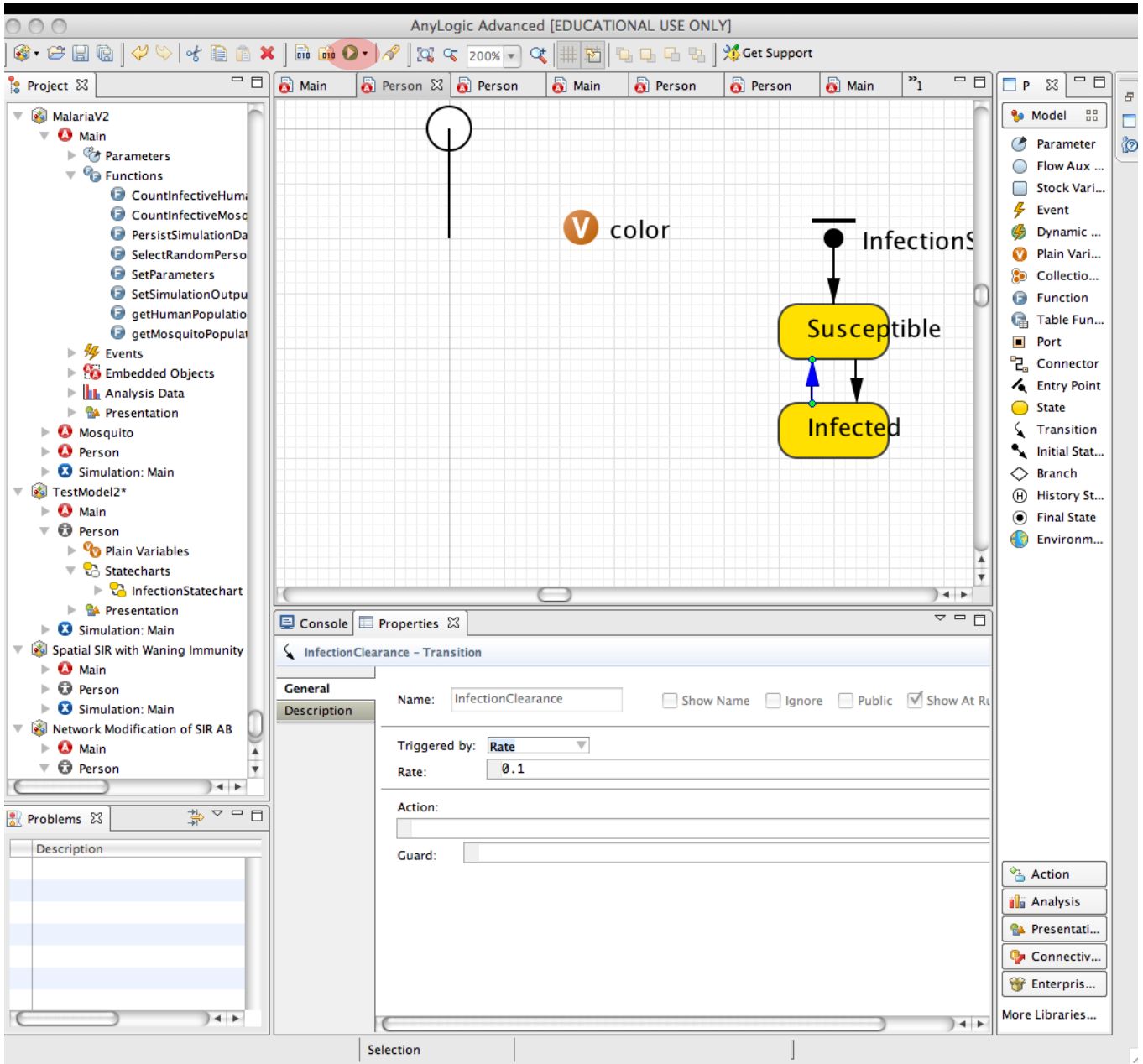
Adding Transition



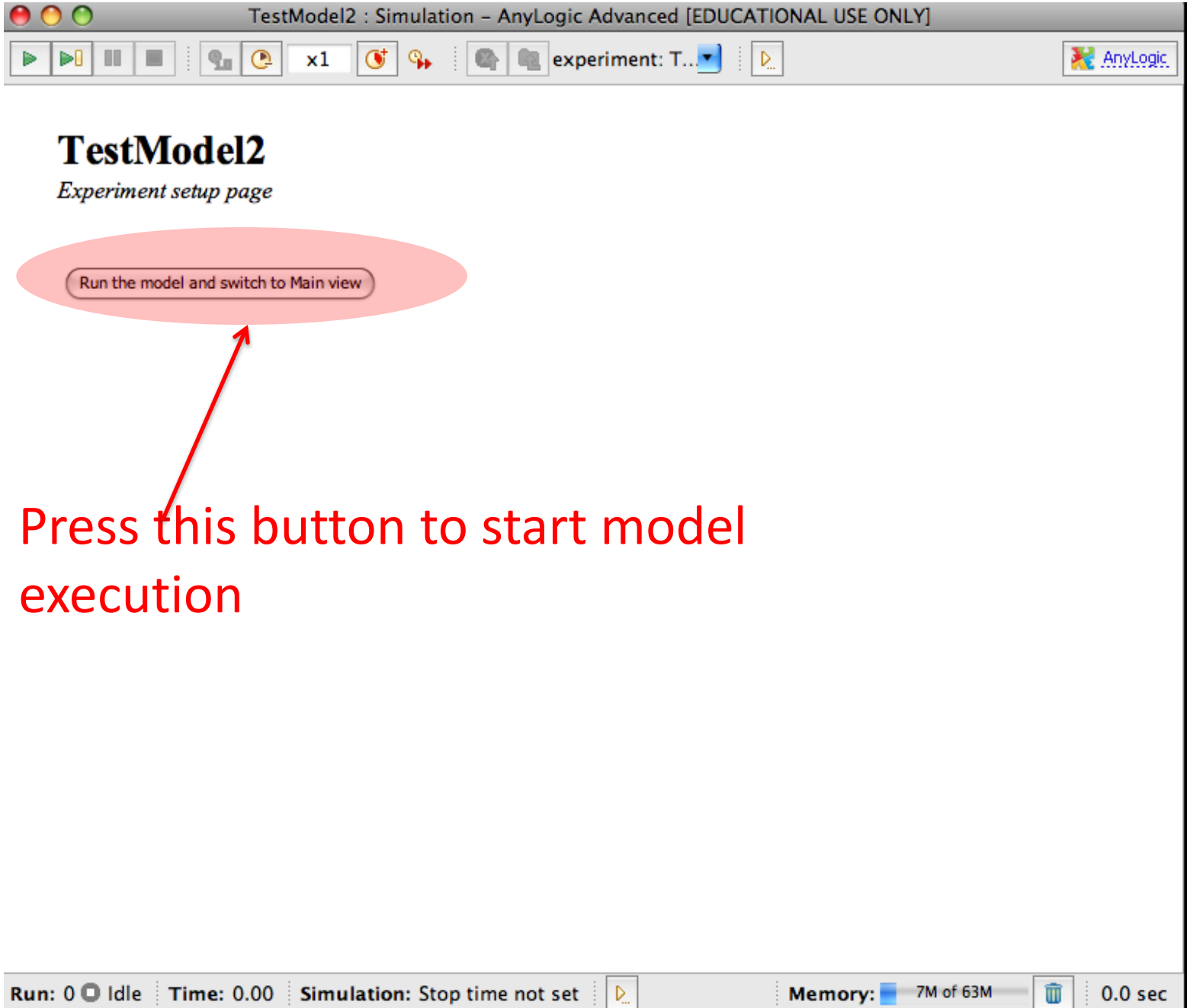
Adding Infection Clearance Transition



