

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 content

The screenshot displays the AnyLogic Advanced interface. The main workspace shows a statechart with three states: 'Susceptible' (yellow oval), 'Infectious Contact' (orange rounded rectangle), and 'Recovered' (grey rounded rectangle). Transitions are labeled 'Infection' (from Susceptible to Infectious Contact) and 'Recovery' (from Infectious Contact to Recovered). A 'statechart' label is at the top, and a 'color' variable is shown next to the Susceptible state.

The 'Infection - Transition' configuration panel is open, showing the following settings:

- Name:** Infection
- Show Name:**
- Ignore:**
- Public:**
- Show:**
- Triggered by:** Message
- Message type:** boolean int double String Other
- Class Name:** Object
- Fire transition:** Unconditionally If message equals If expression is true (use msg for message)
- Action:** (empty field)
- Guard:** (empty field)

The left sidebar shows a project tree with folders for 'Functions', 'Presentation', and 'Simulation: Main'. The 'Problems' window at the bottom left contains several error messages: 'Engine.log cannot be resolved'.

Sending a Message

- (Self-transition because remains in state)

AnyLogic Advanced [EDUCATIONAL USE ONLY]

```
stateDiagram-v2
    [*] --> Susceptible
    Susceptible --> Infectious : Infection
    Infectious --> Recovered : Recovery
    Infectious --> Infectious : Contact
```

Console Properties

Contact - Transition

General

Name: Show Name Ignore Public Show

Triggered by: ▼

Rate:

Action:

```
send( "Infection", RANDOM_NEIGHBOR );
```

Guard:

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

The screenshot displays a statechart editor interface. On the left, a project tree shows a simulation named "Spatial SIR with Waning Immunity" with a "Main" package containing parameters like "ContactRate: 1" and "InfectionProbability". The main workspace shows a statechart with three states: "Susceptible" (yellow), "Infectious" (orange), and "Recovered" (grey). A transition labeled "Infection" connects "Susceptible" to "Infectious". The "Infectious" state has a sub-state "Contact" (orange) with a blue circle around it. A transition labeled "Recovery" connects "Infectious" to "Recovered". A "color" variable is shown near the "Susceptible" state. Below the statechart, the "Properties" window is open for the "Contact - Transition". The "General" tab shows the transition name "Contact" and a "Rate" of $\text{get_Main()}.ContactRate / \text{get_Main()}.InfectionProbability$. The "Action" field contains the code `send("Infection", RANDOM_NEIGHBOR);`, where the string "Infection" is highlighted with a pink box. The "Console" window at the bottom shows several error messages: "Engine.log cannot be resolved".

Receiving a Message: Forwarding Messages on to the Statechart

The screenshot displays the AnyLogic Advanced interface. On the left, a project tree shows the hierarchy: environment > Embedded Objects > Analysis Data > Presentation > Person > Plain Variables > color > Statecharts > statechart > statechart > Susceptible. The main workspace shows a statechart with states: Susceptible (yellow), Infectious (orange), and Recovered (grey). Transitions are labeled: Infection (Susceptible to Infectious), InfectionTransmission (Infectious to Recovered), and Recovery (Infectious to Recovered). A 'getDegree' function is connected to the Recovered state. The Properties window for the 'Person - Active Object Class' is open, showing the 'On Message Received' action set to `statechart.receiveMessage(msg);`. A blue arrow points from this code to the 'statechart' object in the diagram, and a red arrow points from the text below to the code.

Person Main Person

statechart

color

Susceptible

Infectious

Recovered

Infection

InfectionTransmission

Recovery

getDegree

Console Properties

Person - Active Object Class

General

Advanced

Agent

Parameters

Description

Environment defines initial location

Initial coordinates:

X:

Y:

Movement parameters:

Velocity:

Rotation:

On Arrival:

On Message Received: `statechart.receiveMessage(msg);`

On Before Step:

On Step:

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...

Problems

Description	Location

Selection

The **action** for Handling received messages delegates to the Statechart object

Receiving a Message

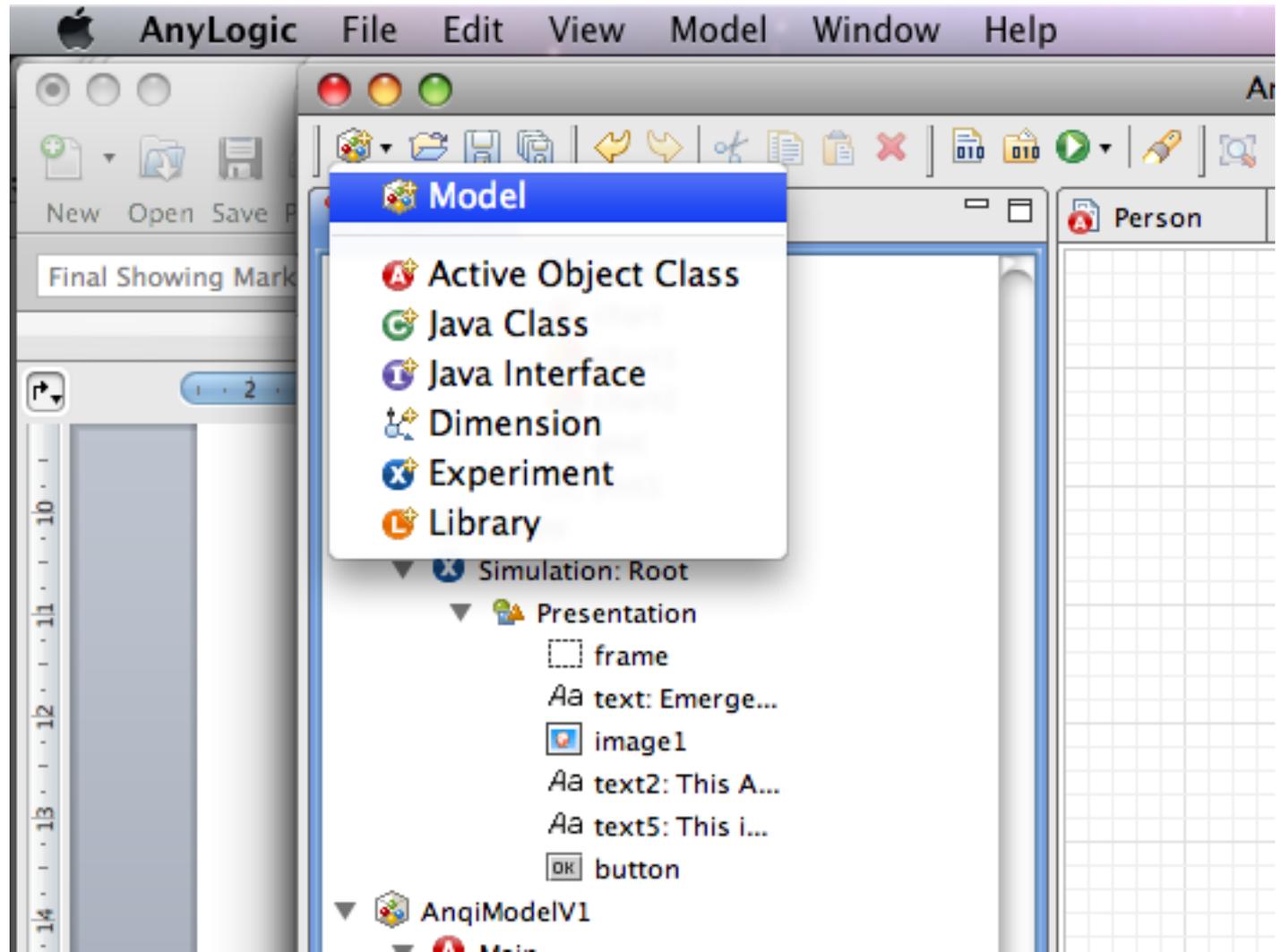
The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace shows a statechart diagram with three states: Susceptible (yellow), Infectious Contact (orange), and Recovered (grey). Transitions are labeled: 'Infection' from Susceptible to Infectious Contact, and 'Recovery' from Infectious Contact to Recovered. A 'statechart' label is at the top, and a 'color' variable is shown on the left. The left sidebar contains a project tree with folders for 'Functions', 'Presentation', and 'Simulation: Main'. The bottom right pane shows the configuration for the 'Infection - Transition'. A red rounded rectangle highlights the 'Triggered by' and 'Fire transition' settings.

