CUAHSI

universities allied for water research

PROCESSING SCHEMATIC NETWORKS IN ARCGIS

August 10, 2012

by:

Dr. Tim Whiteaker (twhit@mail.utexas.edu)
Center for Research in Water Resources
The University of Texas at Austin

Distribution

Table of Contents

INTRODUCTION

Geographic features and the relationships between them can be represented as an interconnected schematic network of links and nodes. Schematic networks can be created in a Geographic Information System (GIS) using tools such as the Arc Hydro Tools. Not just limited to a pretty display, these schematic links and nodes have the potential to serve as the elements at which processes occur in a simulation of some phenomenon. The Schematic Processor unlocks this potential.

The Schematic Processor is a set of ArcGIS (version 10) geoprocessing script tools for processing Arc Hydro schematic networks. The tools give you the ability to associate behavior with schematic features. For example, if you are simulating the movement of bacteria through a stream network, the Schematic Processor can decay the bacteria as they move along schematic links representing stream segments (Figure 1).

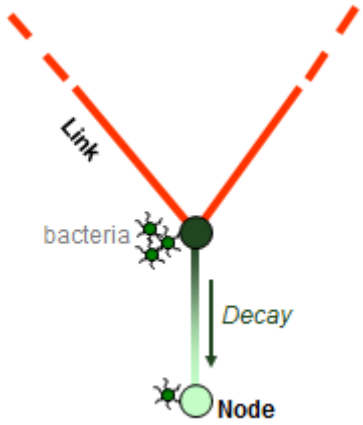

Figure 1 Bacteria decay as they move down the stream network

The behaviors assigned to schematic features are implemented with snippets of Python code called processing ops. A few ops for handling tasks such as simple accumulation of values downstream and first order decay are included with the Schematic Processor. You can also write your own ops to simulate whatever behavior is desired.

In this exercise, you will use the Schematic Processor to simulate the movement of bacteria from subwatersheds through the stream network and finally to the outlet of a watershed. You will write your own processing op, and in doing so, will learn how to create additional ops to extend the power of the Schematic Processor to suit your organization's needs.

GOALS OF THE EXERCISE

This exercise seeks to introduce you to the Schematic Processor. After completing the workshop, you should be able to:

- Understand the conceptual framework behind the Schematic Processor.
- Understand the requirements that a schematic network must meet in order to be compatible with the Schematic Processor.

- Apply the Schematic Processor to a schematic network in ArcGIS using ArcToolbox.
- Write your own processing ops in Python.

Computer Requirements

The Schematic Processor requires ArcGIS 10. An internet connection is required to download the Schematic Processor files and tutorial data.

Installation of the Schematic Processor is part of the exercise procedure and does not require administrator privileges.

A Python code editor or IDE would be useful since you'll be viewing Python files. When the exercise asks you to **open** a Python file, you should open the file for editing in your code editor. Do not actually execute the code. When it's time to execute code, the tutorial will instruct you to **run** the appropriate command.

Conceptual Framework

About Schematic Networks

A schematic network represents features and the connectivity between features using nodes and links, respectively. In Arc Hydro, schematic networks consist of **SchematicNode** and **SchematicLink** features (Maidment, 2002). These features include a unique numerical identifier, called **HydroID**, that can be used to relate one feature to another. SchematicLink features also include **FromNodeID** and **ToNodeID**, which store the HydroID of the SchematicNode at the upstream end of a link and the HydroID of the SchematicNode at the downstream end of a link, respectively. SchematicNode features include a numerical **SrcType** field to indicate the type of feature the node represents, while SchematicLink features include a numerical **LinkType** field to indicate the type of connection the link represents.

