Introduction to the Anylogic Interface & Supporting Concepts

Announcements

- Lecture recording links posted
- Tutorial time: Extended class hours on Tuesday or Thursday
  - Choice will depend on other classes following ours
  - Thursday is likely

The AnyLogic User Interface

Common Configuration

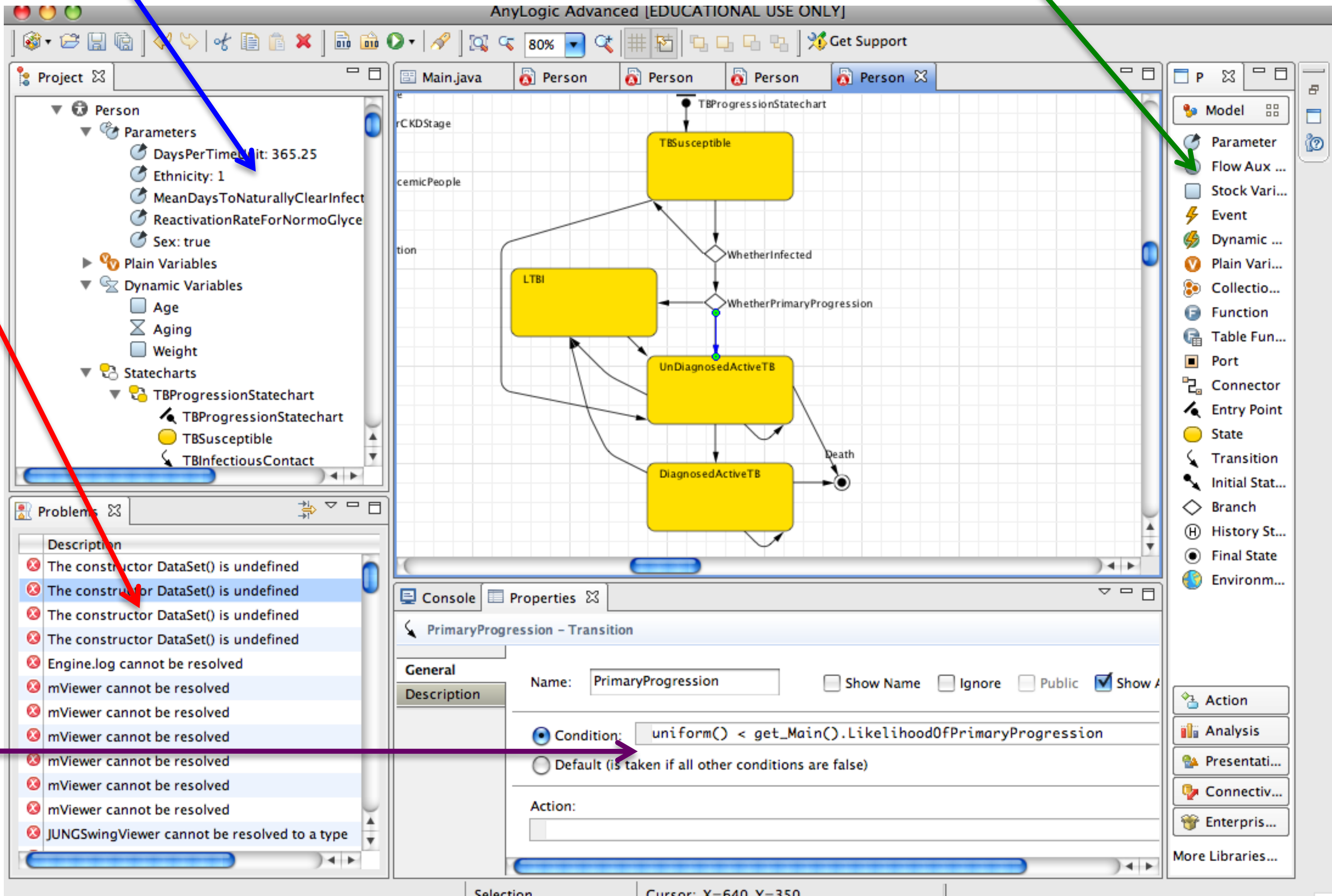
The Project View (overview of projects & components)

Note: Double-Clicking on a Tab opens view as Full-Screen

Palette for adding items to canvas

Problem area (indicates problem Building/ running Model)

Properties area (shows info on selected element in project or palette window)



Description

- The constructor DataSet() is undefined
- The constructor DataSet() is undefined
- The constructor DataSet() is undefined
- The constructor DataSet() is undefined
- Engine.log cannot be resolved
- mViewer cannot be resolved
- mViewer cannot be resolved
- mViewer cannot be resolved
- mViewer cannot be resolved
- mViewer cannot be resolved
- mViewer cannot be resolved
- JUNGSwingViewer cannot be resolved to a type

PrimaryProgression - Transition

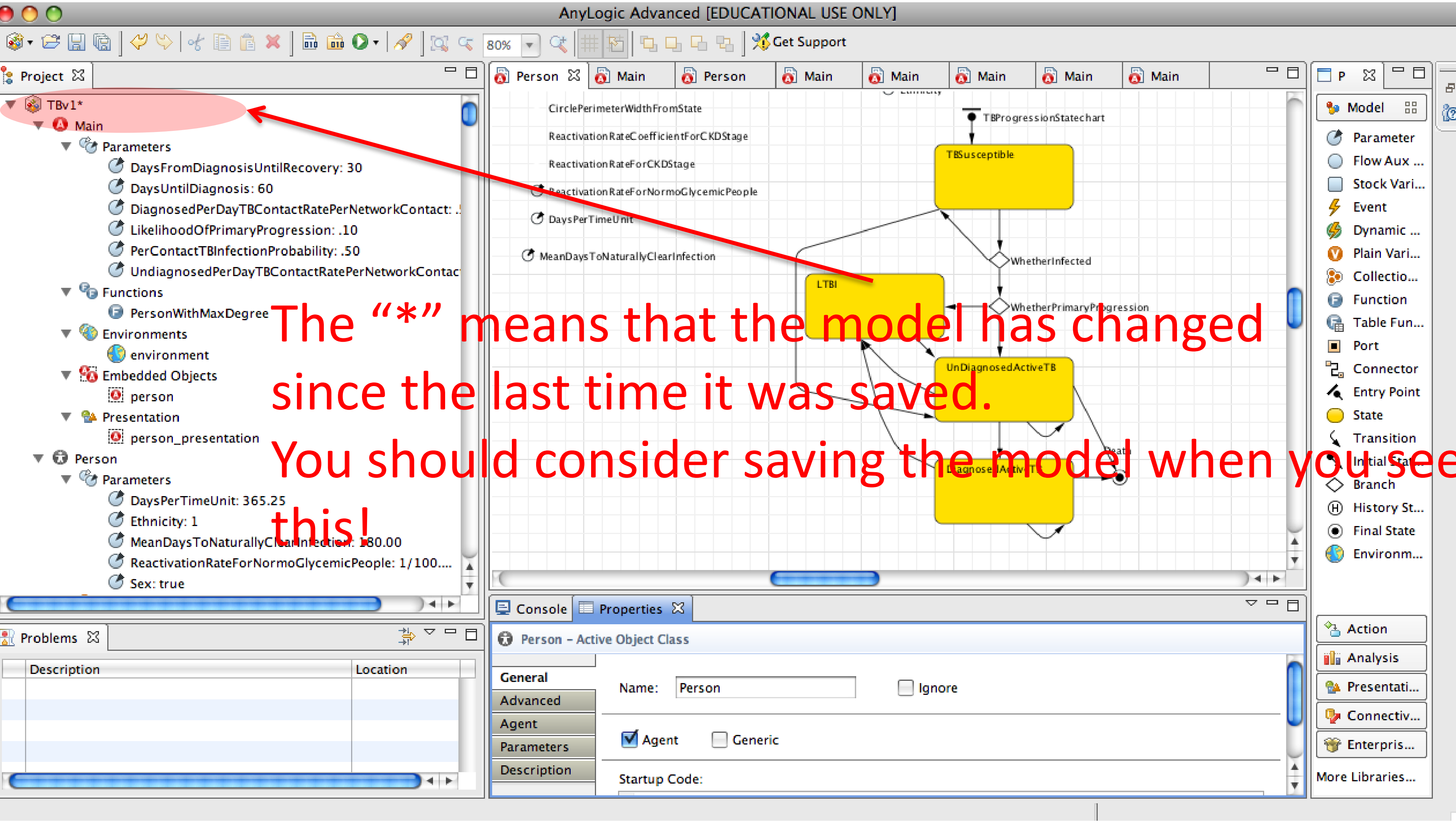
Name: PrimaryProgression □ Show Name □ Ignore □ Public ☑ Show

● Condition: `uniform() < get_Main().LikelihoodOfPrimaryProgression`

○ Default (is taken if all other conditions are false)

Action:

The “Project View” – Hierarchically Shows the Project Components



- TBv1*
  - Main
    - Parameters
      - DaysFromDiagnosisUntilRecovery: 30
      - DaysUntilDiagnosis: 60
      - DiagnosedPerDayTBContactRatePerNetworkContact: .
      - LikelihoodOfPrimaryProgression: .10
      - PerContactTBInfectionProbability: .50
      - UndiagnosedPerDayTBContactRatePerNetworkContac
    - Functions
      - PersonWithMaxDegree
    - Environments
      - environment
    - Embedded Objects
      - person
    - Presentation
      - person_presentation
  - Person
    - Parameters
      - DaysPerTimeUnit: 365.25
      - Ethnicity: 1
      - MeanDaysToNaturallyClearInfection: 180.00
      - ReactivationRateForNormoGlycemicPeople: 1/100....
      - Sex: true

Person – Active Object Class

Name: Person □ Ignore

☑ Agent □ Generic

Startup Code:

The “*” means that the model has changed since the last time it was saved.

You should consider saving the model when you see this!

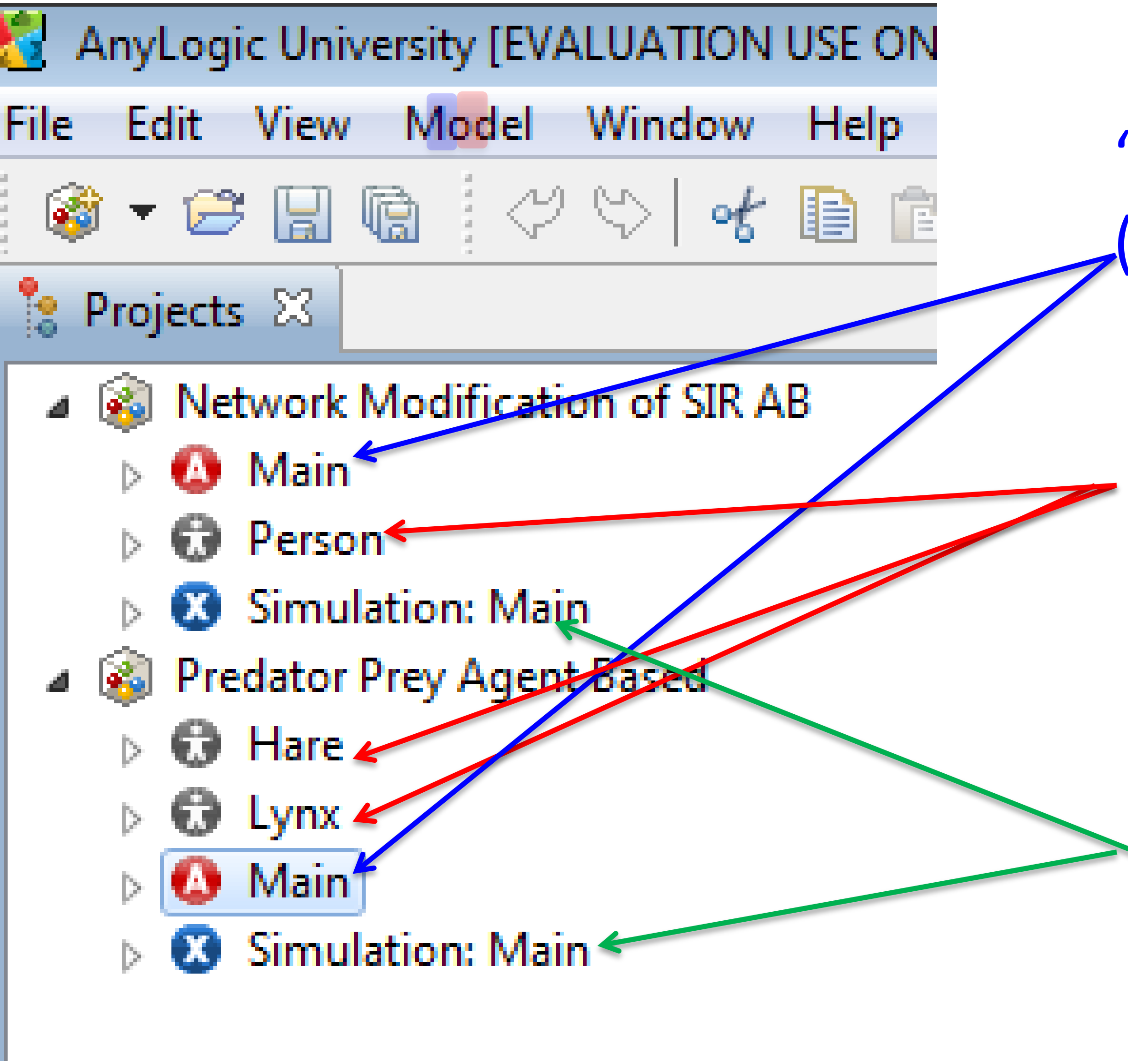
Hands on Model Use Ahead

Load Sample Model:

Predator-Prey Agent Based

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

Example “Classes”



AnyLogic University [EVALUATION USE ON

- Network Modification of SIR AB
  - Main
  - Person
  - Simulation: Main
- Predator Prey Agent Based
  - Hare
  - Lynx
  - Main
  - Simulation: Main

“Main” classes

(Define the “*Stage*”)

“Agent” classes

(Define the *actors*)

“Experiment” classes

(Define the *scenarios*)

Key Customized "Classes"

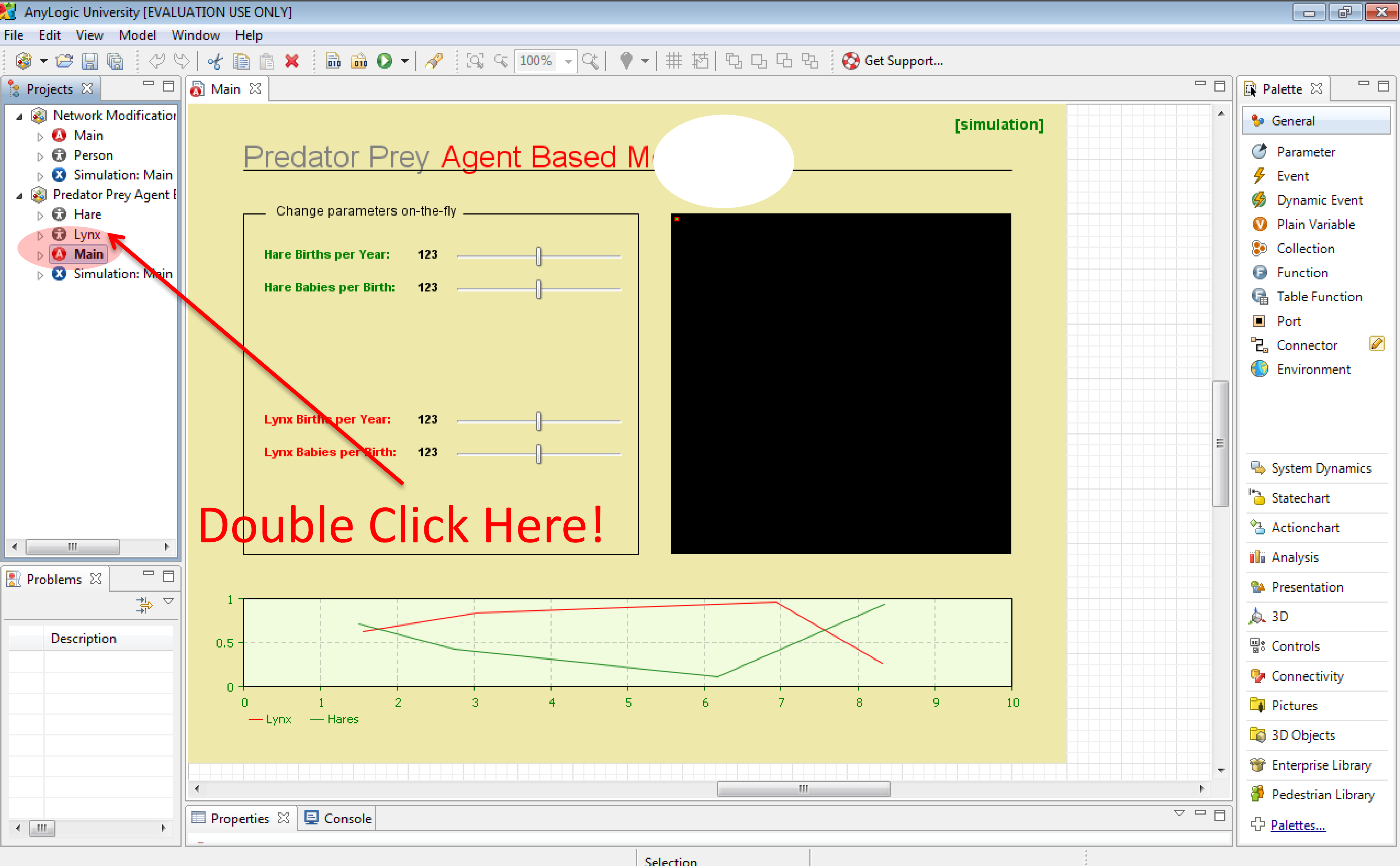
- The structure of the model is composed of certain key user-customized "classes"
- "Main" class
  - Normally just one instance
  - This will generally contain collections of the other classes
- "Agent" classes
  - Your agent classes
  - There are typically many instances (objects) of these classes at runtime

Subclasses of "ActiveObject"

- "Experiment" classes

  These describe assumptions to use when running the model

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



Double Click Here!

