Model Builder Tutorial (Automating Suitability Analysis)

Model Builder in ArcGIS 10.x

Part I: Introduction
Part II: Suitability Analysis: A discussion
Part III: Launching a Model Window
Part IV: A Simple Model
Part V: Saving Models
Part VI: Modifying Models
Part VII: Analysis Tools
Part VIII: Toward a Complex Model
Part IX: Using the Iterator Feature

Note: Some of the graphics in module are from ESRI's ArcGIS Resource Center
The purpose of this laboratory is to launch and use **ModelBuilder**, an application in which you graphically create, edit, and manage models. It is very useful in organizing your inputs, operations, outputs and automating processing. You can use this tool to build a model (like a McHargian Suitability Model) to predict the best areas for some land use.

**Getting Started**

You can start Model Builder from ArcCatalog....and embed your new model into a personal Arc Tool Box.

Start ArcCatalog

Create a new **ArcToolbox**, a custom toolbox where you will house your first model.

**Right Click** on the **folder** where you will store your new tool box.

Here I create one called **Radke_Toolbox.tbx**
Right click on the new toolbox, New → Model to launch a new ModelBuilder window and session.
In the ModelBuilder window you create and program your model.

We will return to this ModelBuilder window after a brief review of Suitability Analysis.
In the past few centuries, human technological innovation has enabled us to yield a great degree of control over nature. This brought about the unprecedented spread of humans across the globe, to places where earlier humans were unable or unwilling to live. However, in our rush to subvert the land to our needs, we stopped paying attention to the natural processes that govern the landscape, at great cost. Our ignorance and blatant disregard of natural systems has led to (among other things) a marked increase in the frequency and severity of disasters, both natural and human-induced.

When we stopped paying attention to the landscape, we built our houses on floodplains, our stadiums on fault lines, our farms in deserts, and our cities in wetlands, and we prided ourselves on our triumph over nature. As a result, our neighborhoods flooded, our farms depleted the water table, our buildings crumbled, and our cities destroyed habitat. In response to our destructive attitude towards nature, a different planning ideology surfaced in the latter part of the twentieth century that advocated for respect and understanding of natural systems. Ian McHarg, one of the eminent theoreticians in this field, wrote the seminal book in Environmental Planning, *Design with Nature*, in 1969 and introduced the concept of *Suitability Analysis*.

The power of Suitability Analysis is that it's concept is taken directly from nature; from the harmony between place and the life forms that inhabit the place. In nature, species inhabit areas that are ideally suited for their needs, and in turn, they contribute to the ecology of that place. McHarg's brilliance was that he translated the idea of ecology and place to investigate the best conditions for human habitation (an ecological land use strategy), and *Suitability Analysis* was born. The original Suitability Analysis used a physical transparent map overlay technique to find the most appropriate locations for human developments. McHarg's method involved superimposing layers of geographical data (environmental and social factors) so that their spatial intersection or relationships could be used in making land use decisions.

*Now we do the same technique with GIS...*

The purpose of the suitability analysis is to aid in the decision making process.
At this point I find it easier (mainly due to the fact that I can easily select files from the Table of Contents while I test out my models) to move to ArcMap to launch a new ModelBuilder window.

In ArcMap add some data.

Click the icon to launch a new ModelBuilder window.

Click the icon to launch a new ArcToolbox window.
Drag and drop (from the **Table_Of_Contents** window) some data layers into the **ModelBuilder** window.

These model elements are known as variables.
Drag and drop (from the ArcToolbox window) some tools into the ModelBuilder window.

There are two types of model elements, **tools** and **variables**, shown here in an organizational chart:
Tools:

Tool elements are represented with rectangles and are created when you add a tool from ArcToolbox. The color of the tool has meaning, as described in the table below.

<table>
<thead>
<tr>
<th>Color</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Tool" /></td>
<td>Not all parameters have been supplied for the tool to run. Tool parameters will be supplied by the user of the model when the model is executed.</td>
</tr>
<tr>
<td><img src="image" alt="Tool" /></td>
<td>All parameters have been supplied. The user does not have to supply parameters when the model is executed.</td>
</tr>
</tbody>
</table>

Variables:

Variables are represented with ovals.

You can think of variables as containers that hold values that can be changed. In the context of a model, a variable can be created and its value used in place of a tool's parameter value.