For example, consider a stream network with watersheds delineated that drain to those streams. When using an Arc Hydro schematic network, the endpoints of stream segments would be represented as nodes and the streams themselves would be represented as links connecting the nodes. Watersheds could also be represented as nodes, and links could connect those watershed nodes with stream endpoint nodes to represent the movement of water from the watershed into the stream network. These links and nodes would consist of these types:

SchematicNode:

SrcType = 1 – Represents watersheds
SrcType = 2 – Represents endpoints of stream segments (one on upstream end and one on downstream end)

SchematicLink:

LinkType = 1 – Connects watershed nodes to stream nodes
LinkType = 2 – Connects stream endpoint nodes (represents the river network)

These SrcType and LinkType values are just examples. You could have any number of values to represent as many types of features as you desire.

Attaching Behavior to Schematic Features

The Schematic Processor gives you the ability to associate behavior with schematic features. For example, consider the case of computing mean annual bacterial load for the outlet of a river basin. Suppose you've computed mean annual load produced by nonpoint sources in the watersheds of the basin. You'd like to move that load through the stream network, accumulating the values downstream until you arrive at a total value at the basin outlet. Since bacteria typically decay as they move through natural systems, you'd also like to consider this decay as the bacteria move downstream.

The accumulation of upstream loads and the decay of bacteria are behaviors that schematic nodes and links exhibit. Adding up loads received from upstream features is a **RECEIVE** behavior, while decaying bacteria before passing the load to the next downstream feature is a **PASS** behavior.

A RECEIVE behavior processes values passed from upstream features as well as any incremental value contributed by the current feature to produce a total value for that feature. A PASS behavior processes the total value for a given feature to produce a passed value, which is sent to the next downstream feature in the network. (See Whiteaker et al., 2006 for more details.)

In the Schematic Processor, these behaviors are implemented via snippets of Python code called processing ops. These ops are associated, by the user, with specific combinations of source type (SrcType and LinkType values), feature type (LINK or NODE), and behavior type (RECEIVE or PASS). Following the example above, the following ops could be created to perform the computation of mean annual bacterial load, given that the values on watershed nodes represent mean annual load contributed by each watershed.

SchematicNode:

SrcType 1 (watershed) – RECEIVE – N/A (these are the most upstream features in the network)
SrcType 1 (watershed) – PASS – Send value unmodified downstream
SrcType 2 (stream node) – RECEIVE – Sum all received values
SrcType 2 (stream node) – PASS – Send value unmodified downstream

SchematicLink:

LinkType 1 (watershed to stream) – RECEIVE – Sum all received values
LinkType 1 (watershed to stream) – PASS – Send value unmodified downstream
LinkType 2 (stream) – RECEIVE – Sum all received values
LinkType 2 (stream) – PASS – Apply bacterial decay and send result downstream

In the exercise that follows, you will learn how to simulate the scenario described above using the Schematic Processor.

Installing the Schematic Processor

Installing the Schematic Processor involves downloading the files to your computer and adding the toolbox to ArcGIS. No administrator privileges are required.

To install the Schematic Processor:

1. Navigate to http://tools.crwr.utexas.edu/.
2. Click the link for **Schematic Processor**.
3. Click the **DOWNLOAD** link.
4. Copy the contents of the downloaded zip file to a convenient location on your computer.
5. Open the **SchematicProcessor** folder.

In this folder you will find documentation, sample data, and the tools themselves.

6. In the **sample_data** folder, open **example.mxd**.
7. If ArcToolbox isn't visible, then click the **ArcToolbox window** button  .
8. If you don't see a toolbox called Schematic Processor Tools, then:
    a. Right-click in the ArcToolbox window and click Add Toolbox.
    b. Navigate to the **SchematicProcess\schematic_processor** folder and open **Schematic Processor Tools.tbx**.

The Schematic Processor Tools toolbox should now be visible in ArcMap (Figure 2).

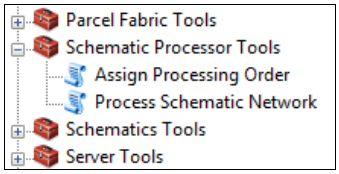

Figure 2 Schematic Processor Tools toolbox

You can add the Schematic Processor Tools toolbox to any ArcMap document, but it just so happens that this example document has the data that you will be working with in this exercise.

