Specifying Agent Properties

Nathaniel Osgood
CMPT 394

February 14, 2013
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: MinimalistNetworkABMMModel
Load in Previously Saved “MinimalistNetworkABMModel”

• Pre-built model is also available
Add Parameters from “Palette” Window

1) Click on the “Model” label in the “Palette” window
2) Click somewhere on the canvas twice (to create 2 parameters)
Setting the 1\textsuperscript{st} Parameter Characteristics

Select the first parameter

Name the parameter “income”

Make sure that the “Type” is marked as a “double” (the Default)
Setting the 2\textsuperscript{nd} Parameter Characteristics

Select the second parameter

Name the parameter “sex”

Make sure that the “Type” is marked as an “int”
“Population” Properties Now Include Parameters
“Recipes” for Determining Agent Characteristics

Income: “uniform(10000, 50000)”

Sex: “uniform_discr(0, 1)”
Model Simulation Opening Screen

InducingAttributeHeterogeneity

Experiment setup page

Run the model and switch to Main view

Navigation drop-down (for browsing model elements & agents during execution)
Model Simulation Opening Screen

If no navigation drop-down is present, click here to Enable it.
Turning on Model Navigation

Click here to Enable Navigation Menu
Browsing Attributes of Population Members

![Simulation Diagram]
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

1) Click here ("Parameter")

2) Click somewhere on the canvas

Click on the "Model" label in the "Palette" window
Set the *Default* Value of the Parameter

Default Value: Use "100"

Type: Use "int" (whole numbers)
Setting the Population Size to be Determined by the Parameter “populationSize”

Replication: Use “population Size” (check any “Replicated” checkbox)

May wish to use Auto-Completion (Control-Space)
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)

This specifies "population Size"
Add a New Experiment

Right-click on project name, select New menu, and then “Experiment”
Name the New Experiment
“LargePopulation”
For “population Size”, Use “500”
Run the New Experiment
Save Model As...

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