“Main” Class

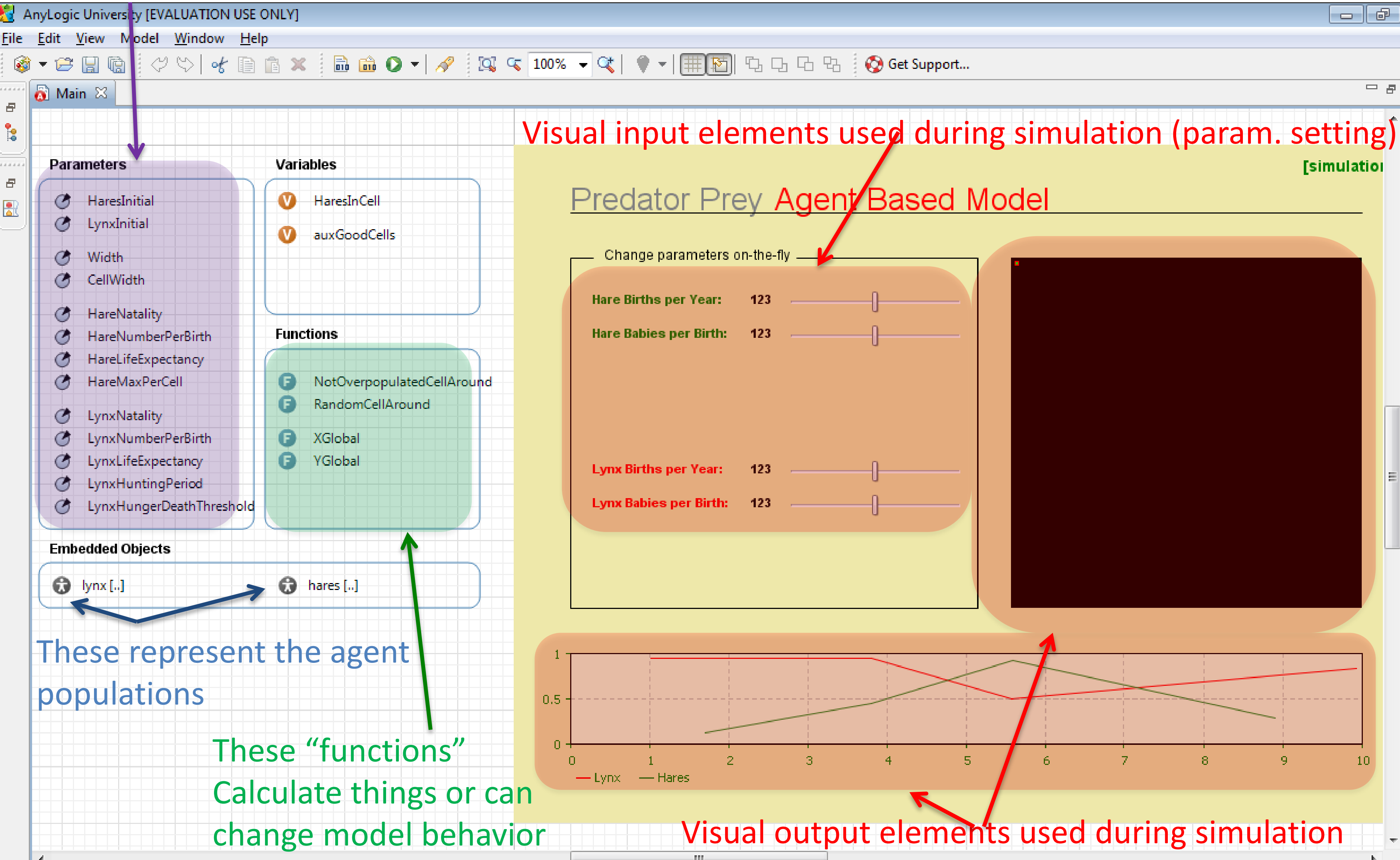
- Defines the environment where agents interact
- Defines interface & cross-model mechanisms
- The Main object normally contains one or more populations of “replicated” agents
  - Each population consists of agents of a certain class (or a subclass therefore), e.g.
    - “Hares”
    - “Lynxes”
  - The agent classes are defined separately from the Main class

Agent Populations in the Main Class

- Through the “Replication” property, the number of these agents can be set
- The “Environment” property can be used to associated the agents with some surrounding context (e.g. Network, embedding in some continuous space, with a neighborhood)
- Statistics can be computed on these agents
- Within the Main class, you can create representations of subpopulations by dragging from an Agent class into the Main class area

Elements of a “Main” Class

These “parameters” specify
static model-wide characteristics



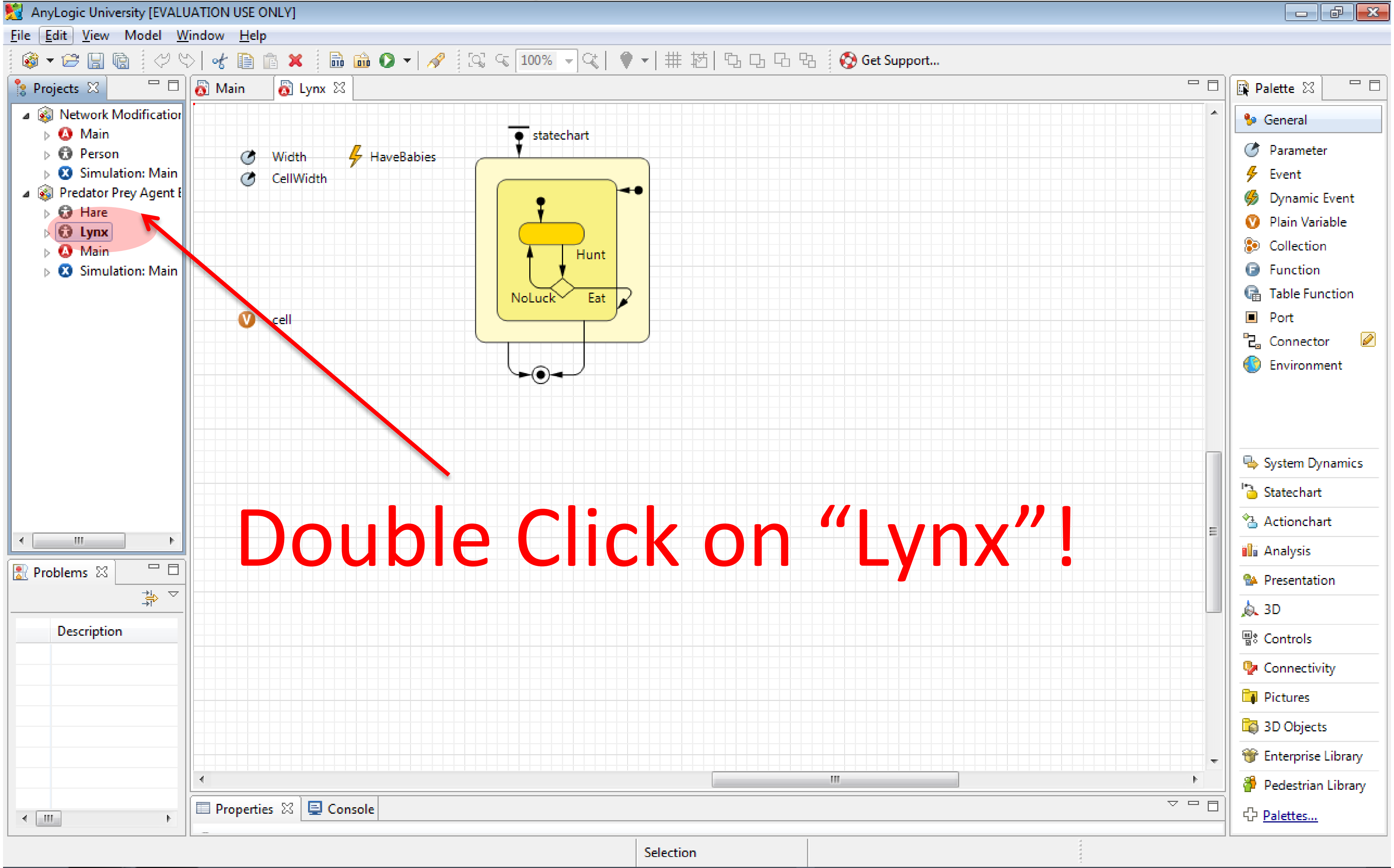
Visual input elements used during simulation (param. setting)

These represent the agent populations

These “functions” Calculate things or can change model behavior

Visual output elements used during simulation

Agent Class Defines the Characteristics & Behaviour of Agent Population Members



Double Click on “Lynx” !

A Critical Distinction: Design (Specification) vs. Execution (Run) times

- The computational elements of Anylogic support both design & execution time presence & behaviour
  - Design time: Specifying the model
  - Execution time (“Runtime”): Simulating the model
- It is important to be clear on what behavior & information is associated with which times
- Generally speaking, design-time elements (e.g. in the palettes) are created to support certain runtime behaviors

A Familiar Analogy

- The distinction between model design time & model execution time is like the distinction between
  - Time of Recipe Design: Here, we’re
    - Deciding what exact set of steps we’ll be following
    - Picking our ingredients
    - Deciding our preparation techniques
    - Choosing/making our cooking utensils (e.g. a cookie cutter)
  - Time of Cooking: When we actually are following the recipe
    - A given element of the recipe may be enacted many times
      - One step may be repeated many times
      - One cookie cutter may make many particular cookies

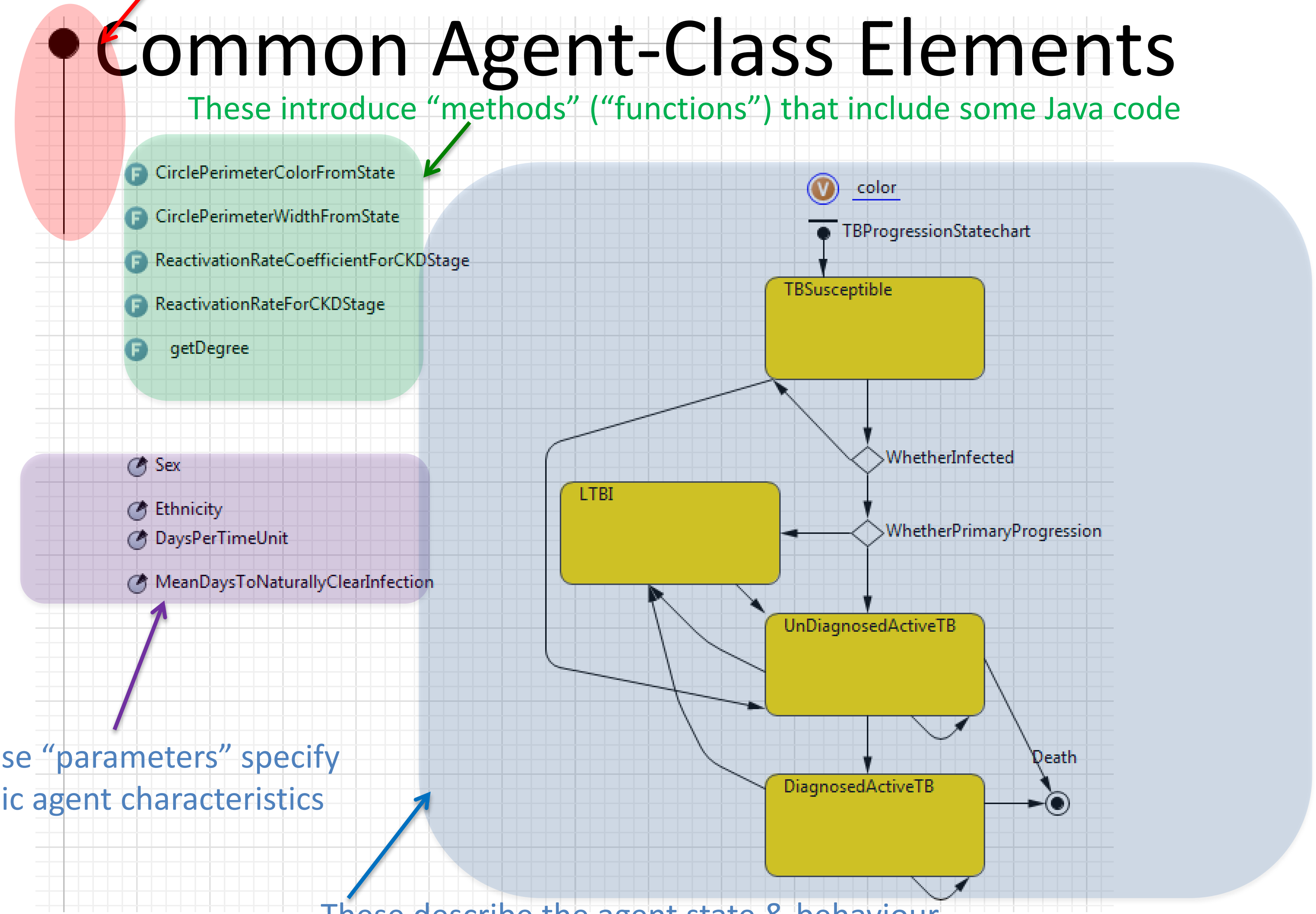
Cooking Analogy to an Agent Class: A Cookie Cutter

- We only need one cookie cutter to bake many cookies
- By carefully designing the cookie cutter, we can shape the character of many particular cookies
- By describing an Agent class at model design time, we are defining the cookie cutter we want to use

This defines the visual elements to be used for this object when it is displayed at runtime.

Common Agent-Class Elements

These introduce "methods" ("functions") that include some Java code

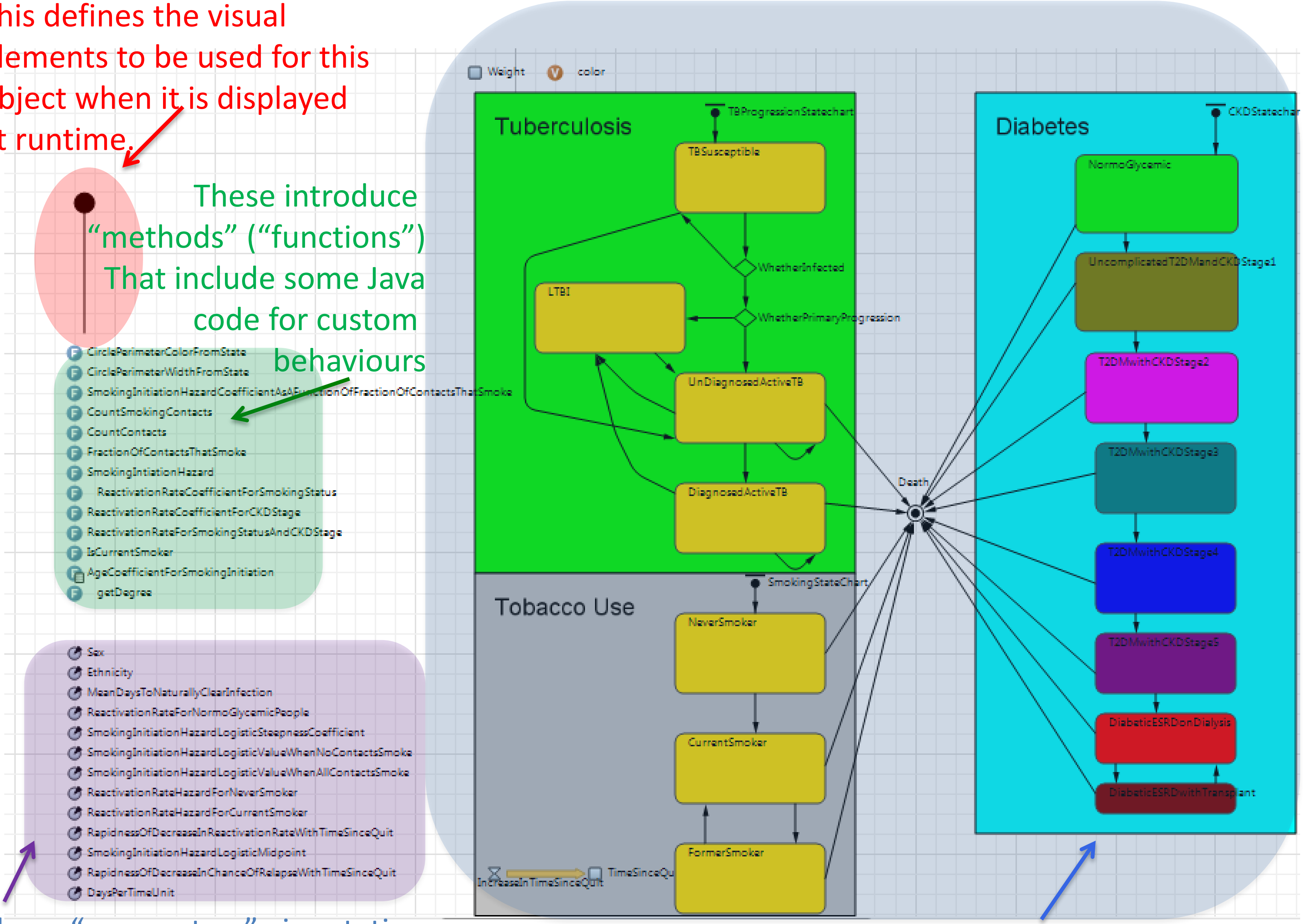

CirclePerimeterColorFromState
CirclePerimeterWidthFromState
ReactivationRateCoefficientForCKDStage
ReactivationRateForCKDStage
getDegree
Sex
Ethnicity
DaysPerTimeUnit
MeanDaysToNaturallyClearInfection
color
TBProgressionStatechart
TBSusceptible
WhetherInfected
LTBI
WhetherPrimaryProgression
UnDiagnosedActiveTB
Death
DiagnosedActiveTB

These "parameters" specify static agent characteristics

These describe the agent state & behaviour – the mechanisms that will govern agent dynamics

This defines the visual elements to be used for this object when it is displayed at runtime.

