THE GEO PROJECT
GEOSPATIAL EDUCATION & OUTREACH

www.geoproject.msstate.edu

GEO TUTORIAL

WORK AUTOMATION IN QGIS USING MODEL BUILDER

Krzysztof Raczynski
Kate Grala
John Cartwright

Geosystems Research Institute
Mississippi State University

NOVEMBER 2024

This work was supported through funding by the National Oceanic and Atmospheric Administration Regional Geospatial Modeling Grant, Award # NA19NOS4730207.

GEOSYSTEMS RESEARCH INSTITUTE, MISSISSIPPI STATE UNIVERSITY, BOX 9627, MISSISSIPPI STATE, MS 39762-9652

The Geospatial Education and Outreach Project (GEO Project) is a collaborative effort among the Geosystems Research Institute (GRI), the Northern Gulf Institute (a NOAA Cooperative Institute), and the Mississippi State University Extension Service. The purpose of the project is to serve as the primary source for geospatial education and technical information for Mississippi.

The GEO Project provides training and technical assistance in the use, application, and implementation of geographic information systems (GIS), remote sensing, and global positioning systems for the geospatial community of Mississippi. The purpose of the GEO Tutorial series is to support educational project activities and enhance geospatial workshops offered by the GEO Project. Each tutorial provides practical solutions and instructions to solve a particular GIS challenge.

WORK AUTOMATION IN QGIS USING MODEL BUILDER

Krzysztof Raczynski (chrisr@gri.msstate.edu)
Kate Grala (kgrala@gri.msstate.edu)
John Cartwright (johnc@gri.msstate.edu)

Geosystems Research Institute,
Mississippi State University

REQUIRED RESOURCES

- QGIS 3+

FEATURED DATA SOURCES

- Dataset used in this tutorial can be downloaded from: https://arcg.is/0XmLiS

OVERVIEW

Work automation is incredibly valuable when dealing with multiple tasks that have a similar workflow. For instance, when you have multiple cases to analyze, but the analysis itself contains the same steps in data processing, or if you are working on a recurring case involving multiple analysis steps but the input data changes. In these instances, it can be excessively time-intensive to run these tasks manually. This is where work automation and the model builder can improve your efficiency. Model builder allows you to control the entire workflow process by combining multiple algorithms into a single tool. It is a built-in module in QGIS that can speed up and simplify the analysis process.

Imagine yourself in the following situation: You are working for the County Media Management Company. There is scheduled work to replace parts of transmission lines within a specified mileage, e.g., between the 2nd and 7th miles. The company is required to identify the parcels that are located within the specified reach and to generate a list of owner names and correspondence addresses for these parcels so they can send appropriate notices before the work starts. This tutorial will demonstrate solving this task using a manual workflow and a model builder approach. We will work on the dataset example representing Oktibbeha County, Mississippi.

STEP 1. PREPARE THE DATA

Before you proceed with the analysis, you need to download and organize the data. To do so, complete the following steps:

A. Use the Featured Data Sources link above to download the data. After downloading the zip archive, unpack it in your work folder. You will see two datasets:
   - *parcels.shp* is a shapefile containing polygon information about land parcels in Oktibbeha County with their addresses.
   - *transmission_lines.shp* is a shapefile containing line geometry associated with transmission line paths within a county area.

B. Add both files to a new project in QGIS software.

C. Explore the *parcels* layer attributes table by selecting the layer in the **Layers** panel, right-clicking it, and choosing Open Attribute Table (or pressing the F6 key).

D. Locate the attribute names that we will need for generating the list:
   - OWNNAME: contains the name of the owner according to the public registry;
   - MAILADD1: corresponding address for the owner;
   - MCITY1: the city associated with the address;
   - MSTATE1: state;
   - MZIP1: the zip code for the address.

