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

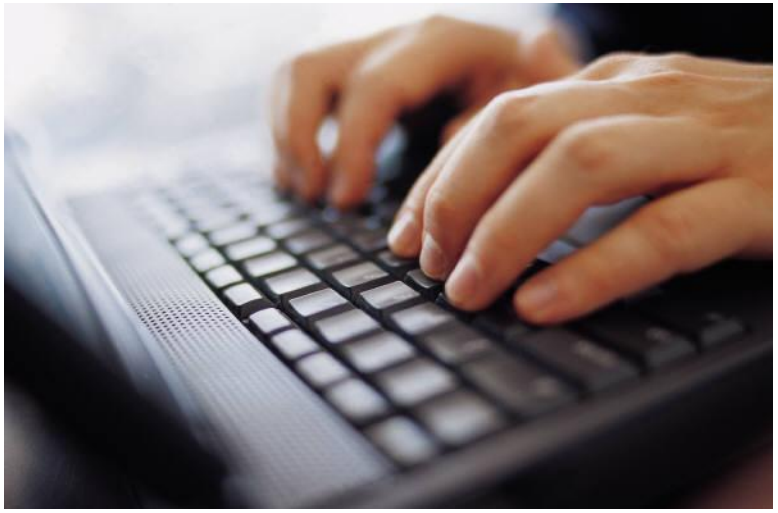
Parameters: Static Quantities

- 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



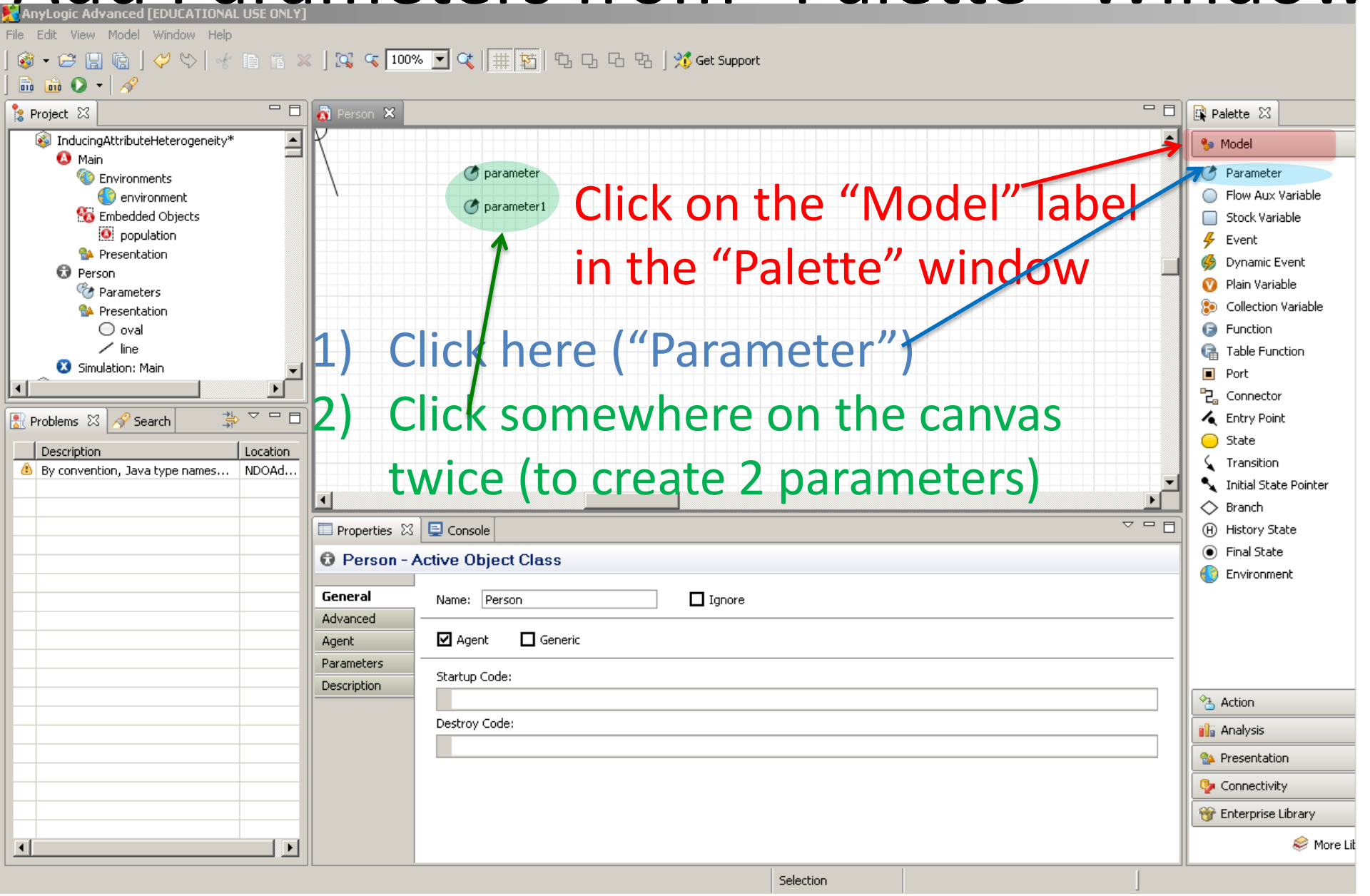
Load Previous Built [& Provided] Model:

MinimalistNetworkABMModel

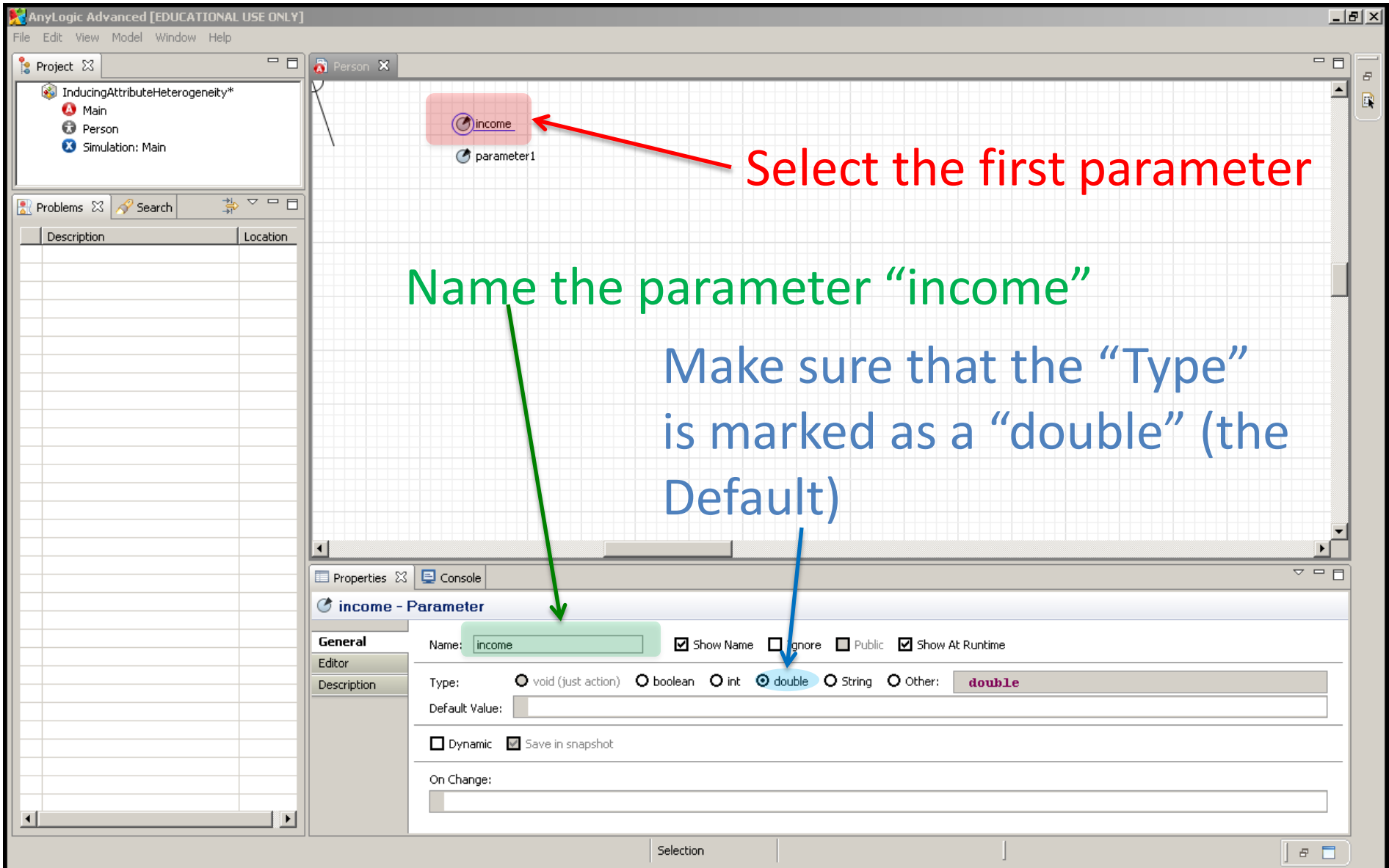
Load in Previously Saved "MinimalistNetworkABMModel"