Running the Tools

In the example ArcMap document that you opened, you can see a simple schematic network connecting subwatersheds and streams. The green subwatershed nodes are located at subwatershed centroids, and they are connected to nodes at the subwatershed outlets via the green dashed links. The red stream nodes are located at the endpoints of stream segments, while the blue links connect the stream nodes. This schematic network was created using the Arc Hydro Tools.

Suppose that runoff from subwatersheds brings bacteria to the stream network. The bacteria decay as it travels along stream segments before finally reaching the stream network outlet. In a real case study, one would estimate the amount of bacterial loading contributed by subwatersheds based on parameters such as land use, animal

populations, and presence of septic tanks. In this example, these numbers have already been tallied for you. Let's take a closer look at these GIS features.

Get to know the data:

1. Open the attribute table for the **Subwatershed** layer.

You can see that each subwatershed feature has been attributed with its area and an estimate of the number of coliform forming units (CFU) per year from a variety of bacteria sources such as cattle. The field called **Total_CFU_yr** stores the sum of all bacterial loads for a given watershed. And like all Arc Hydro features, the **HydroID** field serves as a unique identifier for each feature in this layer.

2. Open the attribute table for the **Node** layer.

The nodes have a HydroID so that they are uniquely identified, and they also have a **FeatureID** that points to the HydroID of the feature that they represent. Some of these FeatureIDs point to subwatersheds, while others point to stream network junction features (not shown in this map). The **SrcType** field indicates the type of feature which the node represents (subwatersheds vs. stream junctions). All of these fields were populated automatically by the Arc Hydro Tools.

The remaining fields are directly related to the Schematic Processor. They include:

- **IncVal** – Incremental value that a feature contributes to the network. Subwatersheds contribute bacteria and have a large value in this field. These values were copied from the Total_CFU_yr field in the subwatershed layer. Stream junctions do not contribute bacteria themselves and thus store a value of zero in this field.
- **TotVal** – Total value for a feature including all values received from upstream features and the current feature's incremental value. This value is zero because the network has not yet been processed.
- **PassVal** – Value that a feature passes to the next downstream feature in the network. This value is zero because the network has not yet been processed.
- **SortOrder** – Indicates the order (ascending) in which features in the schematic network should be processed. You can assign this field manually, or a Schematic Processor tool can do it for you.

3. Open the attribute table for the **Link** layer.

In the Link attribute table you'll see many of the same schematic network fields as described above. Also notice **DecayConst** and **TravelTime** fields (in units of 1/year and year, respectively). One of the processing ops that you'll use in this exercise expects those attributes to be present on the Link layer. They are used in the first-order decay calculation.

Tip
You can add as many fields to schematic network features as needed to support your specific analysis.

4. Close the attribute tables.

As you run the Schematic Processor, you will expect the incremental values from subwatershed nodes to make it into the stream network and travel downstream, with the resulting load values showing up in the TotVal and PassVal fields. Before you can process the network, the Schematic Processor needs to know in what order features should be processed. The Assign Processing Order tool can compute the sort order for you based on FromNodeID and ToNodeID fields on schematic features.

Assigning Processing Order

The Schematic Processor processes features from upstream to downstream. Once a schematic network has been created, you typically only need to determine this processing order once. The Assign Processing Order tool handles this task and writes results to the SortOrder field in the link and node feature classes.

To assign processing order:

1. In the **Schematic Processor Tools** toolbox, double-click **Assign Processing Order** to run that tool. Click Show Help in the tool dialog if you want to see help for the tool and its inputs.
2. Select **Link** as the link features and **Node** as the node features.
3. Click **OK** to run the tool.
4. Close the tool progress dialog when the tool is finished.

The features in the map are labeled according to their processing order. You can now see that upstream features are to be processed before downstream features receive data from them.

Now you are ready to process the network.