E. Note the names of the columns, as we will need them later.

STEP 2. MANUAL ANALYSIS

In order to correctly solve the task given to us, let's first try to solve it manually to make sure we get the correct results. It is always a good idea to prepare an outline for the model and test it before building the final tool. To generate the list of owners and their addresses we must successfully complete four steps in the process:

a) Find the transmission line mileage and the corresponding segment.
b) Identify parcels that are located within that segment.
c) Extract their addresses based on the parcel data table and generate a list.

STEP 2a. FIND TRANSMISSION LINE SEGMENTS

To generate a part of the line located between a given mileage, we can use the ***Line substring*** tool. The tool analyzes the geometry according to its direction and provides a segment that is contained within a given distance. As mentioned, the geometry direction is important here. Let's analyze the direction of our lines by doing the following:

A. Open the Processing Toolbox if it's not already opened (Processing menu: Toolbox or by pressing [CTRL] + [ALT] + [T]).

B. Go to *Vector geometry* and search for the *Extract specific vertices* tool, then open it by double-clicking it. Select *transmission_lines* as the **Input layer** and set **Vertex indices** to 0, -1. The 0 number refers to the first vertex of the line, and the -1 indicates the last one. **Run** the tool.

C. On our map, we can now see two points per line, but at this point, it does not tell us much. **Right-click** on the *Vertices* layer and select **Properties**. Navigate to the **Symbology** tab, then change the first setting (the drop-down menu currently displays a Single symbol) to **Categorized**. Under the **Value** menu, choose *vertex_pos*, then click **Classify** under the central window. You will now see three categories: ***-1***, ***0***, and ***all other values***. We can change the color of the symbols by double-clicking them. Let's set green for position *0* (line beginning) and blue for position *-1* (line end). We can turn off *all other values* in this category. Apply the changes and close the Properties window.

D. Look at the map; what can you see? You probably noticed that the middle line has reversed direction compared to the other two lines. The western and eastern lines have a direction from north to south (green dot of value *0* in the north), while the middle line goes from south to north. It is recommended to always check the geometry direction when dealing with mileage, so there are no surprises during output testing. Based on the current setting, we can expect that one line with a different direction will yield results in the southern part of the county, unlike the other two. We are fine with this setting, but in case you need to change it in the future, you can revert the line by using the ***Reverse line direction*** tool (**Vector geometry**).

E. Let's now search for a second tool called ***Line substring*** in the **Vector geometry** tab of the Processing toolbox. Select *transmission_lines* as the **Input layer**. The code EPSG: 6507 indicates that the layer is in the NAD83/Mississippi East projection. The units of this projection are meters, which allows us to change the distance **unit** in the **Start distance** and **End distance** settings. It is important to understand this dependency, as you will not always be able to change the units. If the input layer is in a different projection, for example, WGS84 (EPSG:4326), the units will be degrees, and the program won't allow you to change anything. In such a case, you either need to convert the projection of the layer or provide the distance recomputed from miles to degrees. In our case, however, the projection allows us to convert units. Let's set **miles** in both cases and set the required frame: ***Start distance: 2 miles***; ***End distance: 7 miles***. We don't need to save the output to a file, so **Run** the algorithm.

F. Once the algorithm is done, go to the main window, and in the properties of the newly created layer (***Substring***), change the style to something that will help you distinguish the output (Fig. 1).

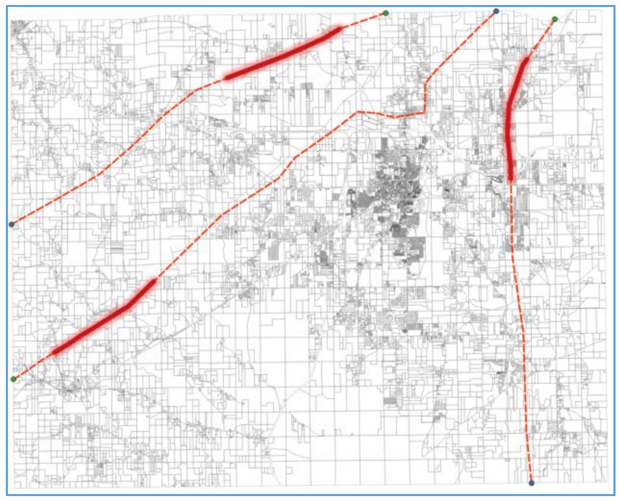

Fig. 1. *The middle transmission line has a south-to-north direction, which can be recognized by the extracted vertices; for this reason, the extracted mileage segment is in the southern section of the county.*