There are two types of variables: data and values. Data variables reference data on disk or in an in-memory layer (such as a layer in the ArcMap table of contents). Values are everything else such as numbers, strings, spatial references, and geographic extents.

Data Variables

There are two types of data variables, project data and derived data.
<table>
<thead>
<tr>
<th>Color</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Data</td>
<td>Project data is data that you add to the model. Typically, it is the result of specifying a dataset to a tool's input parameter.</td>
</tr>
<tr>
<td>Derived Data</td>
<td>Derived data is new data created by a tool in the model.</td>
</tr>
<tr>
<td>Variable</td>
<td>An empty variable has no value.</td>
</tr>
</tbody>
</table>
There are actually two kinds of derived data that you should be aware of.

New derived data is the **SimpleTable.dbf** you see in the above model. The **Create Table tool** creates new data that will be written to disk when you execute the model.

![Diagram of Create Table tool creating SimpleTable.dbf](image)

**In/Out derived data** is the result of a tool that alters the input data rather than creates a new dataset, such as the **Add Field tool**, shown below.

![Diagram of Add Field tool modifying SimpleTable.dbf](image)

In models, you'll recognize **derived data** because the variable name, by default, will have the same name as the input data but with a unique number appended, as in **SimpleTable.dbf (2)** above. Different colors or symbols are not used. A tool dialog that does not have an output data parameter should create an **in/out derived data element** when used in a model.

In ModelBuilder, you cannot designate **in/out derived data** as intermediate data.
For an expansion of this go to the Online ESRI's overview of model concepts and terms.


Connectors

<table>
<thead>
<tr>
<th>Connector Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data connector</td>
<td>connects data and value variables to tools.</td>
</tr>
<tr>
<td>Environment connector</td>
<td>connects a variable containing an environment setting (data or value) to a tool.</td>
</tr>
<tr>
<td>Precondition connector</td>
<td>a connection from a variable to a tool. The tool will execute only when the contents of the precondition variable are current.</td>
</tr>
<tr>
<td>Feedback connector</td>
<td>the output of a tool will be fed back into the same tool as input.</td>
</tr>
</tbody>
</table>
Text Labels

In addition to the variable, tool, and connector model elements, there are text label elements, which are graphical elements for explanatory text in a model. A label is not part of the processing sequence. Labels can be attached to elements or float free in the model diagram.

Adding labels to elements
Right-click the element or connector line and click Create label.
A label with a default name is added. It is attached to the element so that moving the element will move its label as well.
Double-click the label to enter text.

Adding free-floating labels
Right-click the ModelBuilder diagram and click Create label.
A label with a default name is added to the diagram.
Double-click the label to enter text.

To add multiple lines of text:
When editing a label, hold down the Shift key and press Enter to enter a new line.
use the following data base

http://ced.berkeley.edu/faculty/ratt/tool_time/ModelBuilder.gdb.zip

Drag and drop a processing tool onto the model builder sheet.
Flow Links

Select a connect tool and draw in a flow link in your model.

Use a and double click of left button to select the buffer tool box to engage the wizard to modify the analytical operation. This modification will perform the operation in this buffer when it is activated. Any tool wizard is programmed the same.
Fill in the wizard with an output and buffer distance.

Note: The elements are all colored as they are all hard coded at this point. The input variable is liquor store locations, the buffering wizard has the buffer distance set to 500 and all the other parameters are set to the defaults, and the output file B_liquor_buf500.shp is specified.
Execute the Model

Use the button to execute the model and close the window.

Add the newly created shapefile of buffers to the data frame.
Save the Model as **Model_Buffer**

Start a new model (a new model space by clicking the icon ) and add **buffer** tool and add the **B_liquor** stores once again.
Right click on the output name and select **Model Parameter**

A **P** is placed to the side of the output element.

Do the same for the input variable.
Click on buffer tool and set the distance at 500 feet.

Validate your model … and get ready to save.
Save your Model by clicking and navigate to your new tool box and save it.

You can give your model to a friend? It is now an operational tool that can be accessed through ArcCatalog.

Right click on your model and select the Properties tab.
You can name it **Model3** but it is the Label name **Model_BufferParm** that is key here.

You can now open and navigate to the tool in ArcCatalog and **double click** on it to run it.

This time it will ask for the input name and the output name since we have set these up as parameters.