Transition Configuration: Infection - Transition

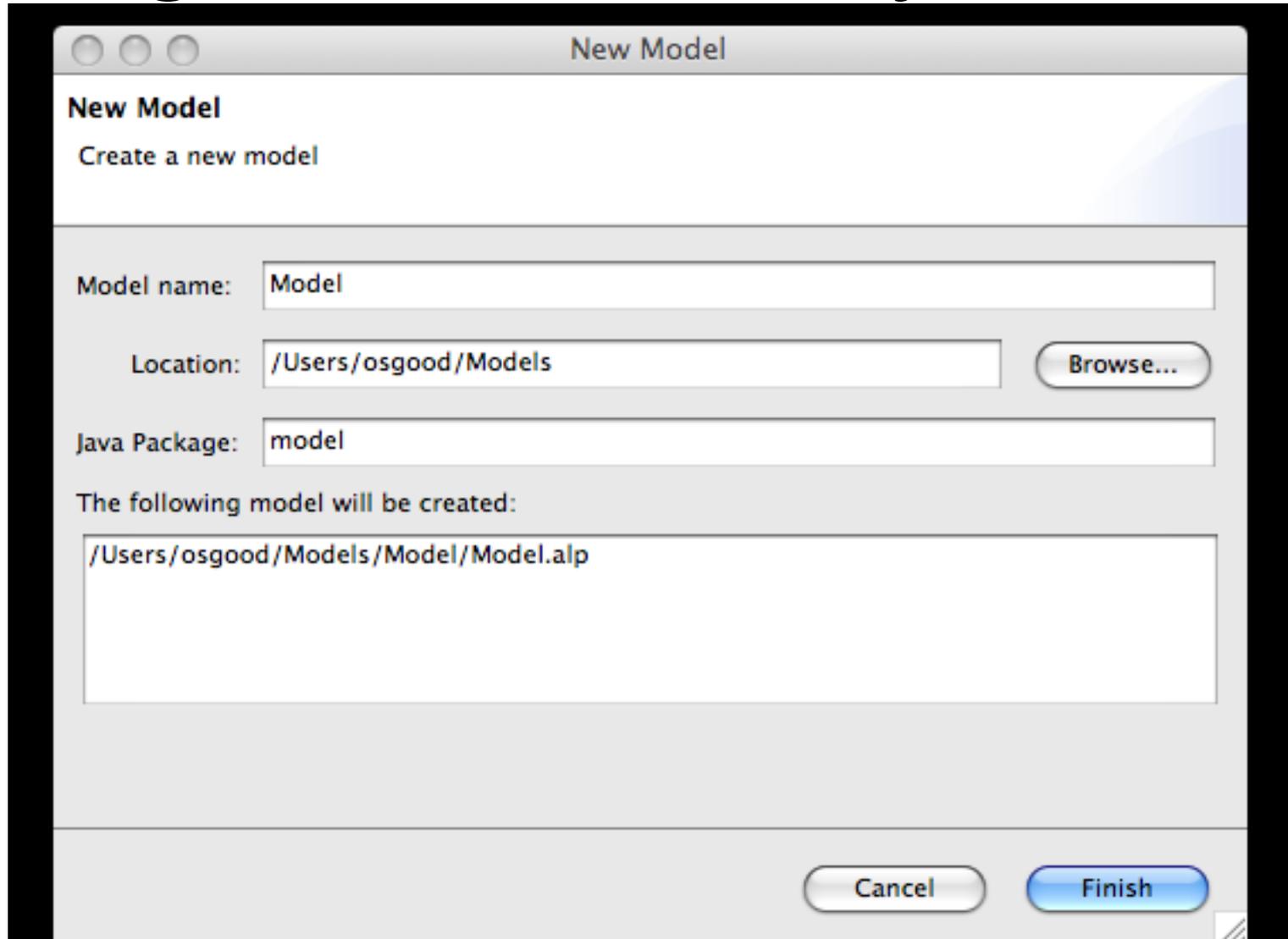
- Name: Infection
- Show Name:
- Ignore:
- Public:
- Show:
- Triggered by: Message
- Message type: boolean int double String Other
- Class Name: Object
- Fire transition: Unconditionally If message equals If expression is true (use msg for message)
- Action:
- Guard:

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

Add a New Model Project



Filling in the Model Project Details



New Model
Create a new model

Model name:

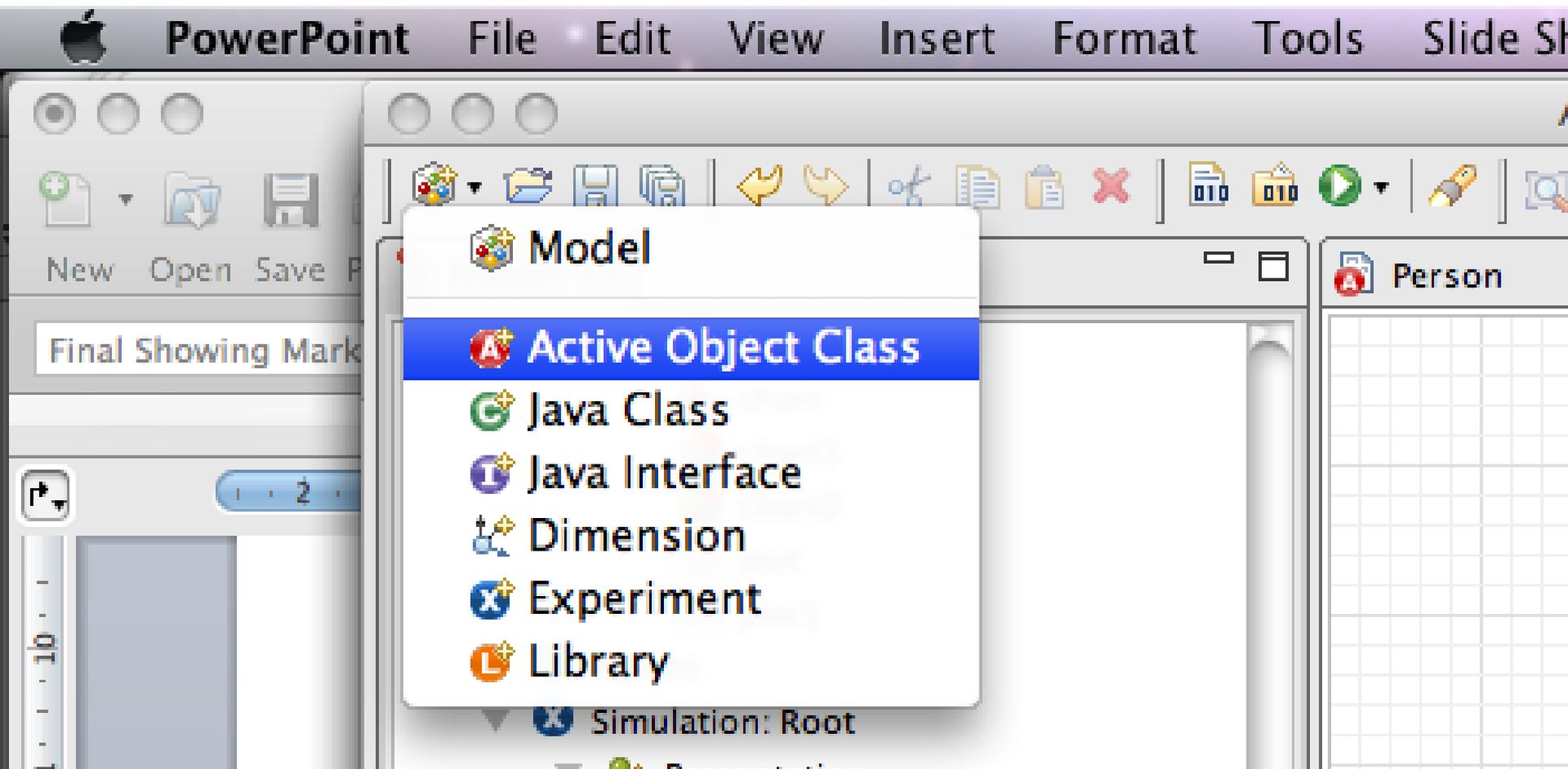
Location:

Java Package:

The following model will be created:

```
/Users/osgood/Models/Model/Model.alp
```

Add an Active Object Class



Filling in the Agent Class Details

New Active Object Class

Active Object Class

Name:

Description:

Cancel Finish

The Updated Project

The screenshot displays a software development environment with the following components:

- Project Tree:** A hierarchical view of the project structure. The 'Person' class under the 'TestModel2*' package is selected and highlighted in blue.
- Problems Window:** A list of error messages, all starting with a red 'X' icon. The errors include:
 - Engine.log cannot be resolved
 - Cannot make a static reference to the non-static field
 - The method getCurrentState() is undefined for the type
 - Type_statechart cannot be resolved
 - The method setModified() is undefined for the type
- Console and Properties Window:** The 'Person - Active Object Class' is selected. The 'Agent' tab is active, showing a message: "Agent properties are only available if the ActiveO Check the Agent checkbox on General tab to con".

Declaring “Person” as an Agent

The screenshot displays a software development environment with several panels:

- Left Panel (Project Explorer):** Shows a tree view of the project structure. The 'Person' class is highlighted under the 'Main' folder.
- Bottom-Left Panel (Problems):** Lists several error messages, including 'Engine.log cannot be resolved' and 'The method setModified() is undefined for the t'.
- Right Panel (Properties):** Shows the 'Person - Active Object Class' properties. The 'Agent' checkbox is checked, and the 'Name' field is set to 'Person'. A red arrow points to the 'Agent' checkbox with the text 'Check this box!'.
- Bottom-Right Panel (Action Bar):** Contains buttons for 'Action', 'Analysis', 'Presentati...', 'Connectiv...', 'Enterpris...', and 'More Libraries...'.

Updated Result

The screenshot displays the AnyLogic Advanced software interface. The main window is titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The interface is divided into several panes:

- Project Tree (Left):** Shows a hierarchical view of the project. The "Person" object is selected, and its icon is highlighted with a red arrow. The icon is a circle with a cross inside.
- Diagram Canvas (Center):** Shows a grid with a single "person" object placed on it. The object's icon is a circle with a cross inside.
- Properties Panel (Bottom Right):** Shows the "Person - Active Object Class" properties. The "Name" field is set to "Person". The "Agent" checkbox is checked, and the "Generic" checkbox is unchecked.
- Model Library (Right):** Shows a list of model components, including "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", and "Environm...".
- Problems Panel (Bottom Left):** Shows a list of errors, all of which are "Engine.log cannot be resolved".

A red arrow points from the text "Note changed icon" to the "Person" object in the project tree. The text "Note changed icon" is written in red.

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

The screenshot shows the AnyLogic Advanced software interface. The main window displays a grid with a red circle and arrow pointing to the origin (0,0). The text overlay reads: "This is the grid origin. To be centred on their spatial location, the Person's presentation items should be placed here." The interface includes a Project tree on the left, a Console and Properties panel at the bottom, and a toolbar at the top. The Properties panel shows the "Person" object class with fields for Name, Agent, and Startup/Destroy Codes.

Project

- Simulation: Main
 - Presentation
- MalariaV2
 - Main
 - Parameters
 - Functions
 - CountInfectiveHumans
 - CountInfectiveMosquitoes
 - PersistSimulationData
 - SelectRandomPerson
 - SetParameters
 - SetSimulationOutputFilename
 - getHumanPopulation
 - getMosquitoPopulation
 - Events
 - Embedded Objects
 - Analysis Data
 - Presentation
 - Mosquito
 - Person
 - Simulation: Main
- TestModel2*
 - Main
 - Person
 - Simulation: Main

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...

Person - Active Object Class

General

Name: Person Ignore

Agent

Agent Generic

Description

Startup Code:

Destroy Code:

Description	Location

Selection | Cursor: X=0, Y=0

Create an Oval at the Origin (Cross-Hairs)

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The interface is divided into several panels:

- Project Panel (Left):** Shows a hierarchical tree structure. The "Main" function is selected under the "MalariaV2" project.
- Canvas (Center):** A grid-based workspace where a blue oval is being created at the origin (0,0). The oval is surrounded by a cross-hair cursor.
- Properties Panel (Bottom Right):** Shows the properties for the selected "oval - Oval". The "Position" is set to X: 0 and Y: 0. The "Radius X" and "Radius Y" are both set to 10. The "Rotation" is set to 0.0. The "Persistent (enable programmatic control)" checkbox is checked.
- Toolbox (Right):** A vertical toolbar containing various shapes and components, including Line, Polyline, Curve, Rectangle, Round Re..., Oval, Arc, Pixel, Text, Image, Group, Button, Check Box, Edit Box, Radio But..., Slider, Combo Box, List Box, File Choo..., Progress ..., CAD Dra..., GIS Map, Connectiv..., and Enterpris....
- Console and Problems Panels (Bottom Left):** The Console panel is empty, and the Problems panel shows a table with columns for "Description" and "Location".

Description	Location

From the Centre of the Oval, Draw a Line

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a diagram with a pink oval and a blue line extending downwards from its center. A red arrow points to the center of the oval, and red text overlaid on the image reads: "The '+' on the end of the line should be at the centre of the oval".

The interface includes a Project tree on the left, a Properties panel at the bottom, and a toolbar on the right. The Properties panel for the selected 'line - Line' object shows the following settings:

Category	Property	Value
General	Position X	0
General	dX	0
Dynamic	Y	0
Dynamic	dY	100
Description	Persistent (enable programmatic control)	<input checked="" type="checkbox"/>

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

The screenshot shows the AnyLogic Advanced software interface. On the left, a project tree shows a simulation named 'MalariaV2' with a 'Main' component. The 'Main' component is selected, and its 'Dynamic' property tab is active in the Properties window. The 'Replication' field is set to 'getConnectionsNumber()'. A blue arrow points from the text 'Make sure you have selected the line by clicking on it!' to a line object in the main workspace. A green arrow points from the text 'Make sure you have selected the "Dynamic" tab!' to the 'Dynamic' tab in the Properties window.