STEP 2b. IDENTIFY PARCELS

Now that we have the transmission line segments identified, we can use this information to extract the parcels located within their range. To do this, we will use the ***Extract by location*** tool:

A. In the **Vector selection** tab of the Processing toolbox, locate ***Extract by location*** algorithm and open it.

B. We will be extracting parcels based on their dependence on transmission line segments; therefore, our **Extract features from input** will be ***parcels*** layer, and **By comparing to the features from** will be ***Substring*** layer that we just calculated. Note that the two layers must be in the same projection, which we can check by comparing the EPSG code in the brackets. The **geometric predicate** (**dependence**) is a condition that the parcels need to meet when compared to lines in order to be selected. In our case, we want the lines to ***intersect*** the parcel, so select this option and then **Run** the tool.

C. In the newly created layer called ***Extracted***, you should see **88 objects** that were extracted from the parcels layer (layer right-click: **Show Feature Count**).

STEP 2c. EXTRACTING ADDRESSES

We now have the information that interests us in the attribute table of the ***Extracted*** layer, and we need to process it to create a semicolon-separated list. To do so, we will use the ***Aggregate*** tool:

A. Open the ***Aggregate*** tool located in the **Vector geometry** tab of the Processing toolbox. Select ***Extracted*** as the **Input layer**. The tool allows you to merge geometries and manipulate its attribute table. We will leverage this second ability to generate the list of addresses.

B. Select the current attributes listed in the *Aggregates* section and remove them using the **Delete selected field** icon. Then use the **Add new field** icon to create a new row representing the new attribute. We will use a semicolon to delimit addresses on our list. To do so, select the following properties:

- **Aggregate Function**: *concatenate*
- **Delimiter**: *;*
- **Name**: *address*
- **Type**: *Text (string)*
- **Length**, **Precision**: don't change, leave at *0*

C. Click on the **Aggregates Expression** ε button in **Source Expression** and use the previously noted names of attributes as input names. Create the expression that will list all the needed information in the mailing format. You can use the **+** operator to add text and fields. Remember that the field name is provided in a double quotation: **"**, while the text is in a single quotation: **'**. Your formula should look like this:

"OWNNAME" + ', ' + "MAILADD1" + ', ' + "MCITY1" + ' ' + "MSTATE1" + ' ' + "MZIP1"

In the above example, we are providing the owner's name read from the OWNNAME attribute, then we add a comma and a space to separate the mailing address (MAILADD1) and city. The city (MCITY1), state (MSTATE1), and zip (MZIP1) are only separated by spaces.

