

# Specifying Agent Properties

Nathaniel Osgood

MIT 15.879

March 7, 2012

# Avoiding a Common Mistake

- AnyLogic projects typically contain a variety of “classes”
- The AnyLogic interface for accessing these classes is deceptively similar
- The semantics of the model will typically be very different depending on whether you add a component to one class or another
- Think about and be very clear as to which class you wish to add an element

# Embedded Objects

- The primary AnyLogic customized classes (Main & Agent classes) contain certain elements
  - Parameters
  - Variables
  - “Actions”
  - Elements of presentations

# Parameters: Static Quantities

- Parameters normally
  - Define constants that represent assumptions
  - Serve as mechanism to *communicate* such assumptions
- In Java, such parameters can have many types
  - Integer, Double precision value, boolean, etc.
- For parameters in the *Main* class, we can override the value of the parameters in an experiment
- Presentation elements associated with an Agent have special “Presentation” tab for their parameters

# Parameters and Communication

- Beyond defining assumptions, parameters in AnyLogic serve as mechanism to *communicate* such assumptions
- This communication takes place from an enclosing object at the point of creation of an enclosed object
  - From an Experiment (scenario) to the single instance of the Main class (as it is being created)
  - From the single instance of the Main class to a particular agent (as it is being created)
  - From a collective agent (e.g. City, Farm) to a particular enclosed agent (Person, Horse) as that enclosed agent is being created



Hands on Model Use Ahead



Load Previous Built [& Provided] Model:  
**MinimalistNetworkABMModel**

# Load in Previously Saved “MinimalistNetworkABMModel”

- Pre-built model is also available





# Setting the 1<sup>st</sup> Parameter Characteristics

The screenshot displays the AnyLogic Advanced interface. On the left, a project tree shows a model named 'InducingAttributeHeterogeneity\*' with a 'Person' component. The main workspace shows a 'Person' component with a parameter named 'income' selected. A red arrow points to this parameter with the text 'Select the first parameter'. Below the workspace, the 'Properties' panel for the 'income - Parameter' is shown. A green arrow points to the 'Name' field, which contains 'income', with the text 'Name the parameter "income"'. A blue arrow points to the 'Type' field, where 'double' is selected, with the text 'Make sure that the "Type" is marked as a "double" (the Default)'. The 'Properties' panel also shows options for 'Show Name', 'Ignore', 'Public', 'Show At Runtime', 'Default Value', 'Dynamic', and 'Save in snapshot'.

Select the first parameter

Name the parameter "income"

Make sure that the "Type" is marked as a "double" (the Default)

# Setting the 2<sup>nd</sup> Parameter Characteristics

The screenshot displays the AnyLogic Advanced interface. On the left, a project tree shows a 'Person' entity with parameters 'income' and 'sex'. The 'sex' parameter is selected. The main workspace shows a grid with 'income' and 'sex' parameters. The 'sex' parameter is highlighted in red. A red arrow points to it with the text 'Select the second parameter'. Below this, a green arrow points to the 'Name' field in the 'sex - Parameter' properties window, which contains the text 'sex'. A blue arrow points to the 'int' radio button in the 'Type' section of the same properties window, which is selected. The 'Type' section also includes options for 'void (just action)', 'boolean', 'double', 'String', and 'Other: int'. The 'Other: int' option is highlighted in pink. The 'Properties' window also shows checkboxes for 'Show Name', 'Ignore', 'Public', and 'Show At Runtime', and a 'Dynamic' checkbox. The 'Save in snapshot' checkbox is checked. The 'On Change' field is empty.

Select the second parameter

Name the parameter "sex"

Make sure that the "Type" is marked as an "int"

sex - Parameter

General

Name: sex  Show Name  Ignore  Public  Show At Runtime

Editor

Description

Type:  void (just action)  boolean  int  double  String  Other: int

Default Value:

Dynamic  Save in snapshot

On Change:

sex - Parameter

# “Population” Properties Now Include Parameters

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a 'Person' object with a 'population' parameter and an 'environment' object. The 'Properties' panel is open for the 'population' parameter, showing the following configuration:

- Name: population
- Checked: Show Name, Show At Runtime
- Ignored: Ignore, Public
- Type: Person
- Package: minimalistnetworkabmmodel
- Environment: environment
- Income: (empty field)
- Sex: (empty field)
- Replication: 100

The 'Action' palette on the right includes options like Action, Analysis, Presentation, Connectivity, and Enterprise Library.