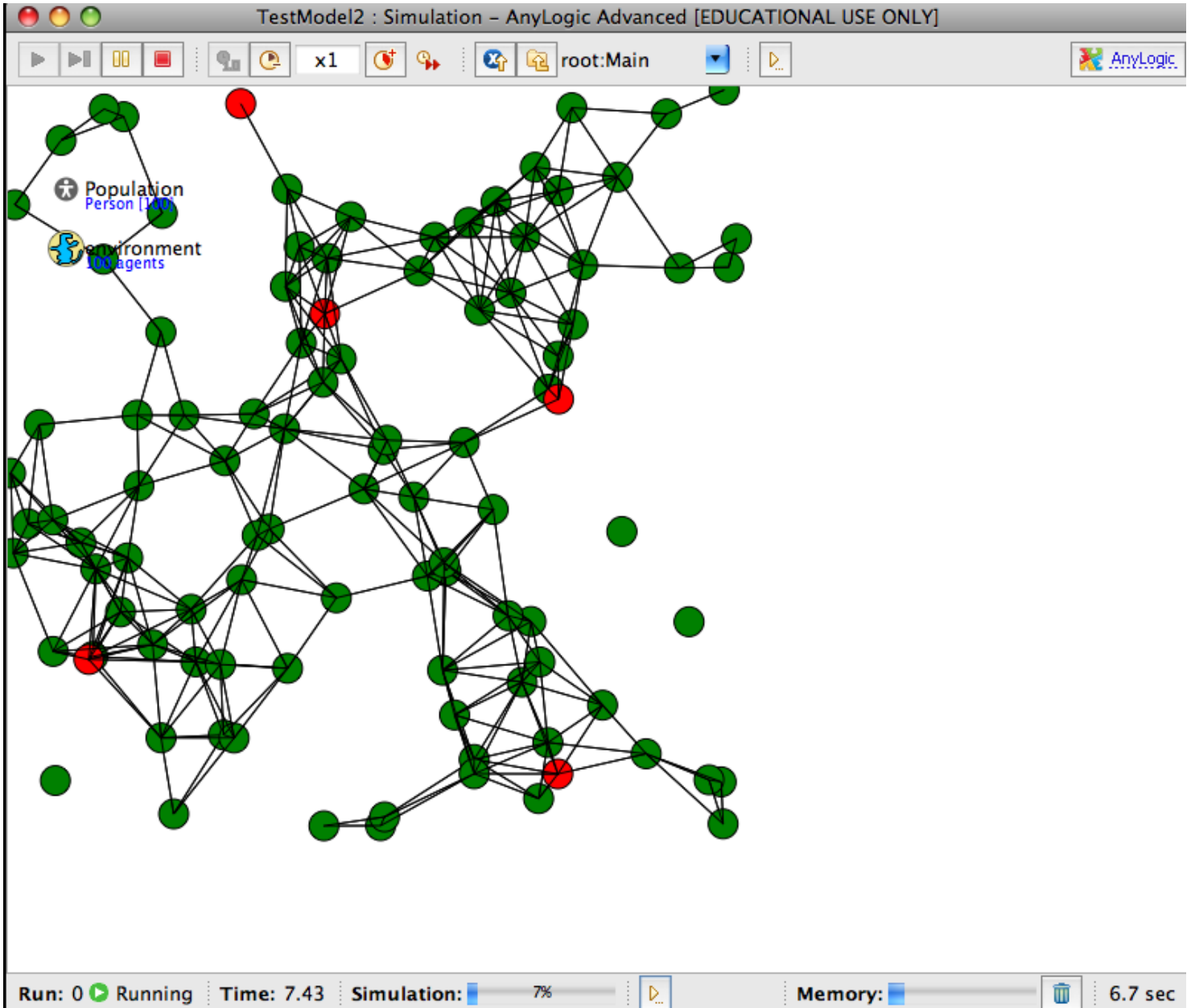
Run the Model!



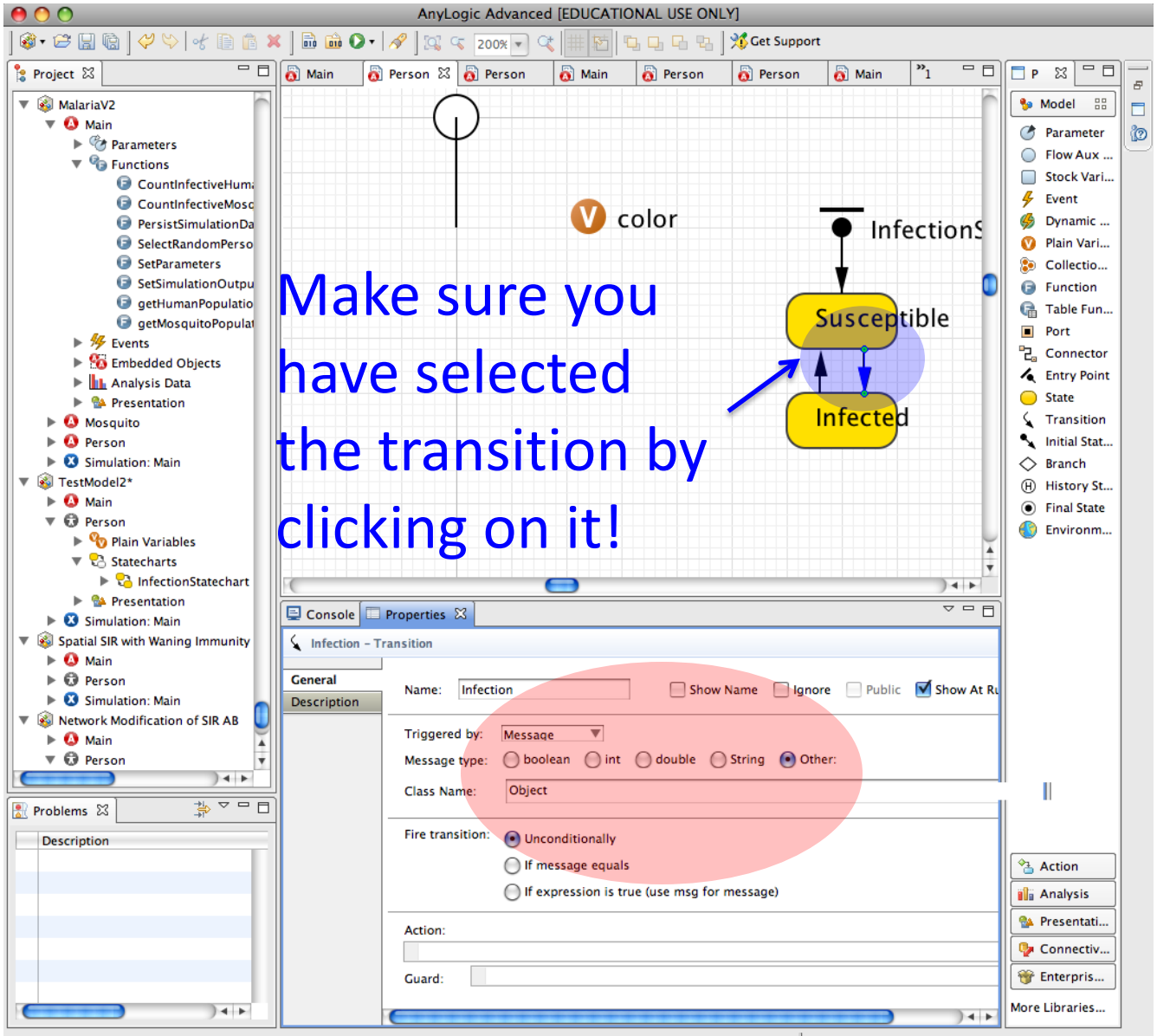
Completing Set-Up



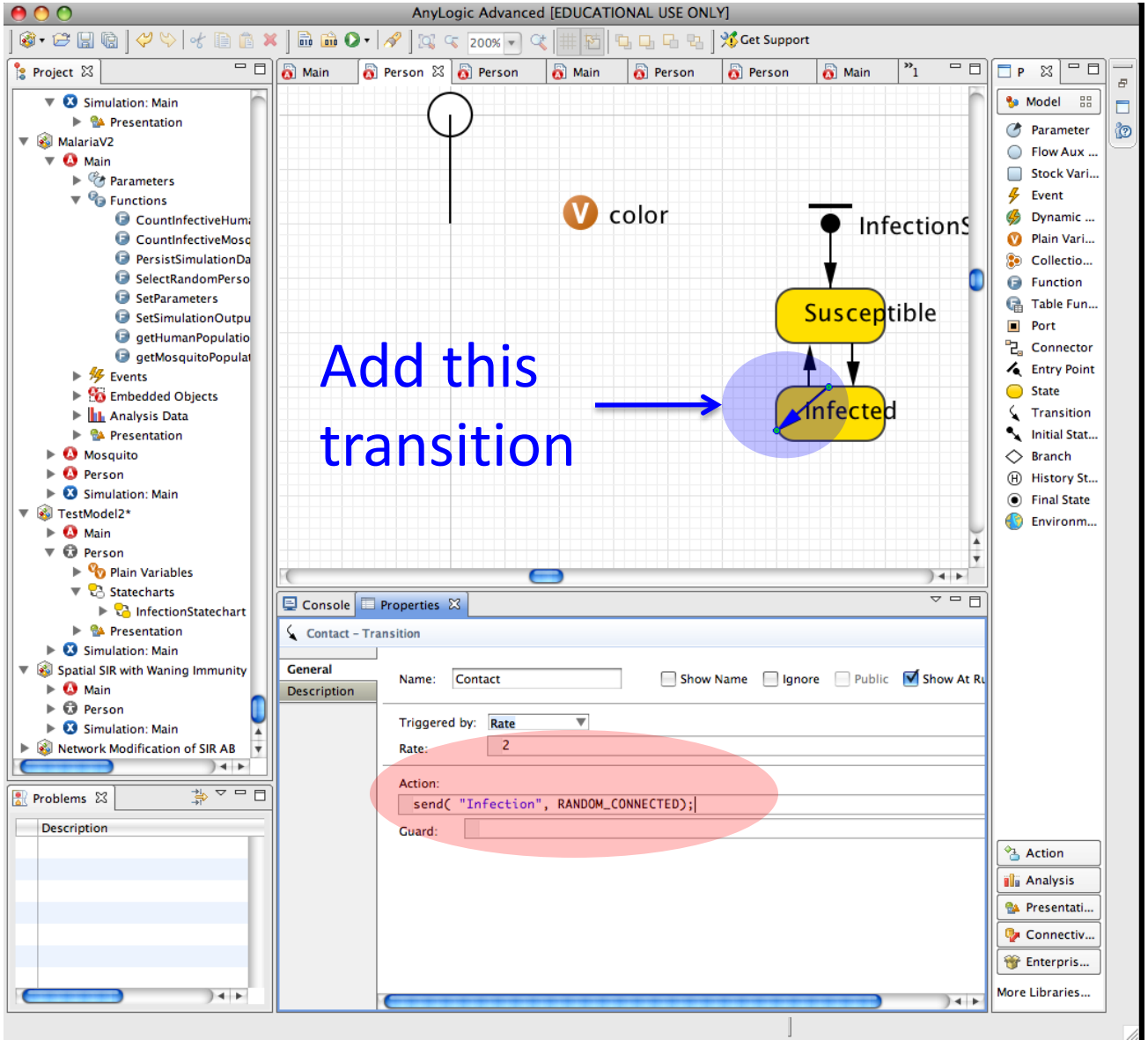
Model Presentation