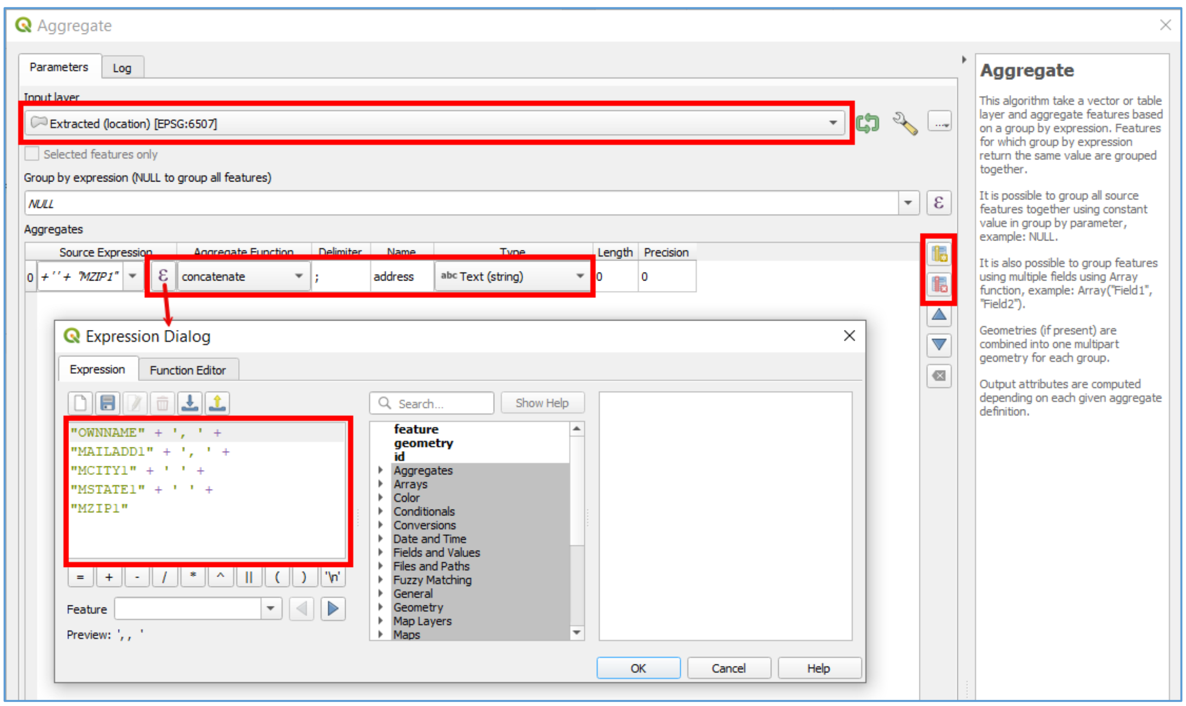

Fig. 2. Setup of the Aggregate tool to generate the formatted list of owners and their addresses.

D. When the expression is set, **Run** the tool without saving the file. It is important not to save the file to a *.shp* file, as the format has a limited length for the text field, and by doing so, the output will be cut. After the tool is done, you can open the attribute table and see the full list of names and addresses in the address attribute. The cell can be copied to external software (e.g., Excel or MySQL) for further processing.

STEP 3. MODEL BUILDER

The process presented in step 2 requires lots of manual work. Now imagine that the mileage changes, or you must include different types of media, e.g., water lines, for the same analysis. And what if your analysis workflow is much longer? In this step, we will use *Model Builder* to create a model that will run all the above steps automatically. This way, we can repetitively use only one tool to recreate all the steps in the analysis. Let's create a new model:

A. Click on the red gear icon at the top of the Processing Toolbox and select Create New Model (Fig. 3).

B. A **Model Designer** window will open. First, in the **Model Properties**, choose the name of the model (Fig. 4). Only after providing the name of the model here will you be able to save the model.

C. Let's think about the structure of our model, starting with the inputs. Considering the above steps in our analysis, we will need four inputs:
   a. **line geometry layer** to provide transmission lines,
   b. **polygon geometry** to provide parcels,
   c. **number** representing the mileage start point,
   d. and another **number** as the mileage end point.

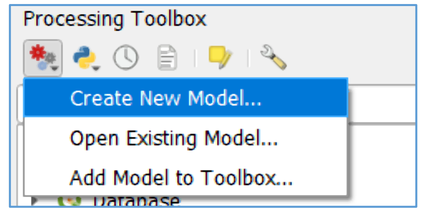

Fig. 3. *Creating new model in QGIS option.*

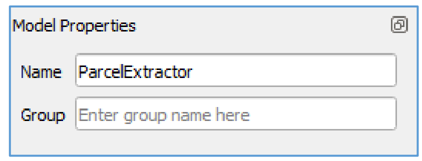

Fig. 4. *Model name needs to be provided in order to save it.*

D. Now that we know what inputs we need, let's add them to the model by selecting the Inputs tab on the left panel. Search for a Number and drag it to the main model window. In the popup, **name** the number as *Mileage Start*, leave **type** as *Float*, and set the **Mandatory** option as required (Fig. 5). Repeat the process for the mileage end point.