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

Make sure you have selected the "Dynamic" tab!

line - Line

Property	Value
Replication:	getConnectionsNumber()
Visible:	<input type="checkbox"/>
X:	<input type="text"/>
Y:	<input type="text"/>
On Click:	<input type="text"/>
Rotation:	<input type="text"/>
Scale X:	<input type="text"/>
Scale Y:	<input type="text"/>
dX:	<input type="text"/>
dY:	<input type="text"/>

Also set the “dX” and “dY” properties

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a simulation model with a vertical line and a circle at the top. The Properties panel is open, showing the configuration for a 'line - Line' object. The 'dX' and 'dY' properties are highlighted in red, indicating they are being set to dynamic expressions.

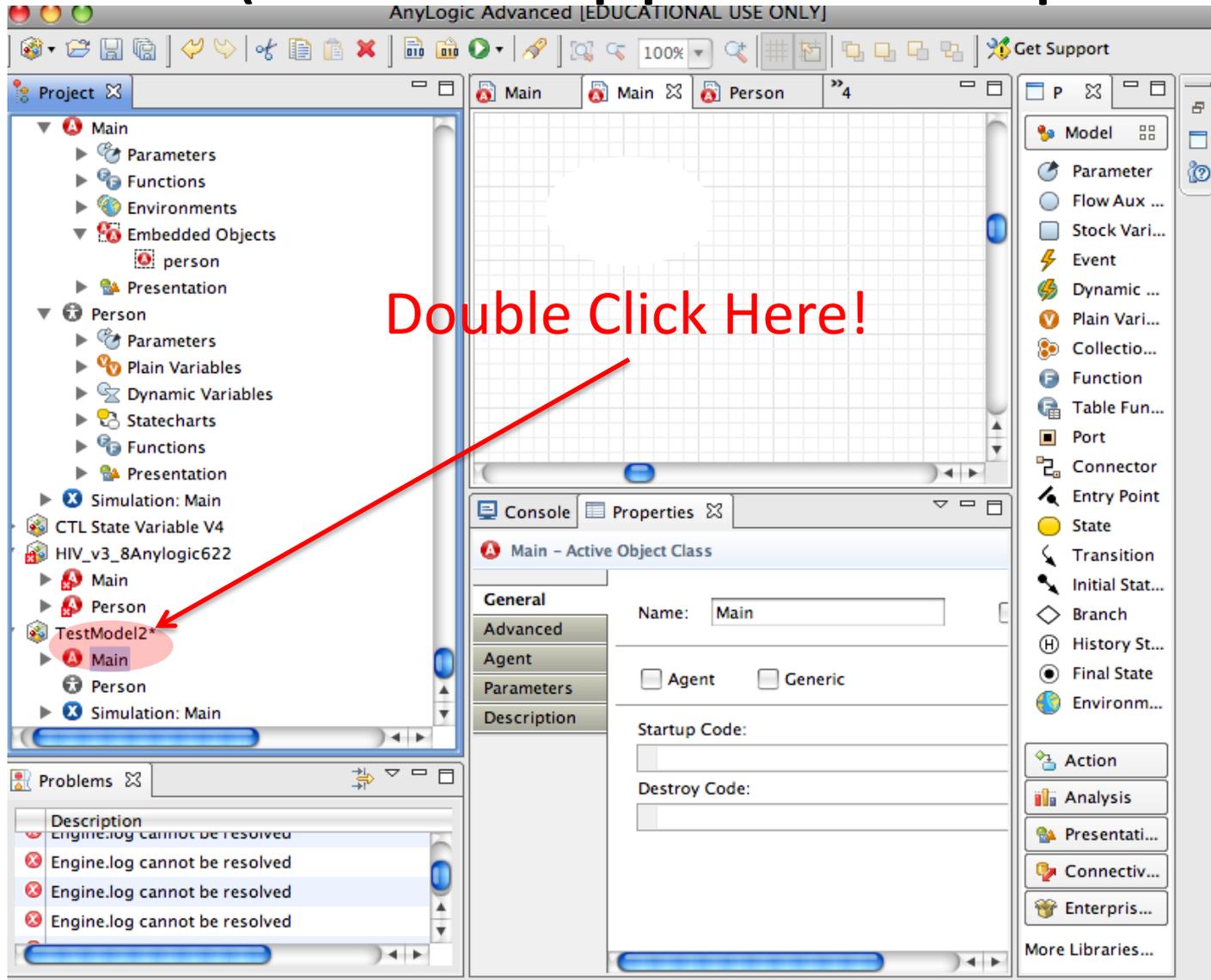
Project Structure:

- button5
- chart
- chart1
- chart2
- plot
- plot1
- Patient
- Simulation: Root
 - Presentation
 - frame
 - Aa text: Emerge...
 - image1
 - Aa text2: This A...
 - Aa text5: This i...
 - button
 - AnqiModelV1
 - Main
 - Parameters
 - AveragellnessDura
 - ContactRatePerNetw
 - CountInitialInfected
 - ImmunityDuration:
 - MeanNetworkConta
 - PerContactInfection
 - TotalPopulation: 10
 - Plain Variables

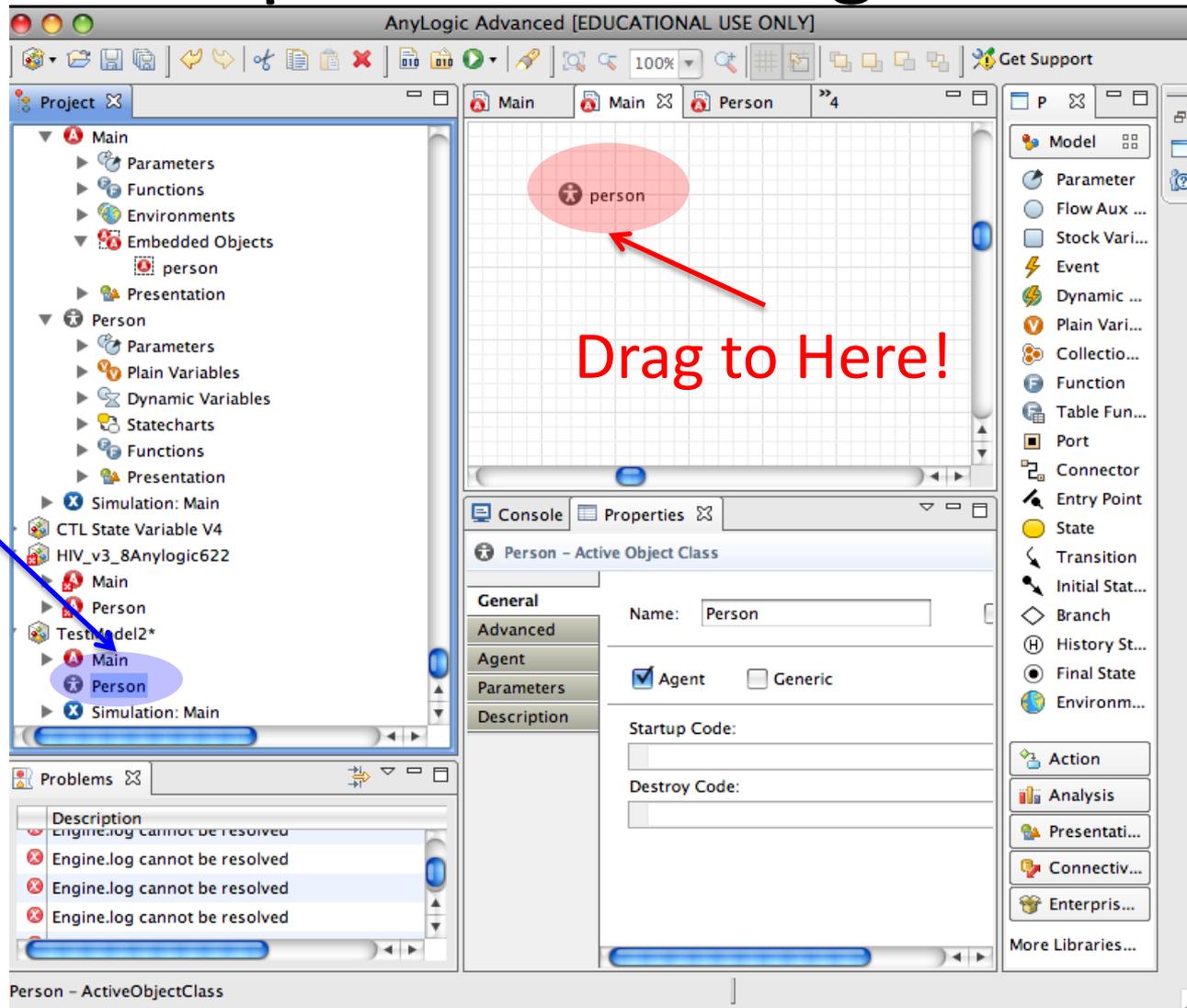
Properties Panel (line - Line):

