

Building User Interfaces for Models

Nathaniel Osgood

Using Modeling to Prepare for Changing
Healthcare Needs

Duke-NUS

April 16, 2014

Lecture Focus: Creating Custom User Interfaces using “Controls”

- “Controls” are “widgets” that allow for obtaining user input
 - These widgets have properties that can be set at both design and run (execution, simulation) time
- By setting the properties of these controls at design time, we can
 - Establish their general logical & visual properties
 - Establish their correspondence with model variables
- These controls can be used by the user during simulation to set assumptions in the model



Hands on Model Use Ahead



Load Previously Built Model:
MinimalistSIRNetworkABM

After change, suggest saving as
“UISupportedMinimalistSIRNetworkABM”

Recall: Hardcoded Exposure Rate

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for a 'Person' object. The statechart starts at an initial state (circle) and transitions to a 'Susceptible' state (yellow rounded rectangle). From 'Susceptible', a transition labeled 'infectionStatechart' leads to an 'Infective' state (yellow rounded rectangle). From 'Infective', a transition leads to a 'Recovered' state (yellow rounded rectangle). A variable 'colorForInfectionState' is also visible in the workspace.

The Properties window at the bottom shows the configuration for the 'pathogenExposure - Transition'. The 'Triggered by' dropdown is set to 'Rate'. The 'Rate' field is highlighted in red and contains the value '0.5'. A red arrow points from the text '“Hard-coded” rate' to this field. The 'Action' field contains the code: `this.send("Infection", RANDOM_CONNECTED);`. The 'Guard' field is empty.

Time units: days

Add a Related Parameter to *Main*

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for a 'Main' state with a 'Final State' circle. A parameter 'exposureHazard' is visible in the workspace. The 'Properties' window is open, showing the configuration for the 'exposureHazard - Parameter'.

exposureHazard - Parameter

- Name: exposureHazard Show name Ignore
- Visible: yes
- Type: double
- Default value: 0.5
- System dynamics array

Below the parameter settings are expandable sections: Value editor, Advanced, and Description.

The bottom status bar shows 'Time units: days' and 'X=...22'.

Setting the Transition to Refer to the Parameter in Main

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for an infection model with three states: Susceptible, Infective, and Recovered. A transition labeled 'pathogenExposure' is shown between the Infective and Recovered states. The Properties window for this transition is open, showing the following configuration:

- Name: pathogenExposu
- Triggered by: Rate
- Rate: `this.get_Main().exposureHazard`
- Action: `this.send("Infection", RANDOM_CONNECTED);`
- Guard: (empty)

A red arrow points from the text 'Note that' to the Rate field in the Properties window. The text 'Note that' is written in red and is positioned above the statechart. The text '“exposureHazard” lives in “Main”. To get a reference to the “Main” object, we call “get_Main()” on ourselves.' is also written in red and is positioned to the right of the statechart.

Note that
“exposureHazard” lives in “Main”. To get a reference to the “Main” object, we call “get_Main()” on ourselves.

Resulting Expression

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for an infection model. The states are Susceptible, Infective, and Recovered, connected by downward arrows. A transition from the Infective state to the Recovered state is highlighted with a blue arrow. The Properties window is open, showing the configuration for the 'pathogenExposure - Transition'.

pathogenExposure - Transition

Name: Show name Ignore

Triggered by:

Rate:

Action:

Guard:

Description

Time units: days X=...64

Reminder: An Explicitly Specified Population Size

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a statechart with two states: 'population [..]' and 'exposureHazard'. The 'population [..]' state is selected, and its properties are shown in the Properties panel below. The Properties panel includes fields for Name, Visible, Initial number of agents, and Initial location (X and Y coordinates).

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects

- UISupportedMinimalistSI
 - Main
 - Person
 - Simulation: Main
 - Diabetes in Saskatoon0
 - Main
 - Person
 - Option Lists
 - Simulation: Main

Statechart

- Statechart Entry Point
- State
- Transition
- Initial State Pointer
- Branch
- History State
- Final State

Properties Progress

population - Person

Name: Show name Ignore

Visible: yes

Single agent Population of agents

Initial number of agents:

Initial location

These settings are applied only if the "User-defined" layout type is set in the "Environment for other agents" properties of the upper level agent.

X:

Y:

UISupportedMinimalistSIRNetworkABM

Time units: days

X=...58

A Parameter Giving the Population Size

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart diagram with a state named 'populationSize' and a parameter named 'populationSize'. A blue arrow points from the parameter icon to the state icon. The 'Properties' window is open, showing the configuration for the 'populationSize - Parameter'.

Properties window: populationSize - Parameter

- Name: Show name Ignore
- Visible: yes
- Type:
- Default value:
- System dynamics array

Below the properties window are sections for 'Value editor', 'Advanced', and 'Description', which are currently collapsed.

The bottom status bar shows 'Time units: days' and 'X=...78'.

Setting the Population to Use the Parameter Value

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart diagram with a state named 'population' and a parameter named 'populationSize'. A blue arrow points from the 'population' state to the 'populationSize' parameter in the Properties window.

Properties window: population - Person

- Name: Show name Ignore
- Visible: yes
- Single agent Population of agents
- Initial number of agents:
- Initial location**
These settings are applied only if the "User-defined" layout type is set in the "Environment for other agents" properties of the upper level agent.
X:
Y:

Palette window: Statechart

- Statechart Entry Point
- State
- Transition
- Initial State Pointer
- Branch
- History State
- Final State

Bottom status bar: Time units: days X=...81

Reminder: The Existing Experiment

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a grid with the text "MinimalistNetworkABMModel" and "Experiment setup page". A button labeled "Run the model and switch to Main view" is visible. The left sidebar shows a project tree with folders like "Agents", "Parameters", and "Simulation: Main". The bottom panel shows the "Simulation - Simulation Experiment" configuration, including fields for Name, Top-level agent, Maximum available memory, and Parameters (exposureHazard, populationSize).

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects

- UISupportedMinimalistSIRNetworkABM
 - Main
 - Agents
 - population
 - Presentation
 - Parameters
 - Variables
 - Links to agents
 - Person
 - Simulation: Main**
 - Diabetes in Saskatoon0
 - Main
 - Person
 - Option Lists
 - Simulation: Main

Person Main Simulation

MinimalistNetworkABMModel
Experiment setup page

Run the model and switch to Main view

Statechart

- Statechart Entry Point
- State
- Transition
- Initial State Pointer
- Branch
- History State
- Final State