# “Recipes” for Determining Agent Characteristics

**Income:** “uniform(10000,50000)”

**Sex:** “uniform\_discr(0,1)”

The screenshot shows a software interface for defining agent characteristics. The window title is "population - Person". The interface includes a sidebar with tabs for "General", "Parameters", "Statistics", and "Description". The "General" tab is active, showing the following fields:

- Name: population
- Type: Person
- Package: minimalistnetworkabnmodel
- Environment: environment
- income\*: uniform(10000,50000)
- sex\*: uniform\_discr(0,1)
- Replication: 100

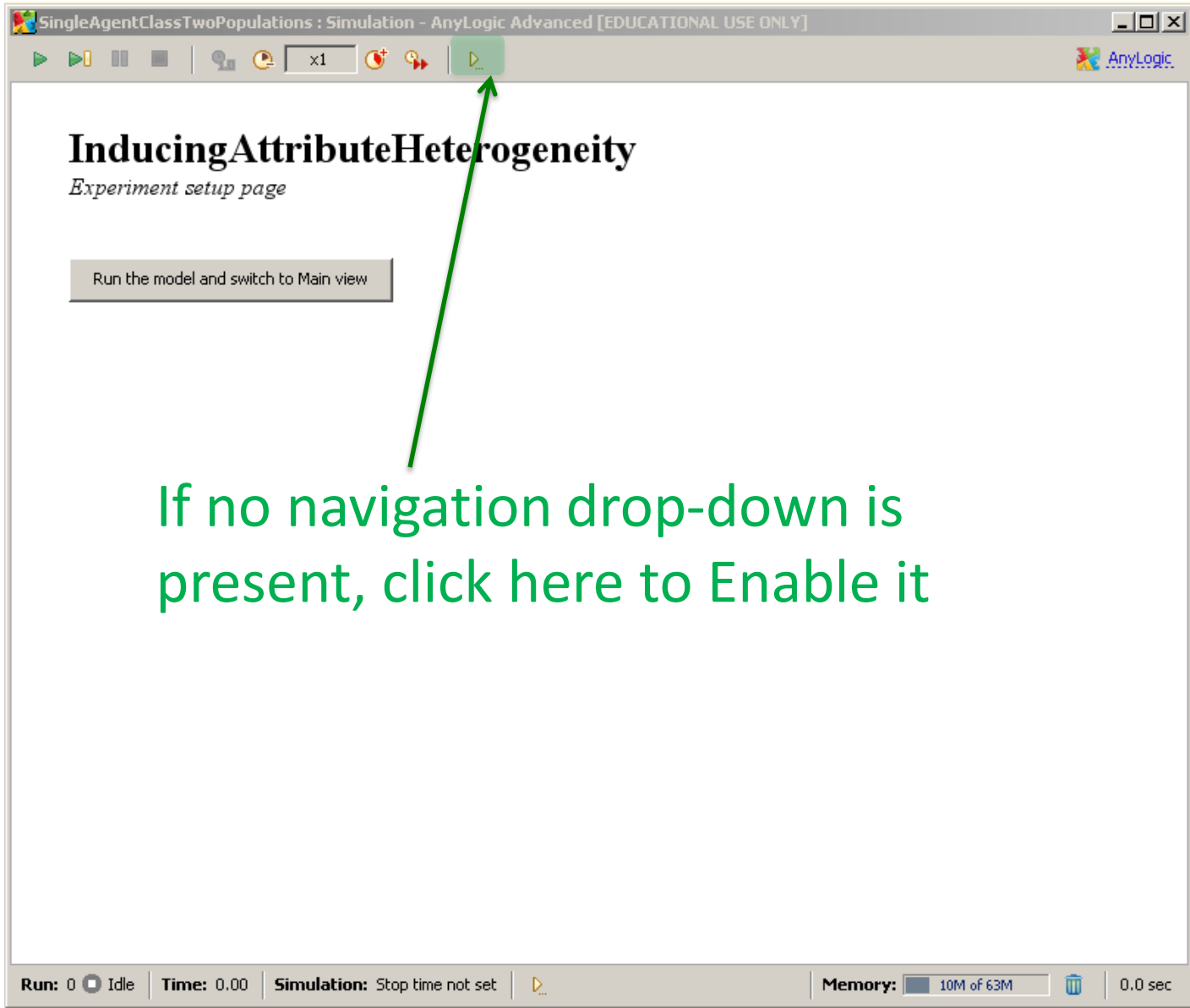
There are also checkboxes for "Show Name", "Ignore", "Public", and "Show At Runtime", along with a "Create Presentation" button. A red arrow points from the text "Income: 'uniform(10000,50000)'" to the 'income\*' field, and a green arrow points from the text "Sex: 'uniform\_discr(0,1)'" to the 'sex\*' field.