- Pre-built model is also available

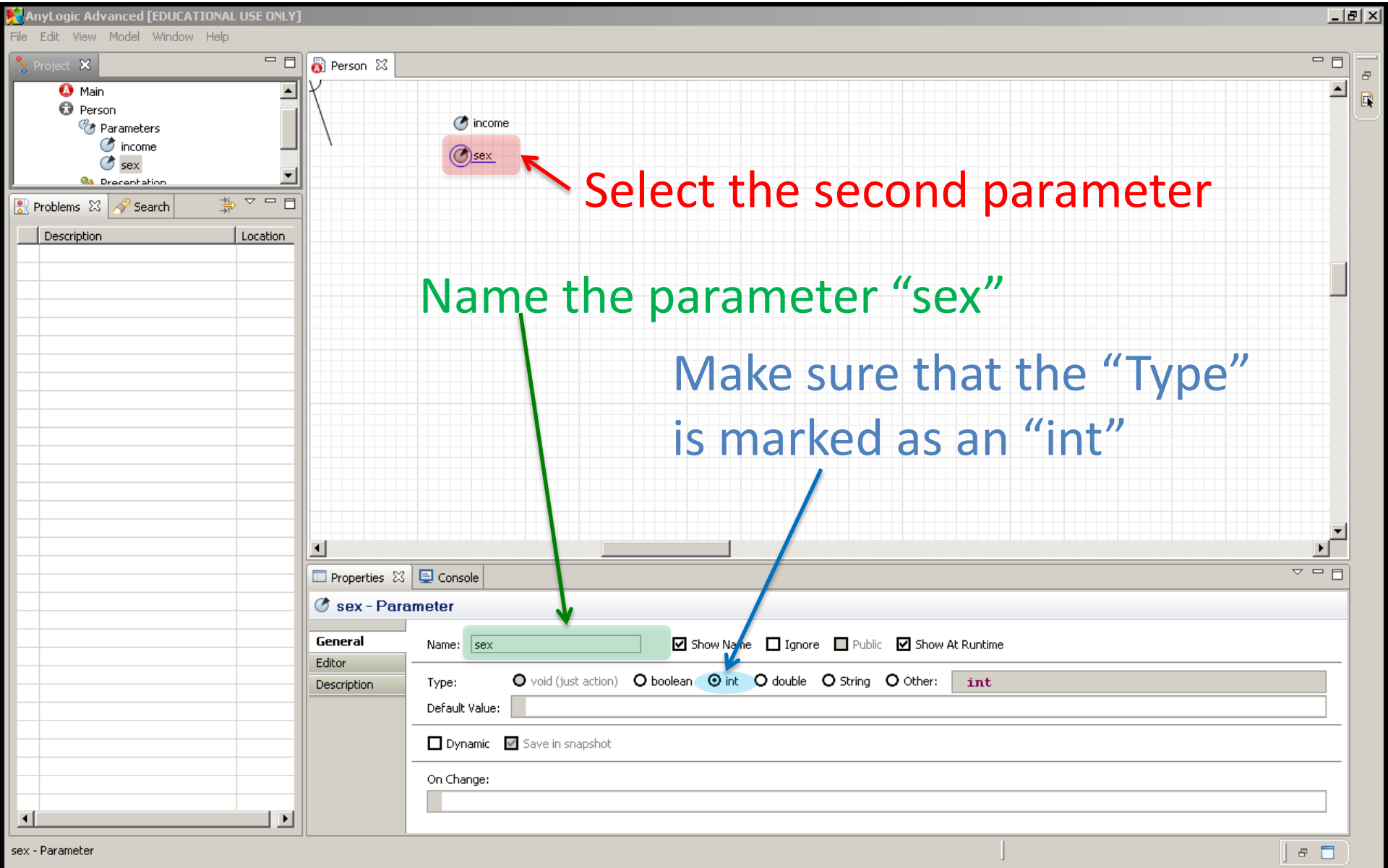
Add Parameters from “Palette” Window



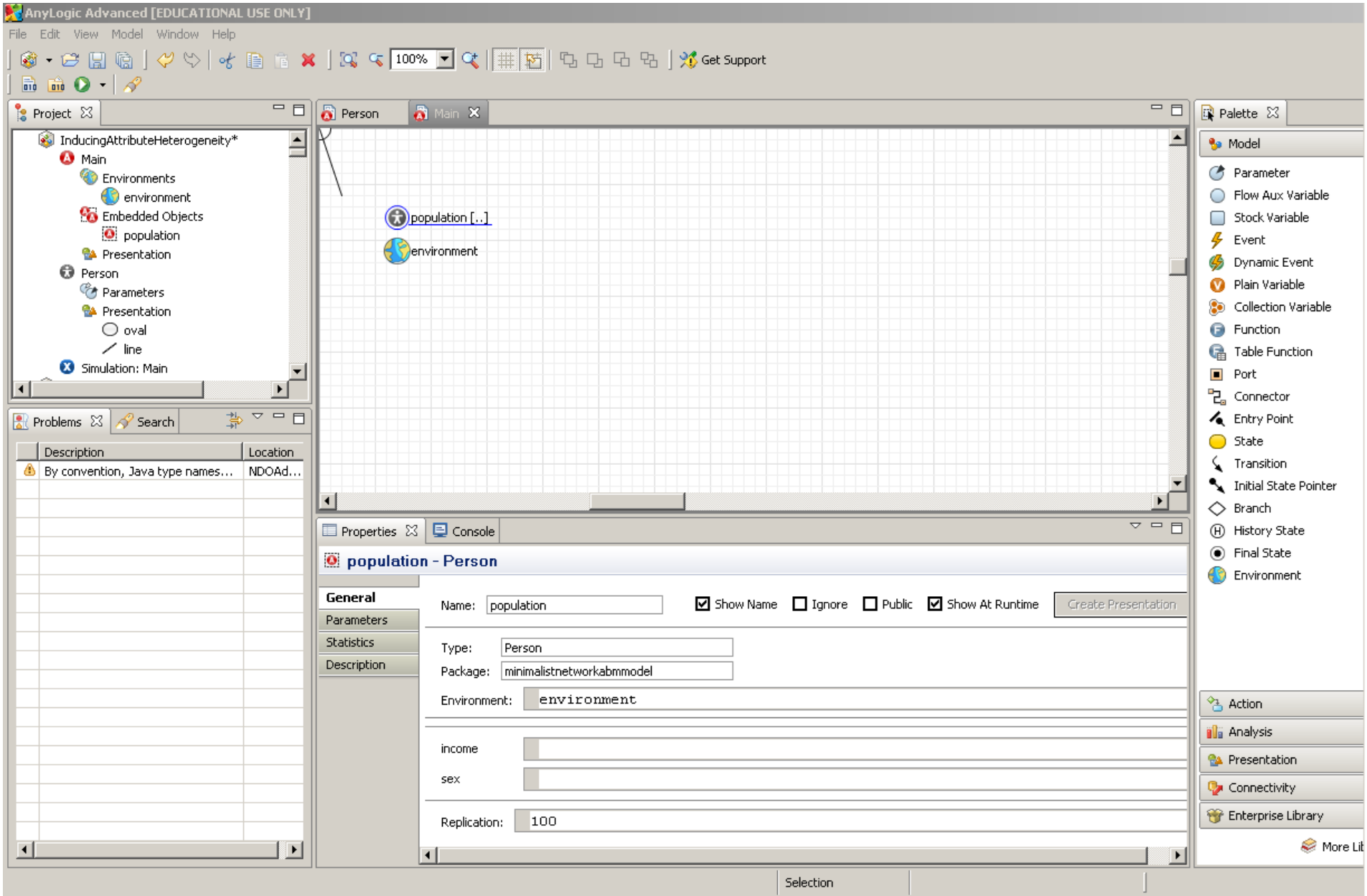
Setting the 1st Parameter Characteristics



Setting the 2nd Parameter Characteristics



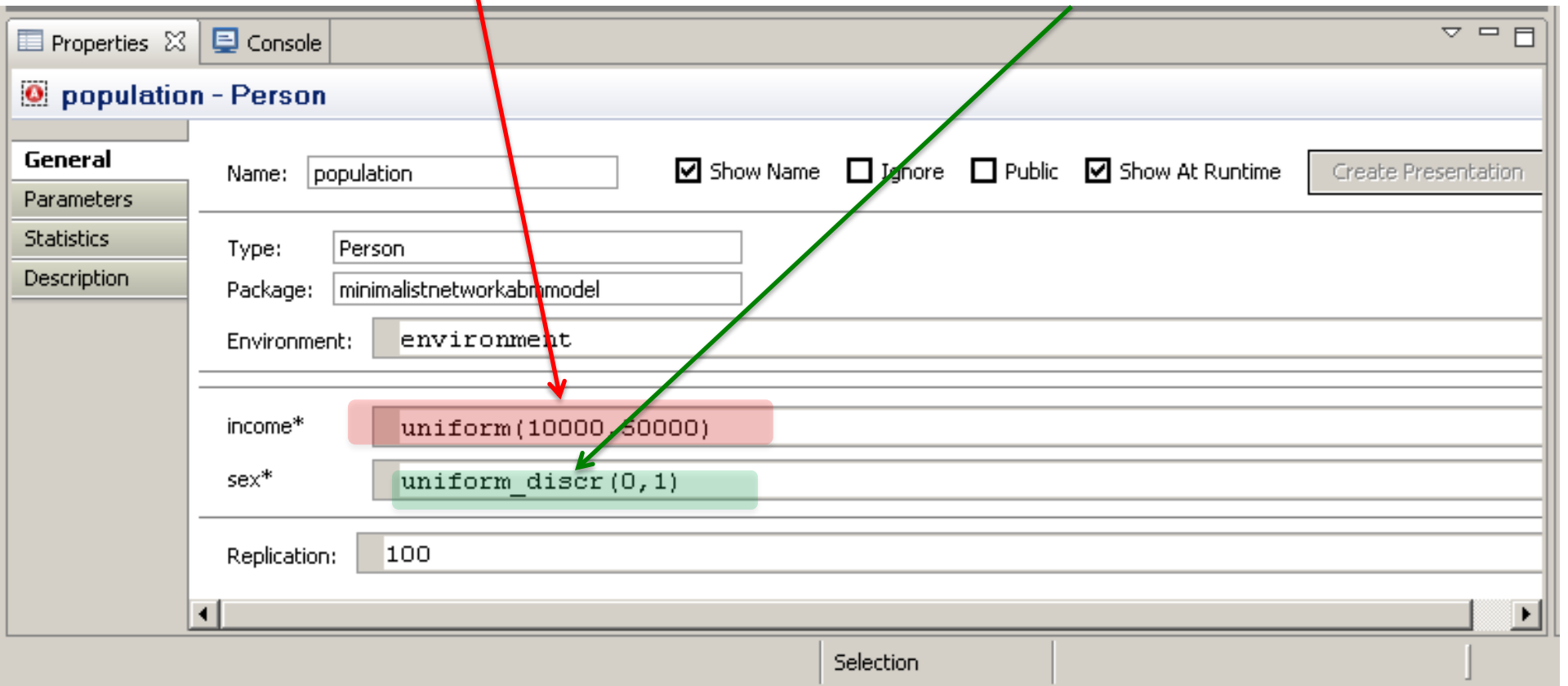
“Population” Properties Now Include Parameters