Processing the Network

Recall that, in this example, subwatershed nodes use the IncVal attribute to store the amount of bacterial load that they contribute. In this portion of the exercise, you will process the network to simulate the accumulation of bacterial loads from upstream to downstream, taking into account bacterial (first-order) decay along the stream segments.

Because simple accumulation of values (i.e., add them up and pass them downstream) is such a common task, this is the default behavior that the Schematic Processor applies to features. This will suffice for most of the interactions between schematic features in this example. The only exception is that you'll specify the **decay** processing op to handle decay along the stream segments as they pass loads to the next downstream feature.

To process the network:

1. Double-click **Process Schematic Network** to run that tool.
2. Select **Link** as the link features. Then select the **IncVal**, **TotVal**, and **PassVal** attributes from the Link layer as the "Link Incremental Value Field", "Link Total Value Field", and "Link Passed Value Field", respectively.
3. Select **Node** as the node features. Then select the **IncVal**, **TotVal**, and **PassVal** attributes from the node layer.

Now you will tell the Schematic Processor which processing ops to use. You only need to specify a decay op for this example.

4. For processing ops, type **decay** (case sensitive!) and click the plus sign to add decay to the list of processing ops. Each op must be associated with a source type, feature type, and behavior type, so you'll specify those items next.
5. For source types, type **2** and click the plus sign.
6. For feature types, type **LINK** and click the plus sign.
7. For behavior types, type **PASS** and click the plus sign.
8. Click **OK** to run the tool.
9. Close the tool progress dialog when the tool is finished.

The labels on stream links should now indicate total and passed values of bacterial load (you may need to zoom in to see the labels). You can use the Identify tool to check out the total value at basin outlet.

Now that the tool is set up, you can imagine how easy it would be to evaluate different strategies to control bacterial load. These strategies would result in different loads associated with each watershed. Once the loads were computed for each watershed, you would assign those loads to the IncVal field of Node layer (or create a new field called Scenario1) and run the tools again to see the result at the basin outlet.

In the next portion of the exercise, you will write your own processing op to simulate the addition of a wetland to the network.

Tip
Tired of filling out the geoprocessing tool inputs each time you run the tool? In cases where most or all of the inputs are the same each time, you can create a new geoprocessing model, drag the tool into the model, fill out the inputs, and save the model. When you run the model, the inputs will already be filled out for you! And be sure to check out ModelBuilder's capabilities for scenario support through iterations and other features.

Writing Your Own Processing Op

Let's suppose a potential strategy for combatting bacteria is to establish a wetland at the stream network confluence just upstream of the the last watershed in the basin (Figure 3). For simplicity's sake, let's **assume the wetland removes 50% of the bacterial load that comes into it**. Of course this is something you could easily calculate by hand, but let's model this strategy by writing your own processing op. After the op is written, you'll need to make a quick edit to the node representing the wetland location in the GIS to distinguish from other nodes in the network.

Note
If you get stuck during the coding, please refer to the finished code in Appendix A: Source Code for wetland.py.

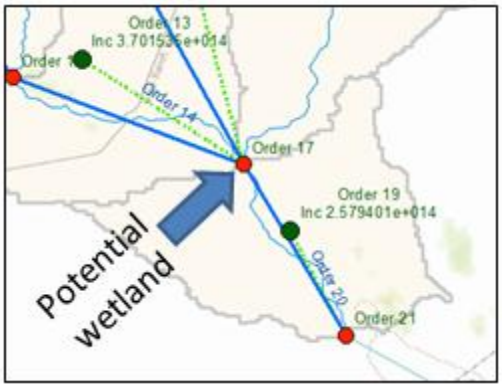

Figure 3 Target location for wetland

> **Note**
> Python script files are ordinary text files. You can use any editor you want to work with them, but it's generally a good idea to use one design to work with Python to take advantage of features such as autocompletion. This exercise is written assuming you are using IDLE, the editor that comes with Python.

The Python documentation has a detailed write-up of what's required to create your own processing op, but this author prefers to work by example (i.e., copy and paste). In this portion of the exercise, you will copy the decay processing op to create your own wetland op.