Properties Progress

Simulation - Simulation Experiment

Name: Simulation Ignore

Top-level agent: Main

Maximum available memory: 64 Mb

Parameters

exposureHazard: = 0.5

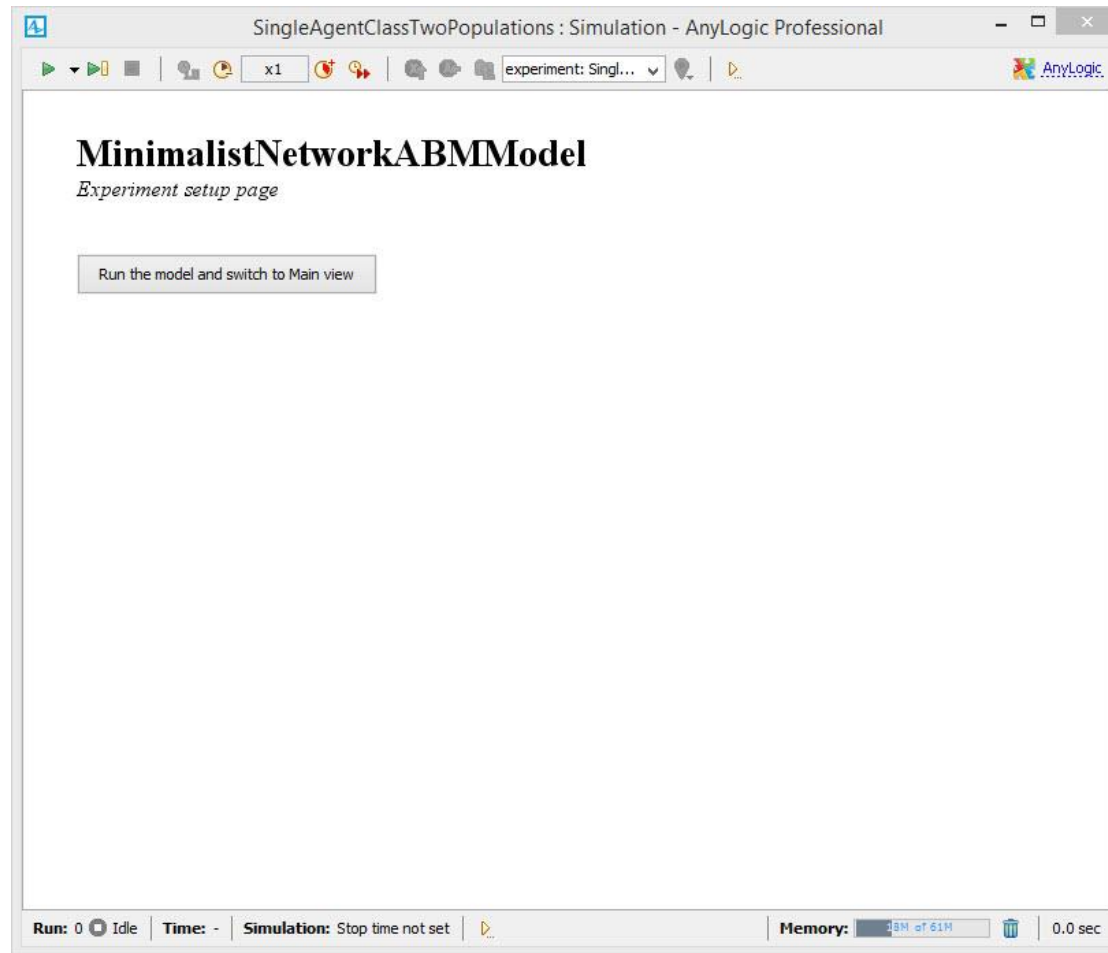
populationSize: = 100

Paste from clipboard

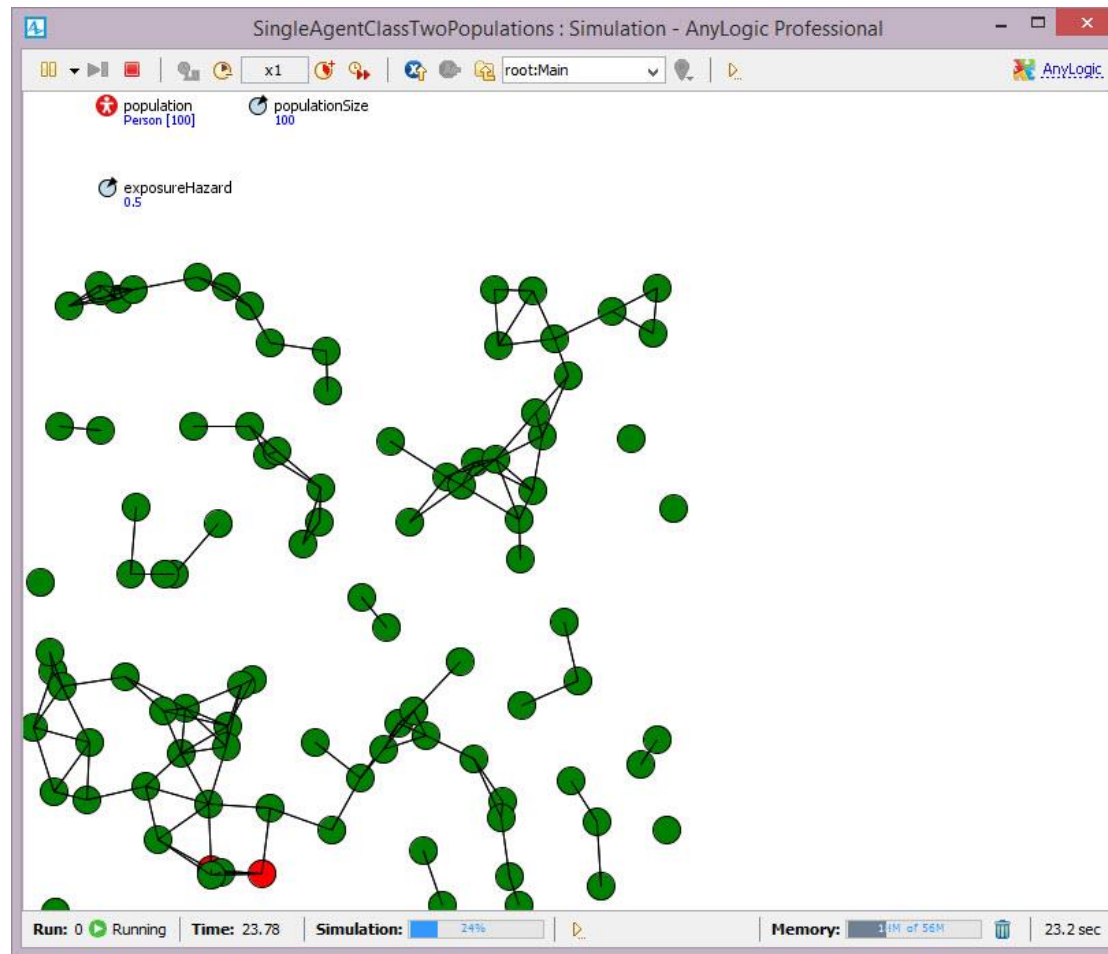
UISupportedMinimalistSIRNetworkABM

Time units: days

Running that Experiment



Reminder: Pushing the Button Shows the Simulation Visualization



Understanding the Button's Actions

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart diagram for a **MinimalistNetworkABMModel**, which is an *Experiment setup page*. A button labeled "Run the model and switch to Main view" is positioned on the diagram. A blue arrow points from this button to the right, indicating a transition to another statechart.

The **Properties** panel for the selected **button - Button** is visible at the bottom. It includes the following configuration:

- Name:** button Ignore
- Label:** Run the model and switch to Main
- Enabled:** `getState() == IDLE`
- Action:**

```
run();
getEngine().getPresentation().setPresentable( getEngine().getRoot() );
```
- Appearance:** (partially visible)

The **Palette** on the right side of the interface lists statechart elements: Statechart Entry Point, State, Transition, Initial State Pointer, Branch, History State, and Final State.

The status bar at the bottom indicates "Time units: days" and "X=...28".

Adding a Slider to Represent the Population Size

The screenshot displays the AnyLogic Professional interface. The main workspace shows a diagram titled "MinimalistNetworkABMModel" with the subtitle "Experiment setup page". A button labeled "Run the model and switch to Main view" is visible. A red arrow points from the text "Fill in this information" to the "Link to:" field in the "sliderPopulationSize - Slider" properties panel. The "Minimum value:" field is highlighted with a red box and contains the value "1". Other fields include "Maximum value:" (1000) and "Default value:" (100). The "Name:" field is "sliderPopulationSize" and "Orientation:" is set to "Horizontal".

Fill in this information

sliderPopulationSize - Slider

Name: sliderPopulationSize Ignore

Orientation: Horizontal Vertical

Link to: _____

Minimum value: 1

Maximum value: 1000

Default value: 100

Enabled:

Setting the Simulation Parameter Values to Use the Slider Setting

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for the **MinimalistNetworkABMModel**, which is an *Experiment setup page*. A slider control is visible in the workspace, and a red arrow points from it to the parameter `sliderPopulationSize.getIntValue()` in the **Simulation - Simulation Experiment** properties panel. The parameter `exposureHazard` is set to `0.5`. The **Simulation** properties panel includes fields for Name, Top-level agent, and Maximum available memory, along with a **Parameters** section.

Simulation - Simulation Experiment

Name: Ignore

Top-level agent:

Maximum available memory: Mb

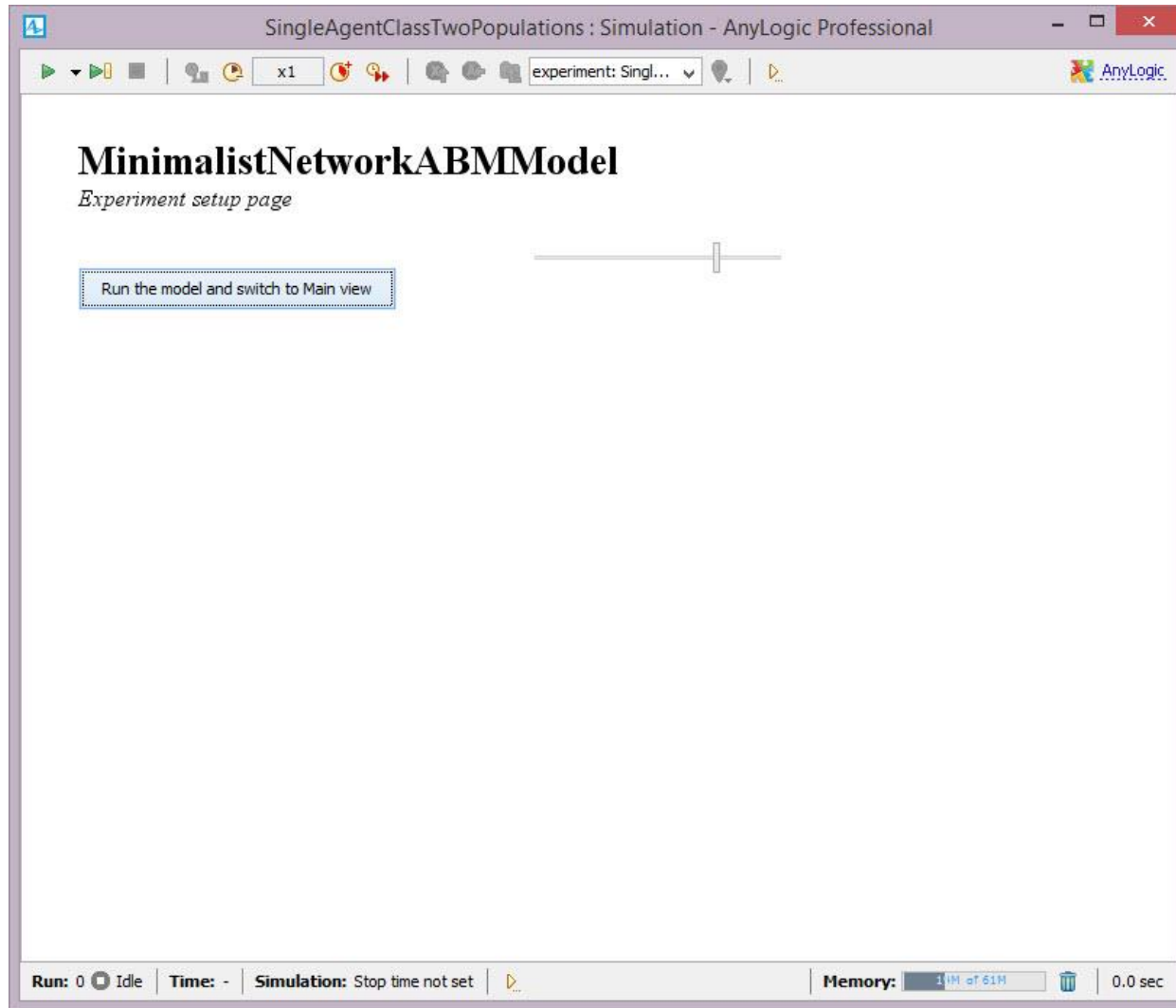
Parameters

exposureHazard: =

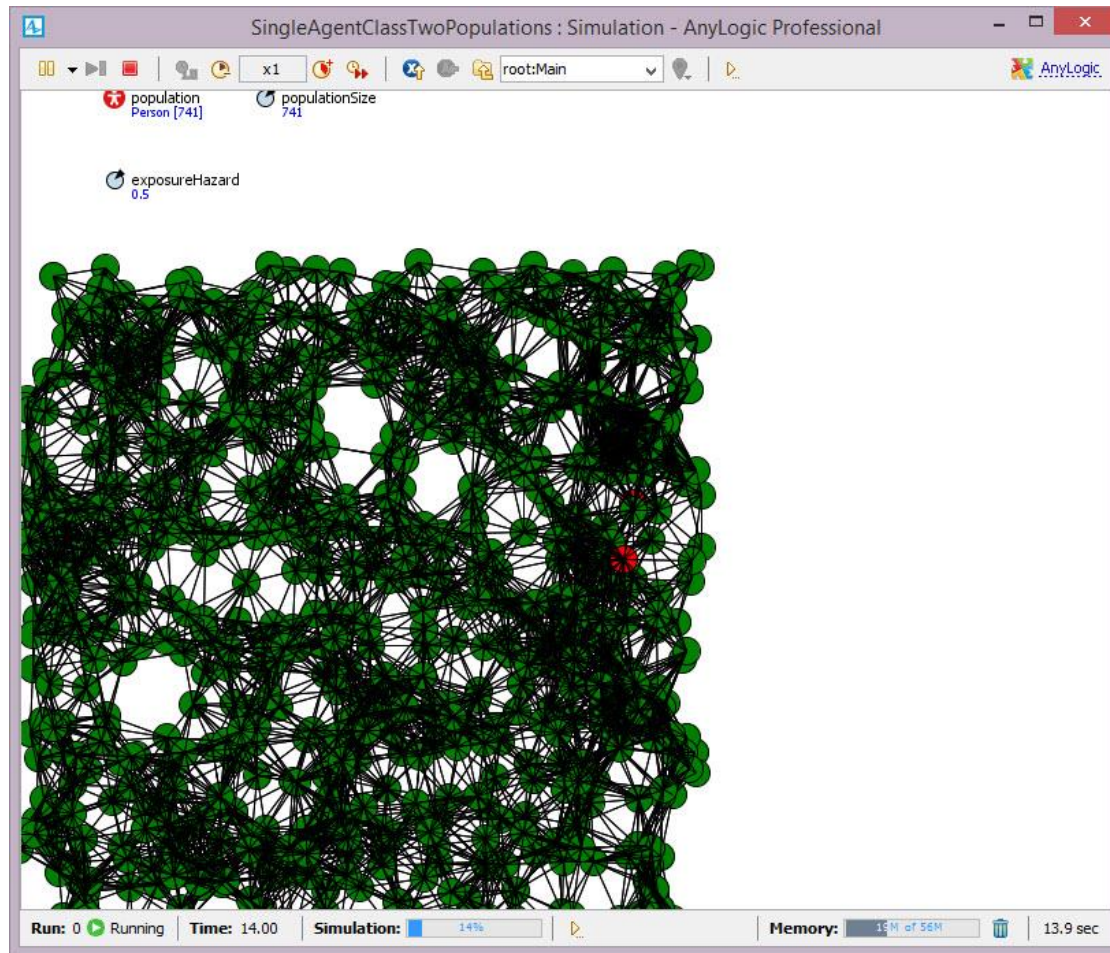
populationSize: =

Time units: days

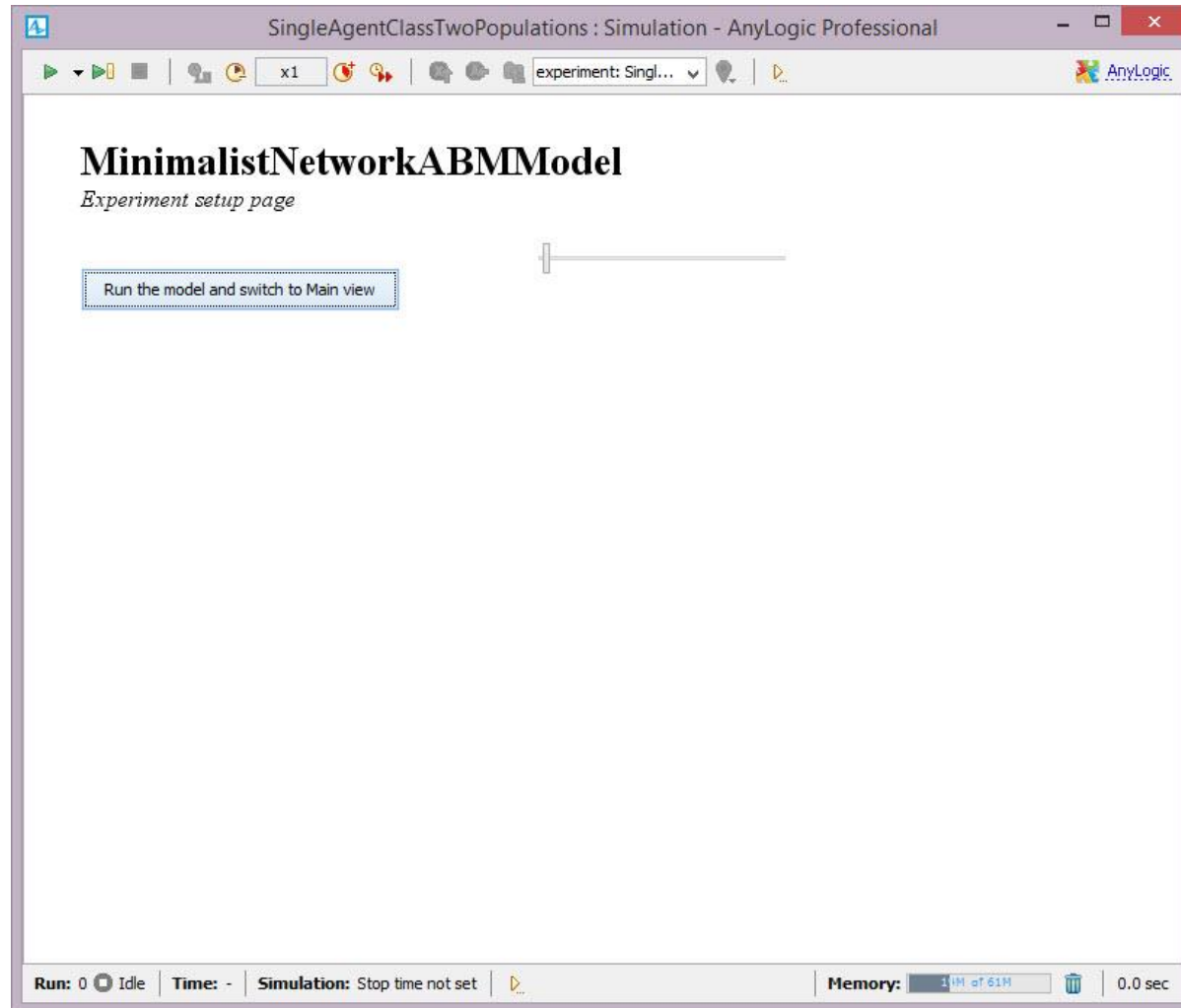
Choosing a High Value on the Slider



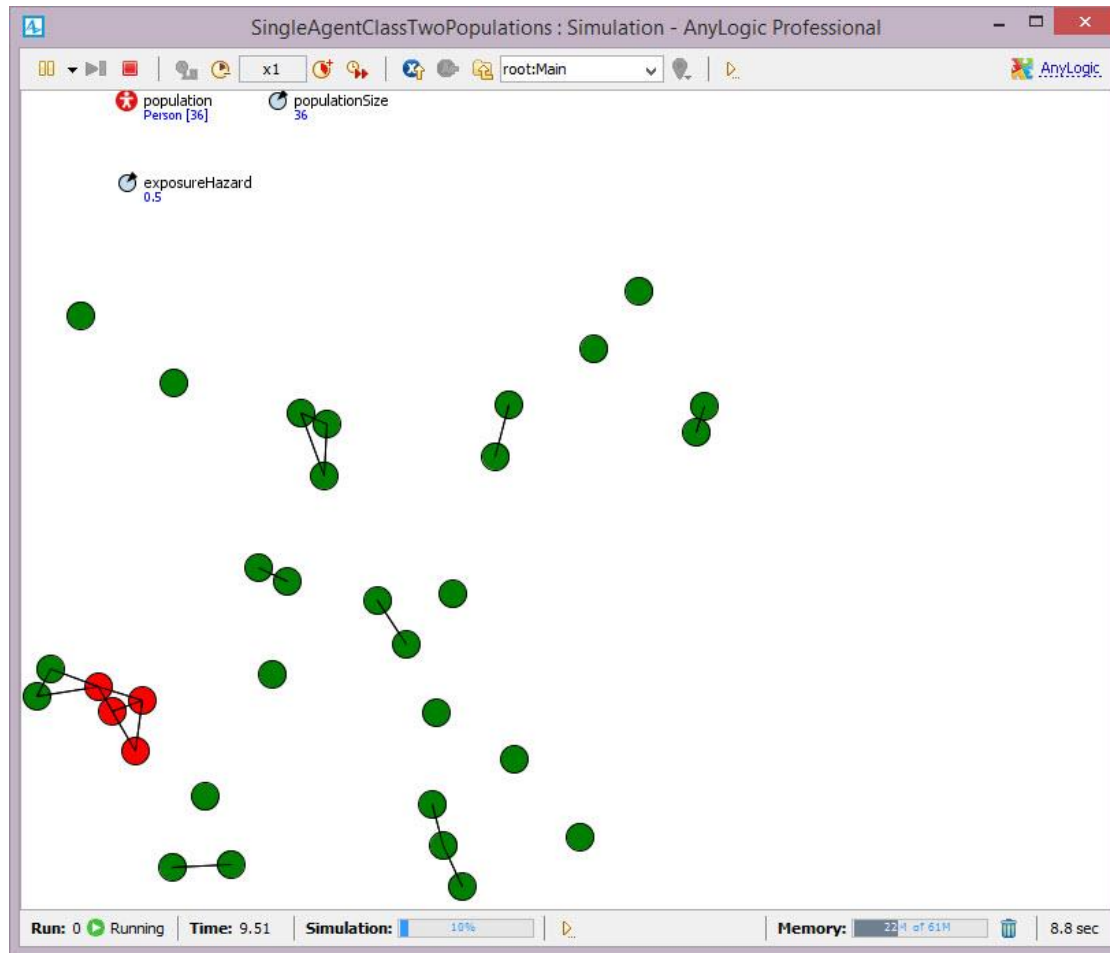
Resulting Network – Large Population



Choosing a Low Value on the Slider



Resulting Network -- Small Population



Adding (Static) Text Labeling Slider

The screenshot displays the AnyLogic Professional interface. The main workspace shows a simulation model titled "MinimalistNetworkABMModel" with the subtitle "Experiment setup page". A slider labeled "Population size" is visible, with a red arrow pointing to it from the text "Fixed text – doesn't change over time". Below the workspace, the Properties panel is open, showing the configuration for a text element named "textPopulationSize - Text". The text content is "Population size", and the color is set to "black".