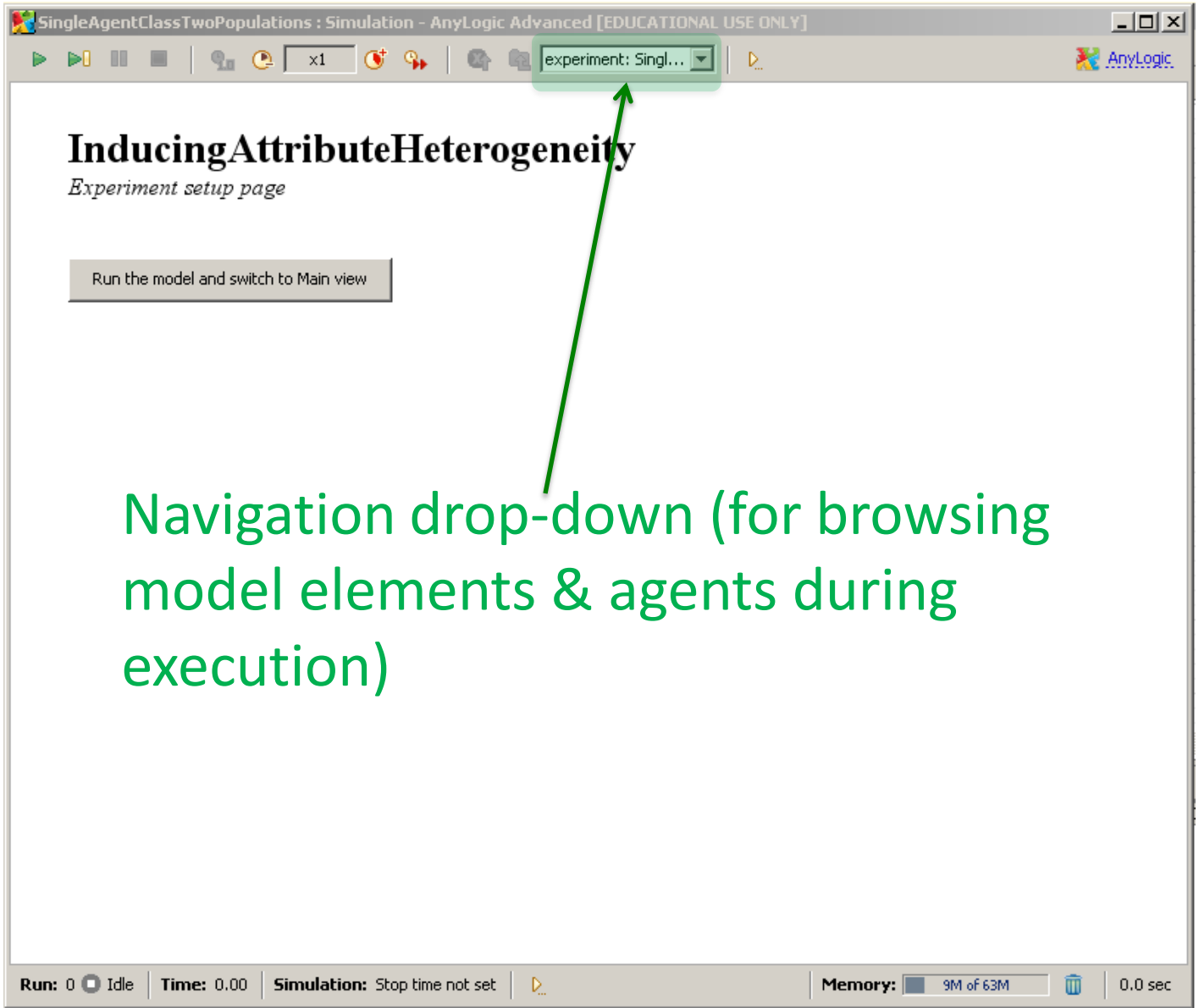
“Recipes” for Determining Agent Characteristics

Income: “uniform(10000,50000)”

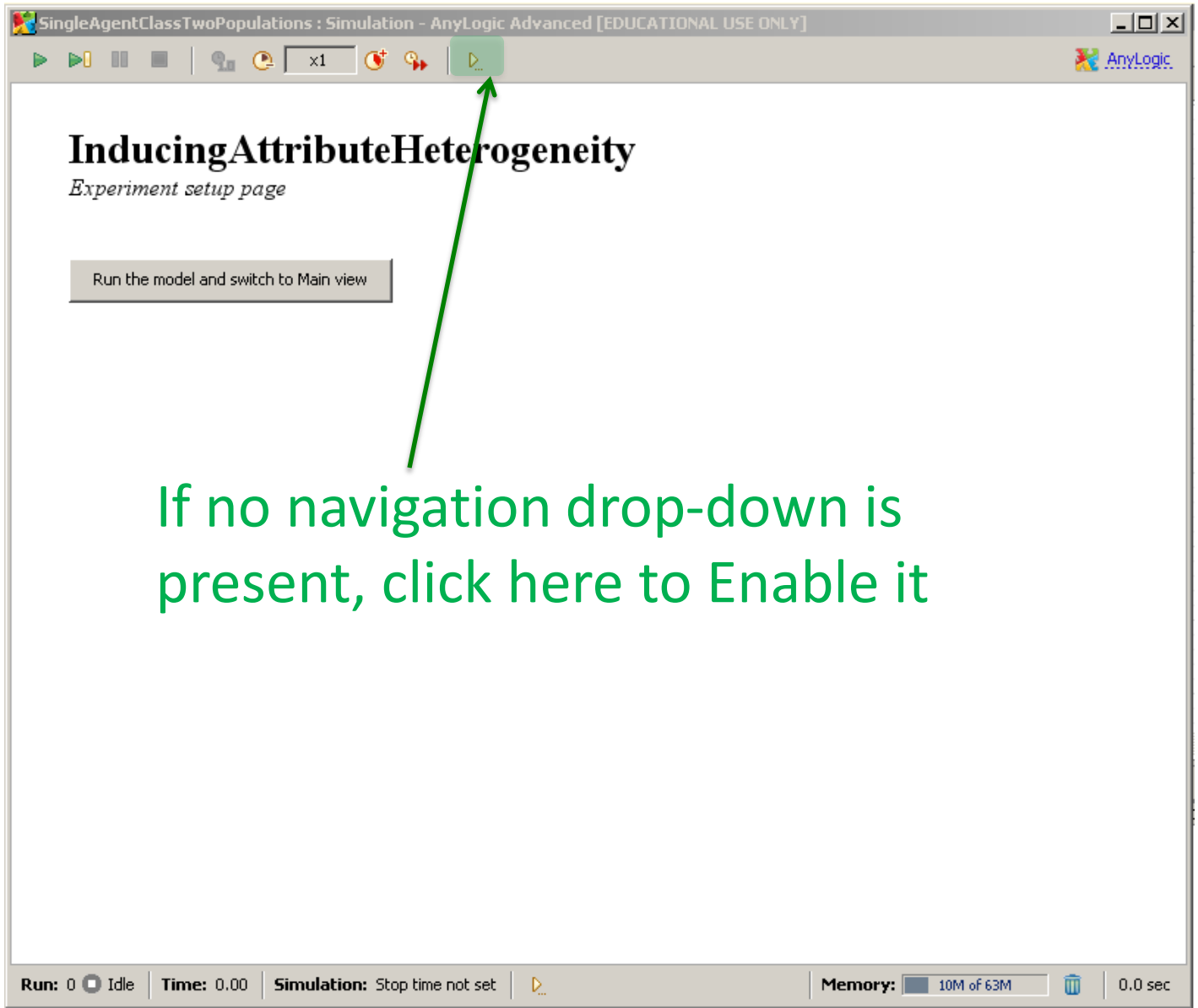
Sex: “uniform_discr(0,1)”