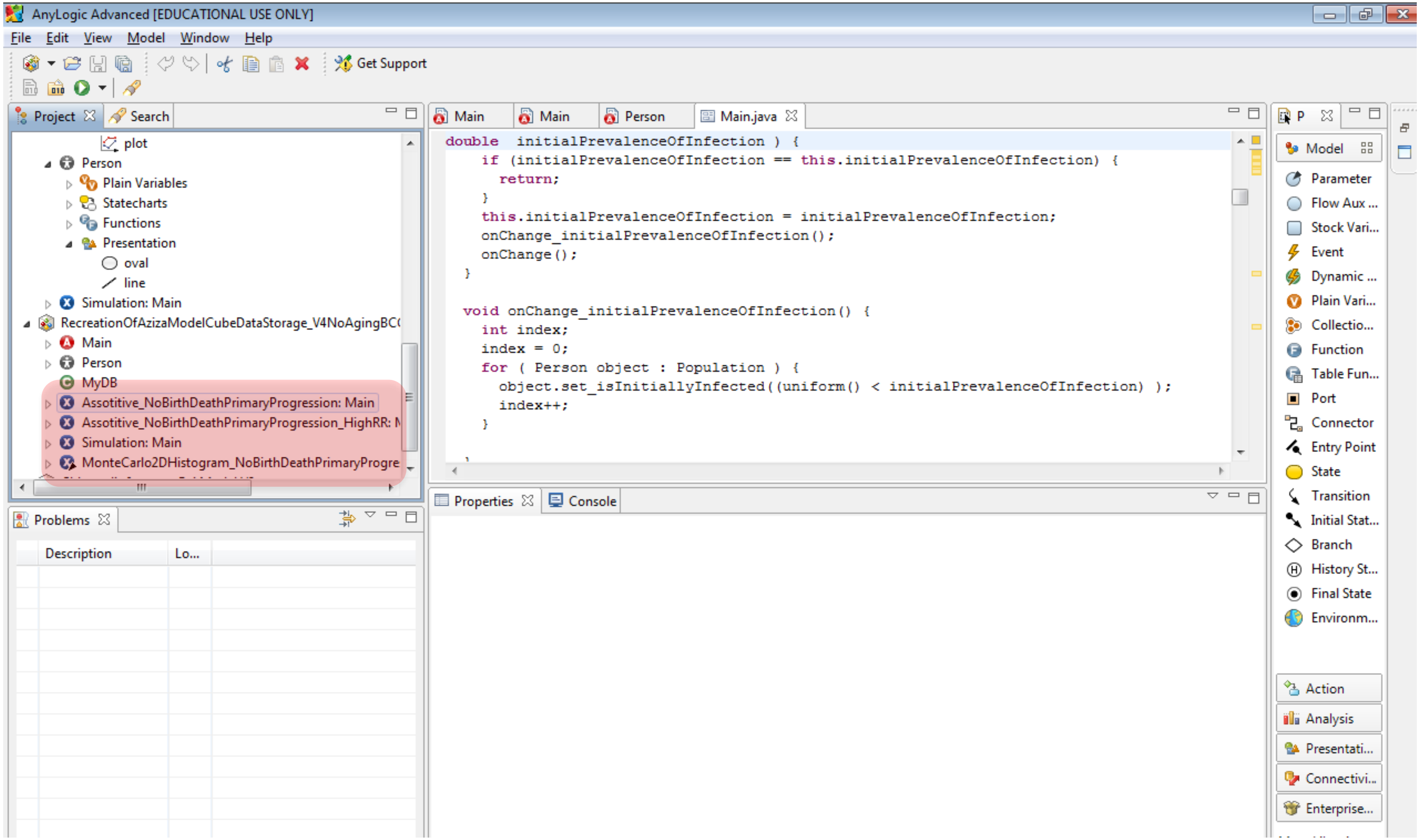
These introduce "methods" ("functions") That include some Java code for custom behaviours


Weight
color
Tuberculosis
TBProgressionStatechart
TBSusceptible
WhetherInfected
LTBI
WhetherPrimaryProgression
UnDiagnosedActiveTB
DiagnosedActiveTB
Tobacco Use
SmokingStateChart
NeverSmoker
CurrentSmoker
FormerSmoker
IncreaseInTimeSinceQuit
TimeSinceQu
Death
Diabetes
CKDStatechar
NormoGlycemic
UncomplicatedT2DMandCKDStage1
T2DMwithCKDStage2
T2DMwithCKDStage3
T2DMwithCKDStage4
T2DMwithCKDStage5
DiabeticESRDonDialysis
DiabeticESRDwithTransplant
CirclePerimeterColorFromState
CirclePerimeterWidthFromState
SmokingInitiationHazardCoefficientAsAFunctionOfFractionOfContactsThatSmoke
CountSmokingContacts
CountContacts
FractionOfContactsThatSmoke
SmokingIntiationHazard
ReactivationRateCoefficientForSmokingStatus
ReactivationRateCoefficientForCKDStage
ReactivationRateForSmokingStatusAndCKDStage
IsCurrentSmoker
AgeCoefficientForSmokingInitiation
getDegree
Sex
Ethnicity
MeanDaysToNaturallyClearInfection
ReactivationRateForNormoGlycemicPeople
SmokingInitiationHazardLogisticSteepnessCoefficient
SmokingInitiationHazardLogisticValueWhenNoContactsSmoke
SmokingInitiationHazardLogisticValueWhenAllContactsSmoke
ReactivationRateHazardForNeverSmoker
ReactivationRateHazardForCurrentSmoker
RapidnessOfDecreaseInReactivationRateWithTimeSinceQuit
SmokingInitiationHazardLogisticMidpoint
RapidnessOfDecreaseInChanceOfRelapseWithTimeSinceQuit
DaysPerTimeUnit

These "parameters" give static characteristics of the agent

These describe the "behaviours" – the mechanisms that will govern agent dynamics

Experiments



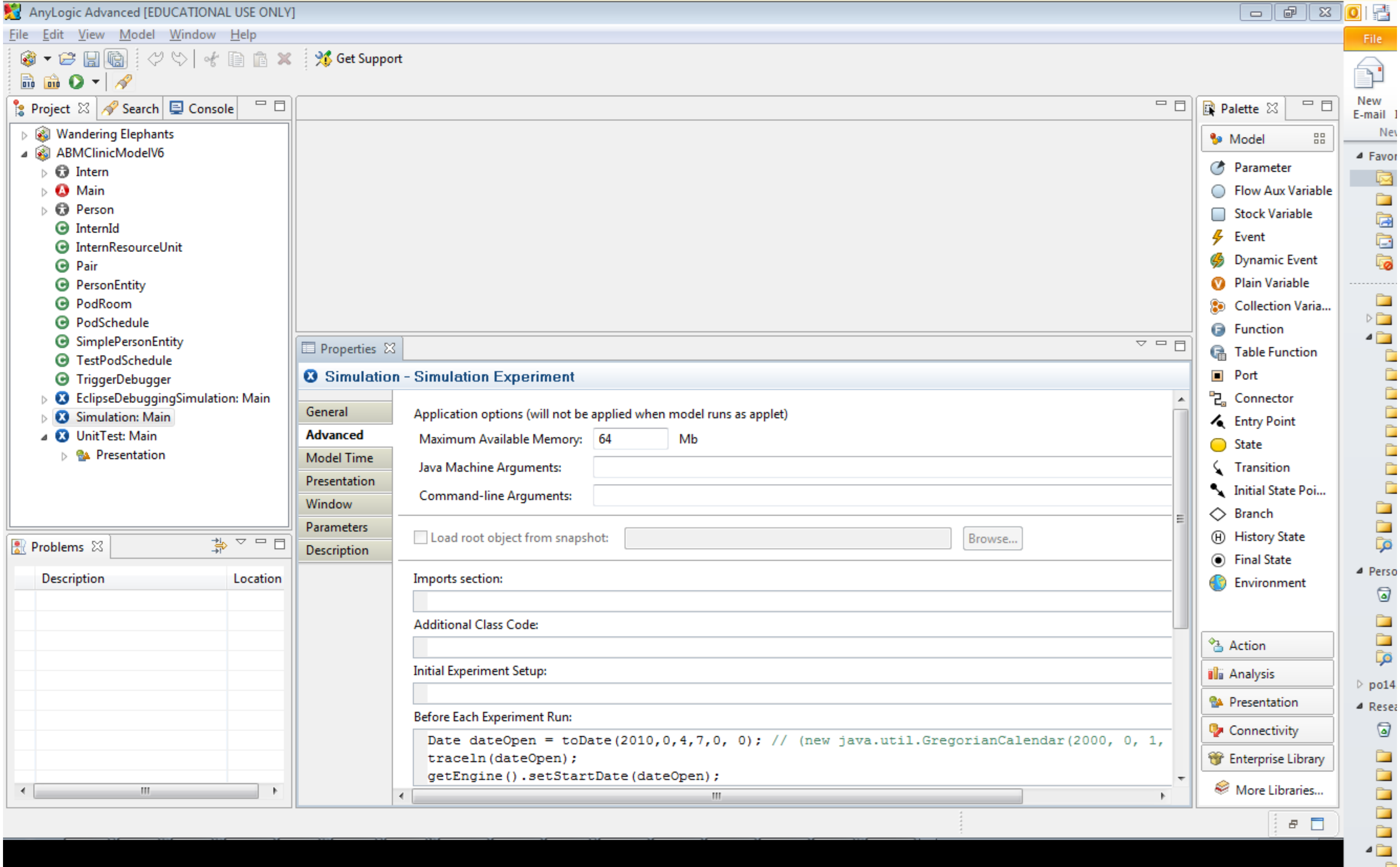
```
double  initialPrevalenceOfInfection ) {
    if (initialPrevalenceOfInfection == this.initialPrevalenceOfInfection) {
      return;
    }
    this.initialPrevalenceOfInfection = initialPrevalenceOfInfection;
    onChange_initialPrevalenceOfInfection();
    onChange();
  }

  void onChange_initialPrevalenceOfInfection() {
    int index;
    index = 0;
    for ( Person object : Population ) {
      object.set_isInitiallyInfected((uniform() < initialPrevalenceOfInfection) );
      index++;
    }
```

Experiment Classes

- Experiment classes allow you to define & run scenarios in which global parameters (i.e. parameters defined in *Main*) may hold either default or alternative values
- Experiment classes are also used to set
  - The time horizon for a simulation
  - Memory limits (important for large models)
  - Details of simulation run
  - Details on random number generation
  - Virtual machine arguments
- "Properties" allow one to set the values for each parameter
- Right click on these & choose "Run" to run such a scenario

Setting Memory & Virtual Machine Arguments



Simulation - Simulation Experiment

Application options (will not be applied when model runs as applet)

Maximum Available Memory: 64 Mb

Java Machine Arguments:

Command-line Arguments:

☐ Load root object from snapshot:

Imports section:

Additional Class Code:

Initial Experiment Setup:

Before Each Experiment Run:

```
Date dateOpen = toDate(2010,0,4,7,0, 0); // (new java.util.GregorianCalendar(2000, 0, 1,
traceln(dateOpen);
getEngine().setStartDate(dateOpen);
```

The Notion of a “Build”

- We prepare a fully specified model to run a simulation using a “build”
  - If all goes well, this translates project to executable Java
  - This may alert you to errors in the project
- A “Compiler” is a tool to convert from a program’s specification (e.g. state charts, Action diagrams, etc.) to a representation that can be executed
  - Normally a compiler is applied to each of several components of a program (e.g. classes)
  - AnyLogic’s “build” process applies a compiler to the components of the AnyLogic model

Cooking Analogy to "Build"ing: Obtaining & Preparing the Ingredients

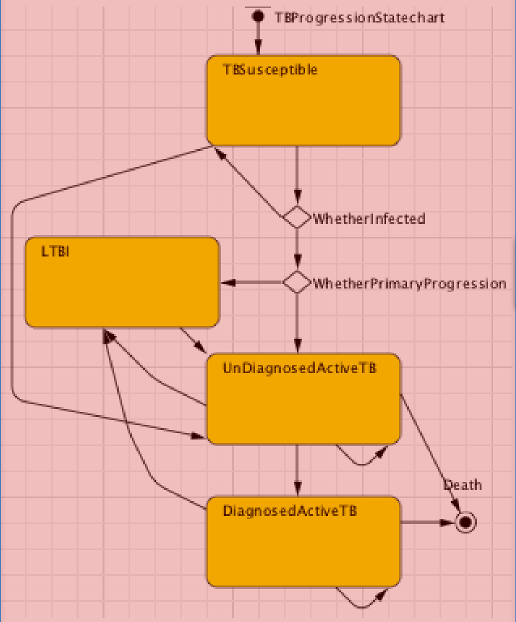
- Before we can actually realize the recipe, we need to go collect & prepare all ingredients
- We're not yet cooking, but what we are doing makes the cooking possible
- The "cooking" here is running the modle

A Bit on “Java”…

- “Java” is a popular cross-platform “object oriented” programming language introduced by Sun Microsystems
- Anylogic is written in Java and turns models into Java
- AnyLogic offers lots of ways to insert snippets (“hooks”) of Java code
- You will need these if you want to e.g.
  - Push AnyLogic outside the envelop of its typical support
    - e.g. Enabling a network with diverse Agent types
  - Exchange messages between Agents
  - Put into place particular initialization mechanisms
  - Collect custom statistics over the population

Stages of the Anylogic Build

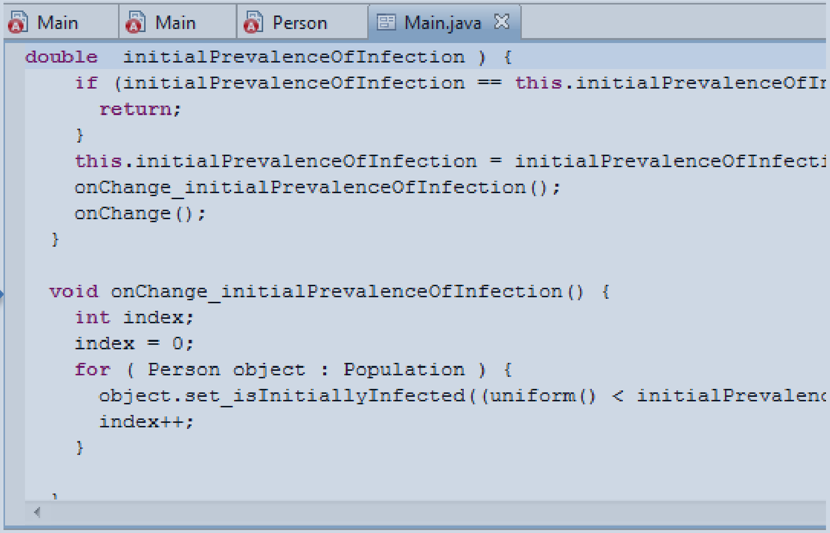
Modification Possible



→

Modification Not Possible

Java Code



```
double  initialPrevalenceOfInfection ) {
    if (initialPrevalenceOfInfection == this.initialPrevalenceOfIn
      return;
    }
    this.initialPrevalenceOfInfection = initialPrevalenceOfInfecti
    onChange_initialPrevalenceOfInfection();
    onChange();
  }

  void onChange_initialPrevalenceOfInfection() {
    int index;
    index = 0;
    for ( Person object : Population ) {
      object.set_isInitiallyInfected((uniform() < initialPrevalenc
      index++;
    }
```

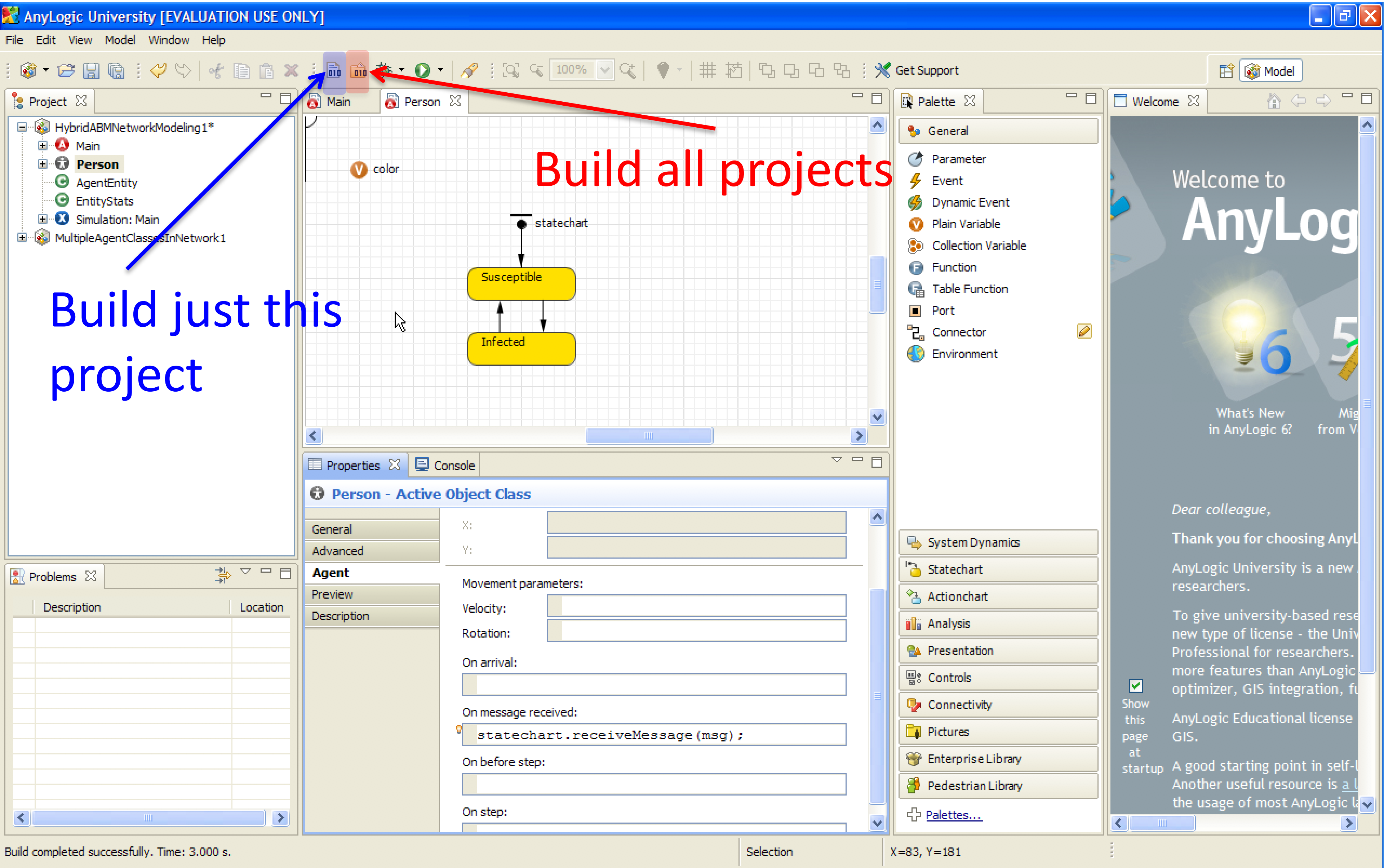
→

JVM Byte Code

Person.class

"Build" Buttons

(One just for this project, one for all projects)



Build all projects

Build just this project

Person - Active Object Class

General
Advanced
Agent
Preview
Description

X:

Y:

Movement parameters:

Velocity:

Rotation:

On arrival:

On message received:

```
statechart.receiveMessage(msg);
```

On before step:

On step:

Dear colleague,

Thank you for choosing AnyL

AnyLogic University is a new researchers.