Property	Value
Replication:	getConnectionsNumber()
Visible:	<input type="checkbox"/>
X:	<input type="text"/>
Y:	<input type="text"/>
On Click:	<input type="text"/>
Rotation:	<input type="text"/>
Scale X:	<input type="text"/>
Scale Y:	<input type="text"/>
dX:	getConnectedAgent(index).getX() - getX()
dY:	getConnectedAgent(index).getY() - getY()

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



Drag from Here!

Drag to Here!

Set the Count of Agents in the Agent Population

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The interface is divided into several panels:

- Project Explorer (Left):** Shows a hierarchical tree structure. The "Person" agent is selected under the "Main" environment.
- Diagram Canvas (Center):** Shows a grid with a single "person" agent icon placed on it.
- Properties Panel (Bottom Right):** Displays the configuration for the selected "person" agent. The "Replication" field is set to "100", which is highlighted with a red oval.
- Console (Bottom Center):** Shows error messages: "Engine.log cannot be resolved".
- Model Library (Right):** Lists various model components such as 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, and Environm.

For Clarity, Rename “Person” to “Population”

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The interface is divided into several panels:

- Project Panel (Left):** Shows a hierarchical tree structure. Under "AnqiModelV1" > "Main", there is a "Parameters" folder containing several parameters: "AveragellnessDuration: 10", "ContactRatePerNetworkContact: 1", "CountInitialInfectedAgents: 10", "ImmunityDuration: 50.0", "MeanNetworkContactsPerPerson: 3", "PerContactInfectionProbability: 0.01", and "TotalPopulation: 1000".
- Main Canvas (Center):** A grid-based workspace showing a single agent labeled "person [..]" with a plus sign icon.
- Properties Panel (Bottom Right):** Displays the properties for the selected "person - Person" agent. The "Name" field is highlighted with a red circle and contains the text "Population". Other fields include "Type: Person", "Package: testmodel2", "Environment:", and "Replication: 100".
- Model Library (Right):** A vertical list of model components including 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, and Environm....
- Console and Problems Panels (Bottom):** The Console panel shows "person - Person". The Problems panel is empty.

Add an Environment

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace shows a grid with a 'Population [..]' element and an 'environment' element. A red arrow points from the text 'Drag to Here!' to the 'environment' element. A blue arrow points from the text 'Drag from Here!' to the 'Environment...' option in the right-hand 'Model' library. The 'Properties' window for the 'Population - Person' element is open, showing fields for Name, Type, Package, Environment, and Replication. The 'Environment' field is currently empty. The 'Project' browser on the left shows a tree structure with folders like 'Simulation: Main', 'MalariaV2', 'Main', 'Parameters', 'Functions', 'Events', 'Embedded Objects', 'Analysis Data', 'Presentation', 'Mosquito', 'Person', and 'TestModel2*'. The 'Problems' window at the bottom left is empty.

Drag to Here!

Drag from Here!

environment

Population [..]

Population - Person

General

Name: Population Show Name

Parameters

Statistics

Description

Type: Person

Package: testmodel2

Environment:

Replication: 100

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...

Edit element's name

Set the Network Type to Use

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a grid with a globe icon labeled "environment" and a plus sign icon labeled "Population [..]". The left sidebar contains a project tree with folders like "Simulation: Main", "MalariaV2", "Main", "Parameters", "Functions", "Events", "Embedded Objects", "Analysis Data", "Presentation", "Mosquito", "Person", "Simulation: Main", and "TestModel2*". The bottom-left pane shows a "Problems" section with a "Description" table.

The "environment - Environment" properties panel is open, showing the following settings:

- Space type: Continuous Discrete GIS
- Width: 500
- Height: 500
- Columns: 100
- Rows: 100
- Neighborhood type: Moore
- Layout type: User-defined Apply on startup
- Network: **Distance based** Apply on startup
- Connections per agent: 2
- Connection range: 75
- Neighbor link fraction: 0.95
- M: 10

The "Network" dropdown menu is highlighted with a red circle, indicating the selected network type.

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

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The interface is divided into several panels:

- Project Tree (Left):** Shows a hierarchical view of the project. The "MalariaV2" project is expanded to show "Main", which contains "Parameters" and "Functions". The "Functions" list includes: CountInfectiveHumans, CountInfectiveMosquitoes, PersistSimulationData, SelectRandomPerson, SetParameters, SetSimulationOutputFilename, getHumanPopulation, and getMosquitoPopulation. Other projects like "Simulation: Main" and "TestModel2*" are also visible.
- Diagram (Center):** A grid-based workspace showing a diagram. A vertical line connects a circle at the top to a "Population" object (represented by a person icon). Below the "Population" object is an "environment" object (represented by a globe icon).
- Properties Panel (Bottom Right):** The "Population - Person" properties panel is open. It has tabs for "General", "Parameters", "Statistics", and "Description". The "General" tab is active, showing:
 - Name: Population
 - Type: Person
 - Package: testmodel2
 - Environment: environment (highlighted with a red oval)
- Toolbox (Far Right):** A vertical toolbar containing various modeling elements such as Model, Action, Analysis, and various shapes like Line, Polyline, Curve, Rectangle, Round Re..., Oval, Arc, Pixel, Text, Image, Group, Button, Check Box, Edit Box, Radio But..., Slider, Combo Box, List Box, and File Choo...

Try Running the Model!

The screenshot displays the AnyLogic simulation environment. The title bar reads "TestModel2 : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]". The interface includes a toolbar with standard simulation controls (play, stop, reset, etc.) and a zoom level of "x1". The main workspace shows a complex network diagram with numerous nodes (circles) and connecting lines. In the top-left corner of the workspace, there are two agent icons: "Population Person [100]" and "environment 100 agents". The bottom status bar provides the following information: "Run: 0" (with a green play icon), "Running" (with a green play icon), "Time: 20.85", "Simulation: 21%" (with a blue progress bar), "Memory: 7M of 63M" (with a blue progress bar), and "20.9 sec".

Adding "Color" Variable

The screenshot shows the AnyLogic Advanced software interface. The main workspace displays a variable named 'color' being added to the model. The properties panel for this variable is visible at the bottom, showing the name 'color', access set to 'public', and type set to 'Color'. The initial value is set to 'BLACK'. Annotations include a red oval around the variable name in the workspace and a blue oval around the 'Color' type in the properties panel.