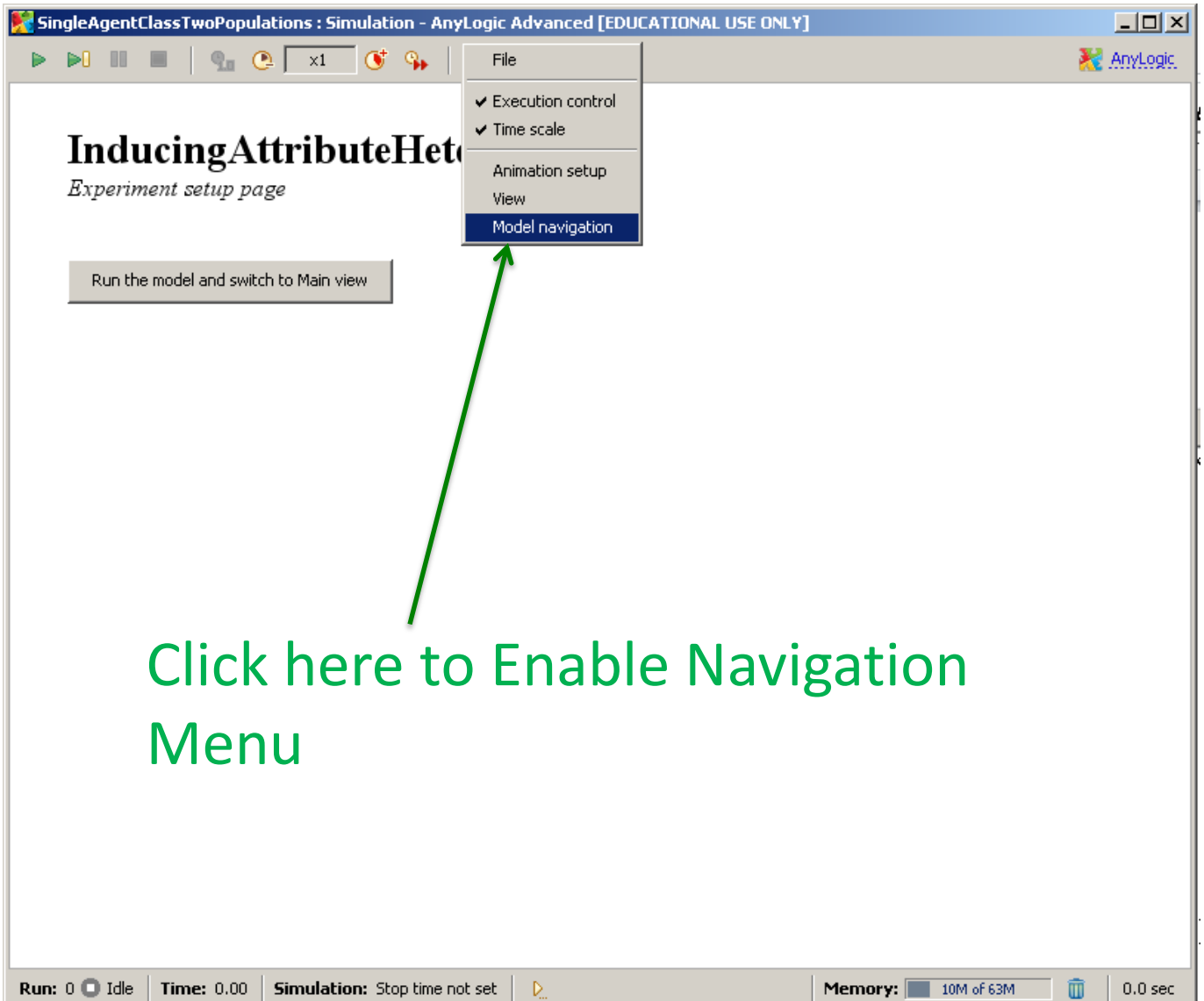
Model Simulation Opening Screen



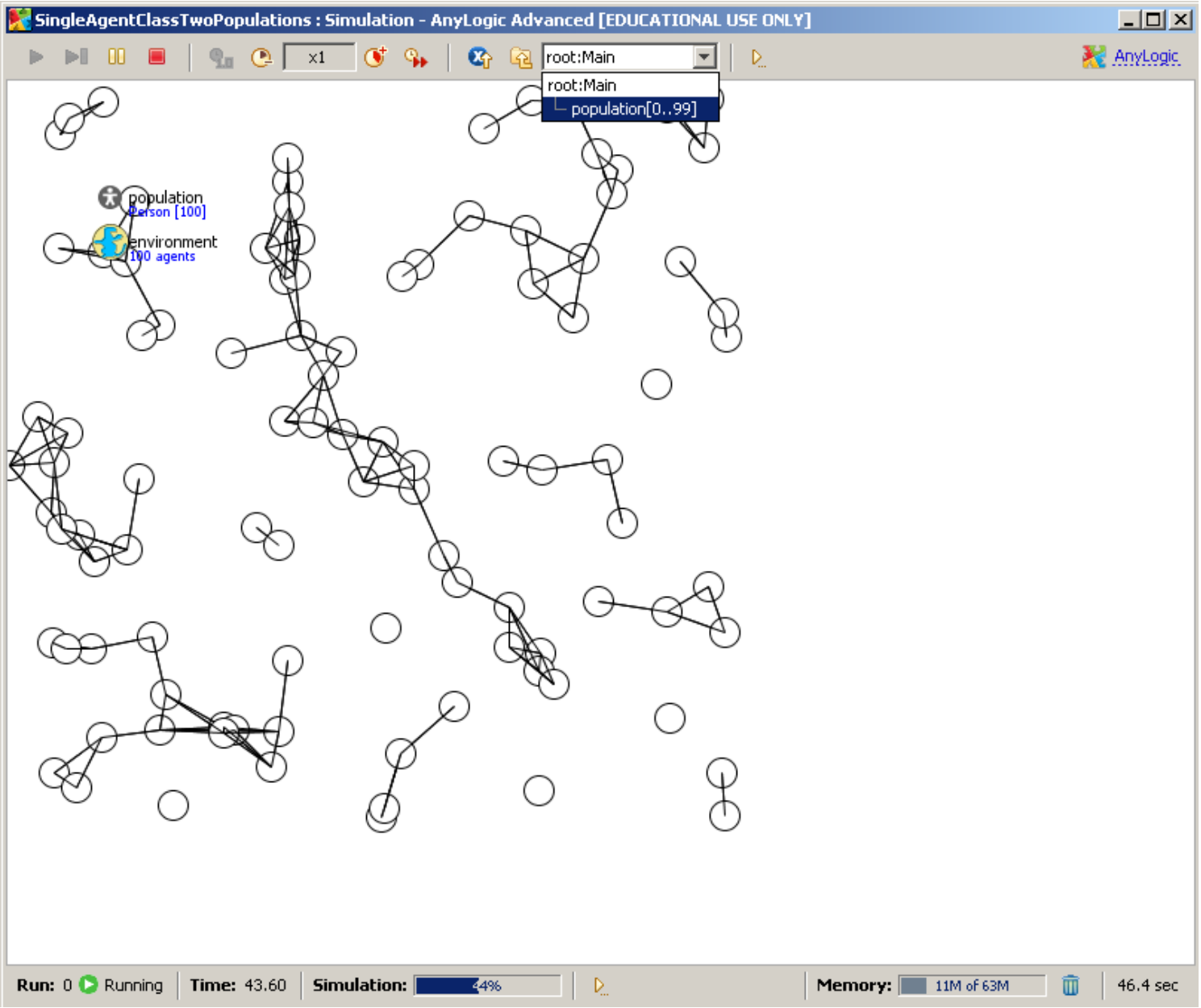
Model Simulation Opening Screen



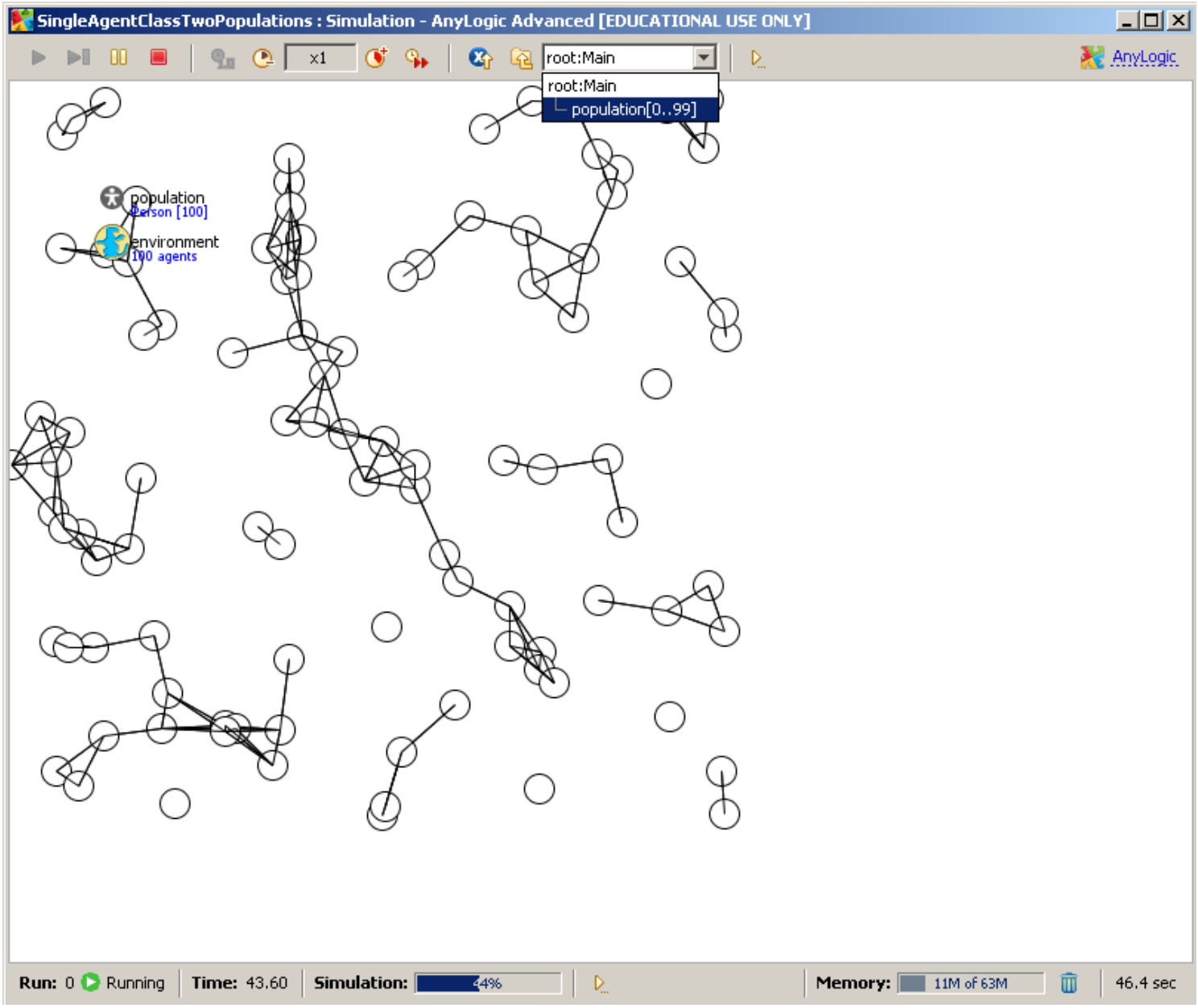
Turning on Model Navigation



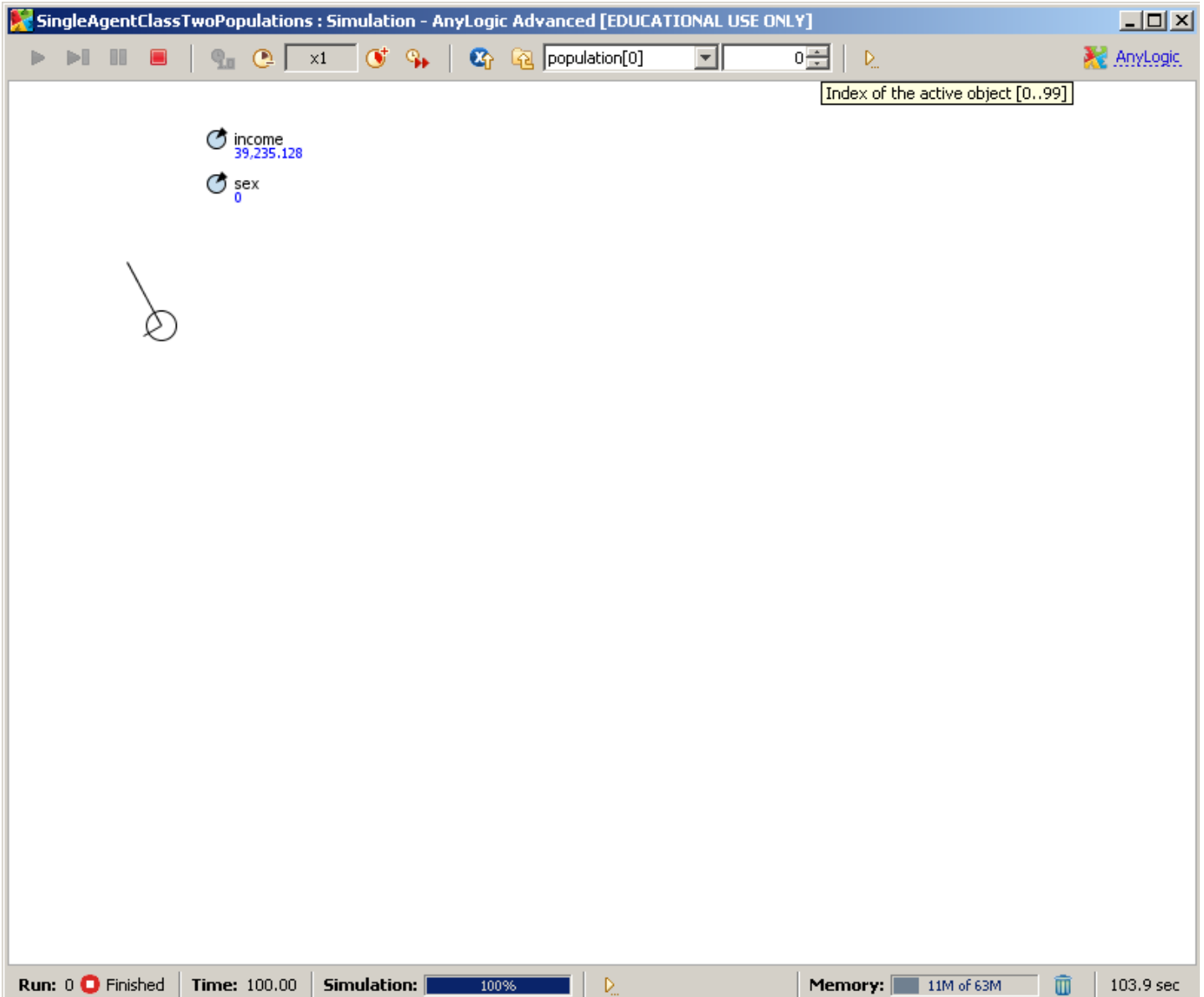
Browsing Attributes of Population Members