To give university-based rese new type of license - the Univ Professional for researchers. more features than AnyLogic optimizer, GIS integration, fu

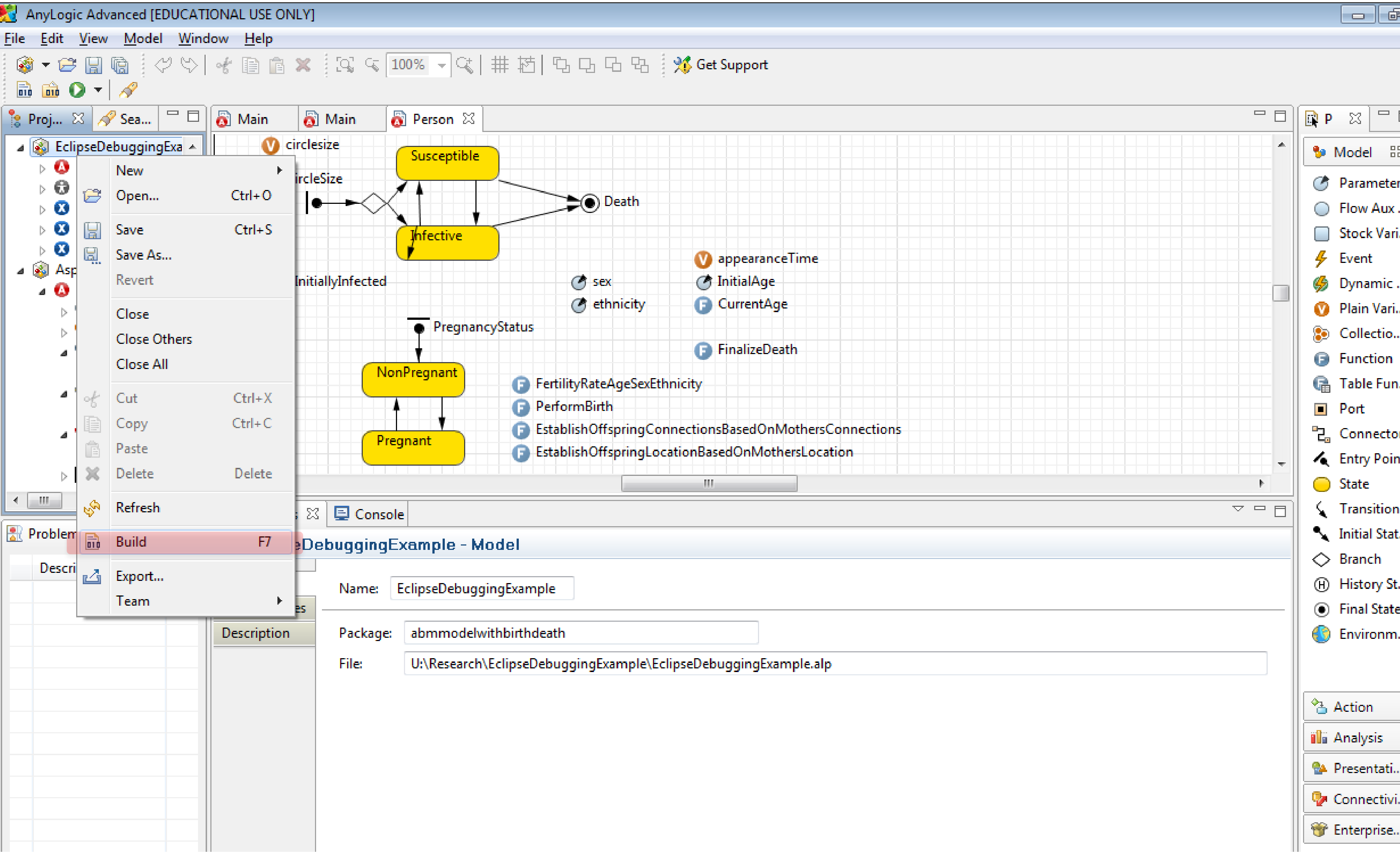
AnyLogic Educational license GIS.

A good starting point in self-l Another useful resource is a l the usage of most AnyLogic la

☑ Show this page at startup

Build completed successfully. Time: 3.000 s.

Alternative: Building via Context Menu



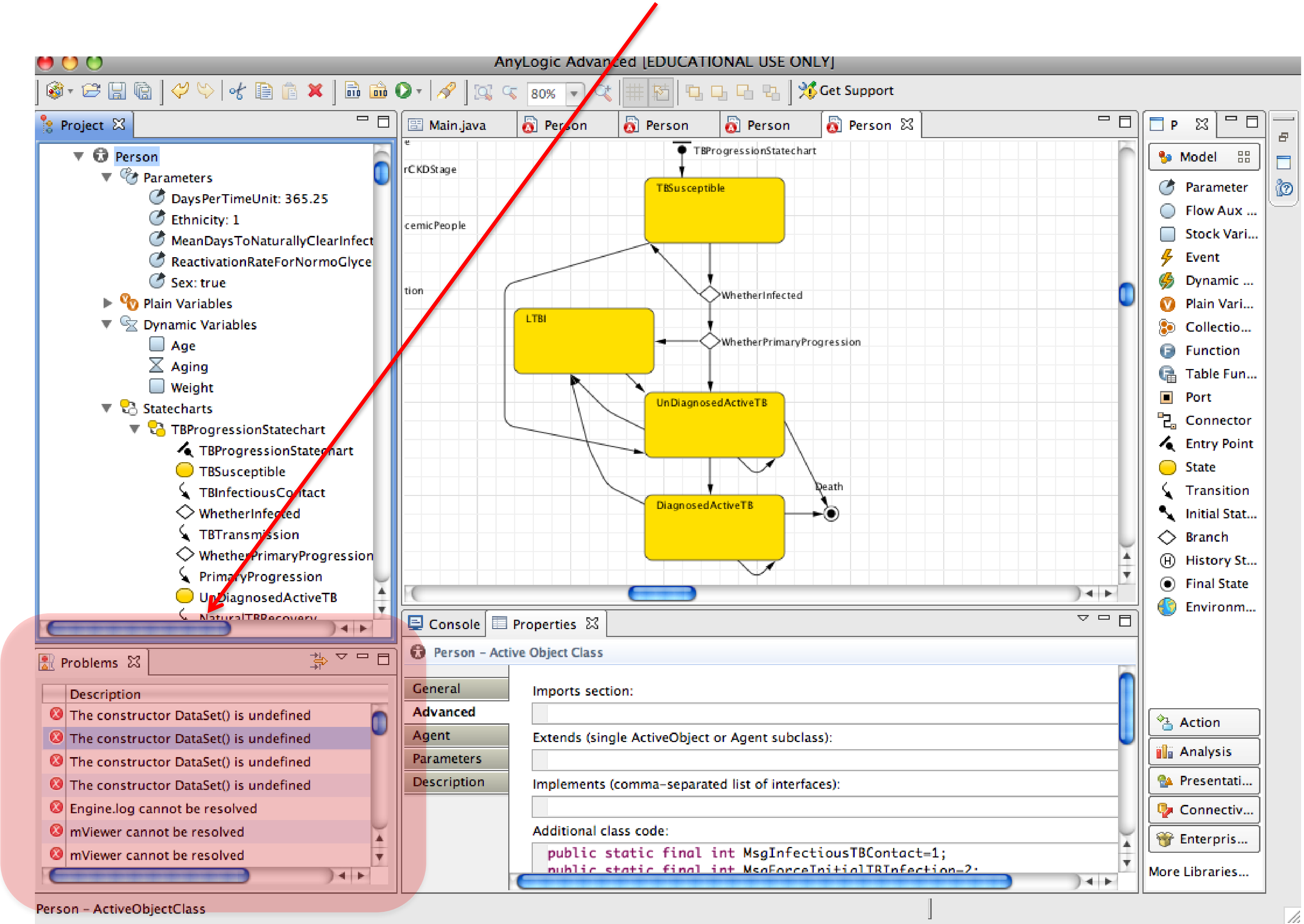
eDebuggingExample - Model

Name: EclipseDebuggingExample

Package: abmmodelwithbirthdeath

File: U:\Research\EclipseDebuggingExample\EclipseDebuggingExample.alp

Builds Gone Bad: The “Problems View”



Problems

Description
The constructor DataSet() is undefined
The constructor DataSet() is undefined
The constructor DataSet() is undefined
The constructor DataSet() is undefined
Engine.log cannot be resolved
mViewer cannot be resolved
mViewer cannot be resolved

Person – Active Object Class

Imports section:

Extends (single ActiveObject or Agent subclass):

Implements (comma-separated list of interfaces):

Additional class code:

```
public static final int MsgInfectiousTBContact=1;
public static final int MsgForceInitialTBInfection=2;
```

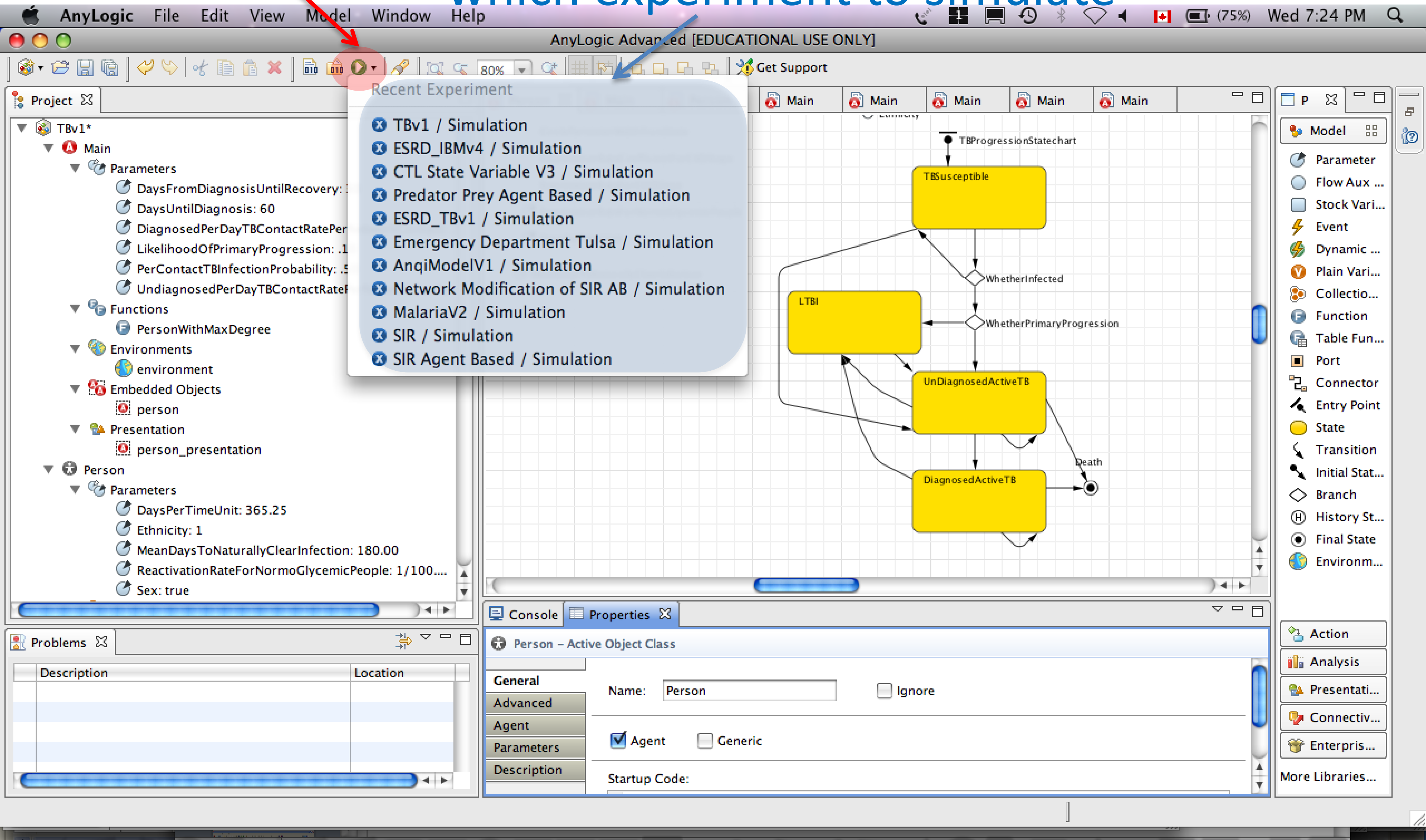
Person – ActiveObjectClass

Builds Gone Good: Model Execution

- The simulation is running
- Time is advancing in steps or as necessary to handle events
- Each agent class will typically have many particular agents in existence
  - Each agent will have a particular state
  - This population may fluctuate
- Variables will be changing value
- Presentation elements will be knit together into a dynamic presentation

Press this button to run an experiment (a simulation)

You can pull down the menu to choose which experiment to simulate



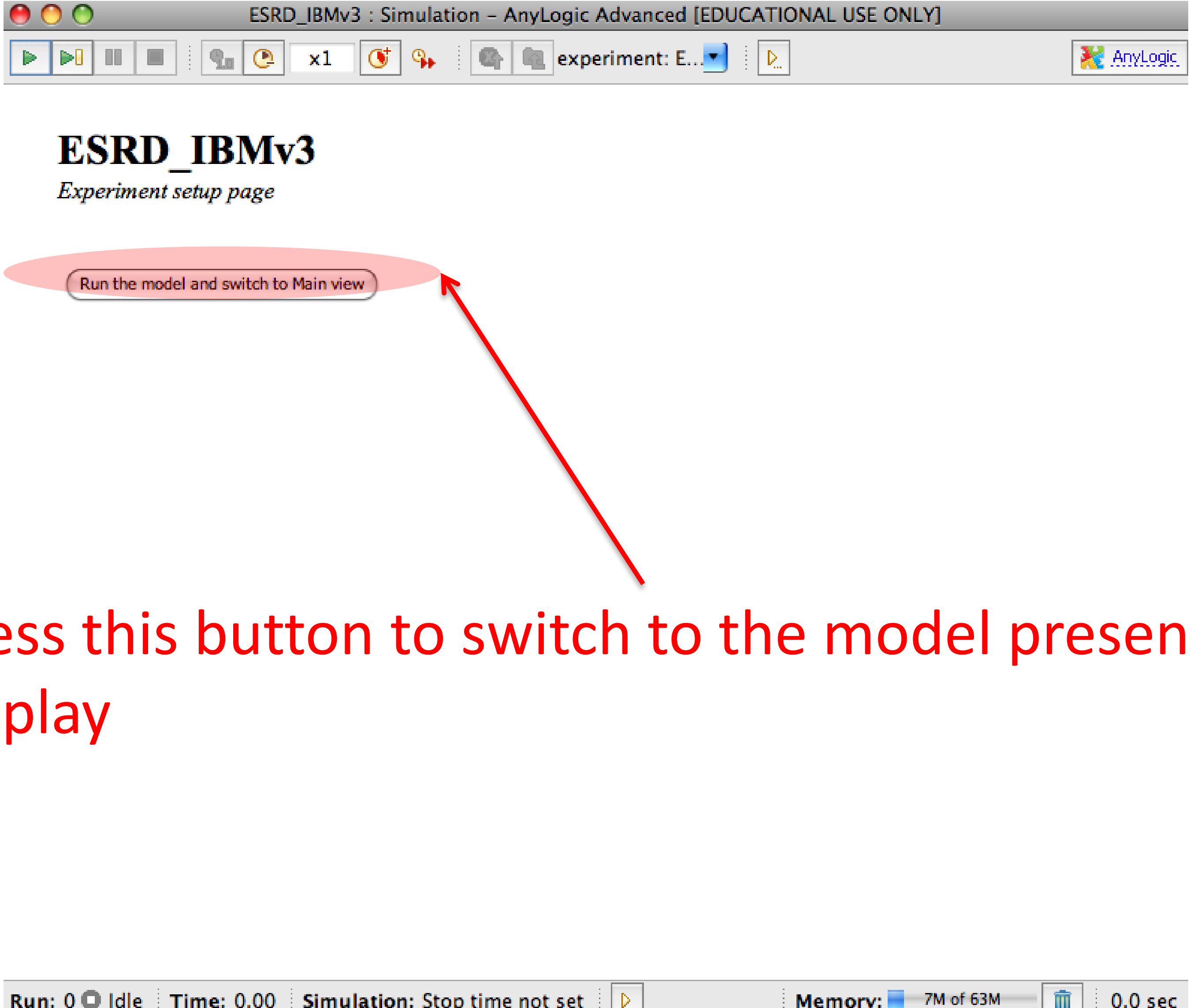
Person – Active Object Class

Name: Person ☐ Ignore

☑ Agent ☐ Generic

Startup Code:

Initial Screen: Experiment Set up (Use to set speed, parameters via UI)



ESRD_IBMv3 : Simulation – AnyLogic Advanced [EDUCATIONAL USE ONLY]

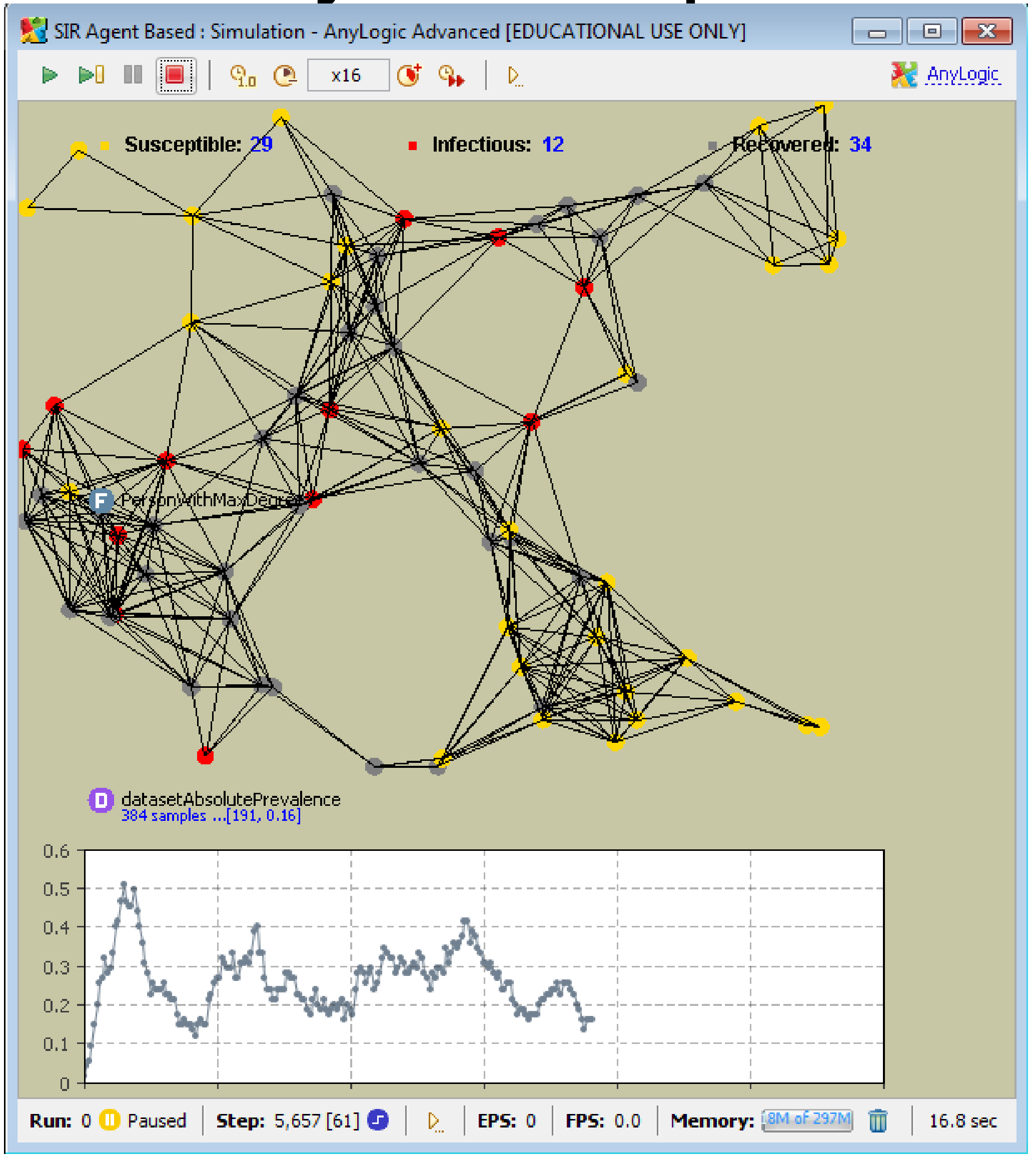
ESRD_IBMv3

Experiment setup page

Run: 0 Idle | Time: 0.00 | Simulation: Stop time not set | Memory: 7M of 63M | 0.0 sec

Press this button to switch to the model presentation display

Presentation of the Model "Main" Object in Operation



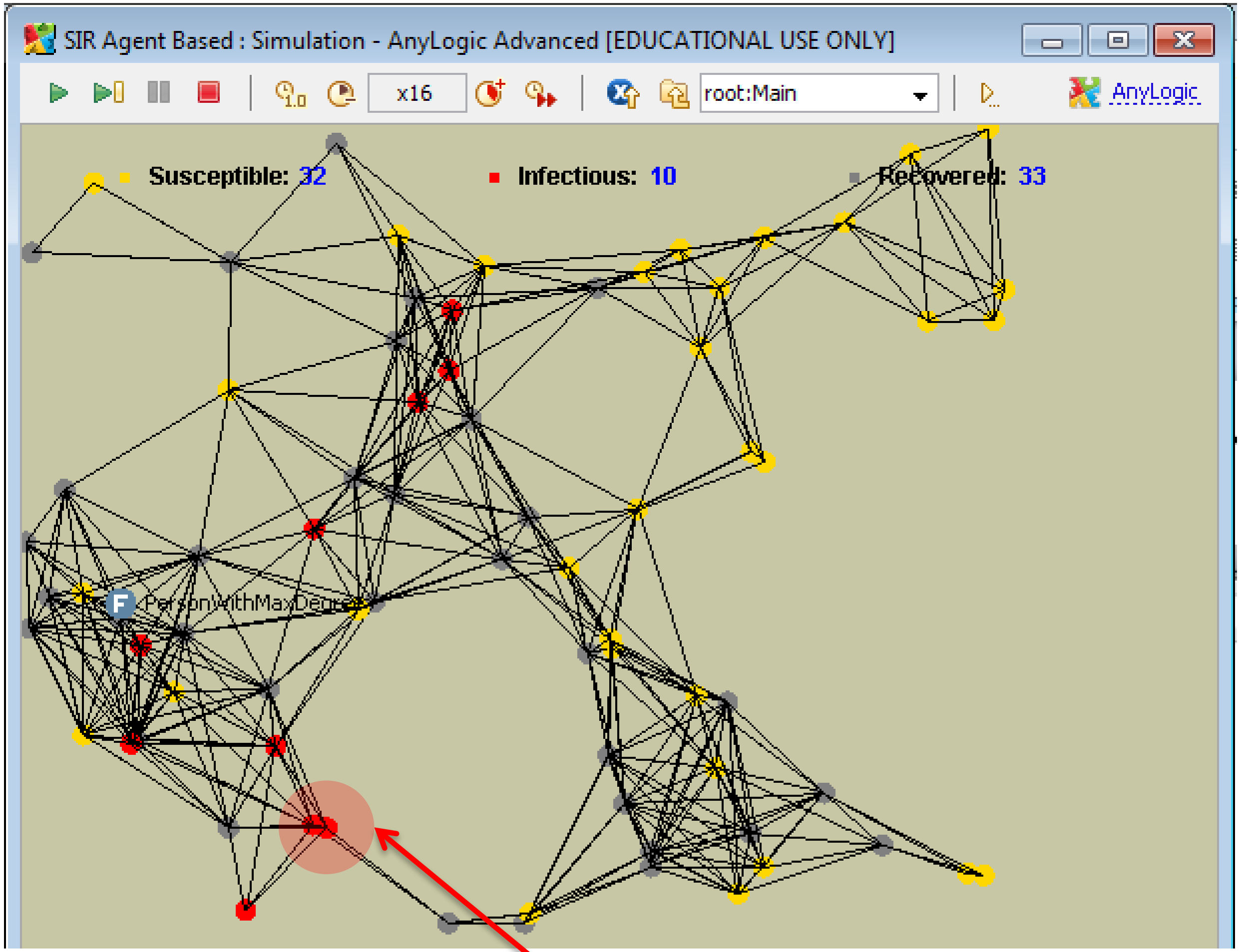
SIR Agent Based : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]
x16
AnyLogic
PersonWithMaxDegree
0.6
0.5
0.4
0.3
0.2
0.1
0

Susceptible: 29 Infectious: 12 Recovered: 34

datasetAbsolutePrevalence
384 samples ...[191, 0.16]

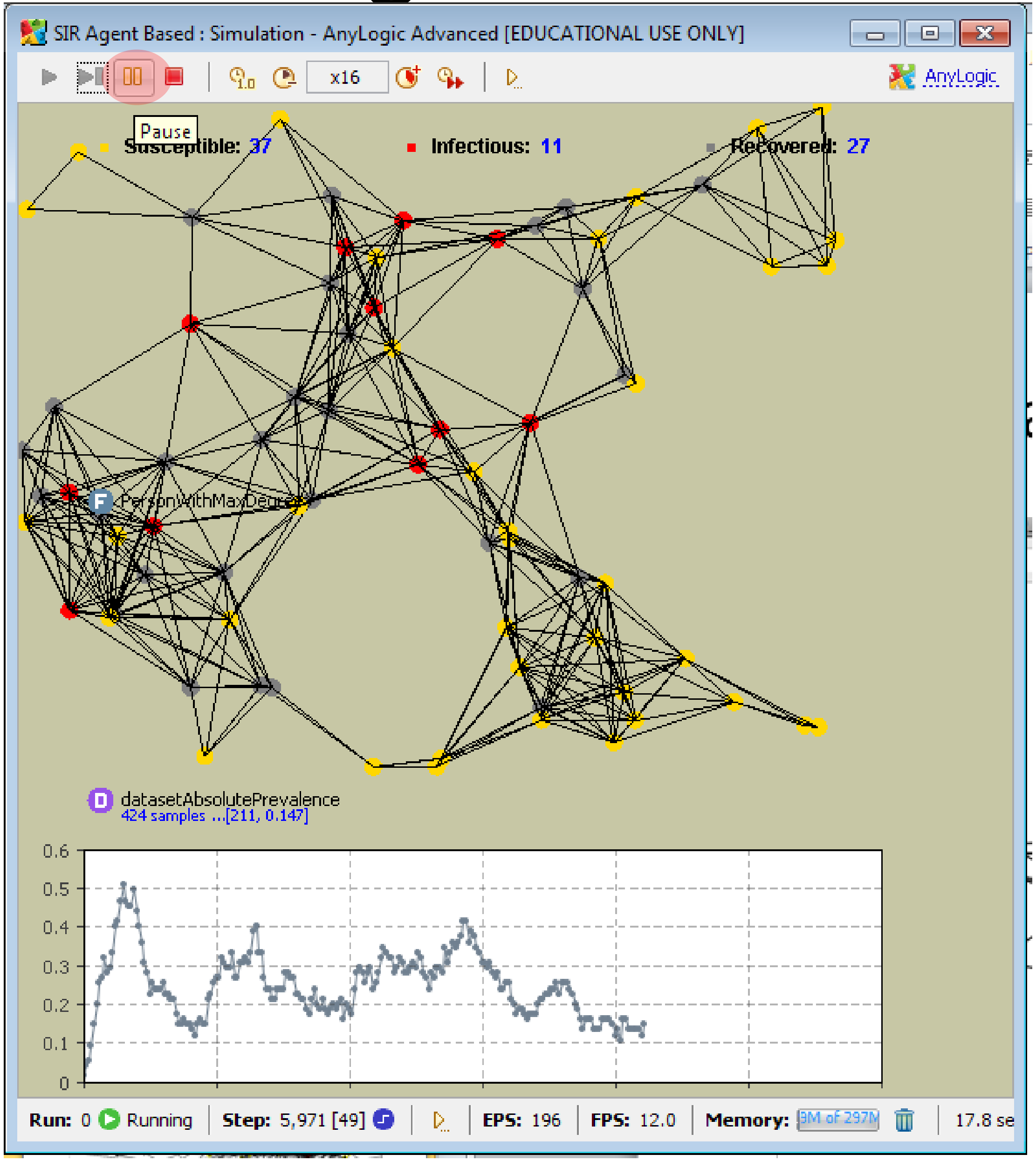
Run: 0 Paused | Step: 5,657 [61] | EPS: 0 | FPS: 0.0 | Memory: 8M of 297M | 16.8 sec

Network Embedding of Agents



Dynamic color updates via agent logic

Pausing the Model



SIR Agent Based : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]
x16
AnyLogic
Pause
PersonWithMaxDegree
0.6
0.5
0.4
0.3
0.2
0.1
0