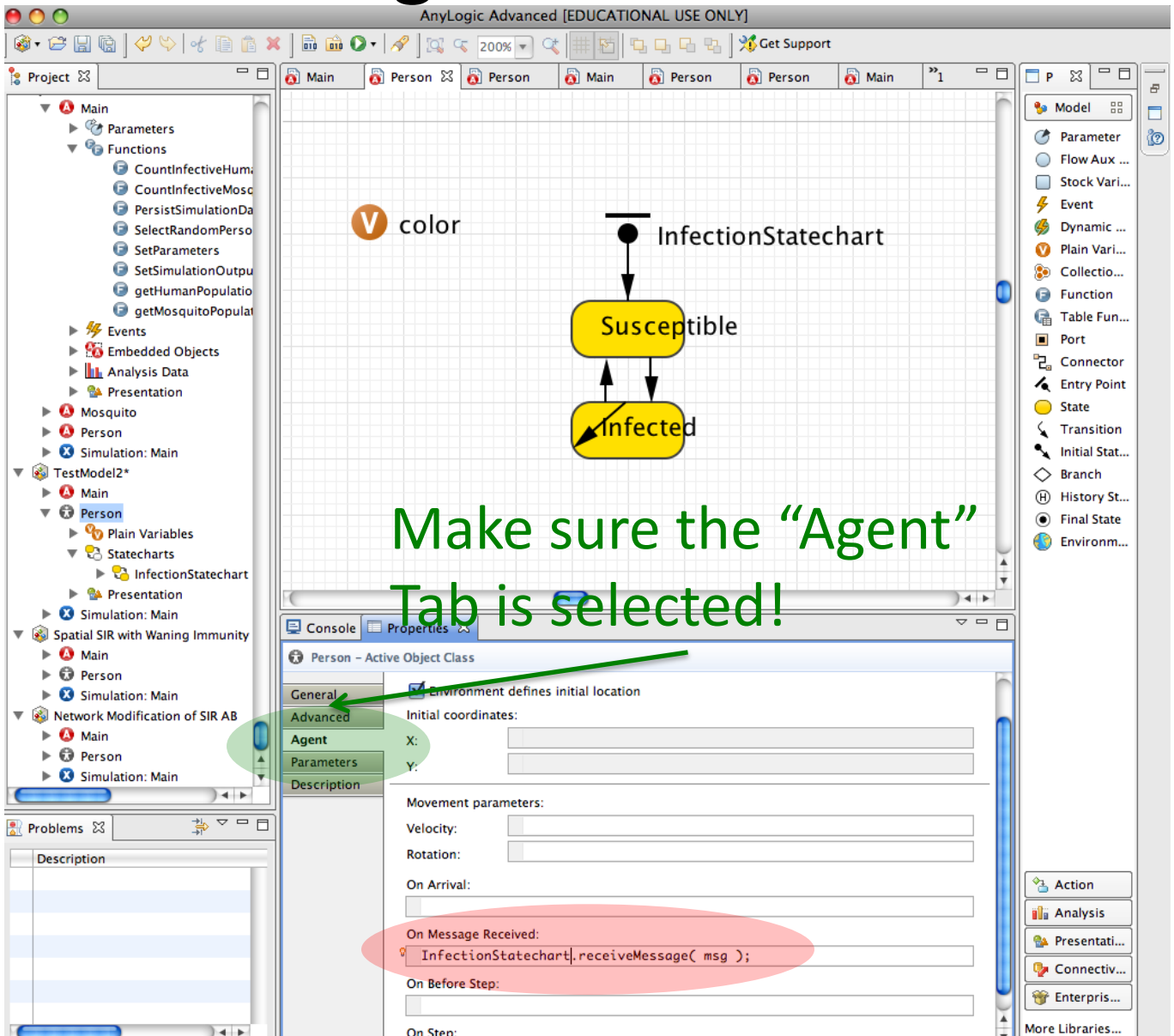
Making Infection Depend on a Message



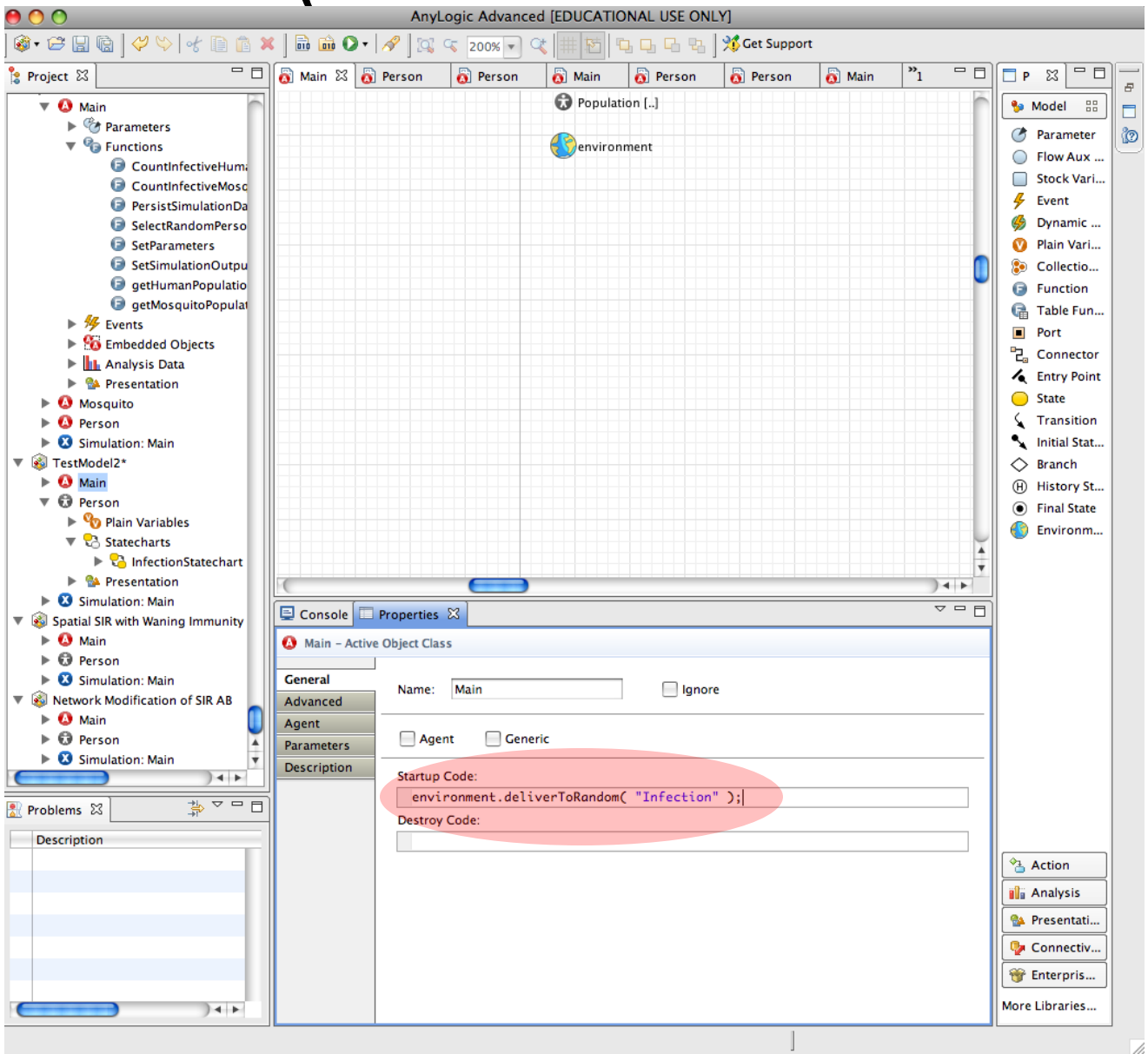
Using a “Contact” Event to Spread Infection



