ModelBuilder Advanced Techniques

Scott Murray
What do you need to know to become proficient at building models?

• **Know how to use geoprocessing tools**
  – Complex operations are accomplished by combining simple operations
  – Learn how to combine tools in the correct order

• **Know how to use ModelBuilder techniques**
Resources for learning Spatial Analysis Concepts

Books:

- The ESRI Guide to GIS Analysis, Volume 1: Geographic Patterns and Relationships. Andy Mitchel
- Geographic Information Systems and Cartographic Modeling. Dana Tomlin
- GIS, Spatial Analysis, and Modeling. David J. Maguire, Michael Batty and Michael F. Goodchild, Editors
Resources for learning tool usage and ModelBuilder techniques

• **ArcGIS Help (webhelp.esri.com):**
  – Geoprocessing\Fundamental tool concepts
  – Geoprocessing\Automating your work with models
  – Geoprocessing\Sharing tools and toolboxes
  – Geoprocessing\Geoprocessing with ArcGIS Server\Geoprocessing service examples
Resources for learning tool usage and ModelBuilder techniques

- **Resources.esri.com: Geoprocessing**
- **Support.esri.com: Geoprocessing**
- **ArcScripts.esri.com: models and scripts**
ModelBuilder Techniques

• Using Model Only Tools
  – Select Data
  – Calculate Value
  – Merge Branch
• Using model iteration
• Using feedback
• Using in-line variables
• Controlling the flow of the model
  – Using pre-conditions
• Using Lists
Making a model a tool

- Creating model elements for parameters
- Setting data type on model elements
- Setting filters on model parameters
- Setting symbology on model parameters
- Using variables for environment setting
- Using the feature set data type to make models interactive
Integrating models

• Using one model in another
• Using geoprocessing services in models
• Integrating external models
Case Study: Drive Time Demographics

- For a location, determine the population within drive time zones
Case Study: Drive Time Demographics

- Use Service Area Network Analysis Layer
- Use Add Locations to add the drive time source
- Use Solve to compute the drive time zones
Case Study: Drive Time Demographics

- Use **Select Data** to get the “polygons” sub-layer from the Service Area Layer
The Select Data tool is used to expose sub-layers or sub-datasets in ModelBuilder

Expose sub-datasets from the results of tools like Solve, Import from CAD, Split

Select Data is a tool that is only useful in ModelBuilder
Other model only tools: Calculate Value, Merge Branch
Case Study: Drive Time Demographics

- Use Make Feature Layer to make the population field a **ratio field** so that Intersect will proportionally split the value by area
Make Feature Layer is your friend

- Can use to get a subset selection without creating a new feature class
- Can rename fields
- Can turn fields off
- Can set fields to be ratio
  - Many tools such as Copy Features and Intersect will use these setting when creating output. The output fields will have the new names, not include fields that are turned off, and do a ratio split when appropriate.

- Used to create a layer for Select by Attribute, Select by Location, and Add Join
Case Study: Drive Time Demographics

- Use Dissolve with statistics fields to sum the population by drive time zones.
How to make the drive time model interactive?
Making the model input interactive

- Use the **Feature set** data type to create a model where the user draws the input on the map
  - Feature class defines the feature type and attribute
  - Layer file defines the symbology
How to control the symbology of the output

• Set the symbology property on the output variable/parameter
How to control the symbology of the output

- The layer file defines the symbology that will be used when a layer is created for the output geodataset.

- Classified legends will update if the classification type is not “manual” (9.3).

- Unique value legends will update if “all other values” is false (9.3).

- See ArcGIS Help: Geoprocessing\Automating your work with models\Using ModelBuilder\Setting symbology for output data.
Case Study: Process temperature data
Case Study: Process temperature data

• World mean temperature data for 1901-2000
  – Celsius
  – Integer value of temperature times 10 to save storage space
• The source data is in netCDF format
• Compute the individual year differences with the baseline
• The baseline is the mean of 1900-1930
• Create a raster catalog and netCDF file of the differences for each year
Case Study: Process temperature data

- Use Make NetCDF Raster Layer to extract the raster for each year
- Subtract the baseline from each year
Case Study: Process temperature data

- Set **model iteration** to run the model for each year
- Get the iteration count from the model variable "NumberOfYears"
Case Study: Process temperature data

• Use **in-line variable substitution** in **Calculate Value** to calculate the netCDF file index and the year.

  • Index = %StartYear% - 1900 + %n%
  • Year = %StartYear% + %n%

• %StartYear% is a model variable
• %n% is a built in variable that indicates the iteration number (starts with 0)
Case Study: Process temperature data

- To make sure all the rasters have the same pixel type (signed integer), use Copy Raster with a negative No Data value and 16 BIT SIGNED type.
- Use in-line variable substitution to put the year in the name of the output raster.
Case Study: Process temperature data

• Create a file geodatabase
• Create a raster catalog in the file geodatabase
  – Use the baseline raster for the coordinate system of the raster column and the geometry column
  – Make the raster catalog un-managed
• Use Workspace to Raster Catalog tool to load the raster catalog
  – Adds in alphabetical order
• Add and calculate the year field on the raster catalog
• Use Raster to NetCDF to generate the netCDF file
How to make the number of iterations a model parameter

• Create a variable of type “Long”
• Set the number of iterations to the variable
• Make the variable a model parameter
In-line variable substitution is used to make parameter values more flexible