Copy the decay op to create the wetland op:

1. Navigate to the **SchematicProcessor\schematic_processor\ops** folder. This folder is where all processing ops are located. To install a new op, simple add a file to this folder.
2. Copy and paste **decay.py** in this folder. Rename the copy to **wetland.py**.
3. Right-click on **wetland.py** and then click **Edit with IDLE**. Or use your editor of choice if that option is not available.

Notice the contents of this file. It has sections as highlighted in Table 1.

Table 1 Structure of the decay processing op

Section	Purpose
Header	Describes the purpose of this op.
Import statements	Imports useful libraries. **decimal.Decimal** enables math with numbers as you might expect, versus the default binary floating point math in Python which performs such arithmetic as 1.1 + 2.2 = 3.3000000000000003. **math.exp** enables raising *e* to the power of x. It is used by the decay function but is not needed for the simple wetland calculation. **schematic_exceptions** helps you raise errors specifically related to the Schematic Processor.
process_values	Required. This is the function that the Schematic Processor calls when it needs your op to simulate RECEIVE or PASS behavior.
get_decimal_value	Reads a value from a field for a given feature. Not required for the wetland op.
validate_inputs	Optional function to make sure inputs to process_values are valid. All this function does is

	makes sure a single value to process was passed from the Schematic Processor.

Since the wetland op is simpler than the decay op, you can delete unneeded portions of code.

To remove unneeded code:

1. Delete the statement `from math import exp`.
2. Delete the **get_decimal_value** function.

Good code comments will help others understand your code and will also help you understand your code six months from now.

To edit the comments:

1. Change the **header** to read `"""Wetland processing op for ArcGIS 10 Schematic Processor tool."""`
2. Edit the docstring for the **process_values** function.
    a. Change the summary to `Simulates 50% removal of constituents via magical wetland.`
    b. Delete the text describing required fields (`This function requires…`) and units of measure.
3. For the **msg** variable in the **validate_inputs** function, replace the word "Decay" with "Wetland" (`Wetland op can only take one value…`). This isn't a comment, but rather a message that is returned if inputs aren't valid.

Tip
As you edit code, make sure you maintain the proper indentation level. Python groups blocks of code together based on indentation level.

Now you're ready to edit the process_values function.

To edit the process_values function:

1. Delete lines of example code or code related to the decay op. These are lines beginning with:
    a. input_class =
    b. feature_id =
    c. decay_constant =
    d. travel_time =
2. Change the line where the result is calculated to be:
```
result = input_value * Decimal('0.5')
```

Tip
The Decimal constructor cannot take floats as a number, which is why the value of 0.5 is passed to the constructor as a string, '0.5'.

In plain English, the process_values function now performs these tasks: First it checks the inputs to see if they are valid. Then it grabs the first input value object. Instead of simple numbers, these input values are in the form of objects called ValueTuples. ValueTuples include the numerical value itself, but also pointers to the feature and feature class that provided the value, just in case your processing op needed to trace back to those features. For the wetland op, all we care about are the numerical values. The next line of code reads that numerical value and converts it to a Decimal, which enables your op to perform precise math without binary floating point artifacts. Once the value is converted, the result is calculated by multiplying the input value by the efficiency of the wetland, or 50%. Finally, the function returns the result to the Schematic Processor.

Your op is finished! Save and close your op. This op should only be applied to nodes representing wetlands, so now you will edit the target node in the GIS to classify it as a wetland node.

Editing the Wetland Node

In this portion of the exercise, you will use standard ArcGIS editing tools to edit the node type for the target node in Figure 3. You will classify node as a wetland so that the Schematic Processor can apply your wetland op just to that node.

To edit the wetland node:

1. In ArcMap, if you can't see the Editor toolbar, click **Customize | Toolbars | Editor** to place a check next to the Editor toolbar.
2. On the Editor toolbar, click **Editor | Start Editing**.
3. Click the target node to select it.
4. On the Editor toolbar, click the **Attributes** button  .
5. In the Attributes window, for **SrcType** input a value of **3** and press ENTER to confirm your edit (Figure 4).