Navigation During Model Execution

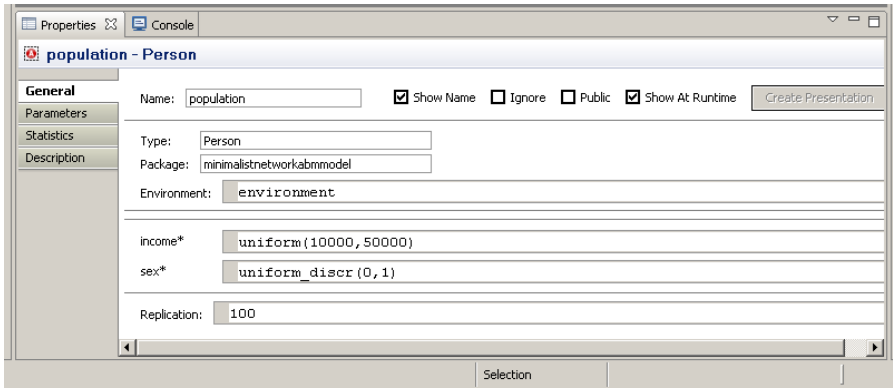


Navigating to View Particular Agents This shows the Attribute Values



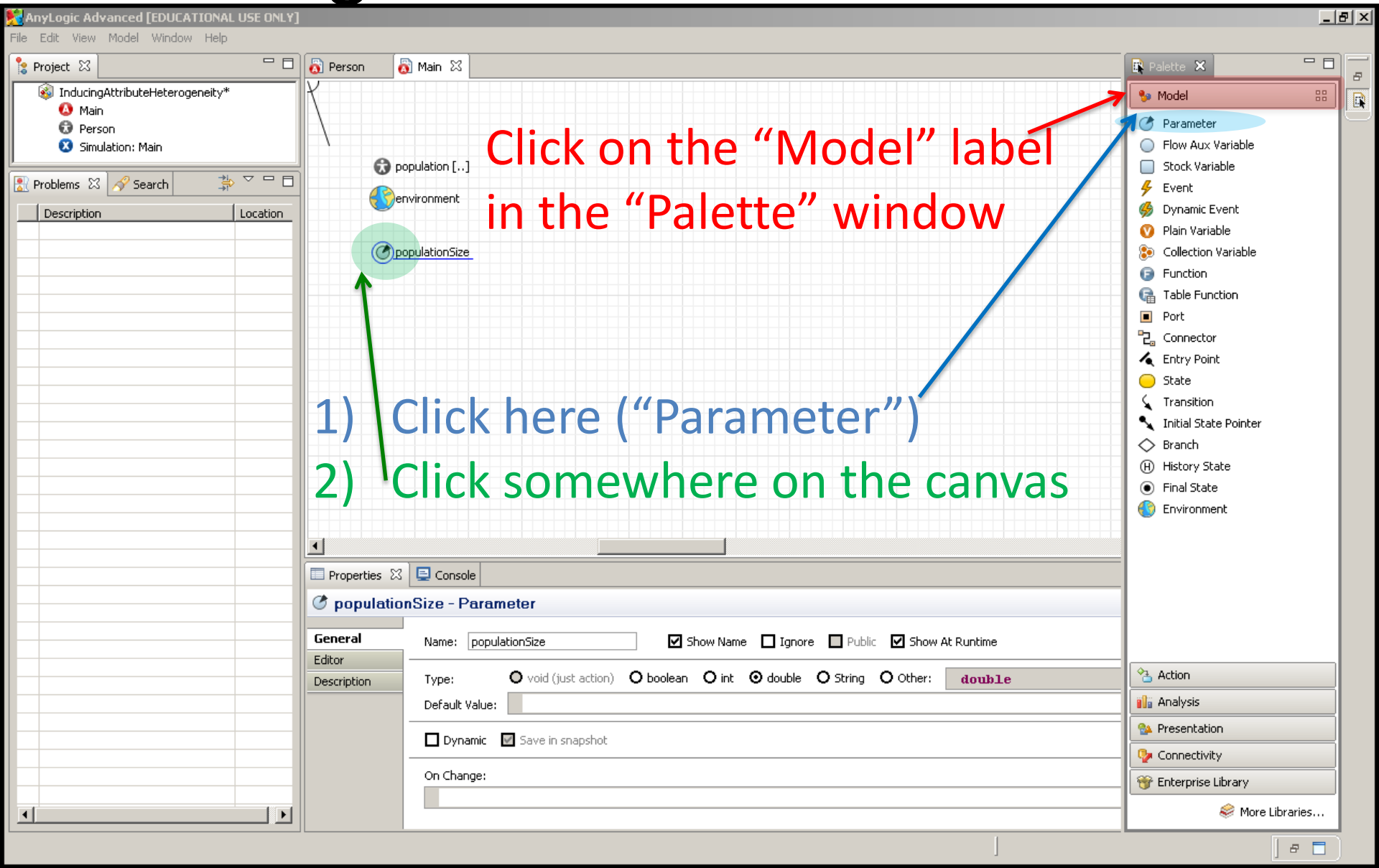
Model-Wide Parameters

- Values for agent parameters are specified by the associated Population

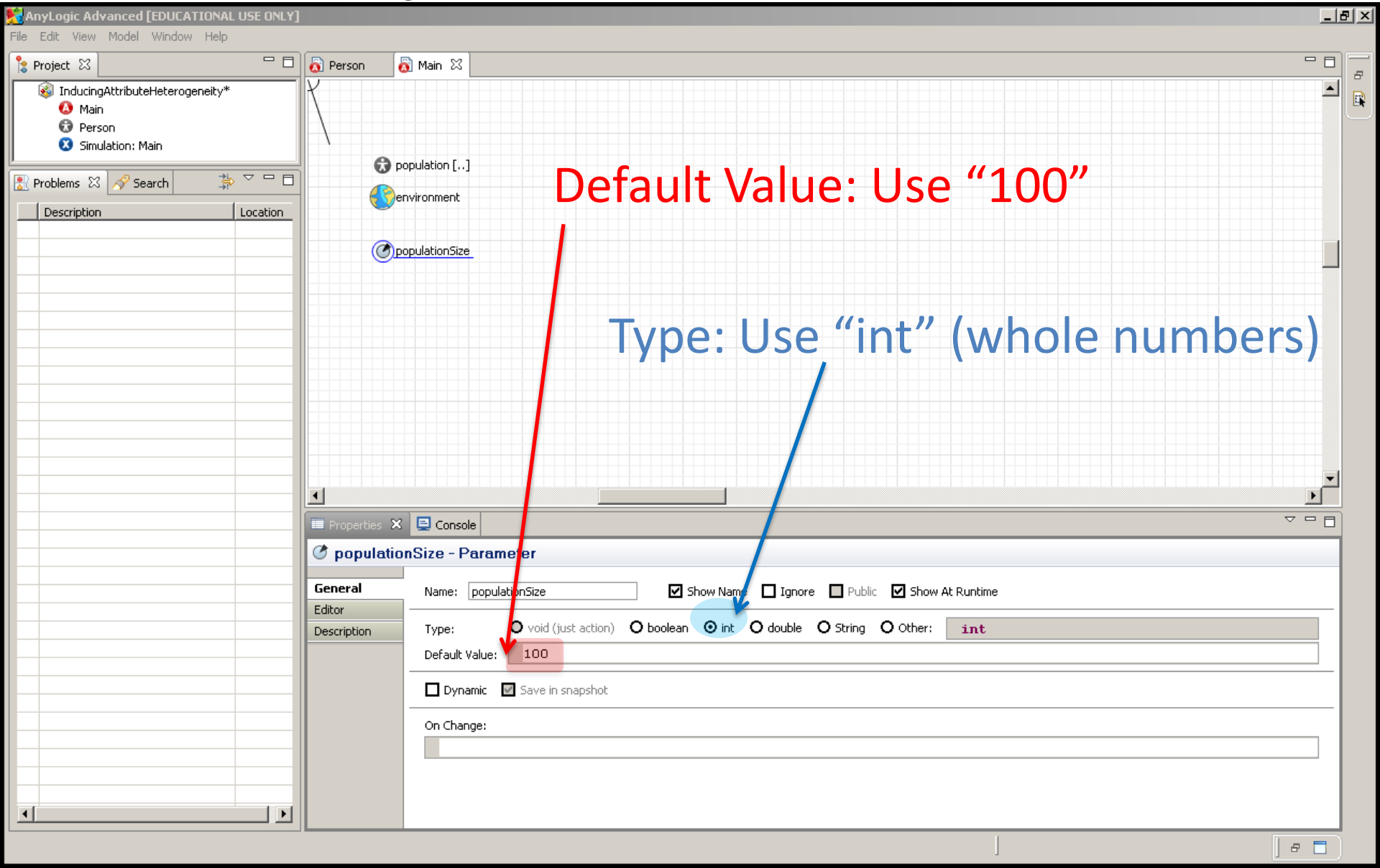


- We can also associate parameters with the “Main” class
  - These parameters can be model-wide quantities (e.g. the size of the population, or the duration of infectiousness to assume for all agents)
  - Values for these parameters are specified by *Experiments*