• Any string or path parameter can include in-line variables

• Use %<keyword>%% to indicate in-line variable

• Keywords:
  – Variable Names
  – Environment setting names
  – Built in keywords
    • %n% is the current iteration number
    • %i% is the current list index
    • %v% is the current series value
In-line Variable Substitution Examples

- **Calculate Value expression**
  - `%StartYear% + %n%

- **Calculate Field expression**
  - `!shape.Area! * %conversion_factor%
  - `[airport_id] = “%airport_id%”

  - **Make sure to include quotes in the expression!**

- **Path to dataset**
  - `%scratchworkspace%\out.shp`
  - `Directory\out%n%`
Calculate Value is your friend

• Can use any Python expression
• Can supply a code block for complex processing
• Can set any output data type
  – Useful for transforming strings into any data type
Case Study: Simulate a volcano

Download: Support.esri.com: Geoprocessing: Models
Case Study: Simulate a volcano

- **Simulation basics:**
  - Use **model iteration**
  - Use **feedback** in the model
  - Use **in-line variable substitution** to name outputs for each iteration

- **Generate random inputs**
  - Random number: Calculate Value tool
  - Random field in a table: Calculate Field tool
  - Random points: Create Random Points tool
  - Random surface: Create Random Raster
  - Random Seed Environment Setting
Case Study: Simulate a volcano

• Equations
  – flowDirection = FlowDirection(Fill(DEM), Force)
  – lavaFlow = if(magmaPipe, max(normal(1) * 5 + lavaLoad, 0), 0)
  – sinusSlope = min(sin(atan(slope(dem))), sin(45))
  – transportCapacity = 0.9 + sinusSlope * 2 * sin(45)
  – flux1 = Flow Accumulation(flowDirection, lavaFlow * transportCapacity)
  – flux = flux1 * depthConversion * if(magmaPipe, 0.15, 1)
  – volcanoSurface = DEM + Flux
  – DEMn = FocalStats(volcanoSurface, Rectangle, 5 5, Cell, Mean)

• Inputs
  – magmaPipe: raster of the lava source
  – DEM: initial DEM
  – lavaLoad: constant (100)
  – depthConversion: constant(0.05)
Case Study: Simulate a volcano

• Map Algebra
  – Use the Spatial Analyst tools
  – For equations with more than one math function use the Single Output Map Algebra tool
How to iterate a model based on values in a table

- Option 1: Load the values into a series variable (9.3)
How to iterate a model based on values in a table

• The variable type is **Series**
  – A series is a list used by model iterations
  – The model iteration determines which value from a series is used
How to iterate a model based on values in a table

- Set the model iteration to run based on the “Value” variable.
- The model will run for each value in the series.
How to iterate a model based on values in a table

• In this example the Select tool will run with a different value for each model iteration.
How to iterate a model based on values in a table

- Option 2: Use a script tool to extract the values from the table.
- Useful if the table may change for each model run

Download: Support.esri.com: Geoprocessing: Models
How to iterate a model based on values in a table

- The Iterate Field Value tool extracts the field value for the current iteration.
- Inputs: Table, field, and iteration number \%n\%
- Outputs: Value and Continue
- Value contains the field value
- Continue is set to false when the end of the table is reached.
How to make the number of iterations based on a condition in the model

- Set model iteration properties to “Run the model until this variable is false”

In this case, use the “Continue” variable
Using Lists

- Useful for repeating the same operations with multiple datasets.
Using Lists

• A process is executed once for each value in a list

• The number of outputs is equal to the maximum number of inputs

• If input lists are not the same length, the last value is reused in the shorter list (It is NOT combinatorial).

• The keyword %i% indicates the current list index
How to do conditional processing

- To control process execution in ModelBuilder
  - Use process *pre-conditions*

- To control processing on a feature by feature basis
  - Use cursor in scripting
  - Use code block in “Calculate Field” tool

- To control processing on a cell by cell basis
  - Use the “Con” tool
Precondition can be used to control the order of processing
Precondition can be used for If/Then branch in a model
How to merge processing branches in a model

- Use the Merge Branch tool
How to do conditional processing by feature

- Example shows using Calculate Field code block to set a class value based on the size of the feature.

```python
def sizeClass(size):
    if (size < 100):
        return 1
    if (size < 500):
        return 2
    return 3
```
How to run an external model from ModelBuilder

• Create a script tool to integrate the model/program

• Script tools can run any executable
  – Program arguments passed via command line
  – External.exe <input> <output>

• Can use scripting language to preprocess and post process programs inputs and outputs

• Use data types you can easily pass between programs
  – Pathname to geodataset, table, or file
  – Simple values: numbers and strings
Wrapping an external model with a script tool

- Example: How to run excel from ModelBuilder
- Create a python script to:
  - Pre-process data: open worksheet and set cell values
  - Execute: calculate
  - Post-process: get cell values

```python
inValue = gp.GetParameterAsText(0)

xlApp.Workbooks.Open("test.xls")
xlApp.ActiveSheet.Cells(1, 1).Value = inValue
xlApp.ActiveSheet.Calculate()
outValue = xlApp.ActiveSheet.Cells(2, 1).Value
xlApp.ActiveWorkbook.Close(SaveChanges=0)
xlApp.Quit()

gp.SetParameterAsText(1, outValue)
```
Use models to organize your work

- You can include one model in another just like any other tool
How to use server tools in ModelBuilder

• Server tools can be used like any other tool in ModelBuilder

• Execution is always synchronous
  – Waits for the tool to execute

• Use the Append tool to load features into a feature set
Additional Support:

- **Resources.esri.com: Geoprocessing**