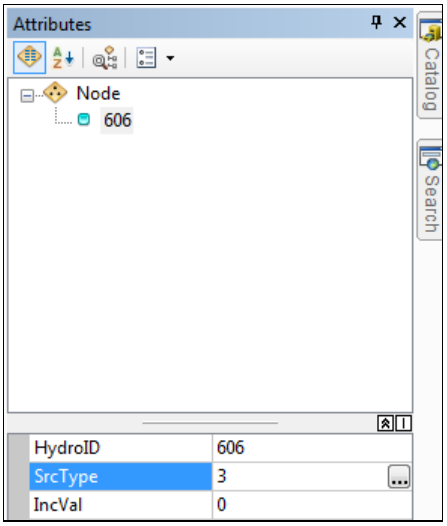

Figure 4 Editing the SrcType attribute

6. Click **Editor | Stop Editing**. Click **Yes** when prompted to save edits.

Notice your node disappeared! You'll need to update the symbology for the node layer to show this new node type.

7. In the table of contents, right-click **Node** and click **Properties**.
8. Click the Symbology tab.
9. Click **Add Values**.
10. Select **3** and click **OK**.
11. Double-click the symbol to change it to something that suits you.
12. To be thorough, you could also click to change the label for this legend item (Figure 5).
13. Click **OK** to confirm changes and close the dialog.

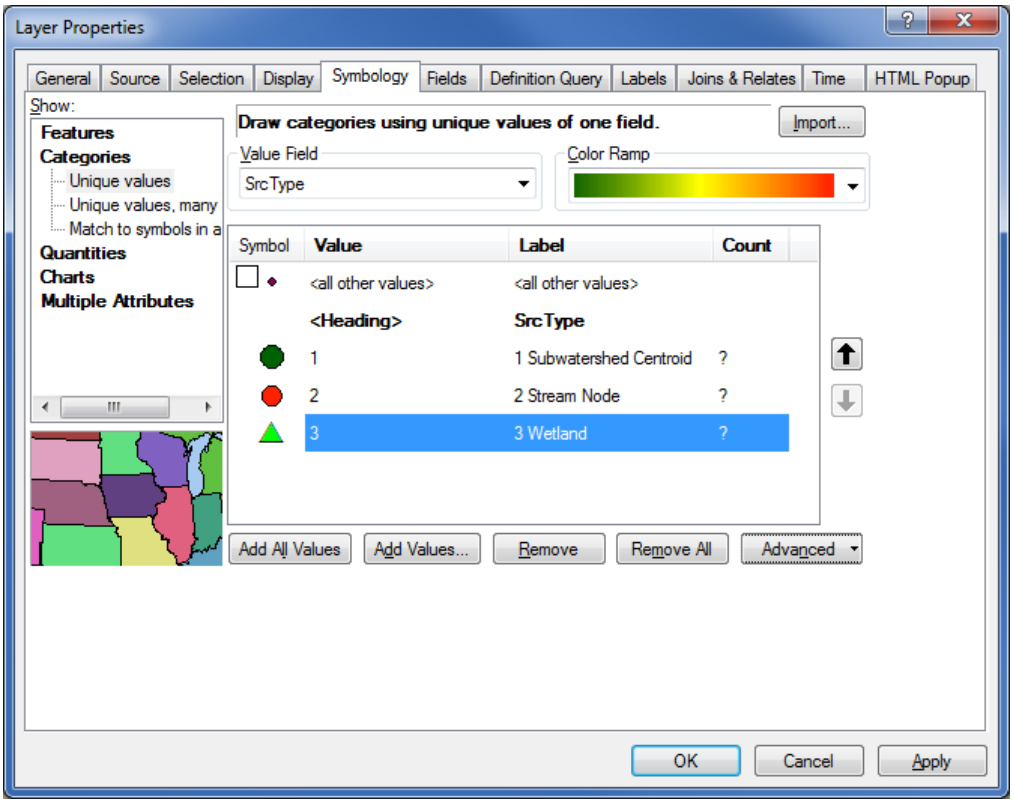

Figure 5 Adding 'wetland' to the node symbology

The node representing the wetland is now clearly visible in the map (shown as a green triangle in Figure 6).