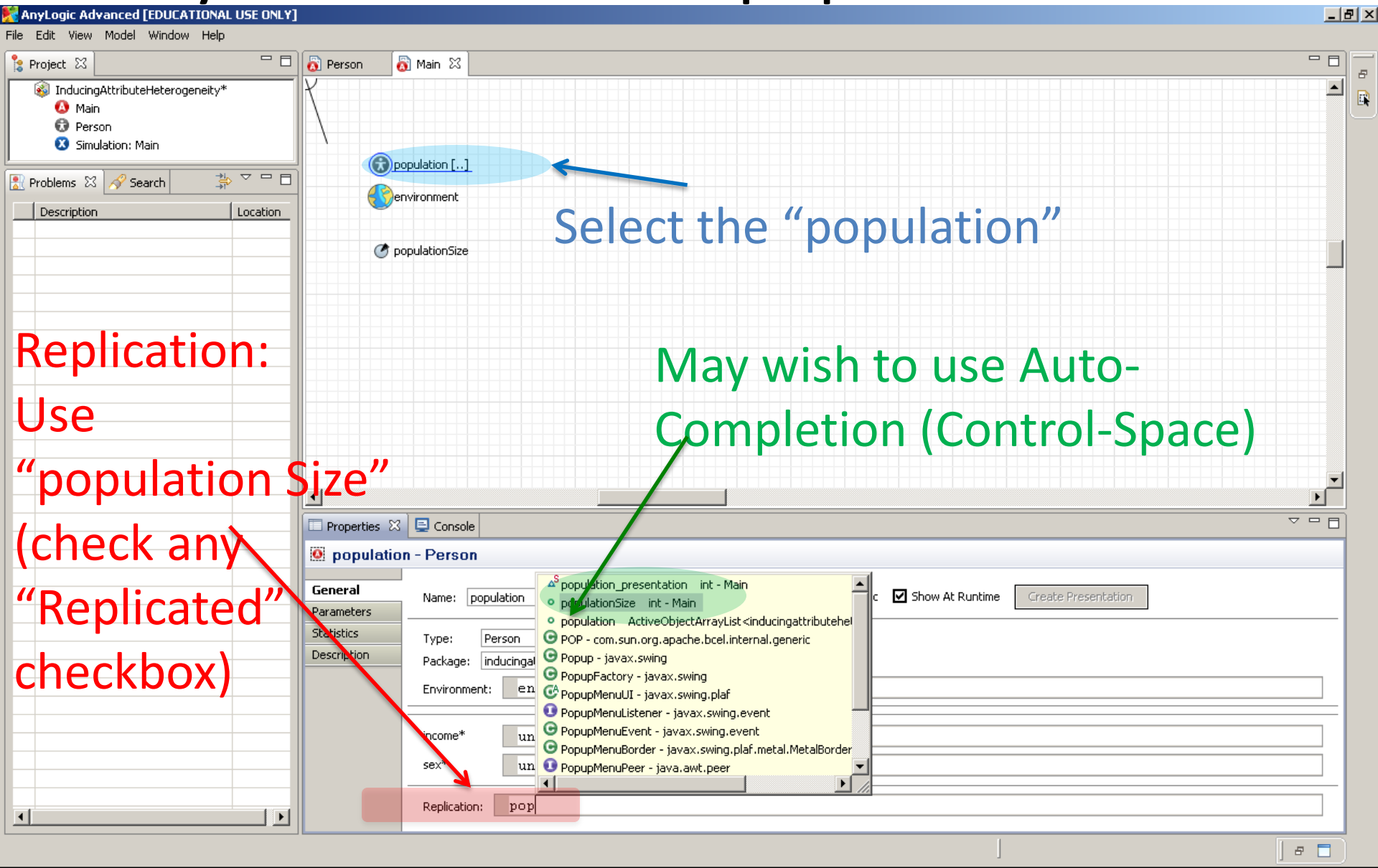
Adding a Model-Wide Parameter



Set the *Default* Value of the Parameter



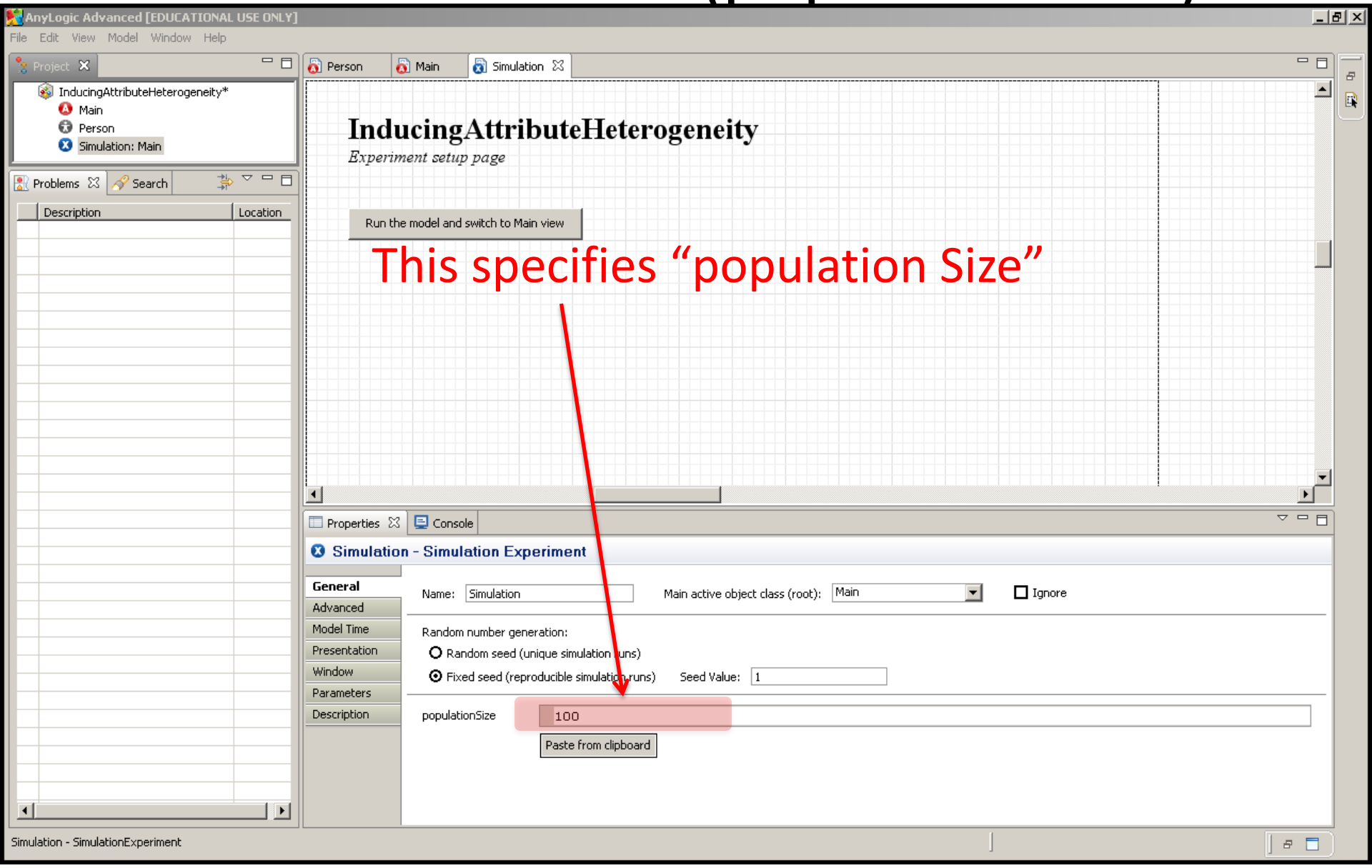
Setting the Population Size to be Determined by the Parameter “populationSize”