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects

- UISupportedMinimalistSI
 - Main
 - Agents
 - population
 - Presentation
 - population_pr
 - slider
 - buttonSeedNe
 - checkboxEnab
 - Parameters
 - Variables
 - Links to agents
 - Person
 - Simulation: Main

MinimalistNetworkABMModel
Experiment setup page

Population size

Run the model and switch to Main view

Fixed text – doesn't change over time

Properties

Progress

Aa textPopulationSize - Text

Name: textPopulationSize Ignore Lock

Visible: yes

Text

Population size

Appearance

Color: black

Alignment: [Left] [Center] [Right]

UISupportedMinimalistSIRNetworkABM

Time units: days

X=...26

Creating a Text Element to Give the Slider Value

The screenshot displays the AnyLogic Professional interface. The main workspace shows a simulation model titled "MinimalistNetworkABMModel" with the subtitle "Experiment setup page". A button labeled "Run the model and switch to Main view" is visible. A red text overlay states: "This text is initially blank, but we'll set it elsewhere to change over time with the slider value". A red arrow points from this text to the "Text" property field in the Properties window.

The Properties window shows the following details for the "textSliderValue - Text" element:

- Name: textSliderValue Ignore Lock
- Visible: yes
- Text:
- Appearance:
 - Color: black
 - Alignment: Left Center Right

The bottom status bar indicates "Time units: days".

Dynamic Properties to Report the Slider Value

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart diagram for a model named "MinimalistNetworkABMModel". A text field is present in the diagram, and a red arrow points from the text in the diagram to the "Text" property in the Properties panel below. The Properties panel shows the "textSliderValue - Text" element with the following settings:

- Name: textSliderValue
- Visible: yes
- Text: `sliderPopulationSize.getIntValue()`
- Appearance: Color: black

The text in the diagram reads: "By setting **this expression** to determine the dynamic value of the text field **“Text”** property, the string associated with this text will automatically change with the slider value".

Example Resulting Output

The screenshot shows the AnyLogic Professional interface for a simulation titled "SingleAgentClassTwoPopulations : Simulation". The main window displays the "MinimalistNetworkABMModel" experiment setup page. A slider control for "Population size" is set to 708. A button labeled "Run the model and switch to Main view" is visible. The status bar at the bottom indicates the simulation is in an "Idle" state with a run time of 0 and a memory usage of 24M of 61M.

SingleAgentClassTwoPopulations : Simulation - AnyLogic Professional

experiment: Singl...

MinimalistNetworkABMModel

Experiment setup page

Population size 708

Run the model and switch to Main view

Run: 0 Idle | Time: - | Simulation: Stop time not set | Memory: 24M of 61M | 0.0 sec

Reflecting on Temporal Specificity of UI Elements

- The user interface component (slider) we created thus has had its value used to set the initial state of the model (the population size)
- User interface components can also be used to vary assumptions dynamically during runtime
 - For example, vary parameter values

Example: Creating a Slider to Dynamically Vary the Infection Hazard

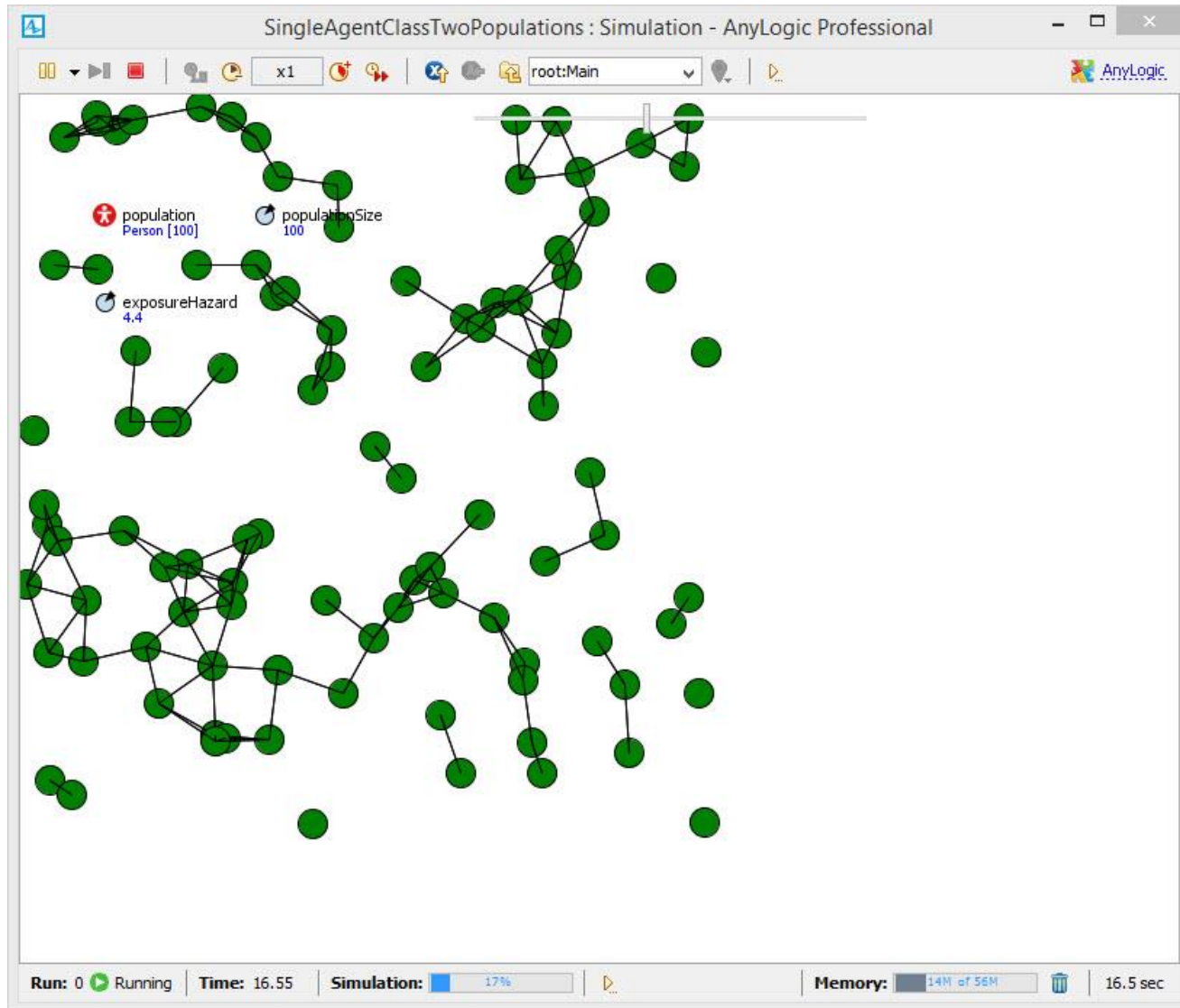
The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart with a transition labeled 'exposureHazard'. A blue text box with arrows pointing to the transition and a slider widget in the Properties panel contains the text: "Link so that changing the slider automatically changes the exposureHazard parameter".

The Properties panel for the 'slider - Slider' widget is visible, showing the following settings:

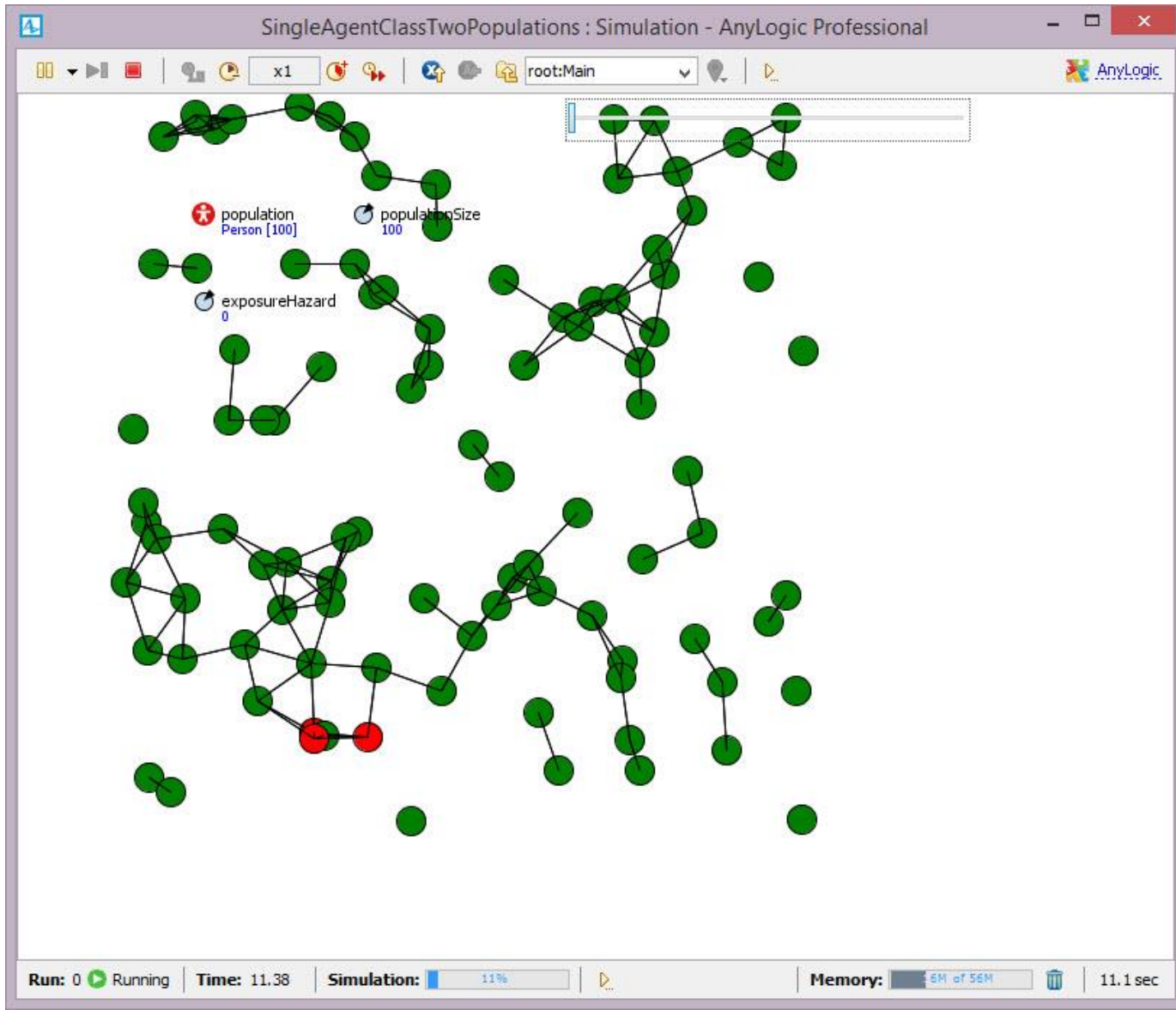
- Name: slider
- Ignore:
- Visible on upper level:
- Orientation: Horizontal Vertical
- Link to: exposureHazard
- Minimum value: 0
- Maximum value: 10

Red text at the bottom left of the image reads: "Set these as the upper and lower bounds of the slider", with a red arrow pointing to the 'Link to' field.

A High Slider Value Leads to a More Rapid Spread



Dropping the Slider Value (Exposure Hazard) to 0 Can Stop the Spread



Recall: The Initial Infection Seed

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for a 'Person' agent. A red text box with the text 'This delivered an infection message to a randomly selected person in the population' is positioned over the statechart. A red arrow points from this text box to the 'On startup' action in the 'Main - Agent Type' properties panel. The 'On startup' action is highlighted with a pink background and contains the code: `deliverToRandomAgentInside ("Infection");`. The 'On destroy' action is currently empty.

This delivered an infection message to a randomly selected person in the population

Main - Agent Type

Name: Ignore

Parameters preview

Agent actions

On startup:
`deliverToRandomAgentInside ("Infection");`

On destroy:

Cut Text from Startup Code for Main

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for the 'Main' agent type. A blue circle at the top left of the statechart grid represents the initial state. A vertical blue line descends from this circle, and a horizontal blue line extends to the right. A 'Seed New Infection' button is positioned on the horizontal line. Below the horizontal line, the variables 'population [..]', 'populationSize', and 'exposureHazard' are listed. The 'Properties' window at the bottom is titled 'Main - Agent Type' and shows the 'Name' field set to 'Main' and an 'Ignore' checkbox. The 'Agent actions' section is expanded, showing 'On startup:' and 'On destroy:' fields.

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects

UISupportedMinimalistSI

- Main
 - Agents
 - population
 - Presentation
 - population_pr
 - slider
 - buttonSeedNe
 - checkboxEnab
 - Parameters
 - Variables
 - Links to agents
 - Person
 - Simulation: Main

Person Main Simulation

Seed New Infection

population [..] populationSize

exposureHazard

Palette

Statechart

- Statechart Entry Point
- State
- Transition
- Initial State Pointer
- Branch
- History State
- Final State

Properties Progress

Main - Agent Type

Name: Main Ignore

Parameters preview

Agent actions

On startup:

On destroy:

Time units: days

Setting the Button to Seed a New Infection

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart with a transition labeled "Seed New Infection" leading to a state labeled "populationSize". A red arrow points from the text "This is the action the button will perform when pushed" to the "Action" field in the button's properties.

This is the action the button will perform when pushed

buttonSeedNewInfection - Button

Name: Ignore Visible on upper level

Label:

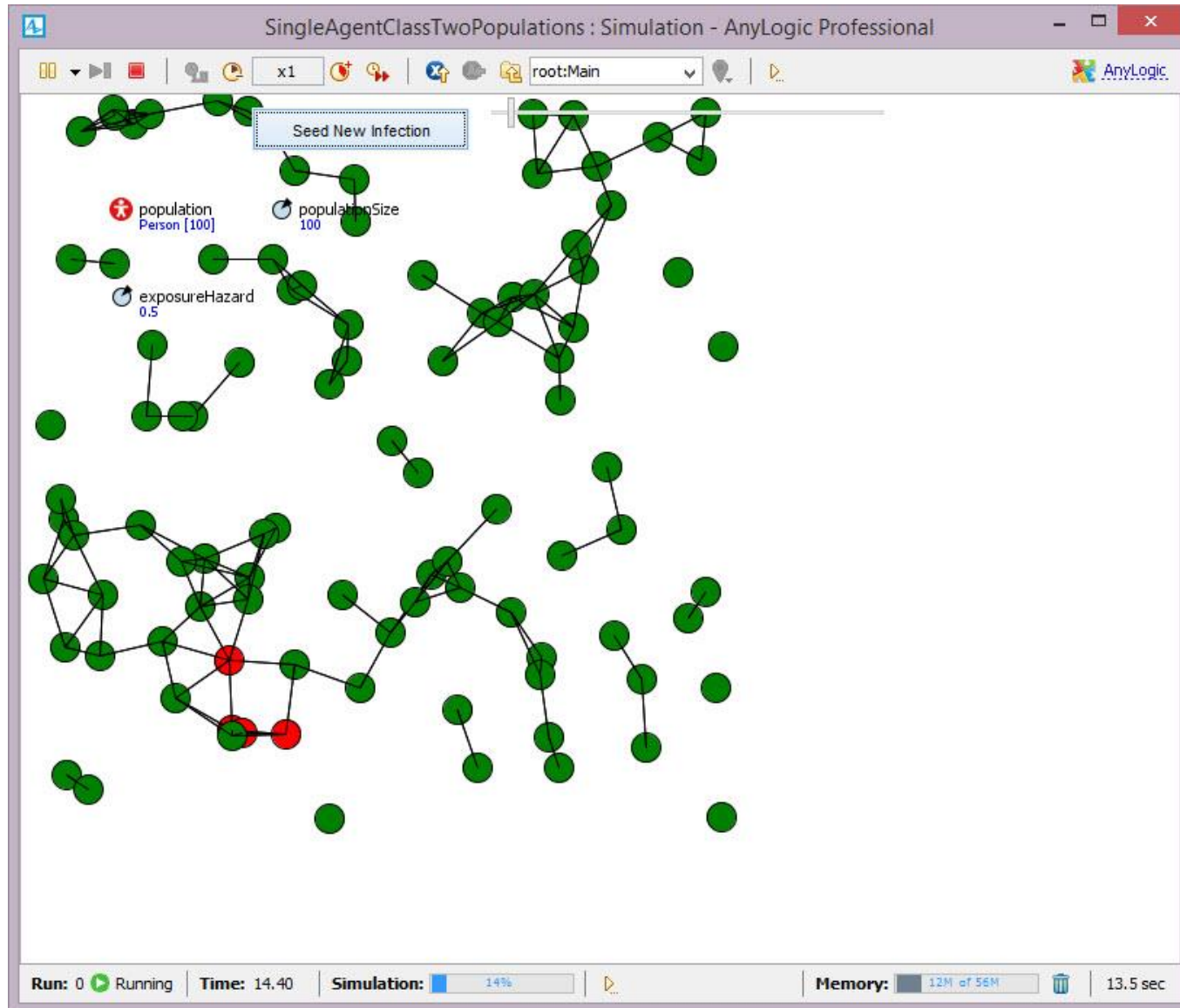
Enabled:

Action

```
deliverToRandomAgentInside ("Infection");
```

Time units: days X=...33

With Multiple Presses, Multiple “Seed” Infections



Add a Contingent Reporting Variable

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart with a state named `isReportingEnabled`. A red arrow points from the text "Initial value should be false" to the `isReportingEnabled` state. Below the workspace, the Properties window is open for the `isReportingEnabled` variable, showing the following settings:

- Name: `isReportingEnabl` (with Show name and Ignore)
- Visible: yes
- Type: `boolean`
- Initial value: `false`

The bottom status bar shows "Time units: days" and "X=...-3".

Contingent Infection Reporting

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects

- UISupportedMinimalistSI
 - Main
 - Agents
 - population
 - Presentation
 - population_pr
 - slider
 - buttonSeedNe
 - checkboxEnab
 - Parameters
 - Variables
 - Links to agents
 - Person
 - Simulation: Main

Person Main Simulation

infectionStatechart

Susceptible

Infective

Recovered

colorForInfectionState

This makes the reporting contingent on the value of isReportingEnabled

Palette

- Statechart
 - Statechart Entry Point
 - State
 - Transition
 - Initial State Pointer
 - Branch
 - History State
 - Final State

Properties Progress

infection - Transition

Triggered by: Message

Message type: Other Object

Fire transition: Unconditionally On particular message If expression is true

Action: `if (this.get_Main().isReportingEnabled) println(this + "has been infected!");`

Guard:

Time units: days

Contingent Recovery Reporting

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for an infection model with three states: Susceptible, Infective, and Recovered. A transition from the Infective state to the Recovered state is highlighted with a red box. A red arrow points from a text annotation to the code in the Properties window for this transition.