This is the name of a Java class!

Make sure this is in lower case!

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

Make Oval “Color” property Use Variable

Make sure you have selected the “Dynamic” tab!

The screenshot shows the AnyLogic Advanced software interface. The main workspace displays a diagram with a blue oval and a line connecting it to a 'Person' object. A blue arrow points to the oval with the text 'Make sure you have selected the Oval by clicking on it!'. The left sidebar shows a project tree with 'MalariaV2' selected, and the 'Dynamic' tab is highlighted in green. The bottom right shows the 'Properties' panel for the selected oval, with the 'Fill Color' property set to 'color' and circled in red. The status bar at the bottom indicates 'Selection' and 'Cursor: X=86, Y=122'.

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

Make sure you have selected the “Dynamic” tab!

oval - Oval

General	Radius X:	<input type="text"/>
Advanced	Radius Y:	<input type="text"/>
Dynamic	Replication:	<input type="text"/>
Description	Visible:	<input type="text"/>
	X:	<input type="text"/>
	Y:	<input type="text"/>
	Fill Color:	color
	On Click:	<input type="text"/>
	Rotation:	<input type="text"/>
	Scale X:	<input type="text"/>

Selection | Cursor: X=86, Y=122

Add Entry Point of State chart

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main workspace shows a statechart diagram on a grid. A red oval highlights a newly added entry point for the statechart, labeled "InfectionStatechart". To the left of the diagram, a variable named "color" is visible. The interface includes a project tree on the left, a console and properties panel at the bottom, and a toolbar on the right. The properties panel for "InfectionStatechart - Statechart Entry Point" shows the following settings:

- General: Name: InfectionStatechart, Show Name, Ignore, Public, Show At R
- Description: Action: (empty field)

The Problems panel at the bottom left shows several error messages:

- Element doesn't belong to any state
- Hanging statechart entry

The status bar at the bottom indicates "Selection" and "Cursor: X=186, Y=63".

Add in “Susceptible” State

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a statechart diagram for an 'InfectionStatechart' object. A new state, labeled 'Susceptible', is being added to the diagram. The state is represented by a yellow oval with a red border, and it is currently highlighted with a red circular selection tool. The diagram also includes a 'color' parameter and a 'Person' object. The left sidebar shows a project tree with various components like 'MalariaV2', 'Person', and 'Simulation: Main'. The bottom panel shows the 'InfectionStatechart - Statechart Entry Point' properties, including a 'Name' field set to 'InfectionStatechart' and checkboxes for 'Show Name', 'Ignore', 'Public', and 'Show At R'. The bottom status bar indicates the cursor position at X=151, Y=127.

AnyLogic Advanced [EDUCATIONAL USE ONLY]

Project: MalariaV2

- Main
 - Parameters
 - Functions
 - CountInfectiveHum...
 - CountInfectiveMosc...
 - PersistSimulationDa...
 - SelectRandomPerso...
 - SetParameters
 - SetSimulationOutpu...
 - getHumanPopulatio...
 - getMosquitoPopulat...
 - Events
 - Embedded Objects
 - Analysis Data
 - Presentation
 - Mosquito
 - Person
 - Simulation: Main
- TestModel2*
 - Main
 - Plain Variables
 - Statecharts
 - InfectionStatechart
 - Unresolved
 - Presentation
 - Simulation: Main
- Spatial SIR with Waning Immunity
 - Main
 - Person
 - Simulation: Main
- Network Modification of SIR AB
 - Main

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...

Console | Properties

InfectionStatechart - Statechart Entry Point

General

Name: InfectionStatechart Show Name Ignore Public Show At R

Description

Action:

Problems

- Element doesn't belong to any state
- Element doesn't belong to any state
- Element doesn't belong to any state
- Hanging transition
- Element doesn't belong to any state
- Hanging transition
- Hanging statechart entry

Selection | Cursor: X=151, Y=127

Connect with Entry Point

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

AnyLogic Advanced [EDUCATIONAL USE ONLY]

Project: MalariaV2

- Main
- Parameters
- Functions
 - CountInfectiveHum...
 - CountInfectiveMosc...
 - PersistsSimulationDa...
 - SelectRandomPerso...
 - SetParameters
 - SetSimulationOutpu...
 - getHumanPopulatio...
 - getMosquitoPopulat...
- Events
- Embedded Objects
- Analysis Data
- Presentation
- Mosquito
- Person
- Simulation: Main
- TestModel2*
 - Main
 - Person
 - Plain Variables
 - Statecharts
 - InfectionStatechart
 - Unresolved
 - Presentation
 - Simulation: Main
- Spatial SIR with Waning Immunity
 - Main
 - Person
 - Simulation: Main
- Network Modification of SIR AB
 - Main

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...

Properties: InfectionStatechart - Statechart Entry Point

General

Name: InfectionStatechart Show Name Ignore Public Show At R...

Description

Action:

Problems

- Element doesn't belong to any state
- Element doesn't belong to any state
- Hanging transition
- Element doesn't belong to any state
- Hanging transition

Selection | Cursor: X=174, Y=130

Fill In Code to Color Green when Enter State

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a statechart diagram with a state named "Susceptible" highlighted in yellow. A variable named "color" is also visible. The "Properties" panel for the "Susceptible" state is open, showing the "Entry Action" field with the code "color=GREEN" entered. The "Fill Color" is set to "Default". The "Problems" panel at the bottom left shows several error messages related to state transitions.

Statechart Diagram:

- Variable: `color`
- State: `Susceptible` (highlighted in yellow)
- Transition: `InfectionStatechart`

Properties Panel (Susceptible - State):

- Name: `Susceptible`
- Show Name:
- Ignore:
- Public:
- Show At Run:
- Fill Color: `Default`
- Entry Action: `color=GREEN`
- Exit Action: (empty)

Problems Panel:

- Element doesn't belong to any state
- Element doesn't belong to any state
- Hanging transition
- Element doesn't belong to any state
- Hanging transition

Adding in “Infective” State

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a state transition diagram with a yellow state labeled "Susceptible" and a red state labeled "Infective". A transition arrow points from "Susceptible" to "Infective". The "Infective" state is highlighted with a red circle. The interface includes a project tree on the left, a toolbar at the top, and a properties window at the bottom. The properties window for the "Infective - State" shows the following details:

Section	Property	Value
General	Name	Infective
	Options	<input checked="" type="checkbox"/> Show Name, <input type="checkbox"/> Ignore, <input type="checkbox"/> Public, <input checked="" type="checkbox"/> Show At Runtime
Description	Fill Color	Default
	Entry Action	
	Exit Action	

The Problems window at the bottom left shows several error messages:

- Element doesn't belong to any state
- Element doesn't belong to any state
- Hanging transition
- Element doesn't belong to any state
- Hanging transition

Adding Transition

The screenshot displays the AnyLogic Advanced interface for an educational simulation. The main workspace shows a statechart diagram with two states: 'Susceptible' and 'Infected', both represented by yellow rounded rectangles. A transition arrow points from 'Susceptible' to 'Infected'. A red arrow points from the 'infection' transition name in the Properties panel to the transition arrow in the diagram. The Properties panel for the 'Infection - Transition' shows the following settings:

- Name: infection
- Triggered by: Rate
- Rate: 0.01
- Action: (empty)
- Guard: (empty)

The Properties panel also includes checkboxes for 'Show Name', 'Ignore', 'Public', and 'Show At Run Time' (checked). The left sidebar shows a project tree with 'MalariaV2' and 'TestModel2*' folders. The bottom-left 'Problems' panel shows two error messages: 'Element doesn't belong to any state' and 'Hanging transition'. The right sidebar contains a 'Model' palette with various components like Parameter, Flow Aux, Stock Vari, Event, Dynamic, Plain Vari, Collectio, Function, Table Fun, Port, Connector, Entry Point, State, Transition, Initial Stat, Branch, Initial S, Final State, and Environm.