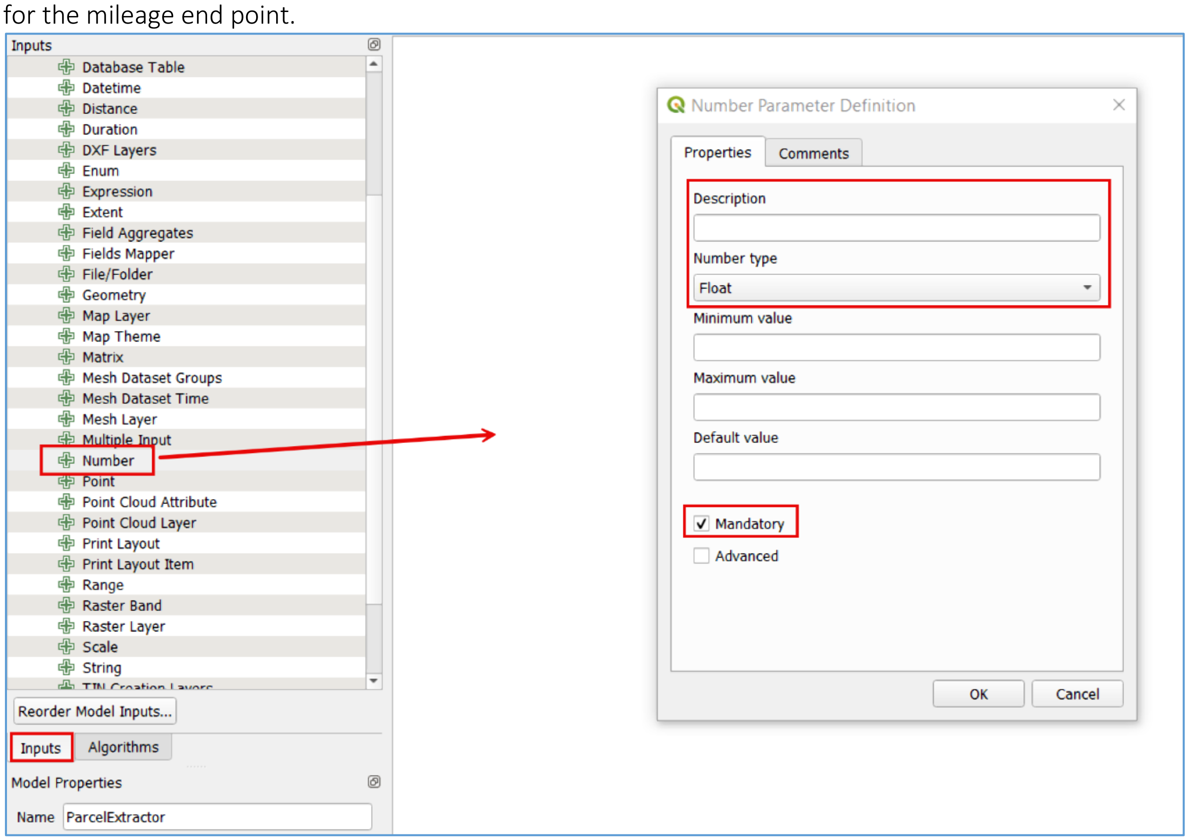

Fig. 5. *Drag and drop the item from the list to add it to the model, then set its parameters in the popup.*

E. Now, search for **vector layer** input and drag it to the model. Set the **name** as *Polygons*, use the **polygon geometry type**, and set the input as **mandatory**. Repeat the process for *lines*, choosing the **line geometry type**.

F. You should now see four inputs in your model presented as light-green rectangles (Fig. 6). You can move them around on the canvas by clicking and dragging them with your mouse.