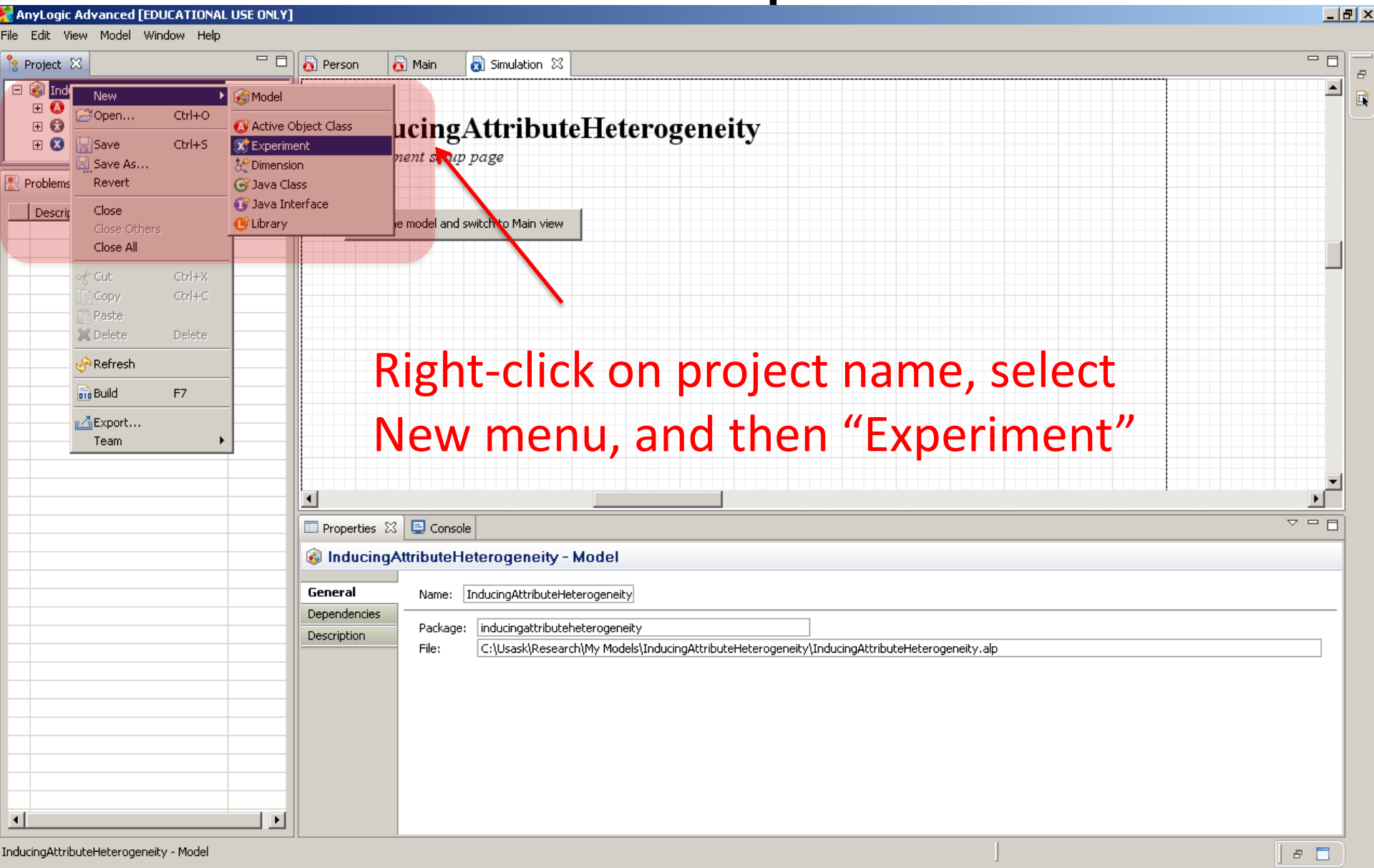
Variables: Dynamic Quantities

- 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

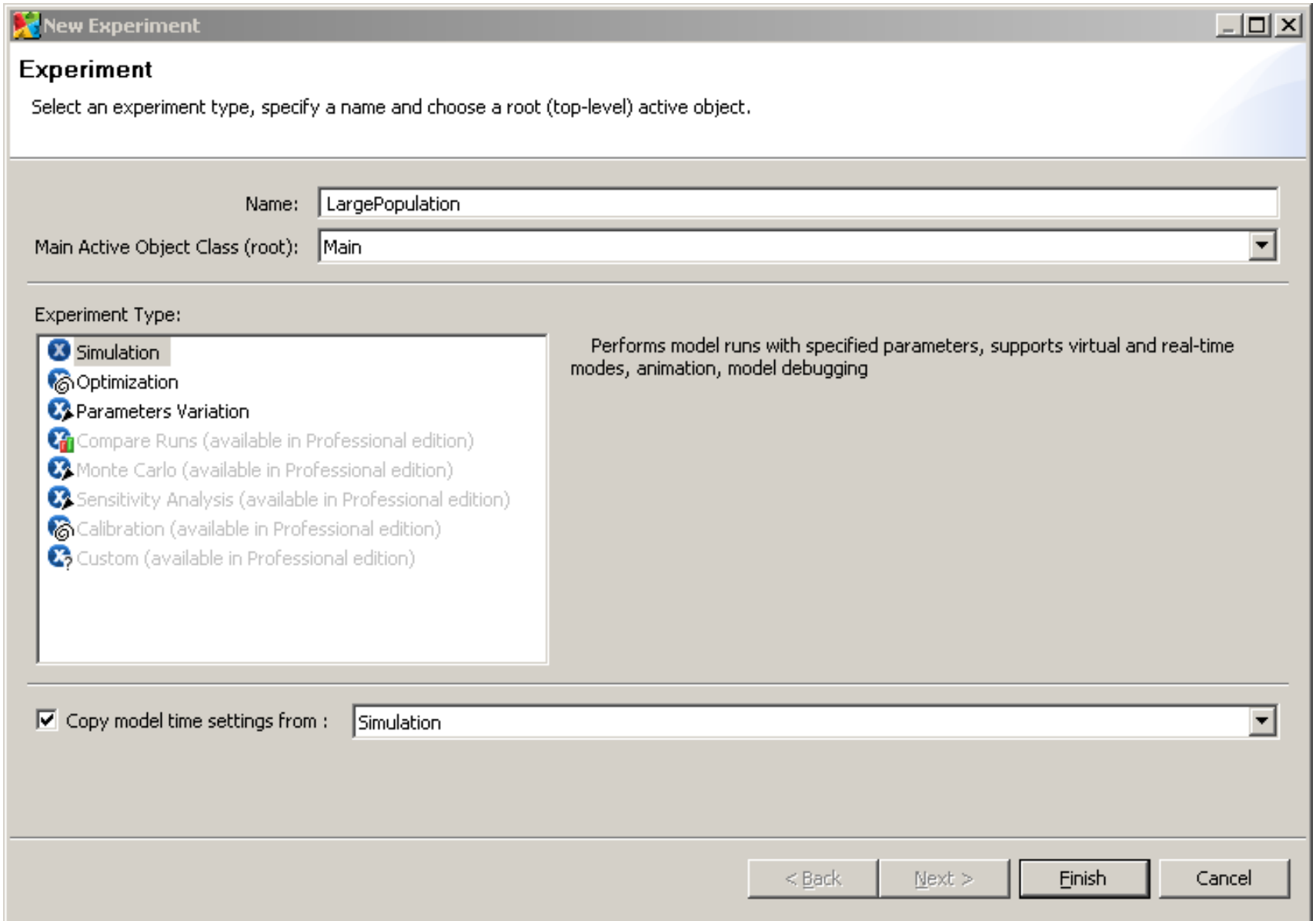
Experiments Now Have Field to Specify Parameter Value (populationSize)