Susceptible: 37

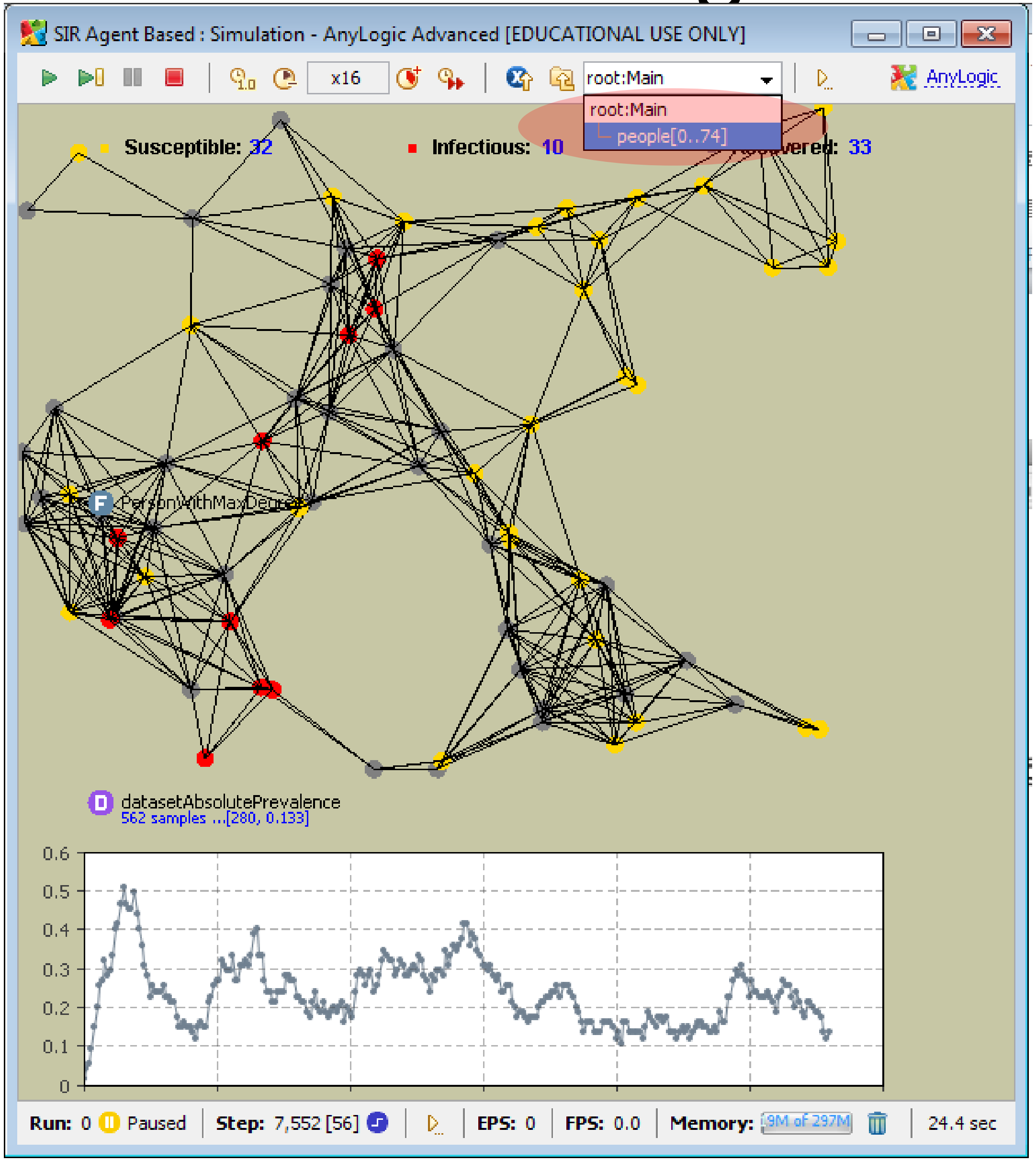
Infectious: 11

Recovered: 27

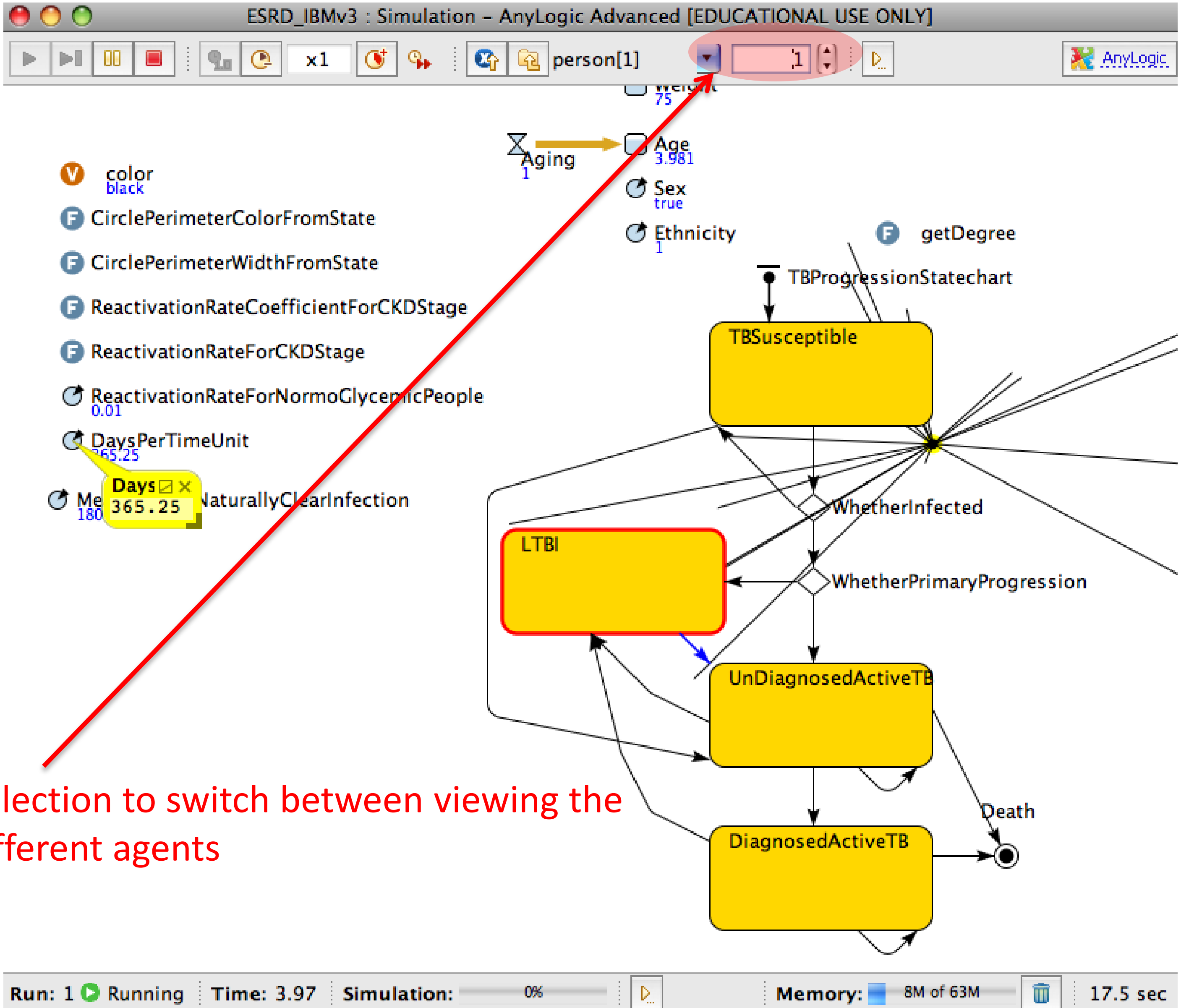
datasetAbsolutePrevalence
424 samples ...[211, 0.147]

Run: 0 Running | Step: 5,971 [49] | EPS: 196 | FPS: 12.0 | Memory: 3M of 297M | 17.8 se

Drill Down from the Model to Particular Agents

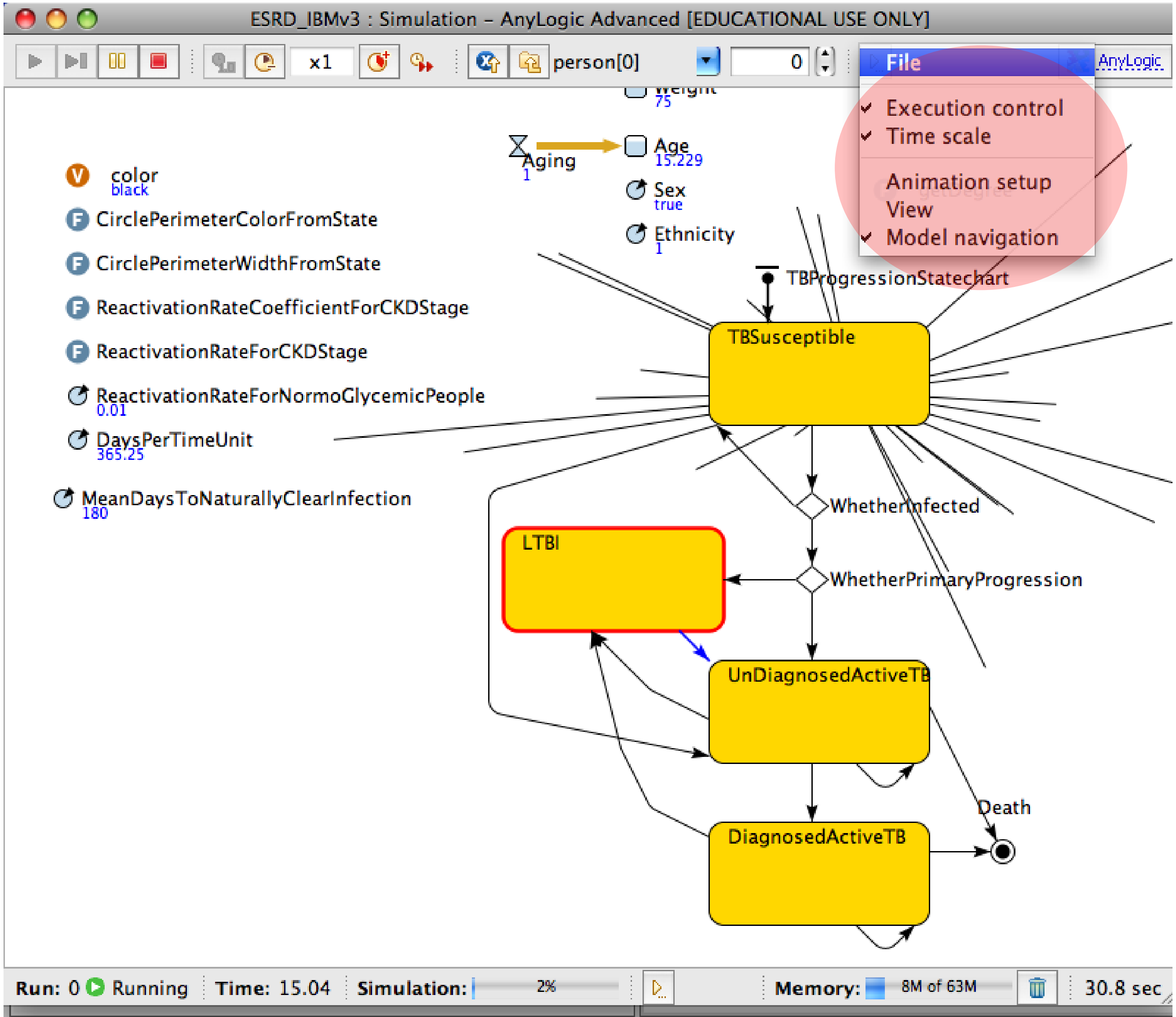


Runtime View of Particular Agent (Drill Down from Previous View)

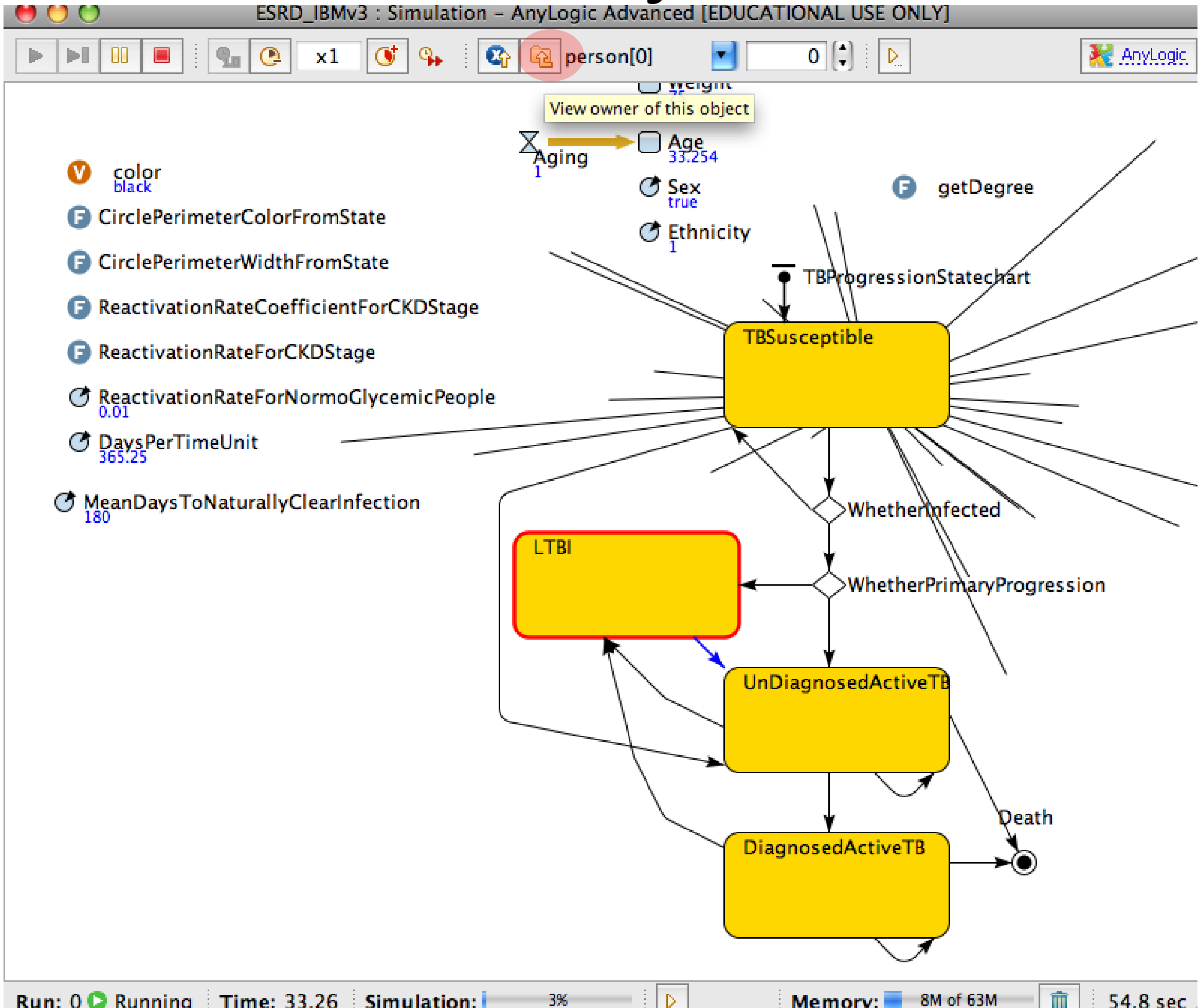


Use this selection to switch between viewing the State of different agents

Customizing the Model Running User Interface



Switching Back to View the Main Object



ESRD_IBMv3 : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]

View owner of this object

color
black

ReactivationRateForNormoGlycemicPeople
0.01

DaysPerTimeUnit
365.25

MeanDaysToNaturallyClearInfection
180

Aging
1

Age
33.254

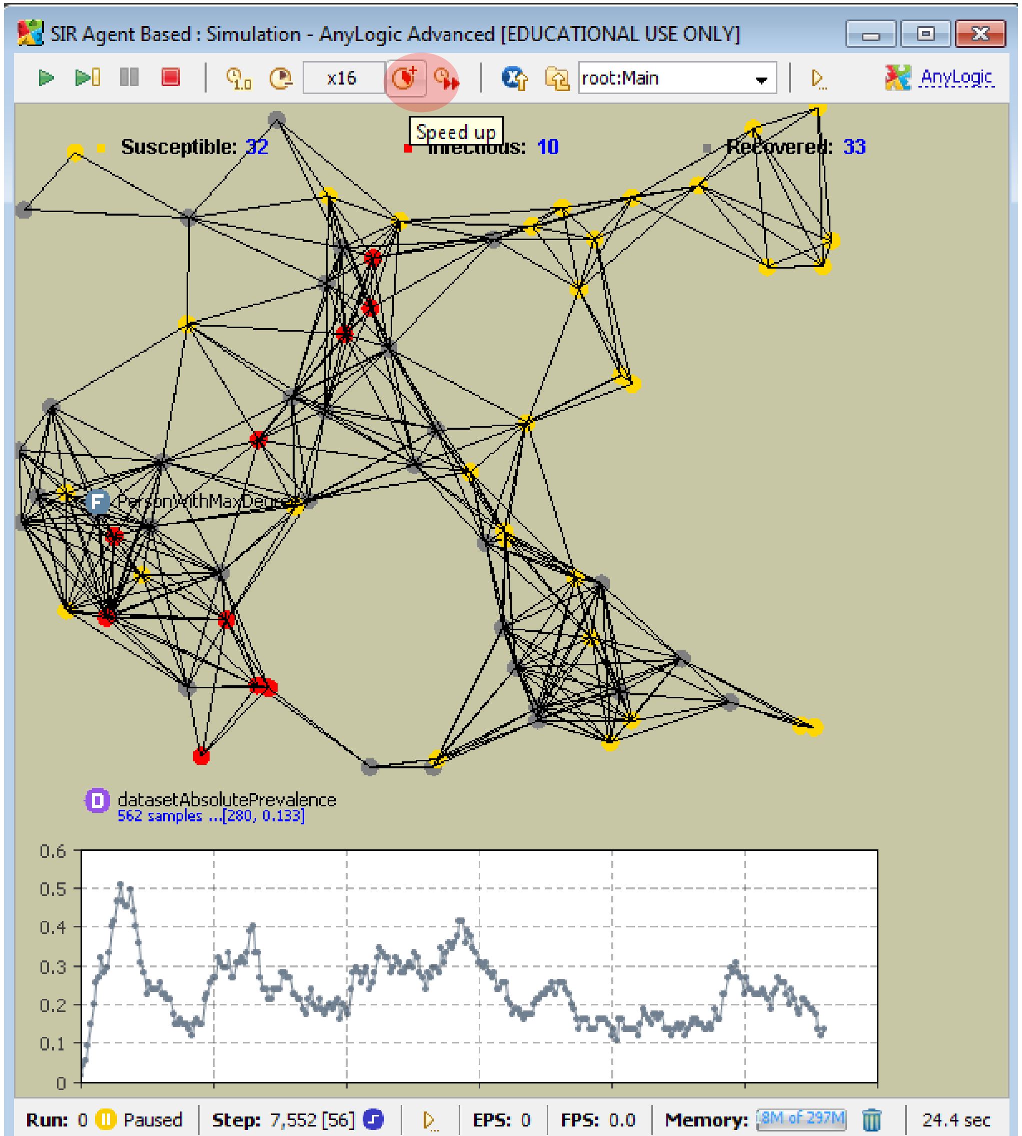
Sex
true

Ethnicity
1

Run: 0 Running | Time: 33.26 | Simulation: 3% | Memory: 8M of 63M | 54.8 sec

x1
person[0]
0
AnyLogic
Weight
CirclePerimeterColorFromState
CirclePerimeterWidthFromState
ReactivationRateCoefficientForCKDStage
ReactivationRateForCKDStage
getDegree
TBProgressionStatechart
TBSusceptible
WhetherInfected
WhetherPrimaryProgression
LTBI
UnDiagnosedActiveTB
DiagnosedActiveTB
Death

Controlling Simulation Speed (Speeding up)



SIR Agent Based : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]

Susceptible: 32

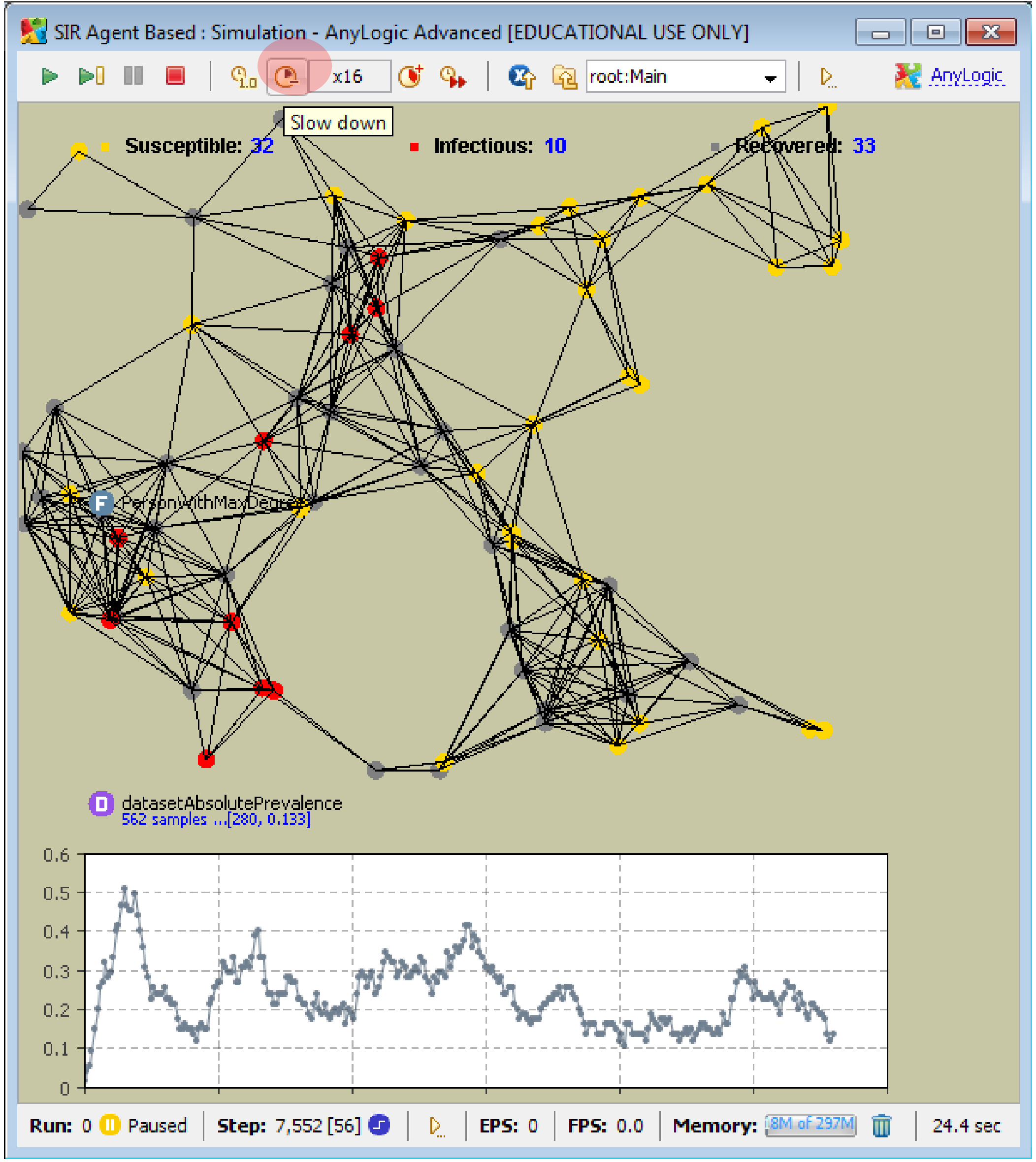
Infectious: 10

Recovered: 33

datasetAbsolutePrevalence
562 samples ...[280, 0.133]