This makes the reporting contingent on the value of isReportingEnabled

recovery - Transition

Triggered by: Timeout

Timeout: 20

Action: `if (this.get_Main().isReportingEnabled) traceIn(this + " has recovered!");`

Guard:

Description

Time units: days X=...25

Enabling Reporting

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart diagram with a transition labeled "Enable Repor...". A red arrow points from this transition to the "Link to" field in the "checkboxEnableReporting - Check Box" properties panel. The "Link to" field is highlighted in red and contains the text "isReportingEnabled".

Link to the "isReportingEnabled" parameter

checkboxEnableReporting - Check Box

Name: Ignore Visible on upper level

Label:

Link to:

Enabled:

Action

Time units: days

Unless Reporting is Enabled (i.e. Checkbox is Checked), No Output

The image displays the AnyLogic Professional interface, split into two main windows. The left window, titled 'Person', shows the model's structure in a tree view on the left and a visual representation of the 'Person' agent on the right. The tree view includes 'Main', 'Agents' (with 'population'), 'Presentation' (with 'checkboxEnabled' checked), 'Parameters', 'Variables', 'Links to agents', 'Person', and 'Simulation: Main'. The visual representation shows a 'Seed New Infection' button and a 'checkboxEnabled' property set to 'checked'. The right window, titled 'SingleAgentClassTwoPopulations : Simulation - AnyLogic Professional', shows the simulation running. It features a network of green agents (representing one population) and a few red agents (representing another population). The simulation is running for 9.95 time units, with a progress bar at 10%. The status bar at the bottom indicates 'Run: 0', 'Time: 9.95', 'Simulation: 10%', and 'Memory: 5M of 56M'. The time units are set to 'days'.

Enabling Reporting Allows Output

The image displays the AnyLogic Professional software interface for a simulation titled "SingleAgentClassTwoPopulations : Simulation - AnyLogic Professional". The interface is divided into several panels:

- Left Panel (Project Explorer):** Shows the project structure under "Main", including "Agents" (population), "Presentation" (population_pr, slider, buttonSeedNe, checkboxEnab), "Parameters", "Variables", "Links to agents", "Person", and "Simulation: Main".
- Center Panel (Diagram):** A network diagram showing a "Seed New Infection" button connected to a "population" agent. The "population" agent is linked to "populationSize" and "exposureHazard". A "checkboxEnab" (isReportingEnabled) is also shown, which is checked.
- Right Panel (Simulation View):** A network of green circular agents connected by lines. A "Seed New Infection" button is visible at the top. Some agents are highlighted in red, indicating they are infected. The "isReportingEnabled" checkbox is checked.
- Bottom Panel (Console):** Displays the following output:

```
anylogic config [Java Application] D:\Program Files\AnyLogic 7 Pro  
root.population[3]has been infected!  
root.population[93]has been infected!  
root.population[23]has been infected!  
root.population[38]has been infected!  
root.population[97]has been infected!  
root.population[87]has been infected!  
root.population[25]has been infected!
```
- Bottom Status Bar:** Shows "Run: 0" (Running), "Time: 12.82", "Simulation: 13%", "Memory: 12M of 55M", and "12.1 sec". The time units are set to "days".

Cleaning Up by Separating the Network Display Space from Other Model Components

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a statechart diagram with a grid background. A red box highlights a statechart entry point on the left side of the diagram, with a red arrow pointing to it. The Properties panel at the bottom shows the configuration for the selected statechart entry point, including its name, visibility, and position.

Statechart Diagram Elements:

- Enable Repor... (checkbox checked)
- Seed New Infection (button)
- population [..] (state)
- populationSize (variable)
- exposureHazard (variable)
- isReportingEnabled (variable)

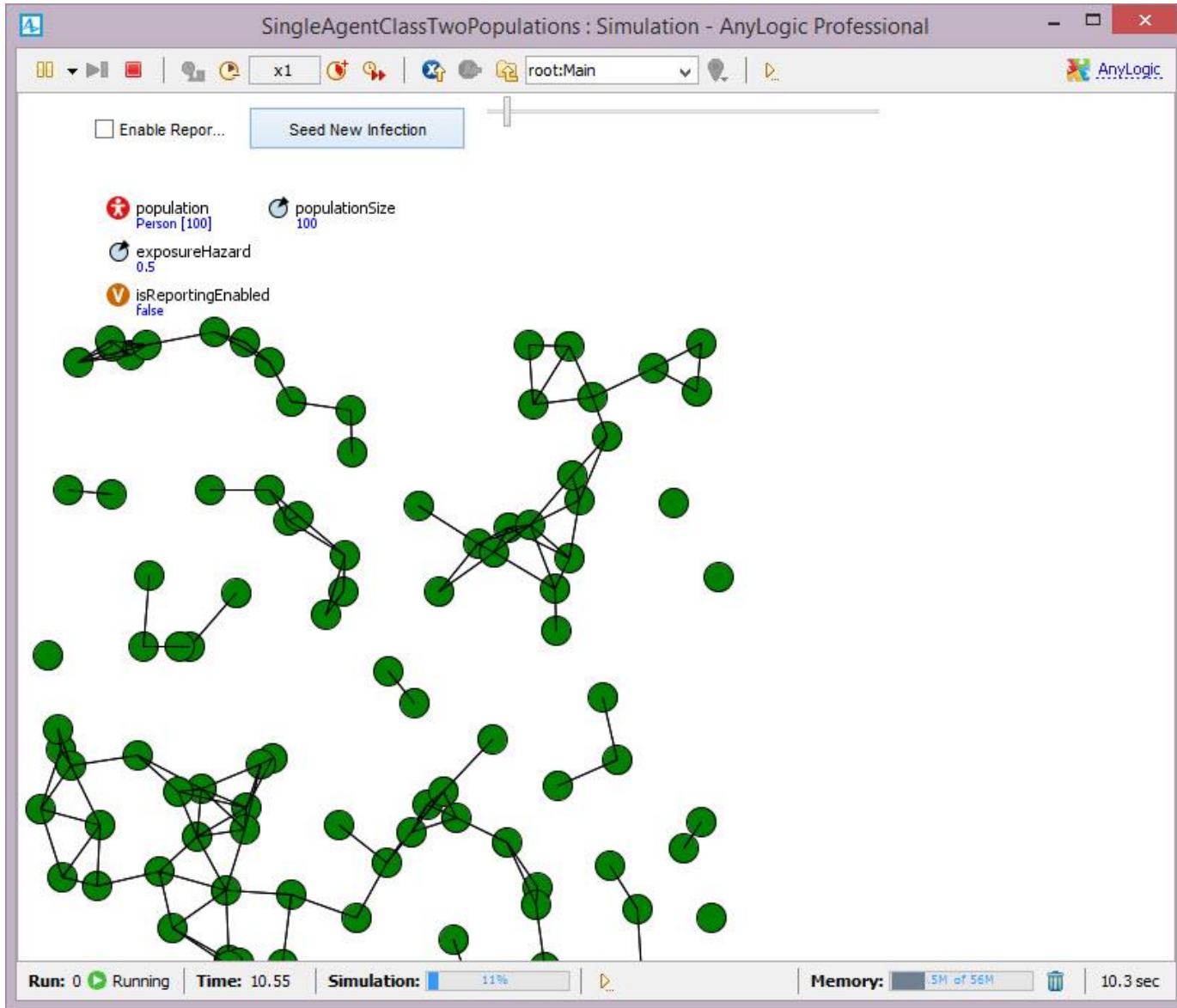
Properties Panel: population_presentation - Agent Presentation

- Name: population_prese Ignore Visible on upper level
- Visible: no yes
- Position and size: Position is defined by owners environment
- X:
- Y:

Red Text Annotation:

This is the display "origin" for the agents. Positive coordinates for the agents will yield locations visually to the right and below this

Resulting Visual Separation





Hands on Model Use Ahead



Load Example Model:

HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkStructure

Recall: "Hardcoded" File Names

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart with a single state and a transition. A red text annotation reads: "This currently 'hardcodes' that we are opening a particular Pajek file". A red arrow points from this text to the code in the Properties window.

Statechart:

- State: population [..]
- Transition: networkFileT
- Transition: networkFileP

Agent actions (Main - Agent Type):

```
On startup:  
//establishNetworkTransitionAndPopulationsFromConnectivityMatrixFile ("C:\\Usask\\Classes\\PajekSampleNetworkFile.txt")  
establishNetworkTransitionsAndPopulationsFromPajekNetworkFile ("C:\\Usask\\Classes\\PajekSampleNetworkFile.txt")  
applyLayout(); // now that established connectivity, perform layout  
  
On destroy:
```

Creating a Parameter to Communicate the Network File Name & Location (“Path”)

The screenshot displays the AnyLogic Professional interface. At the top, the title bar reads "AnyLogic Professional". Below it is a menu bar with "File", "Edit", "View", "Draw", "Model", "Tools", and "Help". A toolbar with various icons is visible. On the left, a "Projects" pane shows a tree view with "HardcodedMinimalistNet", "Main", "Person", and "Simulation: Main". The main workspace shows a statechart diagram with a state named "networkFilePathAndName" and a transition named "population [...]". Below the diagram, a list of actions is visible, including "establishNetworkTransitionsAndPopulationsFromConnectivityMatrixFile", "addConnectionsFromConnectivityMatrixRow", "countAdjacencyValuesSpecified", "createPopulationOfSize", "establishNetworkTransitionsAndPopulationsFromPajekNetworkFile", "parseAndProcessPajekVertices", and "parseAndProcessPajekEdgesAndLabels". On the right, a "Palette" pane shows statechart symbols like "Statechart Entry Point", "State", "Transition", "Initial State Pointer", "Branch", "History State", and "Final State". At the bottom, a "Properties" pane is open for the "networkFilePathAndName - Parameter". It shows the following settings: Name: "networkFilePathAndName" (with a "Show name" checkbox checked and "Ignore" unchecked); Visible: "yes" (checked); Type: "String" (selected in a dropdown menu); Default value: an empty text field; System dynamic: unchecked. A red arrow points from the text "Indicate that this parameter holds a (reference to a) String" to the "String" option in the Type dropdown menu. The bottom status bar shows "Time units: days".