# Model Simulation Opening Screen

The screenshot shows the AnyLogic simulation interface. The title bar reads "SingleAgentClassTwoPopulations : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]". The toolbar includes a speed control set to "x1" and a dropdown menu labeled "experiment: Singl...". The main content area displays the title "InducingAttributeHeterogeneity" and the subtitle "Experiment setup page". A button labeled "Run the model and switch to Main view" is present. The status bar at the bottom shows "Run: 0", "Idle", "Time: 0.00", "Simulation: Stop time not set", "Memory: 9M of 63M", and "0.0 sec". A green arrow points from the text below to the dropdown menu.

Navigation drop-down (for browsing model elements & agents during execution)

# Model Simulation Opening Screen



The screenshot shows the AnyLogic simulation interface. The title bar reads "SingleAgentClassTwoPopulations : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main content area displays the title "InducingAttributeHeterogeneity" and the subtitle "Experiment setup page". A button labeled "Run the model and switch to Main view" is visible. A green arrow points from the text below to a green play button icon in the top toolbar. The bottom status bar shows "Run: 0", "Idle", "Time: 0.00", "Simulation: Stop time not set", "Memory: 10M of 63M", and "0.0 sec".

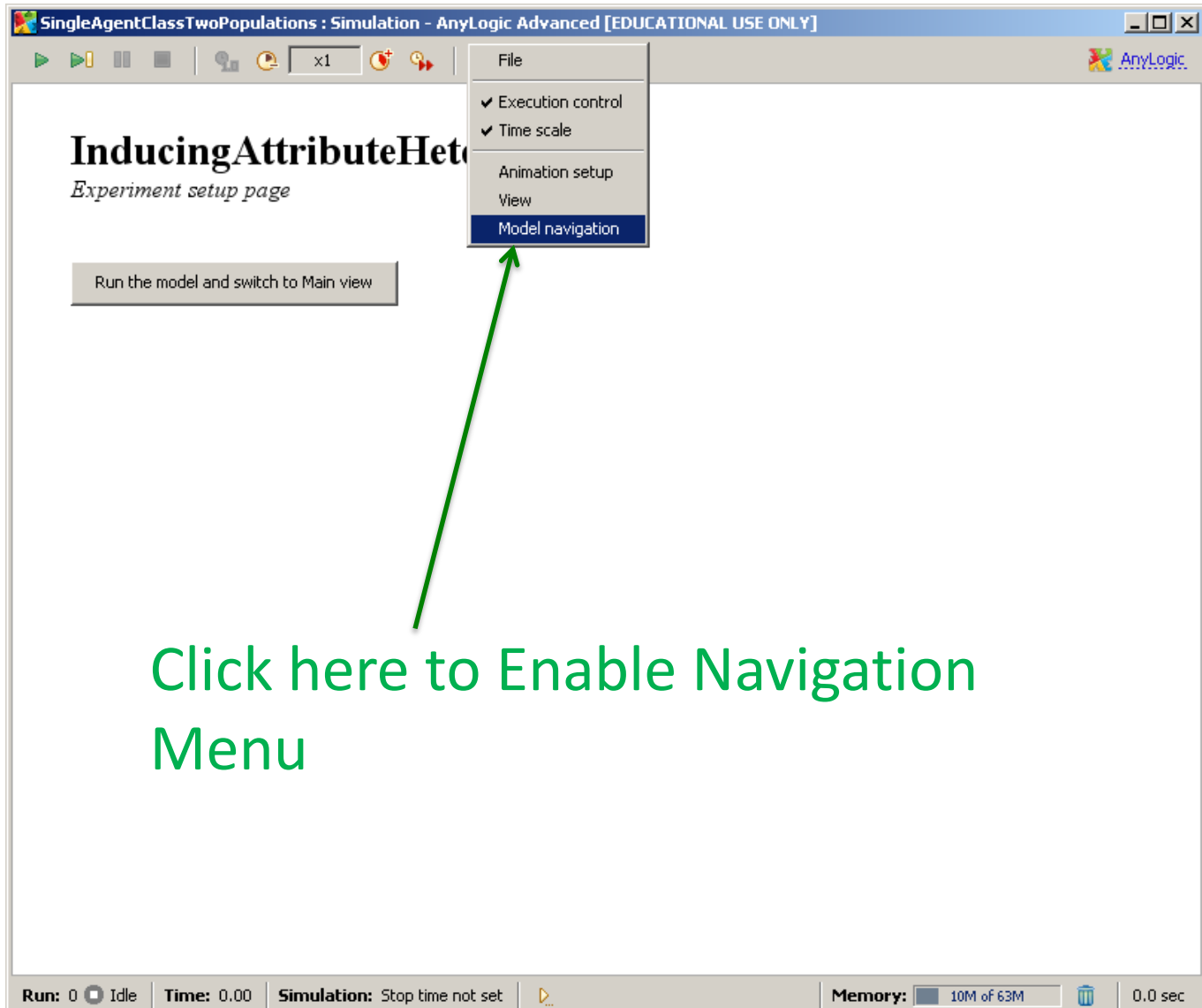
**InducingAttributeHeterogeneity**  
*Experiment setup page*