Run: 0 Paused | Step: 7,552 [56] | EPS: 0 | FPS: 0.0 | Memory: 8M of 297M | 24.4 sec

Controlling Simulation Speed (Slowing Down)



SIR Agent Based : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]

Susceptible: 32

Infectious: 10

Recovered: 33

PersonWithMaxDegree

datasetAbsolutePrevalence
562 samples ...[280, 0.133]

Run: 0 Paused

Step: 7,552 [56]

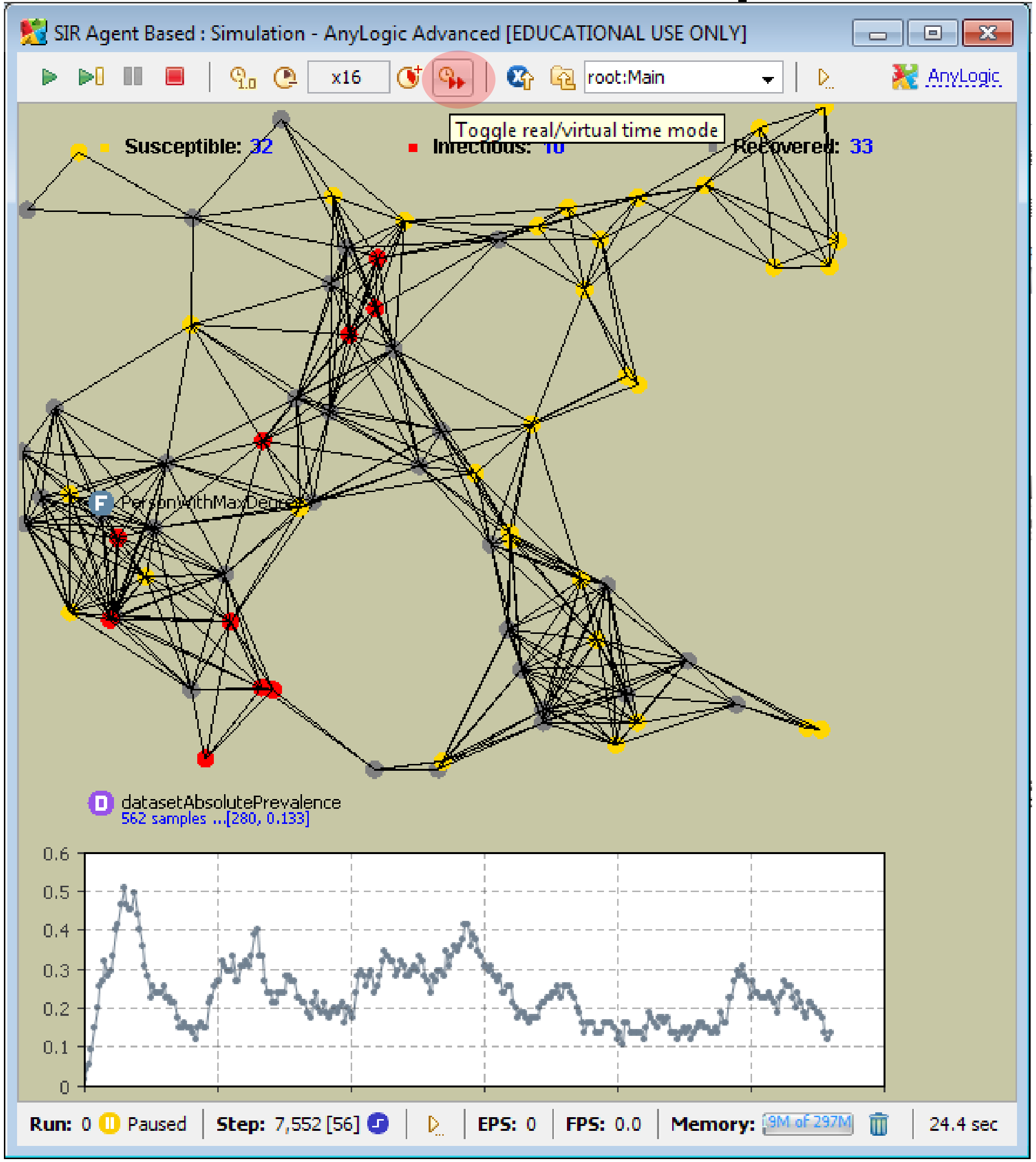
EPS: 0

FPS: 0.0

Memory: 38M of 297M

24.4 sec

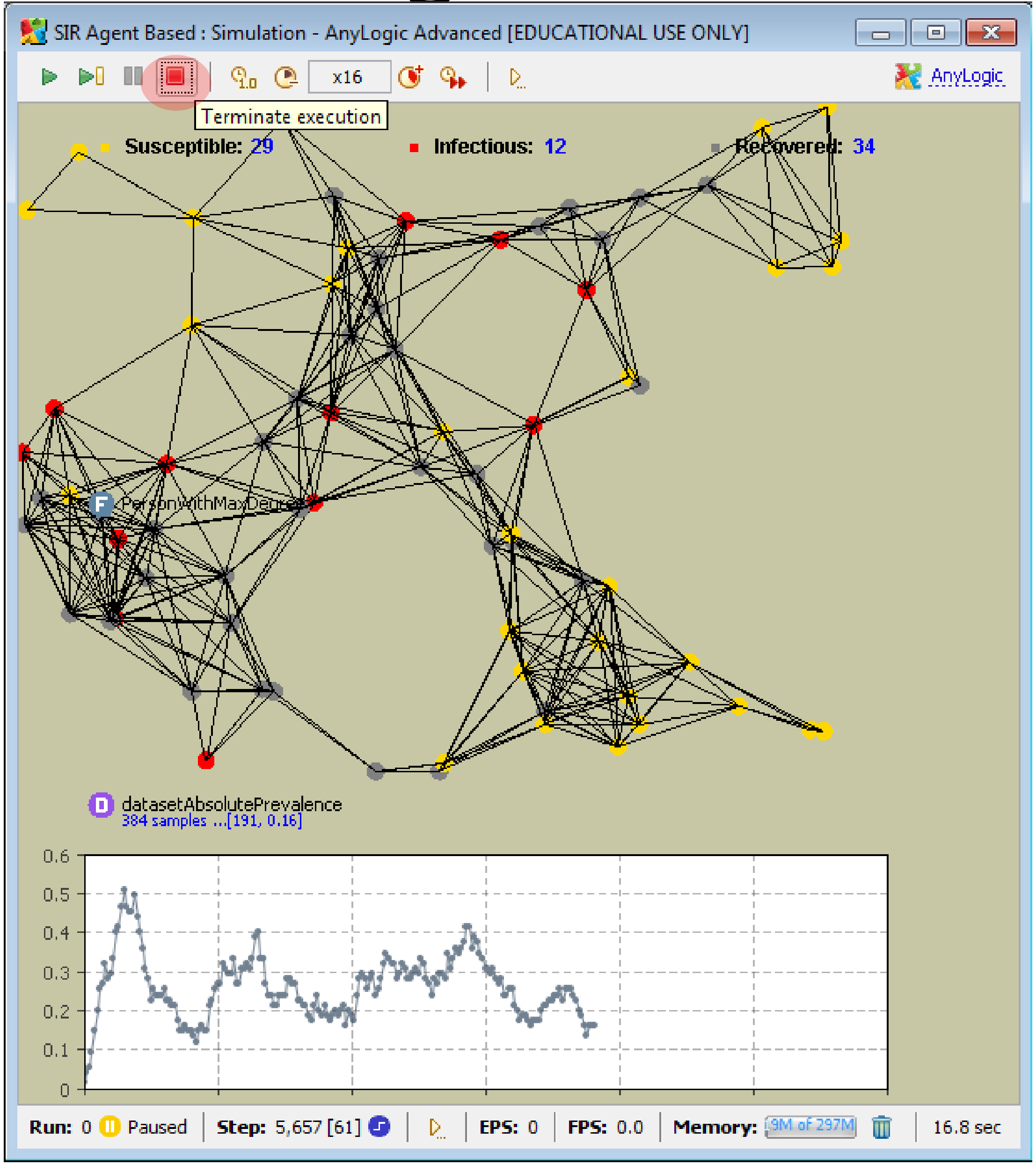
Toggling between Maximum and a Throttled Speed



SIR Agent Based : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]

Run: 0 Paused | Step: 7,552 [56] | EPS: 0 | FPS: 0.0 | Memory: 9M of 297M | 24.4 sec

Terminating Model Execution

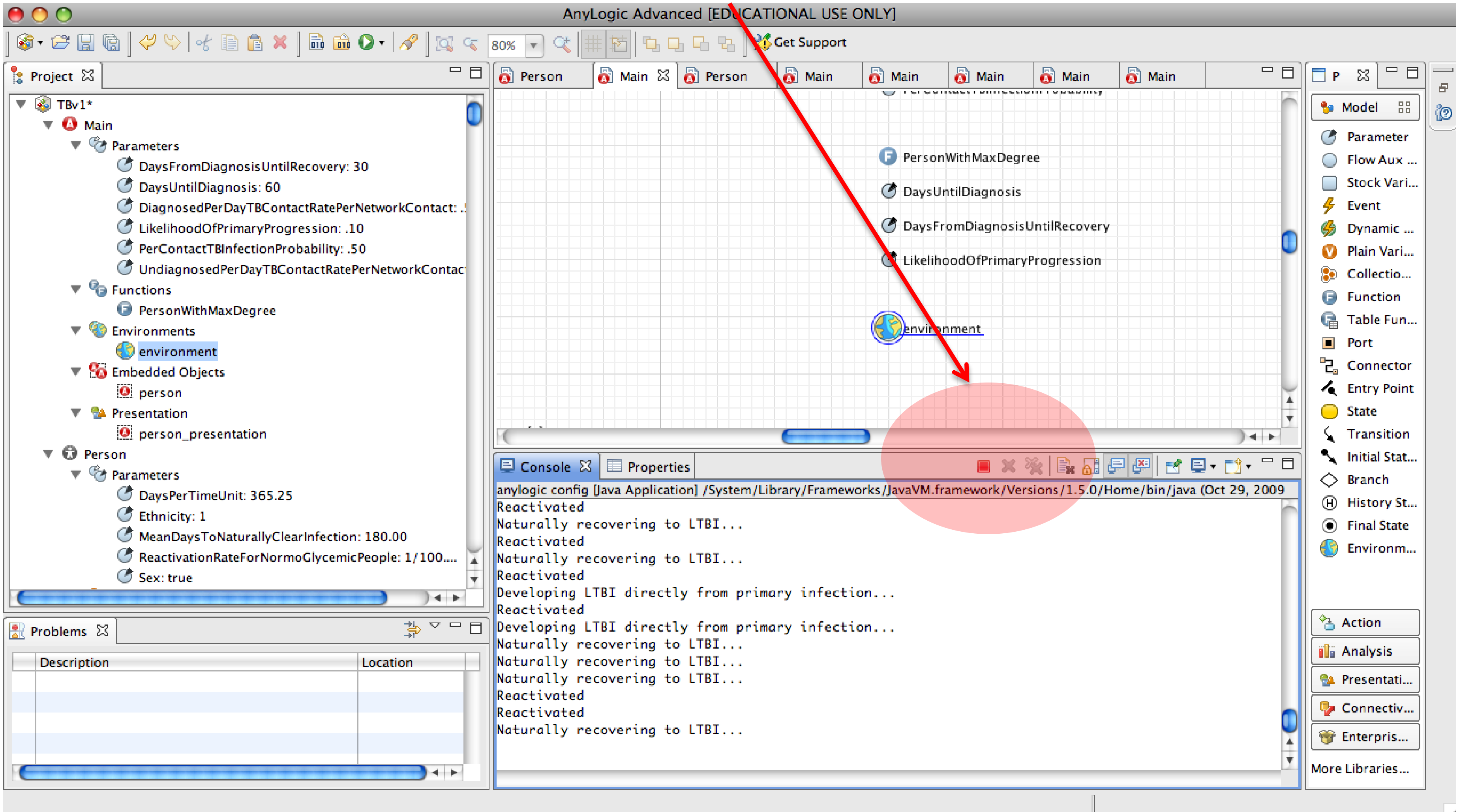

SIR Agent Based : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]
x16
AnyLogic
Terminate execution
PersonWithMaxDegree
datasetAbsolutePrevalence
384 samples ...[191, 0.16]
0.6
0.5
0.4
0.3
0.2
0.1
0

Susceptible: 29 Infectious: 12 Recovered: 34

Run: 0 Paused | Step: 5,657 [61] | EPS: 0 | FPS: 0.0 | Memory: 9M of 297M | 16.8 sec

Another Way to Terminate a Simulation

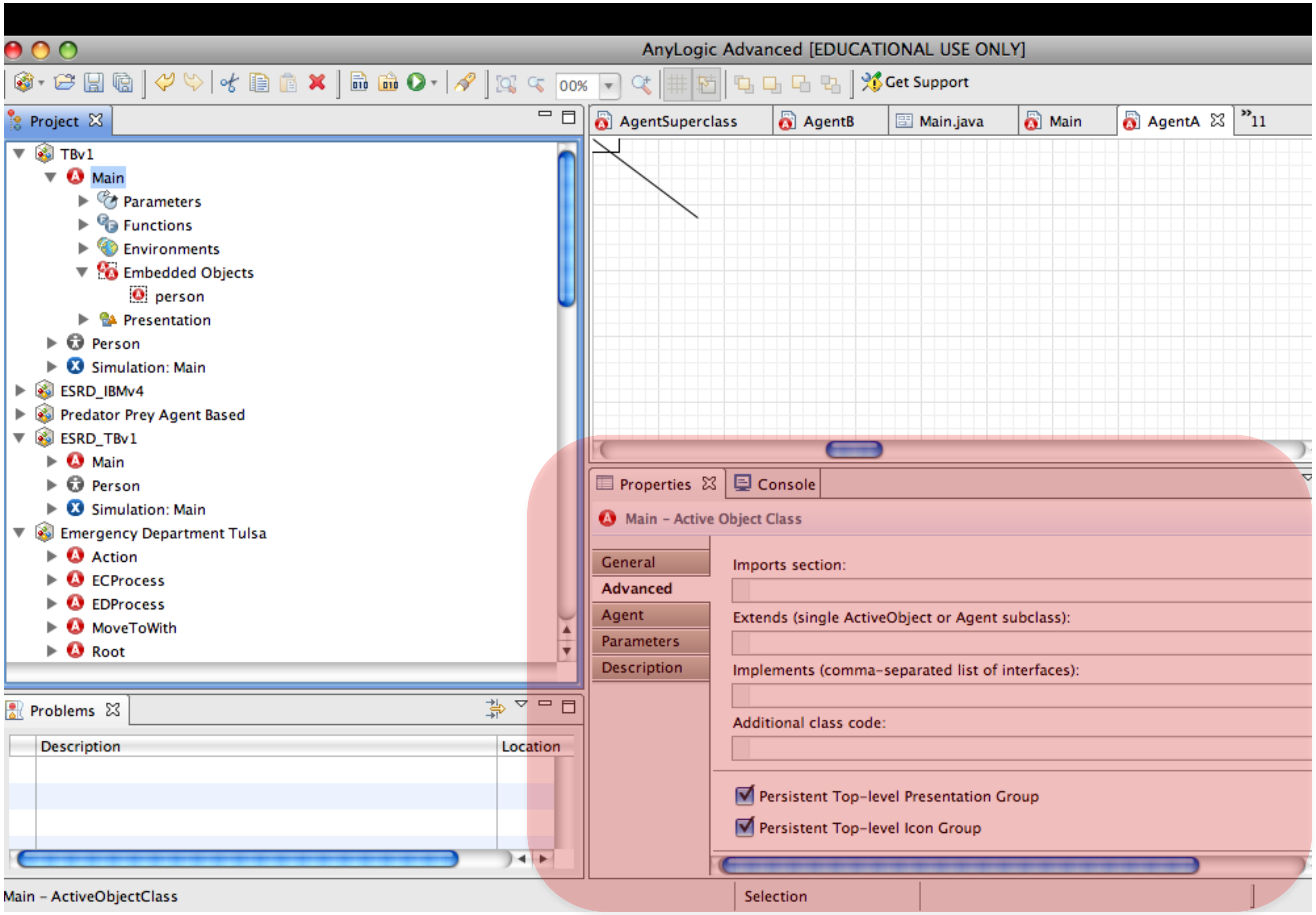
Use this Console “stop’” button to terminate the simulation



```
anylogic config [Java Application] /System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Home/bin/java (Oct 29, 2009
Reactivated
Naturally recovering to LTBI...
Reactivated
Naturally recovering to LTBI...
Reactivated
Developing LTBI directly from primary infection...
Reactivated
Developing LTBI directly from primary infection...
Naturally recovering to LTBI...
Naturally recovering to LTBI...
Naturally recovering to LTBI...
Reactivated
Reactivated
Naturally recovering to LTBI...
```