G. Switch to the **Algorithms** tab and select the first algorithm we used in step 2: ***Line substring*** (under **Vector geometry**). Just like with inputs, drag and drop it into the model. Unlike in step 2, here we won't use hardcoded values, but instead, we will point our algorithm to use model inputs.

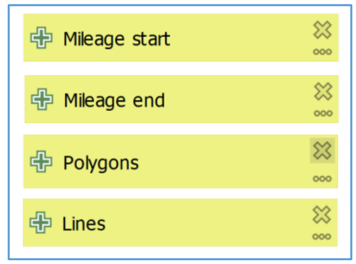

Fig. 6. *Model inputs.*

- Under the **Input layer**, make sure that the first icon is set to **Model Input** (this is represented as a three-gear icon), then select *Lines* (or how you called the input for line geometry);
- Under **Start Distance**, click the icon and change it to **Model Input**. Then select *Mileage start* (or how you called the first number);
- Repeat the process for *end distance* (Fig. 7);
- For **Substring**, if you provide the name in this field, the model will provide you with the additional output from this stage of calculation. Most of the time, you will use this setting for debugging when creating the model, but after the model is set you will remove this output to limit the number of layers created in the process.

H. Click **OK**. You will see how the tool is connected to the three inputs we defined previously. You can rearrange the tiles to increase the readability of the graph.

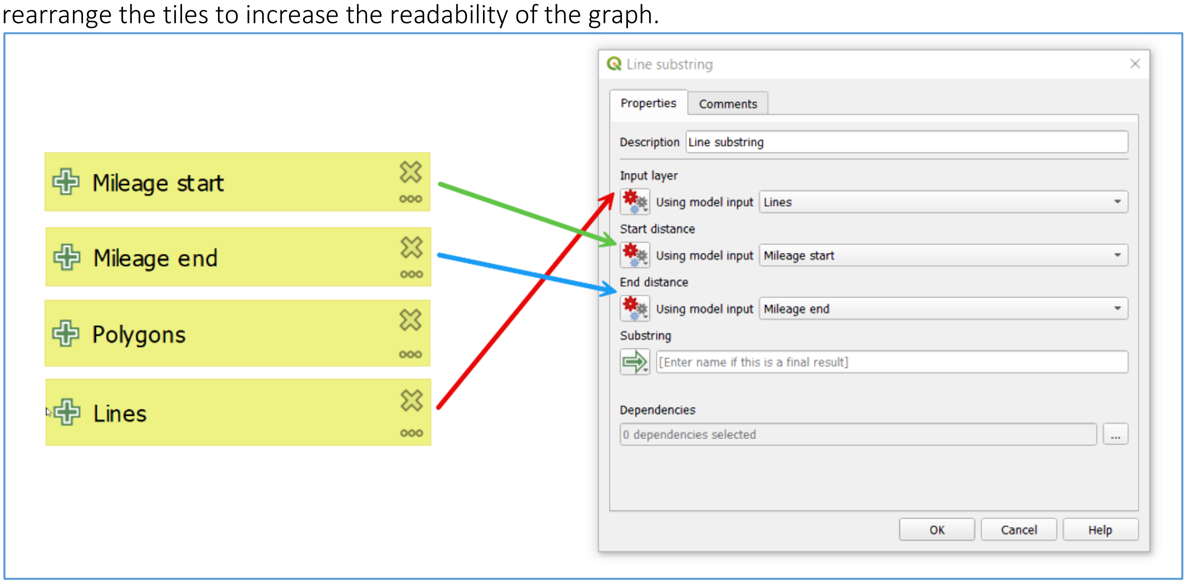

Fig. 7. *When using algorithms inside Model Builder, make sure to correctly setup the inputs to refer to model inputs.*

I. Open the **Vector selection** tab and search for ***Extract by location*** algorithm. Add it to the model. Use **Model Input** under **Extract features from** and set it to the *polygon* input we prepared earlier. For the **Geometric predicate**, click the three-dot icon at the right and make sure that only *intersect* is selected. Change the features from Input to **Algorithm output** by clicking on the icon on the left. Then, in the drop-down menu, select ***Substring***, which we created earlier inside the model. This means that the output of the ***Line substring*** algorithm will be captured by the ***Extract by Location*** algorithm and used as the reference for selection. For learning purposes, provide a name for the output in **Extracted (location)** as *ExtractStep*. Click **OK**. You will see how the *polygon* input and the *line substring* algorithm are now connected with the ***Extract***

by location algorithm. Additionally, a new dark green color output was created, named *ExtractStep*, that we just defined.