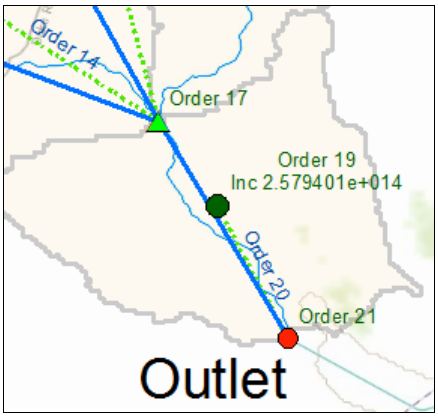

Figure 6 Wetland node (the green triangle) in the map

Now you are ready to process the network.

Running the Tool with Your New Op

It's time to run the Schematic Processor again. This time, you will include your new wetland op. Suppose you've also learned that bacterial decay should be applied during overland flow from watersheds into the stream network (via LinkType 1). Fortunately, decay coefficients and travel times have already been computed for these features. Therefore, you'll apply the decay op to both LinkType 1 (watershed to stream) and LinkeType 2 (stream) features.

There's no need to run the Assign Processing Order tool again. The order hasn't changed since you previously ran the tool.

Before running the Schematic Processor, use the Identify tool to take a note of the total value at the basin outlet. You'll be overwriting that value in the following steps.

Tip
If you don't want to overwrite values, you can add new fields to store total and passed values for this scenario.

To run the Schematic Processor:

1. Double-click **Process Schematic Network** to run that tool.
2. Select **Link** as the link features. Then select the **IncVal**, **TotVal**, and **PassVal** attributes from the Link layer as the "Link Incremental Value Field", "Link Total Value Field", and "Link Passed Value Field", respectively.
3. Select **Node** as the node features. Then select the **IncVal**, **TotVal**, and **PassVal** attributes from the node layer.

Now you will tell the Schematic Processor which processing ops to use. First let's specify the decay op for stream links as before.

4. For processing ops, type **decay** (case sensitive!) and click the plus sign.
5. For source types, type **2** and click the plus sign.

6. For feature types, type **LINK** and click the plus sign.
7. For behavior types, type **PASS** and click the plus sign.

Now specify the decay op for watershed to stream links.

8. For processing ops, type **decay** and click the plus sign.
9. For source types, type **1** and click the plus sign.
10. For feature types, type **LINK** and click the plus sign.
11. For behavior types, type **PASS** and click the plus sign.

Now specify the wetland op.

12. For processing ops, type **wetland** and click the plus sign.
13. For source types, type **3** and click the plus sign.
14. For feature types, type **NODE** and click the plus sign.
15. For behavior types, type **PASS** and click the plus sign.

All ops are now specified (Figure 7).