Fill it in and run the model. This makes it more universal and dynamic.
Part VI: Modifying Models

Edit the model by right clicking on it and selecting **Edit**.
Select the Buffer tool and **right click** to parameterize the distance variable by **Make Variable → From Parameter → Distance (value or field)**.

It produces another input graphic element variable for the buffer tool …

Drag the graphic element variable so the Model is easier to read.
Double click the new graphic element variable.

The distance is fixed at 500 feet.
Right Click the new graphic element variable and select Model Parameter.

Validate your model … , then save the Model as a new model

Execute the model from ArcCatalog and you now see a more universal buff tool with a wizard asking the many questions necessary to successfully run the program …. Putting in new variables on-the-fly. The Default buffer Distance was 500 but you can now change this…. I change it to 1000 feet.
I change the output/input fields and distance parameter.
Up to now we had only taken defaults on the input / output names. We can edit these names and modify our wizard.

This leaves you with a more readable model.
When you now execute it you have a more generic wizard.

That I can use to buffer all kinds of data …such as Toxic sites here.
Open up a new model builder window and drag in the **Select tool**. Parameterize the input and output.

Right click the **Select tool** and parameterize the Selection expression.
Right click on the Expression and model with a parameter.

Validate, save the Model ..... as ....

Run Model_select by double clicking on the icon in ArcCatalog.
use SQL to select a liquor type

```
[LICTYPE1] = '47'
```
Part VIII: Toward a Complex Model

Here I make a relatively complex model looking for suitable sites for housing in the Claremont Canyon.
The results are shown in green below. The small red boxes are the actual houses that exist in what we believe are suitable and likely sustainable given our model.
Part IX: Using the Iterator Feature

IX.1 Insert the Iterator
IX.2 Iterator - using a workspace or Geo-dataBase
IX.3 Select Feature Type to Iterate
IX.4 Add a tool and link the output to the Iterator
IX.5 Run the Model

Using the iterator in ModelBuilder to produce a loop in your model. A loop is a way of repeating a task many times that automates a repetitive action. Note, only one iterator can be used per model.

Use the following data for this exercise.

http://ced.berkeley.edu/faculty/ratt/tool_time/subbasinsALL.gdb.zip

Iteration, often referred to as looping, means to repeat a process over and over with some degree of automation. Iteration is very important because automating repetitive tasks reduces the time and effort required to perform the tasks. With iteration in ModelBuilder, a process can be executed over and over using different settings or data in each iteration. ModelBuilder also provides flexibility in iteration, as an entire model or simply a single tool or process can be executed repeatedly.

Create a new toolbox and empty model.
IX.1 Insert the Iterator

Open the model and insert a **feature classes** iterator (insert → Iterators → Feature Classes)

There are many different ways to include and use an Iterator. See the online help for descriptions:

**IX.2 Iterator - using a workspace or Geo-database**

Open the iterator and add the **sub-basins** geodatabase as the workspace to iterate through.

Open the **subbasinsALL.gdb** Geodatabase
IX.3 Select Feature Type to Iterate

Select "polygon" as the feature type to iterate through (so the iterator will skip the roads file in the GDB which is a line feature class)
IX.4 Add a tool and link the output to the Iterator

Add the **clip** tool to the model and link the **feature class** output from the **iterator** (green) to the clip tool as the **clip feature**.

![Diagram](image)

**Link** the clip tool by **opening** it and populating its parameters.
Use the **feature class** output from the **iterator** (green) as the **clip feature**.
IX.5 Inserting Unique Output File Names

For the **output feature class** create a new GeoDataBase to contain (write to) the output files.

I name mine **output_GDB.gdb** here …

Then use **Subbasin_%Name%_roads** or something to that effect, as the **Output Feature Class** name within the output GeoDataBase.

Note: The **%Name%** command calls the **name** of the **feature class** that is being used, that **iteration** and inserts it in the output file name so that a **unique clip output** is created for each iteration (if you leave it out you will end up with a single output file).
Click

The model is ready to run.....
IX.5 Run the Model

Many of the operations in this iterative modeling take along time as they are looping through and performing many tasks. Some of my models have taken days, if not weeks to run. Start small to perfect your model and then run larger ones. Be patient. Running a model like this could save you weeks of interactive work.
Add the results to your ArcMap data frame.