J. Add the ***Aggregate*** (**Vector geometry**) algorithm to the model. Use **Algorithm output** from the ***Extracted location*** as the **Input layer**. Now repeat the steps from the algorithm setup presented in step 2c: points B–C, setting the attribute and expression. Save the results as ***output***. Once finished, your model should look like in the figure 8.

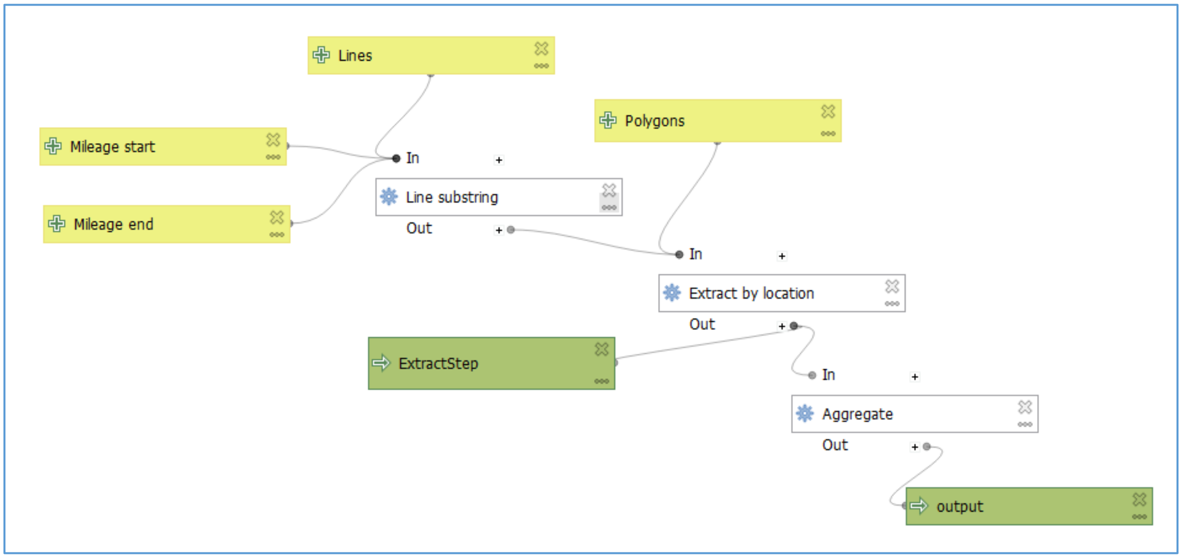

Fig. 8. *Full model scheme, presenting how each component connects to the next.*

K. **Save** your model (choose a location that you will remember) and close **Model Designer**.

You can now run the model from the Processing Toolbox by selecting the **Models** tab and choosing the name of your model. If you don't have the **Models** tab in the toolbox, open the model builder menu at the top of the toolbox (where you created the new model) and select ***Add model to toolbox***. Navigate to the location where you saved the model in the previous step and select it. This will add the **Models** tab to the toolbox. You can run the model just like any other algorithm: provide all the necessary inputs and click **Run**. One thing to note here is that in step 2a, we had the option to choose the unit we wanted to use in the *Line Substring*. When we use the tool inside the model, such an option is no longer available, as our input goes to the *number* variable inside the model before it's passed to the algorithm. For this reason, when running the model, you need to provide the **numbers converted to the default unit used**. This will vary significantly by the region and QGIS version used, with some settings respecting your default unit while others using an input layer unit. The easiest way to verify which is your case is to open the *Line Substring* algorithm from the toolbox and load *transmission_lines* as input. This will set the **default unit** under the **distance** setting. In my case, this was set to feet, which means that when using the model, I will use **Start Mileage** as **10560** (feet, which is 2 miles) and **End Mileage** as **36960** (feet, which is 7 miles). Keep in mind that the inputs in the model window are provided alphabetically; therefore, you are being asked for the end point first and the start point later. Once completed, compare the results from the model output to the results of the manual analysis.