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

Adding Infection Clearance Transition

The screenshot displays the AnyLogic Advanced software interface for modeling a malaria simulation. The main workspace shows a state transition diagram with two states: 'Susceptible' and 'Infected', both represented by yellow rounded rectangles. A transition labeled 'InfectionClearance' is shown as a black arrow pointing from the 'Infected' state to the 'Susceptible' state. A parameter 'color' is also visible in the workspace. The 'InfectionClearance' transition is highlighted with a red circle in the workspace and its properties are shown in the Properties panel below. The Properties panel for 'InfectionClearance - Transition' includes the following fields:

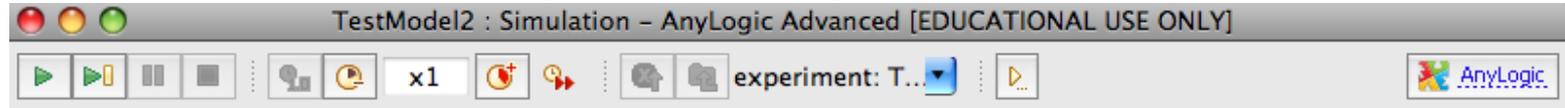
- Name: InfectionClearance
- Triggered by: Rate
- Rate: 0.1
- Action: (empty field)
- Guard: (empty field)

The Properties panel also shows checkboxes for 'Show Name', 'Ignore', 'Public', and 'Show At Run Time' (checked). The left sidebar shows a project tree with folders for 'MalariaV2', 'TestModel2*', and 'Spatial SIR with Waning Immunity'. The bottom status bar indicates 'Selection' and 'Cursor: X=100, Y=207'.

Run the Model!

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main workspace shows a statechart diagram with two states: "Susceptible" and "Infected", both represented by yellow rounded rectangles. A transition labeled "InfectionS" connects the "Susceptible" state to the "Infected" state. A variable "color" is shown in the center of the workspace. The left sidebar contains a project tree for "MalariaV2", with sub-projects like "Main", "Person", and "Simulation: Main". The bottom panel shows the "InfectionClearance - Transition" configuration window, with fields for Name, Description, Triggered by (Rate), Rate (0.1), Action, and Guard. The right sidebar contains a "Model" palette with various components like 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, and Environm. The bottom status bar shows "Selection".

Completing Set-Up



TestModel2

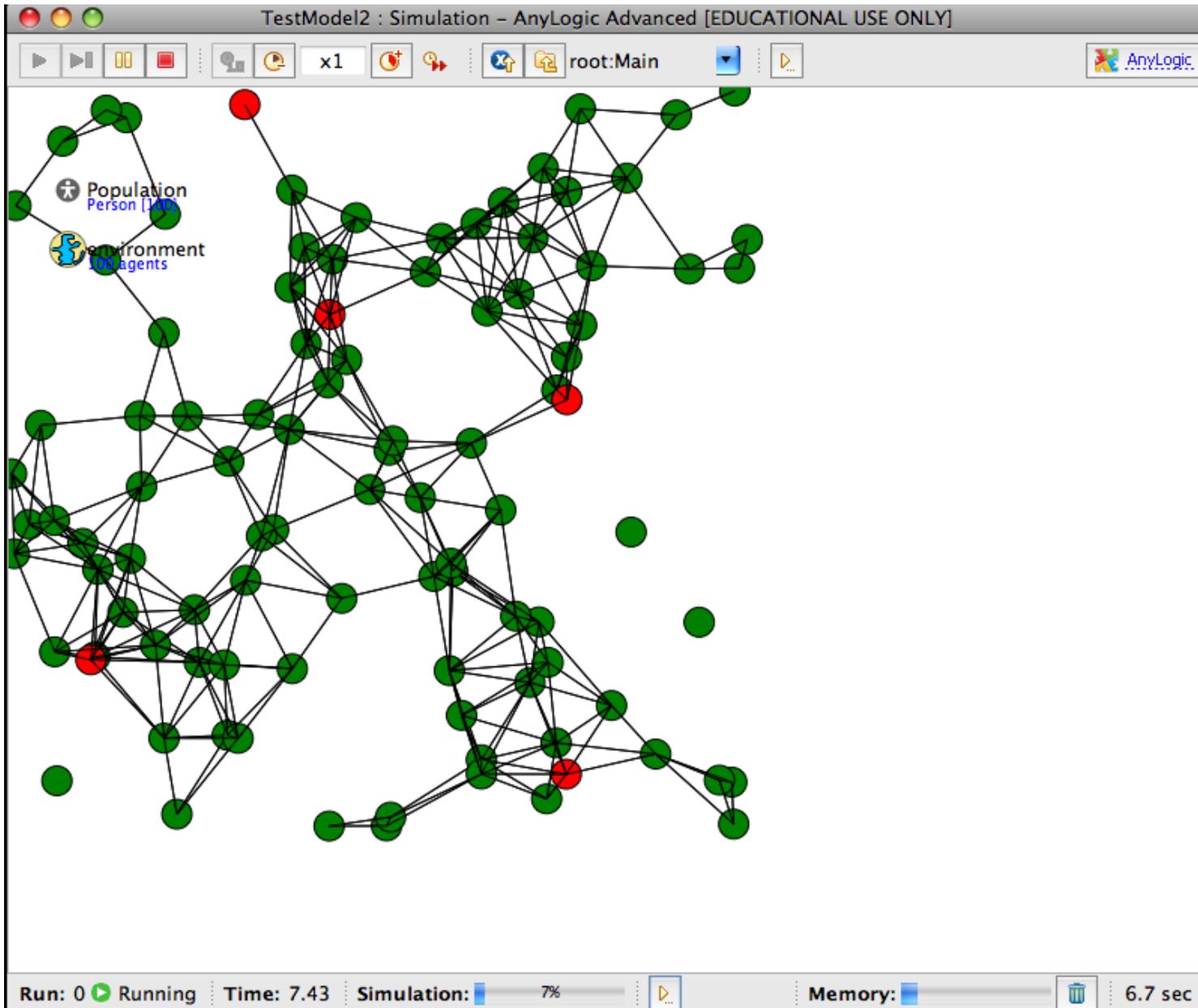
Experiment setup page

Run the model and switch to Main view

Press this button to start model execution

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

Model Presentation



Making Infection Depend on a Message

The screenshot displays the AnyLogic Advanced interface. On the left, a project tree shows a model named 'MalariaV2' with a statechart for 'InfectionStatechart'. The main workspace shows a statechart with two states: 'Susceptible' and 'InfectionS' (partially visible). A transition between them is highlighted with a blue arrow. The 'InfectionS' state is labeled 'color' with a 'V' icon. Below the workspace, the 'Infection - Transition' configuration panel is open. The 'Triggered by' dropdown is set to 'Message'. The 'Message type' is set to 'Other'. The 'Fire transition' section has 'Unconditionally' selected. A red oval highlights the 'Message' type and 'Other' options.