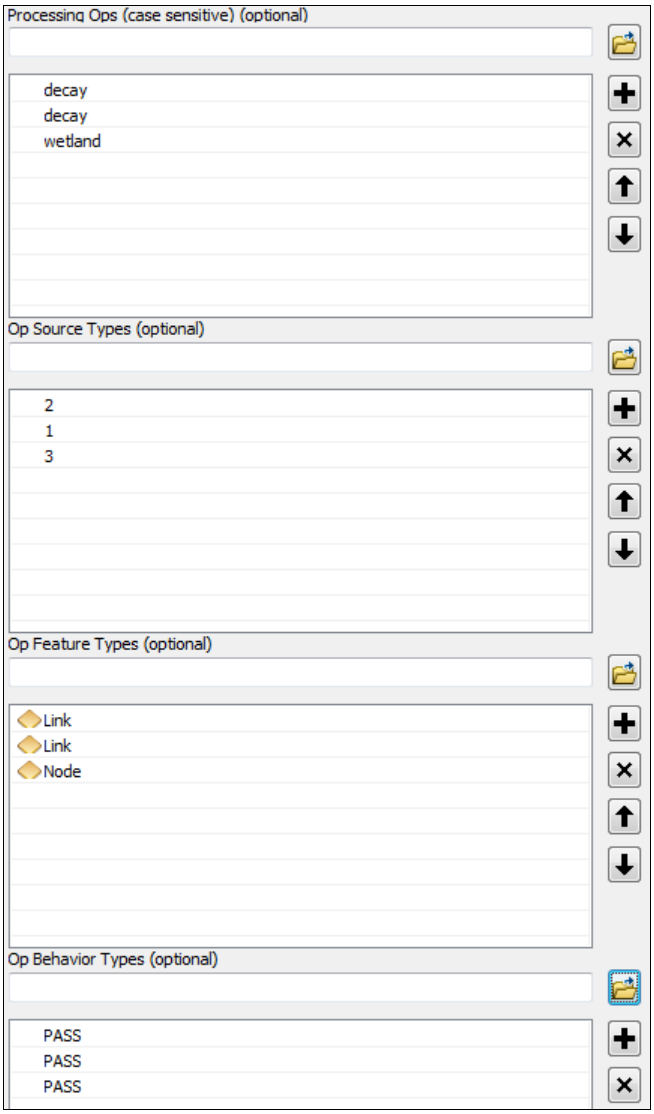

Figure 7 Processing ops specified in tool dialog

16. Click **OK** to run the tool.
17. Close the tool progress dialog when the tool is finished.

Zoom in and use the Identify tool to view the results of the operation.

Congratulations! You have completed the exercise. Feel free to try the Schematic Processor on more complex networks or to write your own processing ops.

For more complex applications of the Schematic Processor see Whiteaker et al., 2006 and Johnson, 2009.

ACKNOWLEDGEMENTS

The author would like to thank Dr. Stephanie Johnson for her contributions to this work.

APPENDIX A: SOURCE CODE FOR WETLAND.PY

```
#!/usr/bin/env python
"""Wetland processing op for ArcGIS 10 Schematic Processor tool."""

from decimal import Decimal

import schematic_exceptions as ex

def process_values(row, input_values):
    """Simulates 50% removal of constituents via magical wetland.

    Arguments:
        row -- input row representing schematic feature
        input_values -- list of ValueTuple objects
    Remarks:
        ValueTuple objects have these attributes:
        value -- numerical value to process
        source_class -- class containing the feature providing the value
        feature_id -- unique identifier of the feature providing
                      the value

        This function only processes the first value found.

    """

    # Validate inputs
    validate_inputs(input_values)

    # Get parameters from input row
    v = input_values[0]
    input_value = Decimal(str(v.value))

    result = input_value * Decimal('0.5')

    return result

def validate_inputs(input_values):
    """Checks values for valid inputs.

    Arguments:
        input_values -- list of ValueTuple objects

    """

    # Only one ValueTuple present
    if len(input_values) != 1:
        msg = "Wetland op can only take one value as input.  " + \
              "Values provided: " + str(len(input_values)) + ".  " + \
              "Please make sure this op is used only for PASS behavior."
        raise ex.OpError(msg)
```

REFERENCES

Johnson, S. (2009). “A General Method for Modeling Coastal Water Pollutant Loadings.” Ph.D. Dissertation, The University of Texas at Austin, Austin, Texas.

Maidment, D. R. (2002). Arc hydro: GIS for Water Resources. Redlands: ESRI Press.

Whiteaker, T., D. R. Maidment, J. L. Goodall, M. Takamatsu (2006). “Integrating Arc Hydro Features with a Schematic Network.” Transactions in GIS 10(2) pp. 219-237. (DOI: 10.1111/j.1467-9671.2006.00254.x)