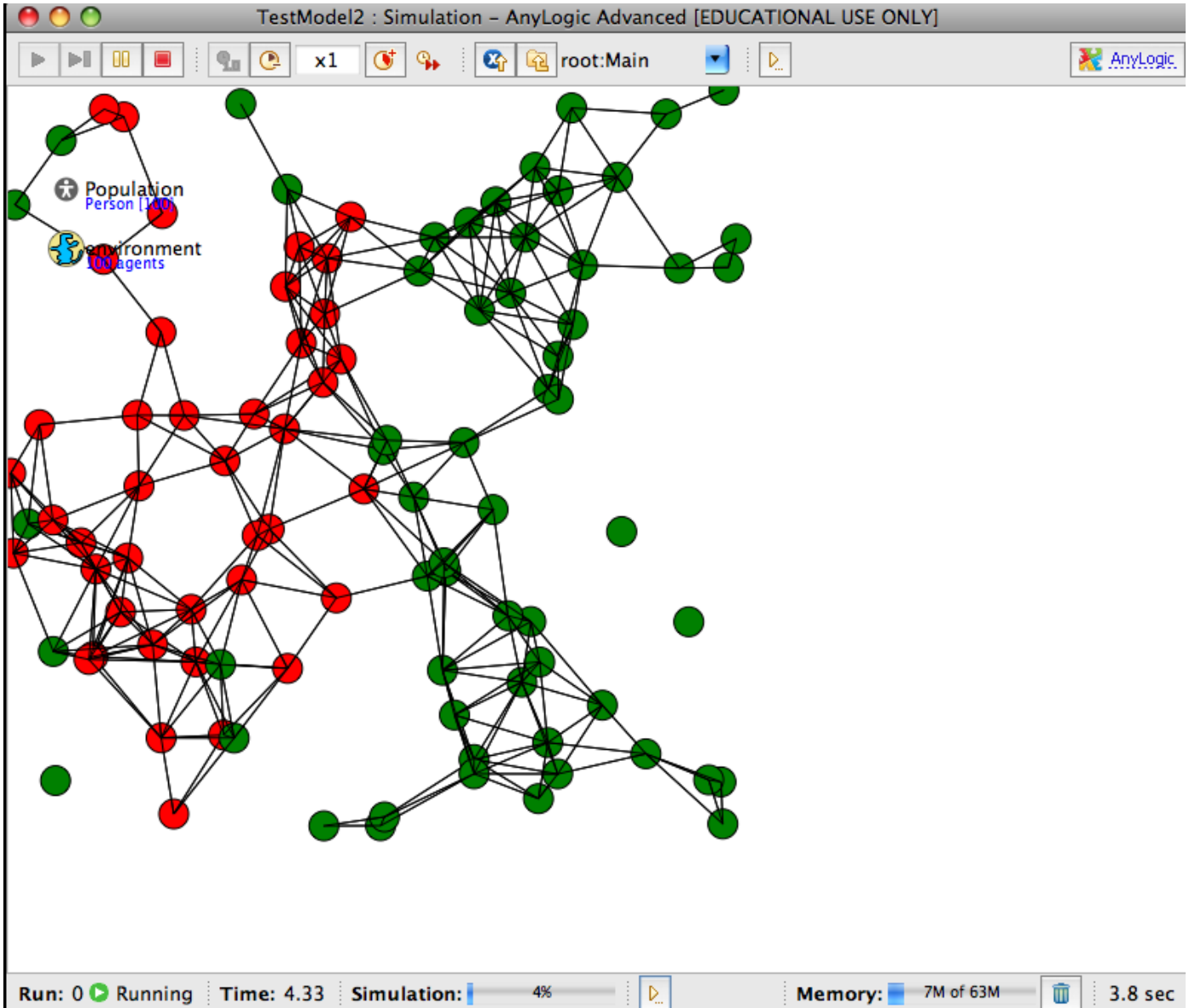
Setting “Person” so forwards Infection Message to Statechart



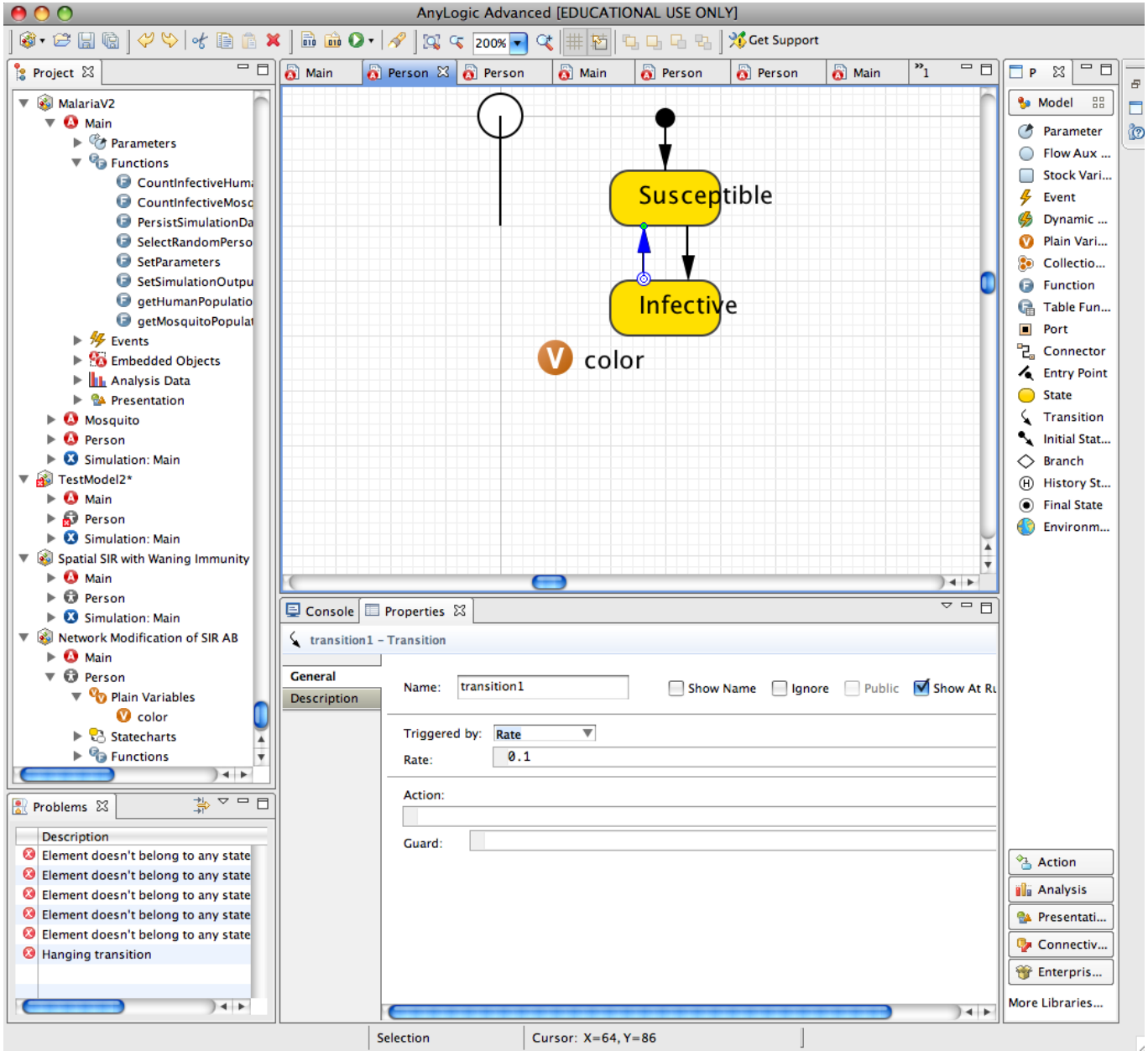