STEP 4. DEBUGGING [OPTIONAL]

You might receive an **error** stating that the **substring geometry is invalid**. If this is the case, you need to modify the model. Simply right-click the model in the toolbox and select **Edit Model**.

A. In the **Model Designer** *Algorithms* tab, find the ***Fix Geometries*** tool (**Vector Geometry**) and add it to the model.

B. As the **Input layer**, use the **Algorithm output** from the ***Line substring***. Click **OK**.
C. Double-click the ***Extract by location*** algorithm and change the **By comparing to the features** from the output of the ***Fix geometries*** algorithm.
D. If you want to **remove** the additional output that we have created during the ***Extract by location*** step, simply click the **x** icon in the top right corner of the *ExtractStep* output.
E. Save the model and rerun it.

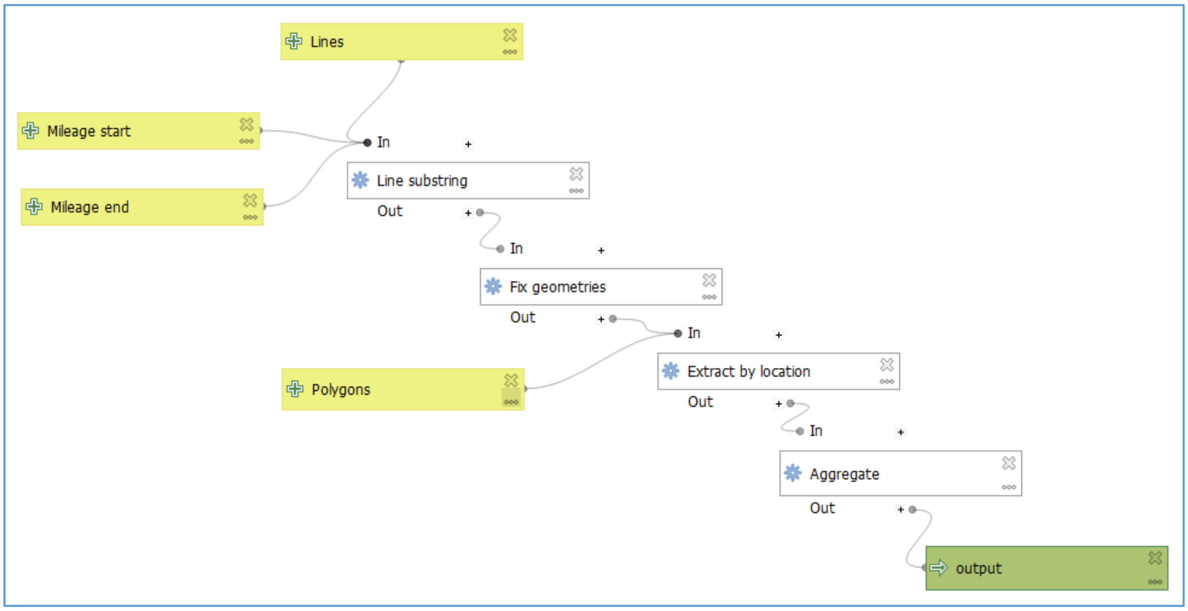

Fig. 9. *Final model with geometry fixing after substring generation.*

At this point, you are ready to present the outcomes to the County Media Management Company. Also, the next time you receive a similar assignment, you can simply edit the model created above and complete the assigned task in no time. This concludes our tutorial. Feel free to experiment and modify the model or start creating new ones to expand your processing abilities and optimize your workflows.