Run the model and switch to Main view

If no navigation drop-down is present, click here to Enable it

Run: 0 Idle Time: 0.00 Simulation: Stop time not set Memory: 10M of 63M 0.0 sec

# Turning on Model Navigation



# Browsing Attributes of Population Members

The screenshot displays the AnyLogic simulation environment. The window title is "SingleAgentClassTwoPopulations : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]". The interface includes a toolbar with various simulation controls and a dropdown menu showing "root:Main". A pop-up window is open over the "root:Main" element, displaying the attribute "population[0..99]".

The simulation area shows a network of agents represented by circles connected by lines. A legend in the top-left corner identifies the elements:

- population Person [100]
- environment 100 agents

The status bar at the bottom provides simulation details:

- Run: 0 Running
- Time: 43.60
- Simulation: 44%
- Memory: 11M of 63M
- 46.4 sec



# Navigation During Model Execution

The screenshot displays the AnyLogic simulation environment for a model named "SingleAgentClassTwoPopulations". The interface includes a toolbar with navigation controls (play, pause, stop, reset) and a zoom level of "x1". The main workspace shows a complex network diagram with nodes and edges. A tooltip is visible over a node, showing the path "root:Main" and "population[0..99]". On the left side, a legend identifies "population" as a class with 100 instances and "environment" as a class with 100 agents. The status bar at the bottom indicates the simulation is "Running", has taken "43.60" seconds, and is at "44%" completion. It also shows "Memory: 11M of 63M" and a runtime of "46.4 sec".

# Navigating to View Particular Agents

## This shows the Attribute Values

The screenshot displays the AnyLogic simulation environment. The window title is "SingleAgentClassTwoPopulations : Simulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]". The toolbar includes navigation and control icons, a zoom level of "x1", and a dropdown menu showing "population[0]". A status bar at the top right indicates "Index of the active object [0..99]".