Setting Startup Code So Initially Infects a Random Person (so start with 1 infective)



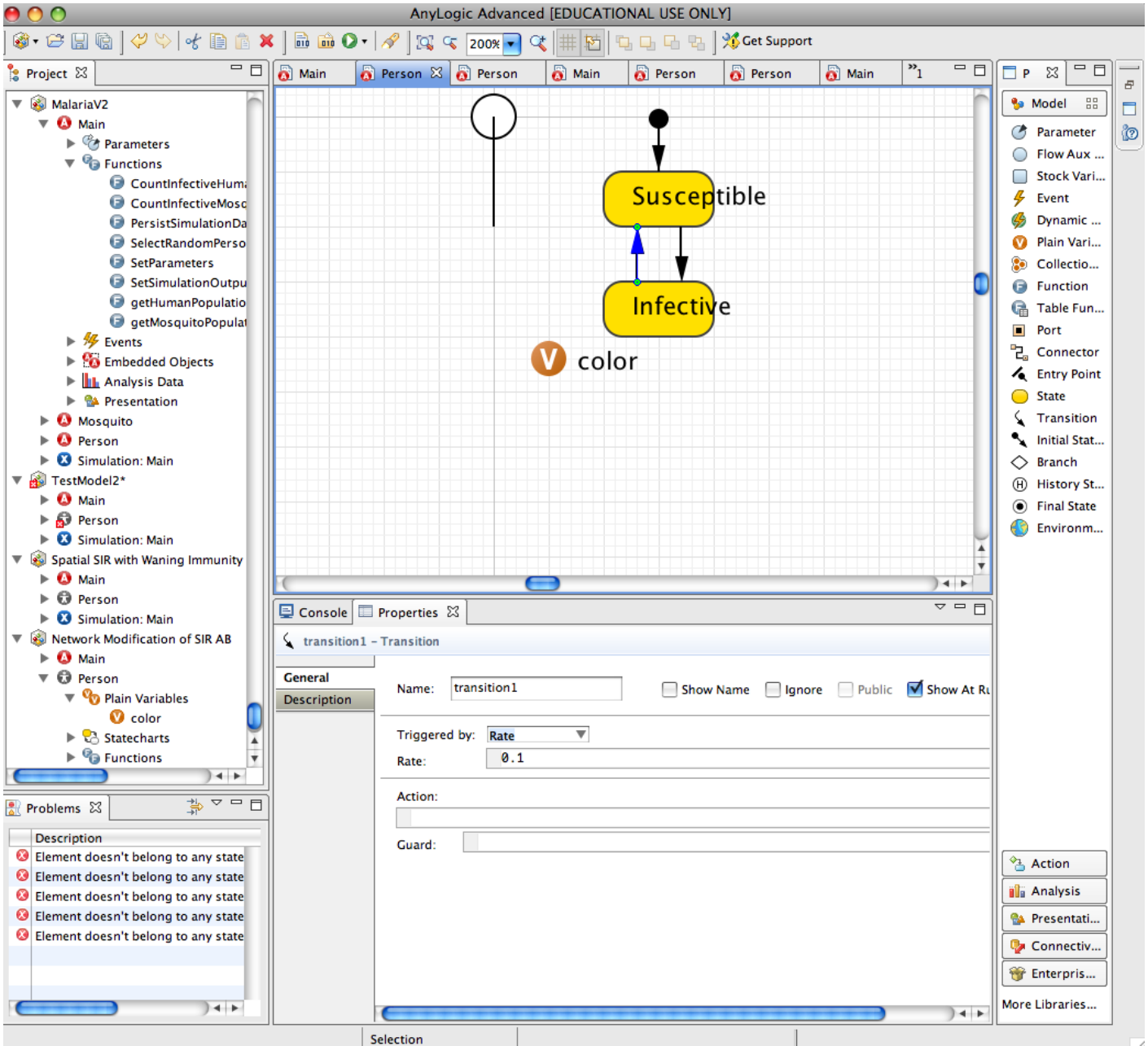
Infection Percolation over the Network



Tip: Beware Loose Connections



Corrected


AnyLogic Advanced [EDUCATIONAL USE ONLY]
Get Support
Project
MalariaV2
Main
Parameters
Functions
Events
Embedded Objects
Analysis Data
Presentation
Mosquito
Person
Simulation: Main
TestModel2*
Spatial SIR with Waning Immunity
Network Modification of SIR AB
Plain Variables
color
Statecharts
Functions
Problems
Description
Element doesn't belong to any state
Main
Person
Susceptible
Infective
color
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