Indicate that this parameter holds a (reference to a) String

networkFilePathAndName - Parameter

Name: networkFilePathAndName Show name Ignore

Visible: yes

Type: String

Default value:

System dynamic

Value editor

boolean

int

double

String

Main

Person

Other

Time units: days

Creating an “Enum” to Encode the Possible Types of the Specified File

The screenshot displays the AnyLogic Professional interface. At the top, the title bar reads "AnyLogic Professional". Below it is a menu bar with "File", "Edit", "View", "Draw", "Model", "Tools", and "Help". A toolbar with various icons is visible. On the left, a "Projects" panel shows a tree view with "HardcodedMinimalistNet", "Main", "Person", and "Simulation: Main". The main workspace shows a statechart with a single state and a transition. A red arrow points from the text "Specifies legal types of files" to the statechart. Below the workspace, the "Properties" window is open, showing the "Main - Agent Type" section. It contains the following code:

```
Imports section:  
import java.io.*;  
  
Implements (comma-separated list of interfaces):  
  
Additional class code:  
enum NetworkFileType { Pajek, ConnectivityMatrix };  
 Parameterized type
```

The "Additional class code" section is highlighted with a red background. At the bottom right, the status bar shows "Time units: days".

Creating a Parameter to Encode the Network File Type

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart diagram with a parameter 'networkFileType' connected to a state. A red text box with an arrow pointing to the parameter's value in the Properties window contains the following text:

Specifies that this parameter encodes the legal types of files (as specified by the NetworkFileType enumeration)

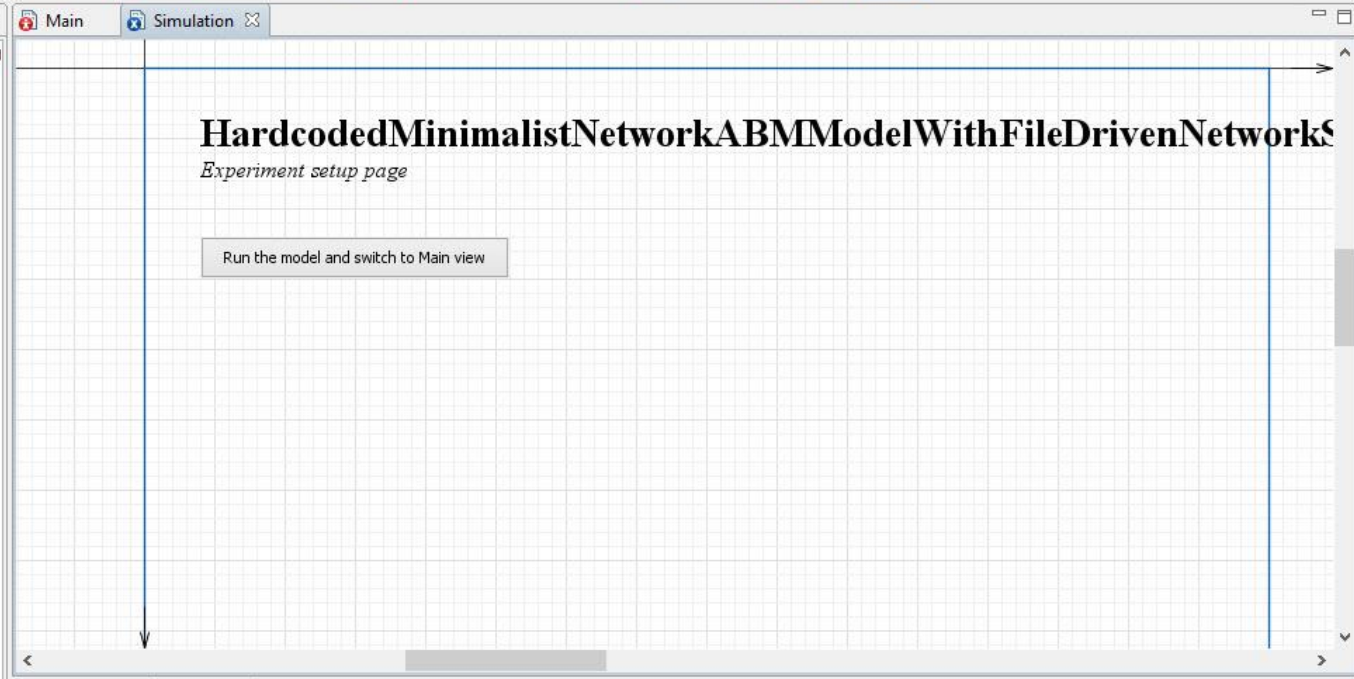
The Properties window for the 'networkFileType - Parameter' is visible at the bottom, showing the following configuration:

- Name: networkFileType Show name Ignore
- Visible: yes
- Type: Other
- Default value:
- System dynamics array

The Statechart Palette on the right lists various elements including Statechart Entry Point, State, Transition, Initial State Pointer, Branch, History State, and Final State.

Projects

- HardcodedMinimalistNet
- Main
- Person
- Simulation: Main



Palette

Statechart

- Statechart Entry Point
- State
- Transition
- Initial State Pointer
- Branch
- History State
- Final State

Properties

radioButtonsFileTypes - Radio Buttons

Name: radioButtonsFile Ignore

Orientation: Vertical Horizontal

Referring to the External Java Swing Library

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart with the following text:

```
HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkS
Experiment setup page
```

Below the statechart, a red text overlay reads: **Choose "Add" to add the reference to the "swing-layout-1.0.1.jar" file, which contains the Java library containing the "JFileChooser" control (dialog box). [This is freely downloadable; some other versions may also do]**

The Properties panel at the bottom shows the model configuration for "HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkStructure - Model". Under the section "Jar files and class folders required to build the model:", the following table is visible:

Location
swing-layout-1.0.1.jar

A red arrow points from the text overlay to the "swing-layout-1.0.1.jar" entry in the table.

Adding a Reference to the Java “Swing” File Chooser Component

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for an experiment titled "HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkS". A red text annotation is overlaid on the statechart, stating: "We add a reference here to the 'JFileChooser' control, since we need to use it". A red arrow originates from this text and points to the "Imports section" of the "Simulation - Simulation Experiment" properties panel. In this section, the line `import javax.swing.JFileChooser;` is highlighted with a red background. Other sections in the properties panel include "Additional class code:" and "The following options will not be applied when the model runs as applet:". The bottom status bar indicates "Time units: days".

HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkS
Experiment simulation page

Run the model and switch to Main view

We add a reference here to the "JFileChooser" control, since we need to use it

Simulation - Simulation Experiment

Advanced Java

Imports section:

```
import javax.swing.JFileChooser;
```

Additional class code:

The following options will not be applied when the model runs as applet:

Java machine arguments:

Time units: days

Adding a Button “buttonSelectFile”

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for a model titled "HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkSt". The statechart is currently in the "Experiment setup page" view. A button labeled "Select File" is being added to the statechart. The Properties window at the bottom shows the configuration for the "buttonSelectFile - Button":

- Name: Ignore
- Label:
- Enabled:

The Action window at the bottom is currently empty.

Add an EditBox

editboxNetworkFilePathAndName

The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for a model titled "HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkStr". The statechart includes a state "Experiment setup page" with a button "Run the model and switch to Main view". Below this, there is a "Select File" button and two radio buttons: "Pajek File" (selected) and "Connectivity Matrix File". A blue-bordered edit box is positioned over the "Pajek File" radio button. The edit box is titled "editboxNetworkFilePathAndName - Edit Box" and contains the following fields:

- Name: Ignore
- Link to:
- Minimum value:
- Maximum value:
- Default value:
- Enabled:

The interface also shows a "Palette" on the right with statechart elements like "Statechart Entry Point", "State", "Transition", "Initial State Pointer", "Branch", "History State", and "Final State". The "Properties" and "Progress" panels are visible at the bottom, and the status bar indicates "Time units: days".

Adding a Label for the Filename

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a statechart titled "HardcodedMinimalistNetworkABMModelWithFileDrivenNetwork" with the subtitle "Experiment setup page". A red arrow points from the text "This is static text" to the "Network Input File:" label in the statechart. The Properties panel at the bottom shows the selected element, "textFileName - Text", with fields for Name, Visible, and Text. The Text field contains "Network Input File:". The Appearance section is also visible.

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects Main Simulation

HardcodedMinimalistNetworkABMModelWithFileDrivenNetwork
Experiment setup page

Run the model and switch to Main view

Select File Network Input File:

Pajek File
 Connectivity Matrix File

This is static text

Properties Progress

Aa textFileName - Text

Name: textFileName Ignore Lock

Visible: yes

Text

Network Input File:

Appearance

HardcodedMinimalistNetworkAB...thFileDrivenNetworkStructure Time units: days X=...28

Logic to Set the File Name

Here, we open the Dialog box, which returns a value indicating the result.