In the main workspace, two attributes are listed for the selected agent:

- income: 39,235.128
- sex: 0

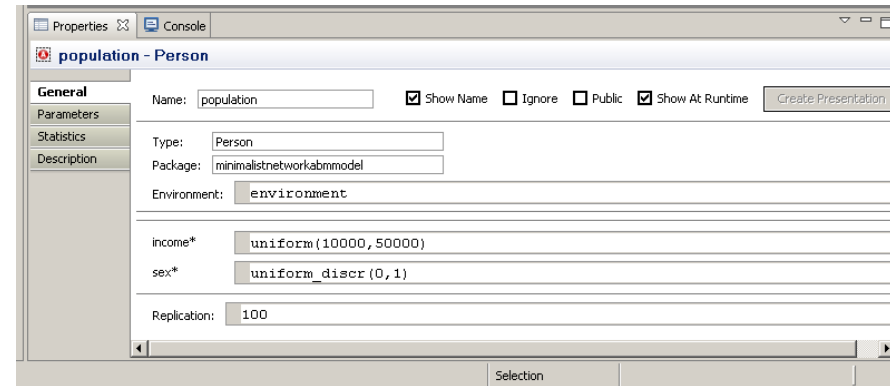
A mouse cursor is positioned over a small circle on the workspace, indicating the selected agent.

The bottom status bar shows the following information:

- Run: 0 Finished
- Time: 100.00
- Simulation: 100%
- Memory: 11M of 63M
- 103.9 sec

# Model-Wide Parameters

- Values for agent parameters are specified by the associated Population
- We can also associate parameters with the “Main” class
  - These parameters can be model-wide quantities (e.g. the size of the population, or the duration of infectiousness to assume for all agents)
  - Values for these parameters are specified by *Experiments*



# Adding a Model-Wide Parameter

Click on the "Model" label in the "Palette" window

- 1) Click here ("Parameter")
- 2) Click somewhere on the canvas

populationSize - Parameter

**General**

Name:   Show Name  Ignore  Public  Show At Runtime

Type:  void (just action)  boolean  int  double  String  Other: **double**

Default Value:

Dynamic  Save in snapshot

On Change:

# Set the *Default Value* of the Parameter

The screenshot displays the AnyLogic Advanced interface. On the left, a project tree shows the hierarchy: InducingAttributeHeterogeneity\* > Main > Person > Simulation: Main. The main workspace shows a diagram with a parameter 'populationSize' highlighted. The Properties panel at the bottom is open to the 'populationSize - Parameter' configuration. The 'Name' field is 'populationSize'. The 'Type' is set to 'int', and the 'Default Value' is '100'. A red arrow points from the text 'Default Value: Use "100"' to the '100' in the Default Value field. A blue arrow points from the text 'Type: Use "int" (whole numbers)' to the 'int' radio button.

Default Value: Use "100"

Type: Use "int" (whole numbers)

populationSize - Parameter

Name: populationSize  Show Name  Ignore  Public  Show At Runtime

Type:  void (just action)  boolean  int  double  String  Other: int

Default Value: 100

Dynamic  Save in snapshot

On Change:

# Setting the Population Size to be Determined by the Parameter "populationSize"

The screenshot shows the AnyLogic Advanced software interface. The main workspace displays a diagram with three elements: "population [...]", "environment", and "populationSize". A blue arrow points to the "population [...]" element with the text "Select the 'population'".

Below the workspace, the "Properties" window is open for the "population - Person" element. The "General" tab is selected, showing the "Name" field set to "population". The "Replication" section at the bottom has a dropdown menu open, showing a list of options. The "populationSize" option is highlighted in green, and a green arrow points to it with the text "May wish to use Auto-Completion (Control-Space)".

On the left side of the interface, there is a "Project" browser showing a tree structure with "Main", "Person", and "Simulation: Main". Below it is a "Problems" browser with a search bar and a table with columns "Description" and "Location".

Red text annotations are present on the left side: "Replication: Use 'population Size' (check any 'Replicated' checkbox)". A red arrow points from this text to the "Replication" dropdown menu in the Properties window.

# Variables: Dynamic Quantities

- Variables are used for time-varying quantities
- Note that some variables (e.g. stocks) are defined using other “primitive” objects directly supported by AnyLogic
- As with parameters, variables support many types
- If we want to create an instance variable with a particular class, we should do it with a variable
  - Declaring things using variables (rather than in code) gives us the option of browsing these things at runtime

# Experiments Now Have Field to Specify Parameter Value (populationSize)

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows the 'InducingAttributeHeterogeneity' experiment setup page. A red arrow points from the text 'This specifies "population Size"' to the 'populationSize' field in the 'Simulation - Simulation Experiment' configuration window. The field contains the value '100'. The configuration window also shows other settings like 'Name: Simulation', 'Main active object class (root): Main', and 'Seed Value: 1'.