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

General

Name: Infection Show Name Ignore Public Show At Run

Description

Triggered by: Message

Message type: boolean int double String Other:

Class Name: Object

Fire transition: Unconditionally If message equals If expression is true (use msg for message)

Action:

Guard:

Setting "Person" so forwards Infection Message to Statechart

Message to Statechart

The screenshot displays the AnyLogic Advanced interface. The main workspace shows a statechart for 'InfectionStatechart' with two states: 'Susceptible' and 'Infected'. A variable 'color' is also visible. The left sidebar shows a project tree with 'Person' selected under 'TestModel2*'. The bottom panel shows the 'Person - Active Object Class' configuration. The 'Agent' tab is selected, and the 'On Message Received' event is configured with the code: `InfectionStatechart.receiveMessage(msg);`. A green arrow points to the 'Agent' tab, and a red oval highlights the code in the 'On Message Received' field.

Make sure the "Agent" Tab is selected!

```
InfectionStatechart
```

```
graph TD; Start(( )) --> Susceptible(Susceptible); Susceptible --> Infected(Infected); Infected --> Susceptible;
```

```
Person - Active Object Class
```

Tab	Value
General	<input checked="" type="checkbox"/> Environment defines initial location
Advanced	Initial coordinates: X: <input type="text"/> Y: <input type="text"/>
Agent	Movement parameters: Velocity: <input type="text"/> Rotation: <input type="text"/>
Parameters	On Arrival: <input type="text"/>
Description	On Message Received: <code>InfectionStatechart.receiveMessage(msg);</code>
	On Before Step: <input type="text"/>
	On Step: <input type="text"/>

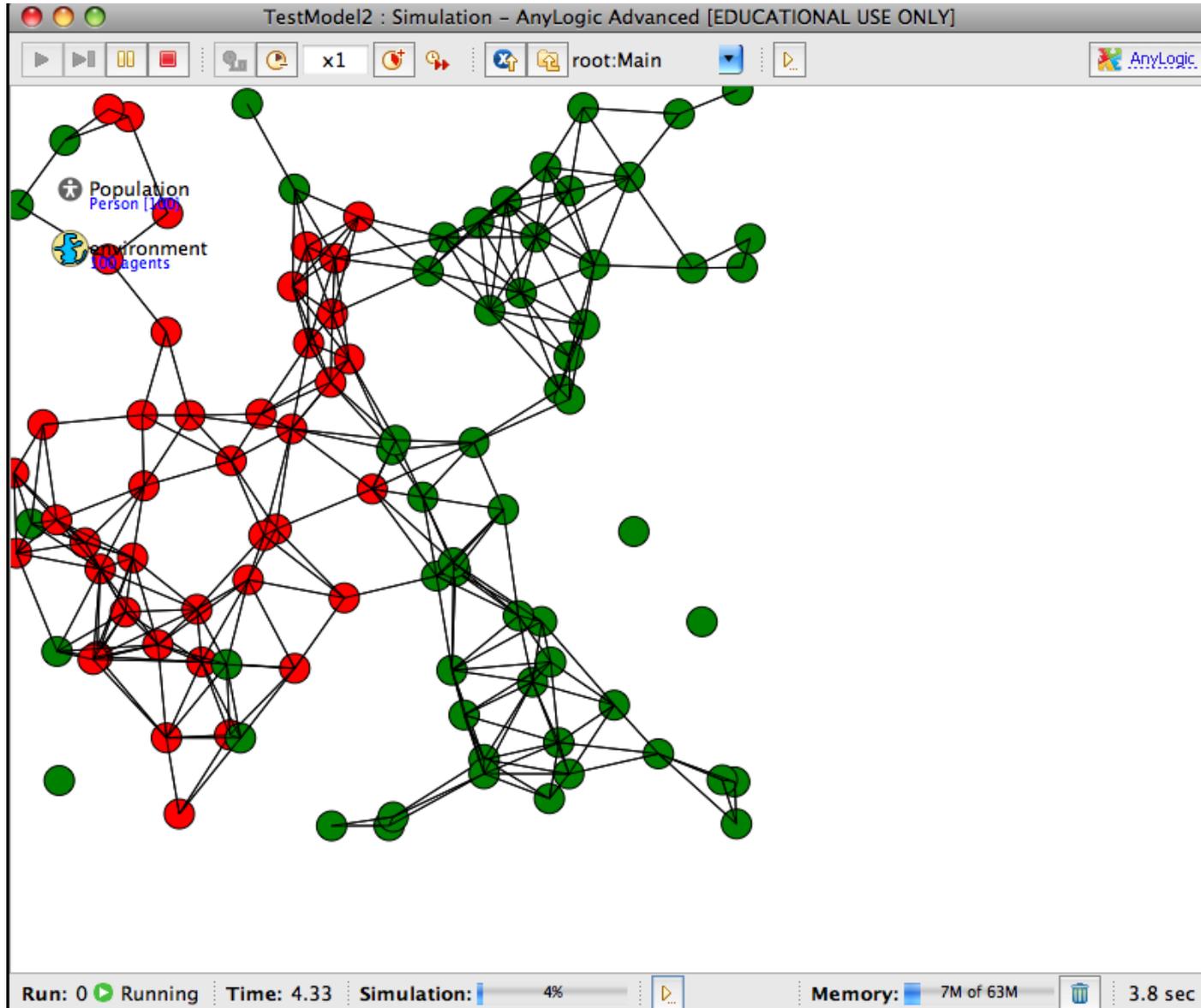
Setting Startup Code So Initially Infects a Random Person (so start with 1 infective)

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a diagram with a 'Population [..]' and an 'environment' object. The left sidebar contains a project tree with various models and objects. The bottom right panel, titled 'Main - Active Object Class', is open to the 'Description' tab. In this tab, the 'Startup Code' field contains the following code:

```
environment.deliverToRandom( "Infection" );
```

This code is highlighted with a red oval. The 'Destroy Code' field is currently empty. The interface also shows a 'Console' and 'Properties' panel above the code editor, and a 'Problems' panel at the bottom left.

Infection Percolation over the Network



Tip: Beware Loose Connections

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace shows a state transition diagram with two states: 'Susceptible' and 'Infective', both represented by yellow rounded rectangles. A transition arrow points from 'Susceptible' to 'Infective'. A variable 'color' is shown below the transition. A loose connection is visible as a vertical line extending upwards from the transition area. The left sidebar shows a project tree for 'MalariaV2'. The bottom console window displays the following error messages:

- Element doesn't belong to any state
- Hanging transition

The Properties window for 'transition1 - Transition' is open, showing the following configuration:

- Name: transition1
- Triggered by: Rate
- Rate: 0.1
- Action: (empty)
- Guard: (empty)

The status bar at the bottom indicates 'Selection' and 'Cursor: X=64, Y=86'.

Corrected

The screenshot displays the AnyLogic Advanced software interface for a malaria model. The main workspace shows a state transition diagram with two states: 'Susceptible' and 'Infected'. A transition arrow points from 'Susceptible' to 'Infected', with a variable 'color' (represented by a 'V' icon) associated with it. The 'Person' entity is visible in the background. The left sidebar shows a project tree with folders for 'MalariaV2', 'TestModel2*', 'Spatial SIR with Waning Immunity', and 'Network Modification of SIR AB'. The bottom panel shows the 'transition1 - Transition' properties, including 'Name: transition1', 'Triggered by: Rate', and 'Rate: 0.1'. The 'Problems' panel at the bottom left lists several errors: 'Element doesn't belong to any state'.

AnyLogic Advanced [EDUCATIONAL USE ONLY]

Project: 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

transition1 - Transition

General

Description

Name: transition1 Show Name Ignore Public Show At R...

Triggered by: Rate

Rate: 0.1

Action:

Guard:

Problems

- Element doesn't belong to any state

Selection