Examples of Where to Insert Code Object Properties

- “Advanced”



Imports section:

Extends (single ActiveObject or Agent subclass):

Implements (comma-separated list of interfaces):

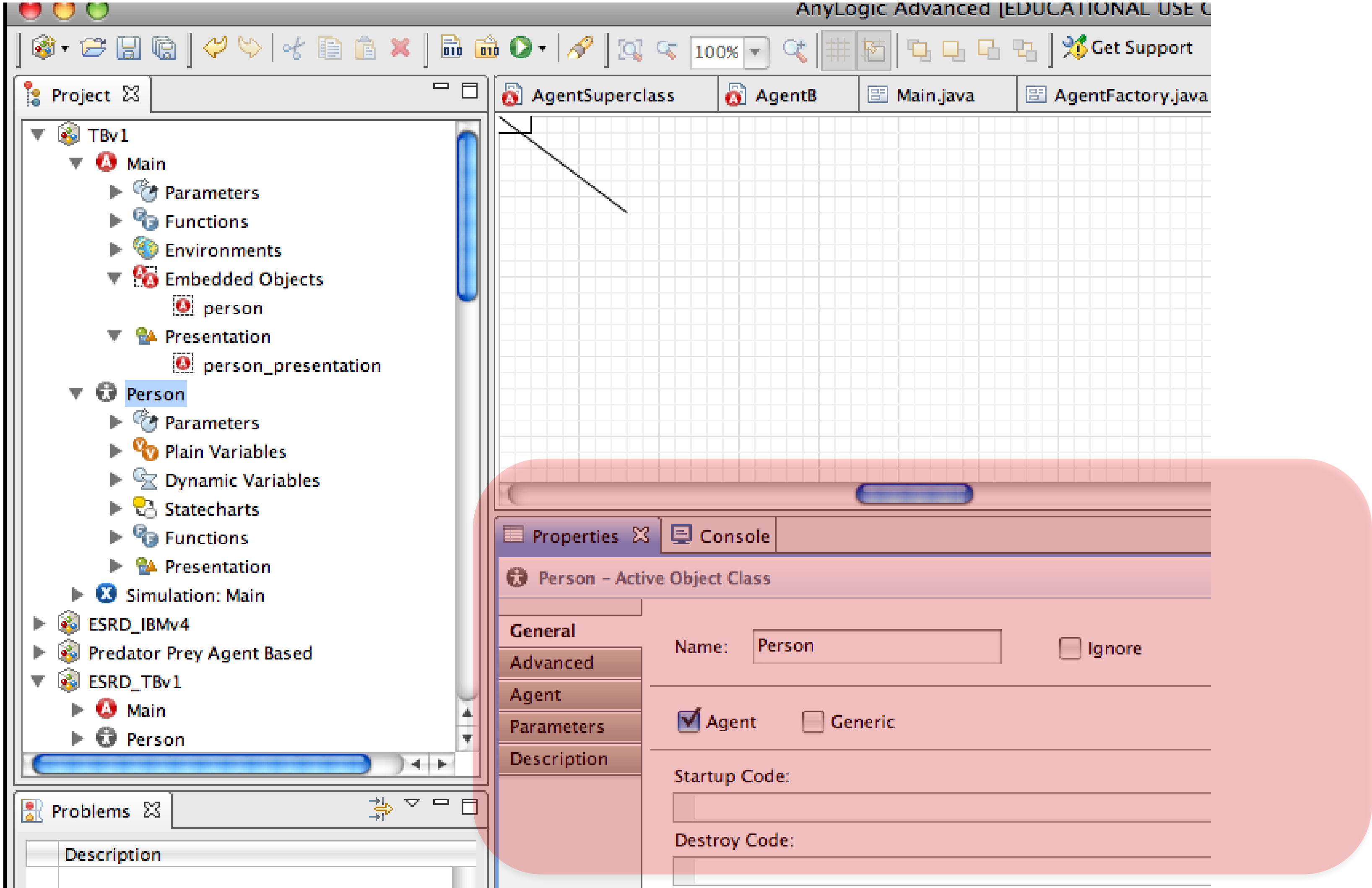
Additional class code:

☑ Persistent Top-level Presentation Group

☑ Persistent Top-level Icon Group

Examples of Where to Insert Code Object Properties

- “General”



Person – Active Object Class

Name: Person ☐ Ignore

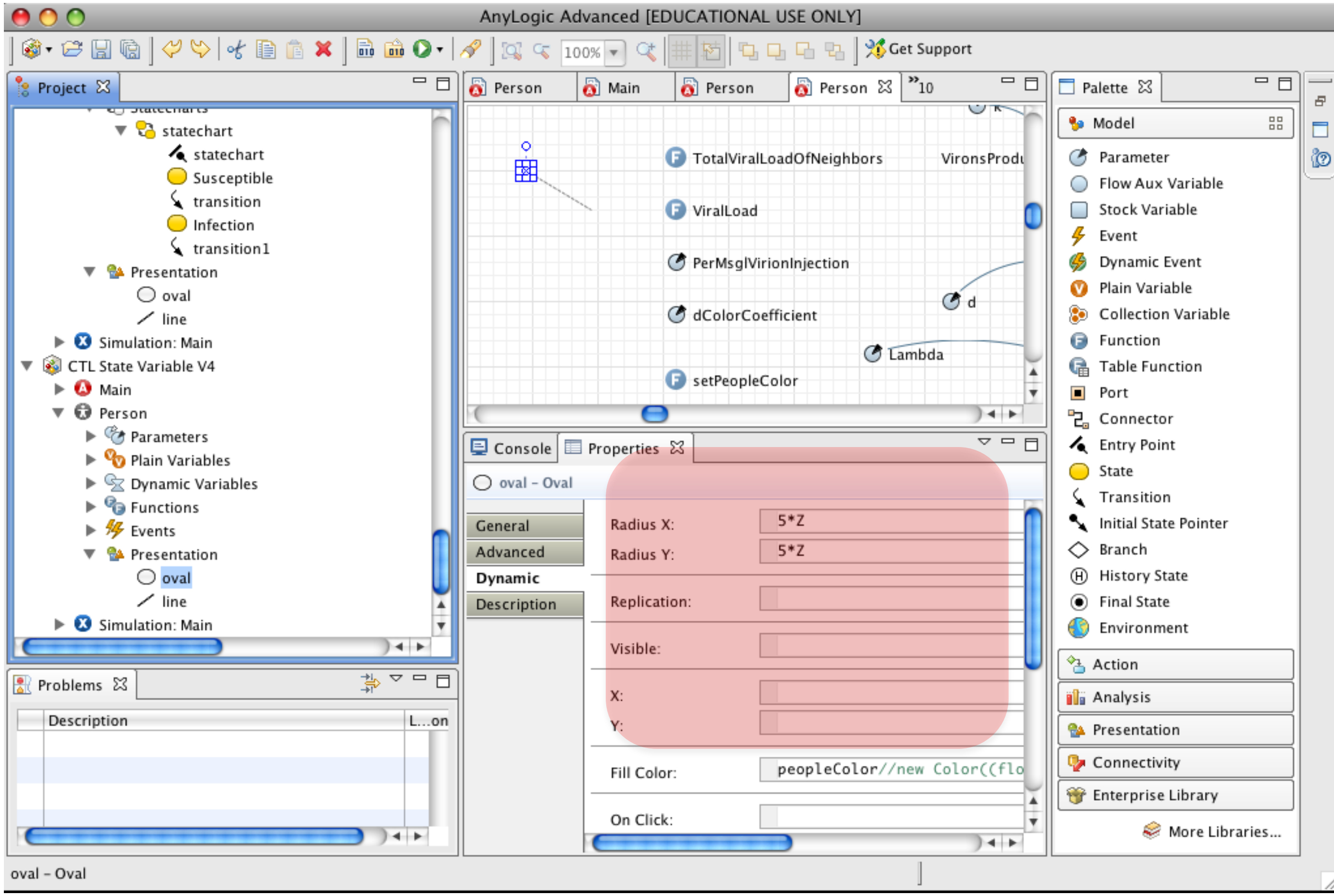
☑ Agent ☐ Generic

Startup Code:

Destroy Code:

Example of Where to Insert Code Presentations Properties

- “Dynamic” properties of presentation elements (especially of Agents)



oval - Oval

Radius X: 5*Z

Radius Y: 5*Z

Replication:

Visible:

X:

Y:

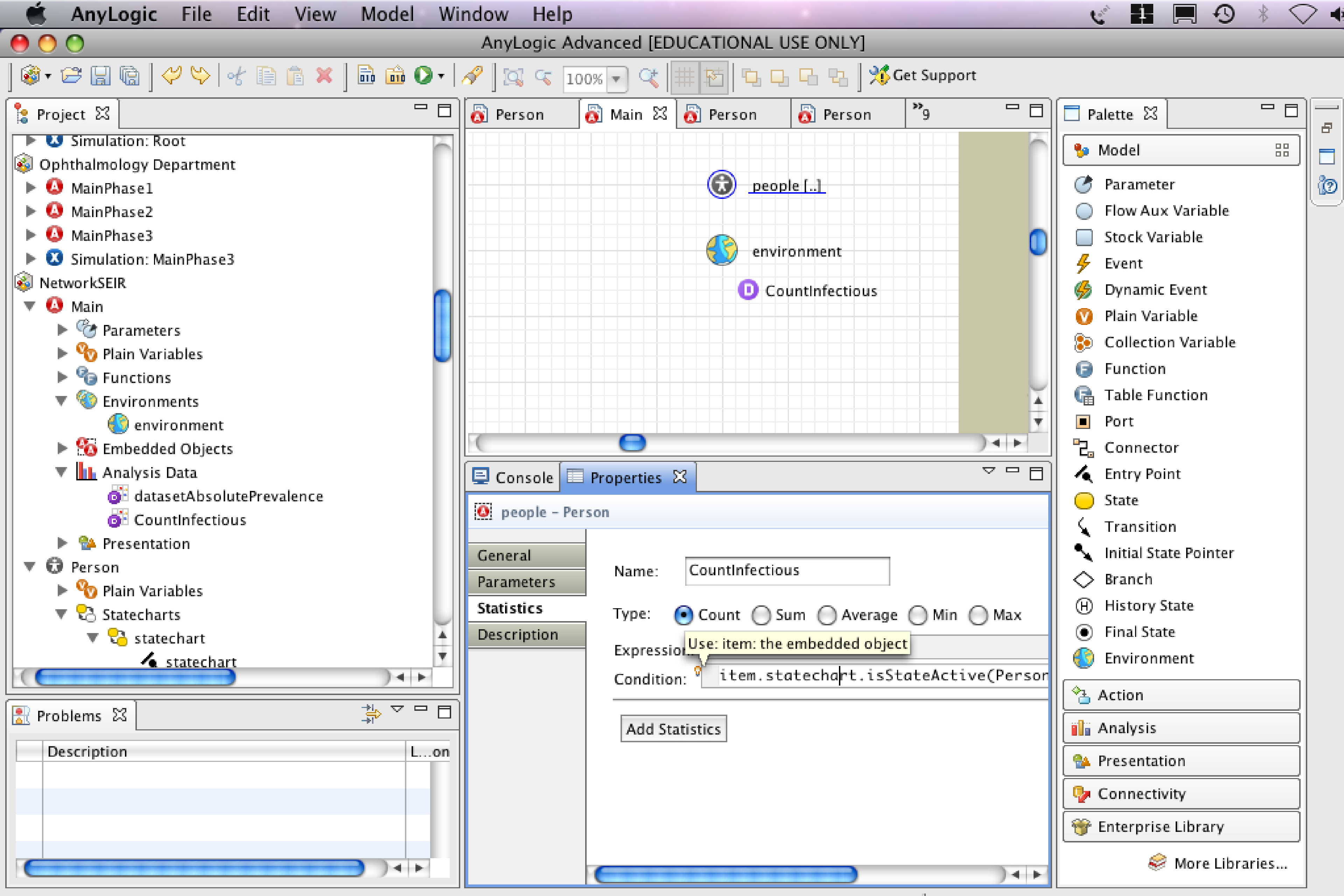
Fill Color: peopleColor//new Color((flo

On Click:

Tips to Bear in Mind While Writing Code

- Click on the “light bulb” next to fields to get contextual advice (e.g. on the variables that are available from context
- While typing code, can hold down the Control key and press the “Space” key to request autocompletion
  - This can help know what parameters are required for a method, etc.
- Java is case sensitive!
- Can press “Control-J” to go to the point in Java code associated with the current code snippet
- Can press “build” button after writing snippet to increase confidence that code is understood

Example of Contextual Information



people – Person

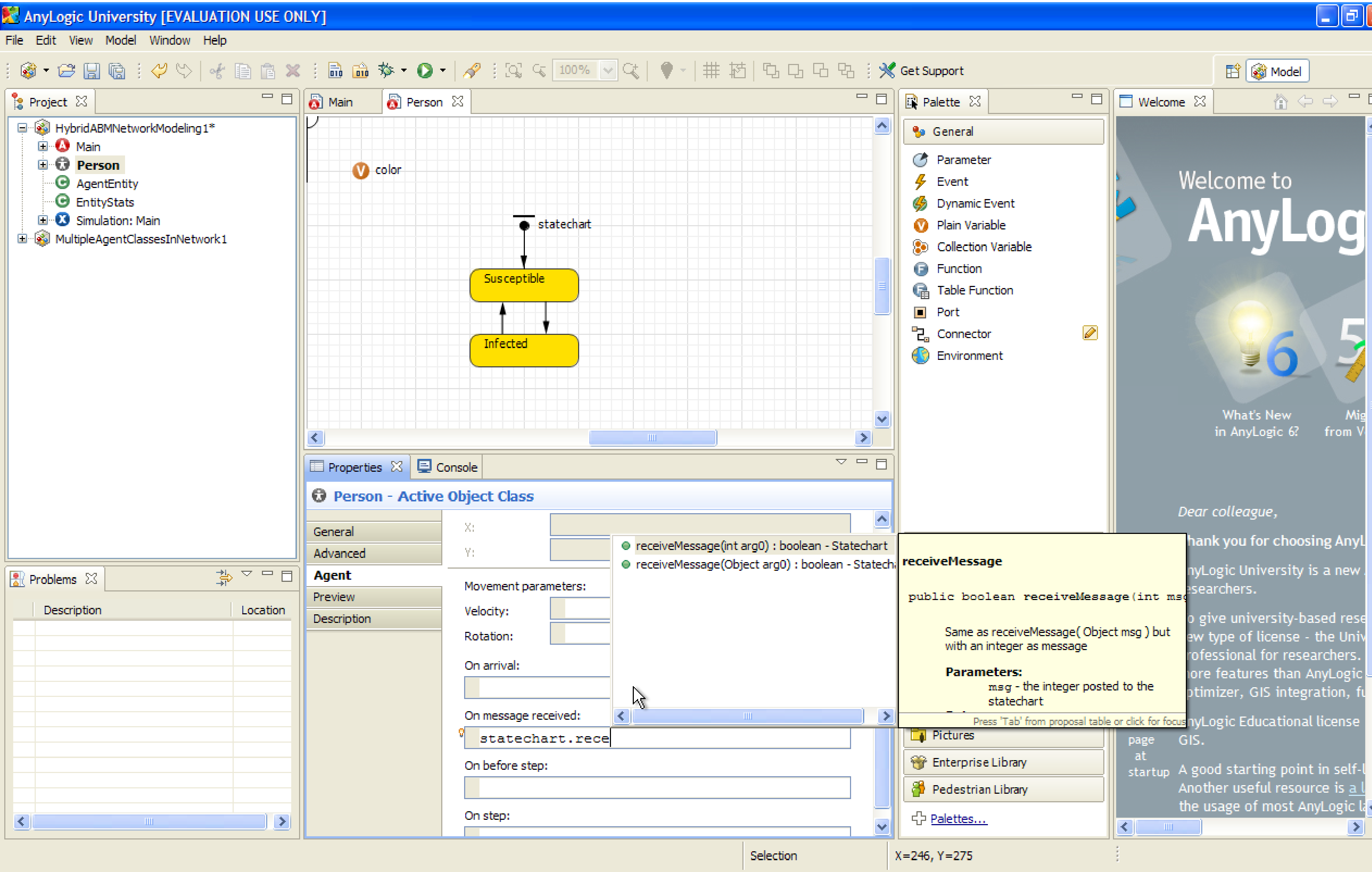
Name: CountInfectious

Type: ● Count ○ Sum ○ Average ○ Min ○ Max

Expression: Use: item: the embedded object

Condition: item.statechart.isStateActive(Person

Autocompletion Info (via Control-Space)



On message received: `statechart.rece`

- receiveMessage(int arg0) : boolean - Statechart
- receiveMessage(Object arg0) : boolean - Statech

receiveMessage

```
public boolean receiveMessage(int msg
```

Same as receiveMessage(Object msg) but with an integer as message

Parameters:

msg - the integer posted to the statechart

Press 'Tab' from proposal table or click for focus

Dear colleague,

hank you for choosing AnyL

nyLogic University is a new
esearchers.

o give university-based rese
ew type of license - the Univ
rofessional for researchers.
ore features than AnyLogic
ptimizer, GIS integration, fu

nyLogic Educational license
GIS.

page at startup

A good starting point in self-
Another useful resource is a l
the usage of most AnyLogic la