Run the model and switch to Main view

This specifies "population Size"

Simulation - Simulation Experiment

General

Name: Simulation Main active object class (root): Main  Ignore

Random number generation:

Random seed (unique simulation runs)

Fixed seed (reproducible simulation runs) Seed Value: 1

populationSize 100

Paste from clipboard



# Add a New Experiment

The screenshot shows the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace displays a project named "InducingAttributeHeterogeneity" on a grid background. A right-click context menu is open over the project name, with the "New" option selected, which has opened a sub-menu. In this sub-menu, the "Experiment" option is highlighted. A red arrow points from the red text overlay to the "Experiment" option. The Properties panel at the bottom shows the details for the "InducingAttributeHeterogeneity - Model", including its name, package, and file path.

Right-click on project name, select New menu, and then "Experiment"

InducingAttributeHeterogeneity - Model

General

Name:

Dependencies

Package:

Description

File:

# Name the New Experiment

## “LargePopulation”

**New Experiment**

**Experiment**

Select an experiment type, specify a name and choose a root (top-level) active object.

Name:

Main Active Object Class (root):

Experiment Type:

- Simulation
- Optimization
- Parameters Variation
- Compare Runs (available in Professional edition)
- Monte Carlo (available in Professional edition)
- Sensitivity Analysis (available in Professional edition)
- Calibration (available in Professional edition)
- Custom (available in Professional edition)

Performs model runs with specified parameters, supports virtual and real-time modes, animation, model debugging

Copy model time settings from :

< Back   Next >   Finish   Cancel

# Setting the New Experiment Assumptions

The screenshot shows the AnyLogic Advanced software interface. The main workspace displays the 'InducingAttributeHeterogeneity' experiment setup page. A red text overlay reads: "For 'population Size', Use '500'". A red arrow points from this text to the 'populationSize\*' field in the 'LargePopulation - Simulation Experiment' configuration panel, which is highlighted with a red background and contains the value '500'. The configuration panel also shows the 'Name' as 'LargePopulation', 'Main active object class (root)' as 'Main', and 'Seed Value' as '1'. The 'Random number generation' section has 'Fixed seed (reproducible simulation runs)' selected.

AnyLogic Advanced [EDUCATIONAL USE ONLY]

File Edit View Model Window Help

Project x

InducingAttributeHeterogeneity\*

- Main
- Person
- LargePopulation: Main

Problems x Search

Description	Location
-------------	----------

Person Main Simulation LargePopulation x

## InducingAttributeHeterogeneity

Experiment setup page

Run the model and switch to Main view

**For "population Size", Use "500"**

Properties Console

### LargePopulation - Simulation Experiment

**General**

Name: LargePopulation Main active object class (root): Main  Ignore

Advanced

Model Time

Presentation

Window

Parameters

Description

Random number generation:

Random seed (unique simulation runs)

Fixed seed (reproducible simulation runs) Seed Value: 1

populationSize\* 500

Paste from clipboard

LargePopulation - SimulationExperiment

# Run the New Experiment

The screenshot displays the AnyLogic software interface for a simulation titled "InducingAttributeHeterogeneity : LargePopulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main workspace shows a dense, interconnected network graph with numerous nodes and edges. On the left side, there is a hierarchical tree view with three visible items: "Environment" (containing 200 agents), "population", and "agent". The top toolbar includes standard simulation controls (play, pause, stop, reset) and a dropdown menu currently set to "root:Main". The bottom status bar provides real-time simulation data: "Run: 0" (with a green play icon), "Running" (with a green play icon), "Time: 8.55", "Simulation: 9%" (with a blue progress bar), "Memory: 10M of 63M" (with a blue progress bar), and "8.9 sec".

# Save Model As...

- Use “Save As” on the file menu to save the model as “InducingAttributeHeterogeneity”.