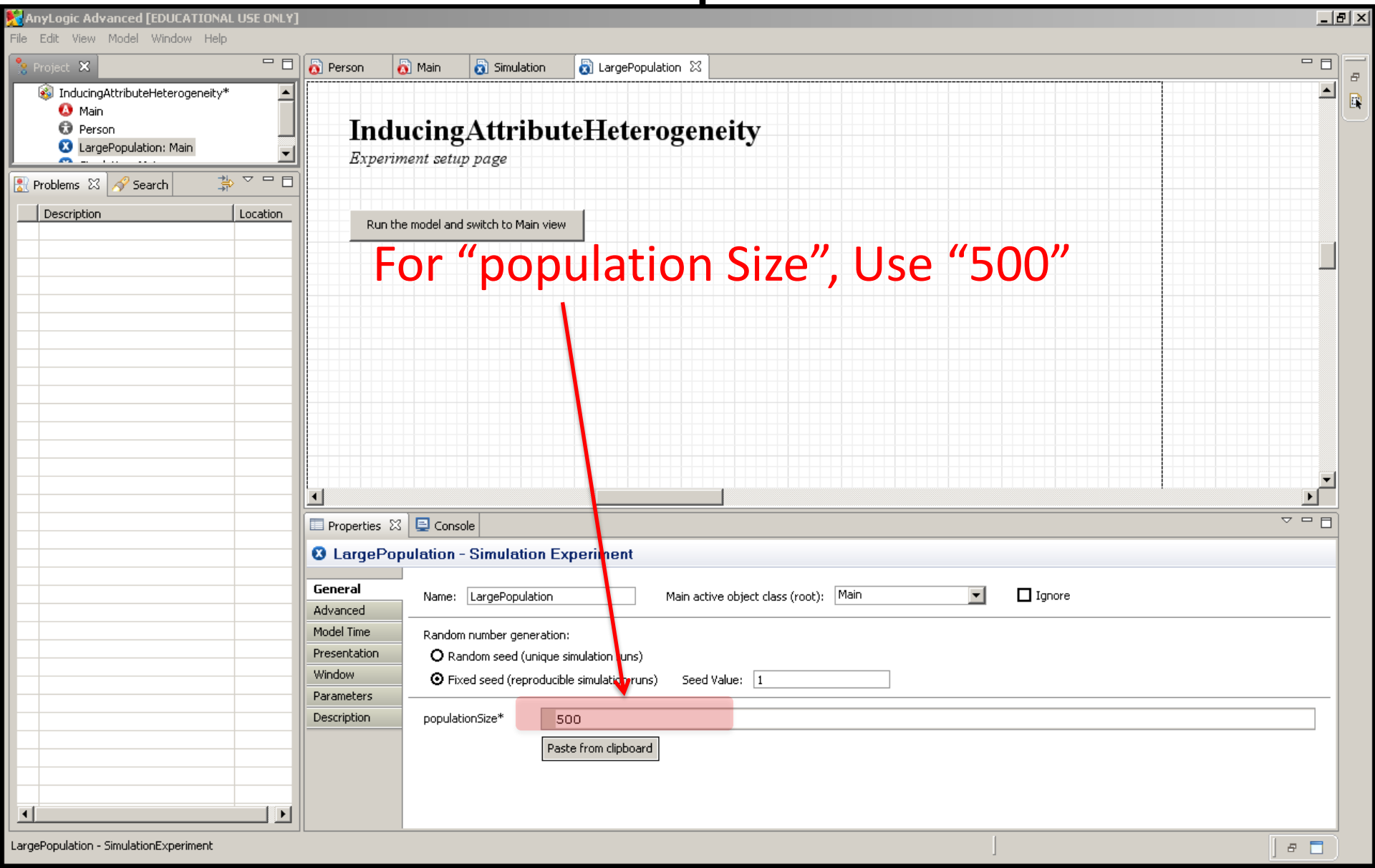
Add a New Experiment



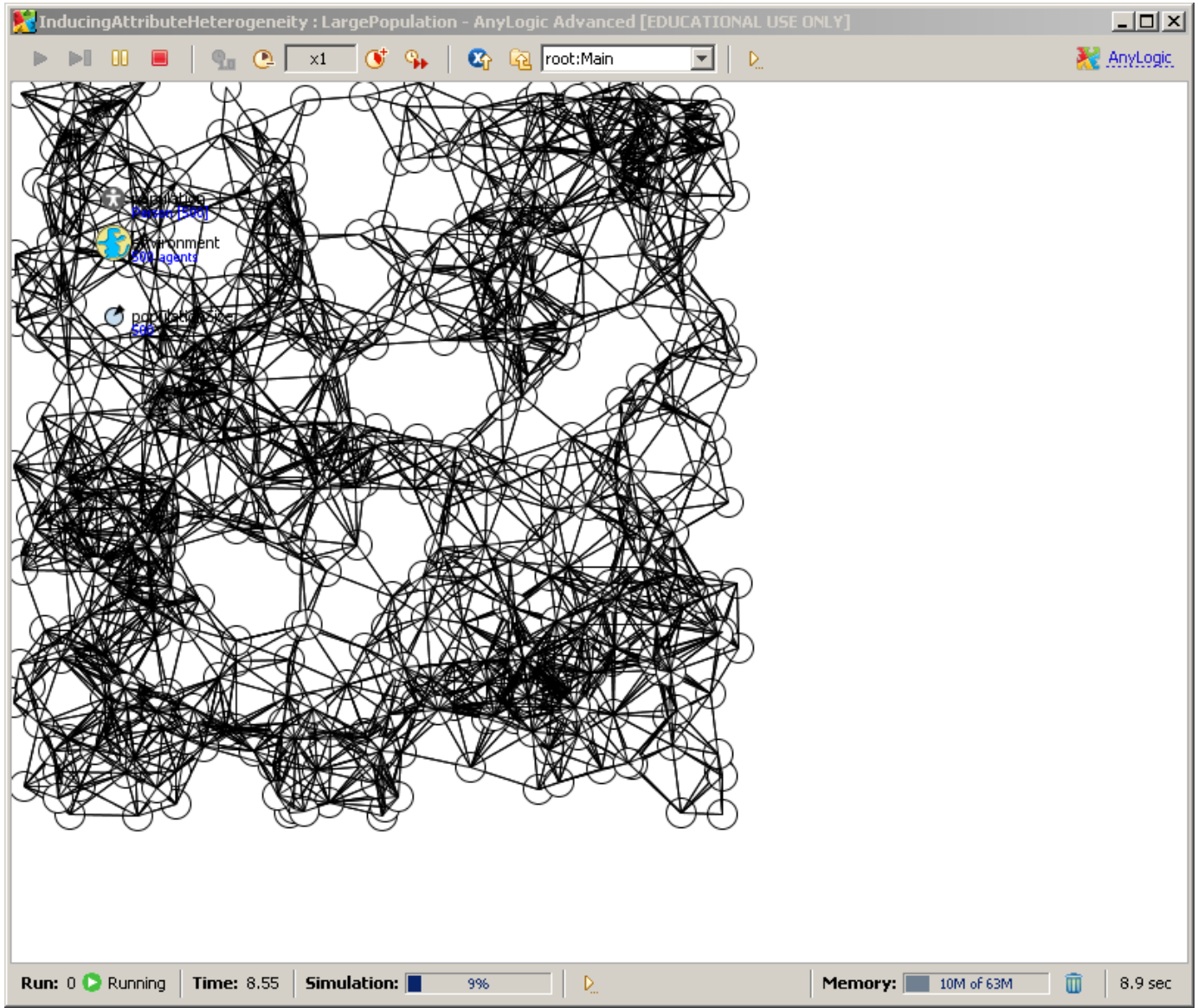
Name the New Experiment “LargePopulation”



Setting the New Experiment Assumptions



Run the New Experiment



Save Model As…

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