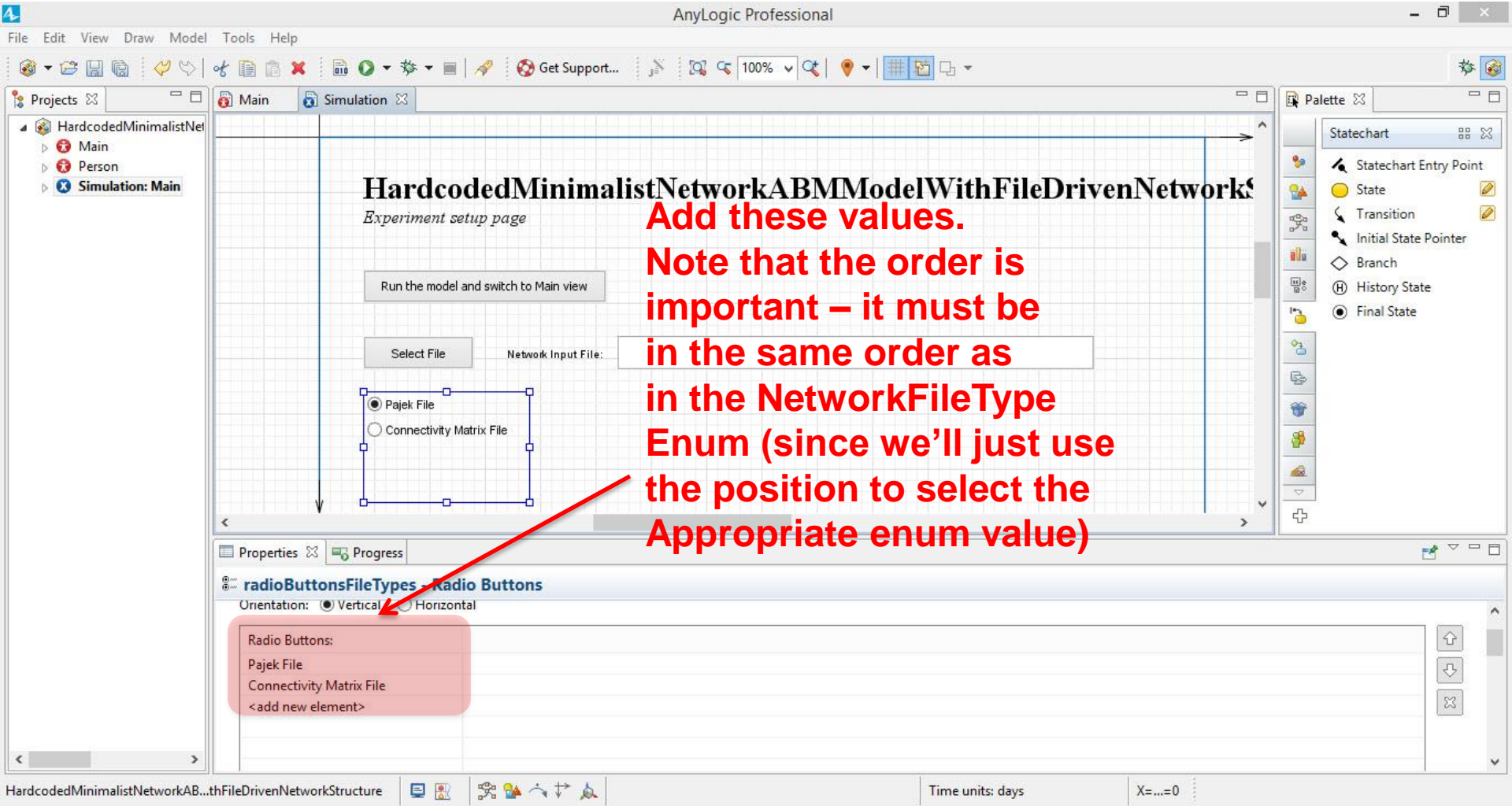
If this return value indicates that the user wants to go ahead with the chosen file...

...set the text in the Editbox to The "fully Qualified" filename (including path)

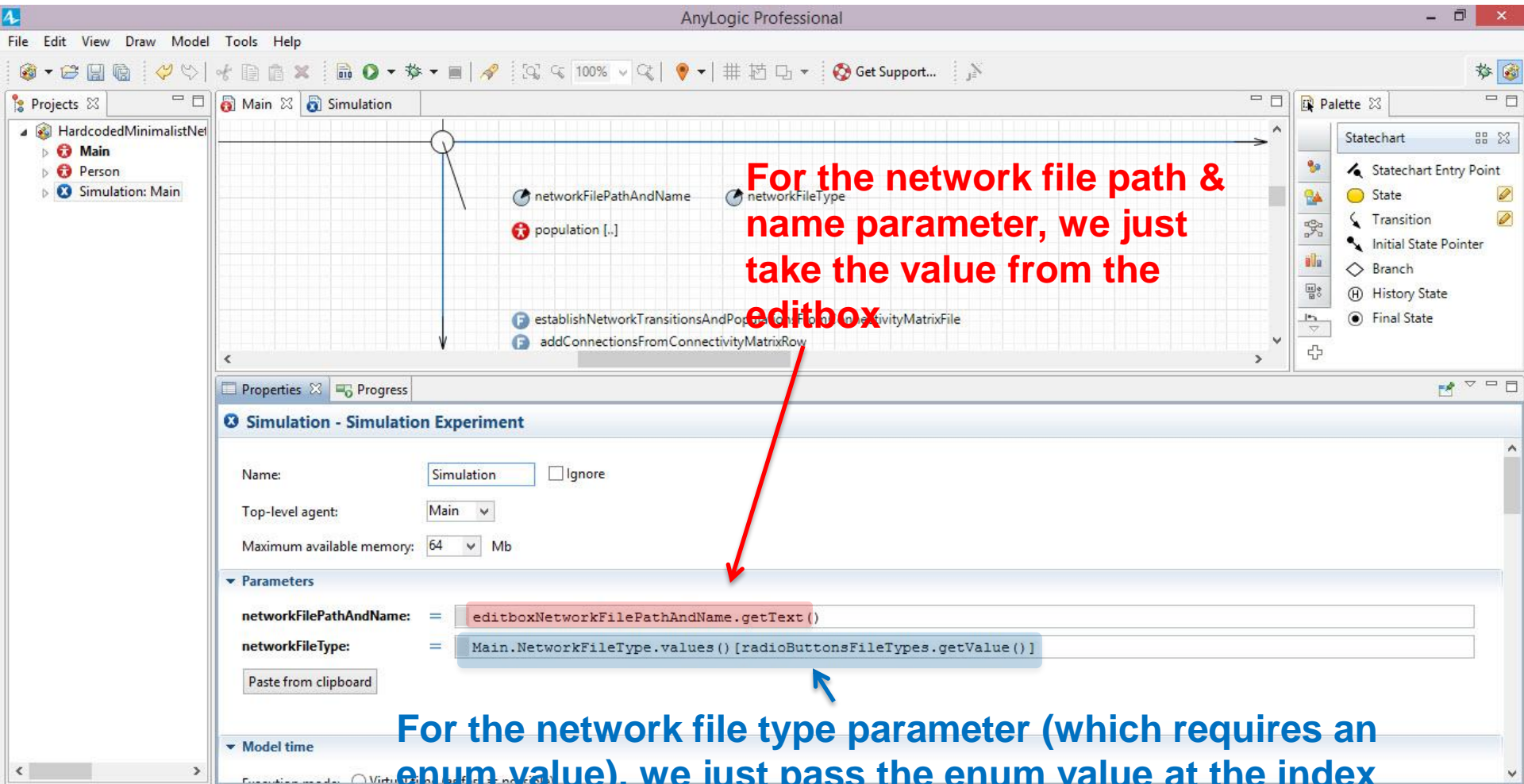
The screenshot displays the AnyLogic Professional interface. The main workspace shows a statechart for an 'Experiment setup page' with a 'Select File' button. The Properties window is open for the 'buttonSelectFile - Button' component, showing its 'Action' property. The code editor contains the following Java code:

```
JFileChooser c = new JFileChooser();  
// Demonstrate "Open" dialog:  
int iDialogResult = c.showOpenDialog(null);  
if (iDialogResult == JFileChooser.APPROVE_OPTION)  
    editboxNetworkFilePathAndName.setText(c.getCurrentDirectory().toString()+"\\"+c.getSelectedFile().getName());
```

Annotations include a red arrow pointing from the text 'Here, we open the Dialog box...' to the `showOpenDialog` method call, and a blue arrow pointing from the text 'If this return value indicates...' to the `if` statement. A green arrow points from the text '...set the text in the Editbox...' to the `setText` method call.



Add these values.
Note that the order is important – it must be in the same order as in the NetworkFileType Enum (since we'll just use the position to select the appropriate enum value)



For the network file path & name parameter, we just take the value from the editbox

For the network file type parameter (which requires an enum value), we just pass the enum value at the index given by the "Radio button" (the first enum value [i.e. at index 0] if the first radio button is selected, the second if the second radio button is selected)

Startup Code for Main

For the startup code for Main, we call the appropriate method to process the specified file, where the identity of that method is indicated by the specified NetworkFileType

The screenshot displays the AnyLogic Professional interface. The top window shows a statechart with parameters `networkFilePathAndName` and `networkFileType`, and a state `population [..]`. Below the statechart, the `Agent actions` section is expanded to show the `On startup:` code block, which is highlighted in a pink rounded rectangle. The code block contains the following code:

```
System.err.println("'" + networkFilePathAndName + "' is a " + networkFileType + " file.");
switch (networkFileType)
{
    case Pajek:
        establishNetworkTransitionsAndPopulationsFromPajekNetworkFile(networkFilePathAndName);
        break;
    case ConnectivityMatrix:
        establishNetworkTransitionsAndPopulationsFromConnectivityMatrixFile(networkFilePathAndName);
        break;
    default:
        throw new RuntimeException("Unexpected networkFileType " + networkFileType);
}
applyLayout(); // now that established connectivity, perform layout
```

The interface also shows a `Projects` panel on the left with a tree view containing `HardcodedMinimalistNet`, `Main`, `Person`, and `Simulation: Main`. A `Palette` panel on the right lists statechart elements like `Statechart Entry Point`, `State`, `Transition`, etc. The bottom status bar indicates `Time units: days`.