

An Application in GIS for a Sanitary Sewer Overflow Emergency Response Program



**MGIS Capstone Project
Pennsylvania State University**

**Author:
Mike Pritchard**

**Date:
7/22/2014**

**Faculty Advisor:
Dr. Jan Oliver Wallgrün**

Abstract

The Medina County Sanitary Engineers Office (MCSE) operates and maintains approximately 550 miles of sanitary sewer line. One of the key components of an effective operational plan is to have a plan in place to respond in the event of a Sanitary Sewer Overflow (SSO). The sanitary trunk sewer (30" or larger pipe that acts as the main artery of the sewer system) has a great deal of flow passing through at any given time. To help in the emergency response plan in the case of a SSO on this trunk sewer, a solution was needed to integrate the existing GIS system into the plan. The goal of this project was to develop a GIS tool that uses existing GIS data to create a report for assistance in the response to a SSO. This report contains information meeting MCSE needs (estimated ERU's, number of customers, diversion options, necessary or unusual equipment), environmental factors, emergency contacts, and other valuable information in the event of an SSO. The form is generated through a geoprocessing service on ArcGIS Server which was created using Python and Model Builder. The script considers location and user defined parameters to generate the information on the report. In addition reference maps are generated to accompany the report. The GIS tool is designed to run through an existing web mapping application (MCSE uses flex viewer for ArcGIS) as a geoprocessing widget. The focus of the tool is to be quick and simple to use but also contain relevant, detailed information to assist in the response to an SSO. Although the tool is designed to meet MCSE needs, the design also takes into consideration how other organizations could implement the tool.

Table of Contents

Project Background	4
Project Goals	7
Literature Review	8
Components of Project	10
HTML Report	11
Geoprocessing Workflow	14
Primary GIS Layers	15
Workflow Model	16
Python Geoprocessing Script	17
Running the SSO Tool	19
ArcGIS Server Geoprocessing Service	20
Packaged Data	23
Project Conclusions	25
References	28

Project Background

The Medina County Sanitary Engineer's office is a public utility managing approximately 550 miles of sanitary sewer line within Medina County, OH (Figure 1). The system is divided into four districts based on treatment plant service areas. The primary district (Sewer District 500) is served by the Liverpool Wastewater Treatment Plant located in the northeast portion of the county. This plant treats the sewage from Medina City and surrounding residential and commercial areas as well as western and southern Brunswick City. Figure 2 provides an overview of the sewer district and the tributary sewers of district 500.

The core of the population served in sewer district 500 is a considerable distance away from the treatment plant. To properly transport this sewage to the treatment plant, two trunk sewers were built mostly through easements often a considerable distance from streets and corresponding right-of-ways. Medina County uses the term trunk sewer to define any sewer line that is over 30" in diameter and the primary role is a transport line. Most connections along these lines are tributary sewers and not individual laterals. The primary trunk sewer starts close to the center of Medina City at a diameter of 36" and extends approximately 7 miles where it meets the treatment plant with a diameter of 60" (Figure 2). There is also a smaller trunk sewer that starts in western Brunswick City at a size of 30" and extends to the treatment plant where the diameter increases to 36".

These two trunk sewers are a critical part of the sanitary sewer system as all sewage going to the Liverpool treatment plant pass through some portion of these sewers. Routine maintenance is performed such as easement clearing along these sewer lines and regular inspections are completed on a regular basis. In addition televising of the sewers has been performed to identify areas of concern inside the pipe. Medina County uses this information in conjunction with a GIS to be proactive in preventing failures along the trunk sewer.

A necessary management component of a sanitary sewer system is to have a plan in place to handle sanitary sewer overflows (SSO). An example of an SSO occurrence would be a sanitary pipe collapse and the sewage ultimately flowing to the surface as the sewer network is blocked. This will result in a serious hazard potentially damaging the environment and causing health concerns. For Medina County, a plan is already in place to report and respond to an SSO, but the trunk sewer provides some unique issues not present in the rest of the system. Based on the location, size and material of the pipe, a quick and organized approach to an SSO on the trunk sewer is difficult. The flow within these systems is also considerably higher resulting in diversion challenges and potential environmental impacts. Medina County is currently being proactive in trying to avoid a potential collapse but the risk is always there for a collapse to happen resulting in a SSO.

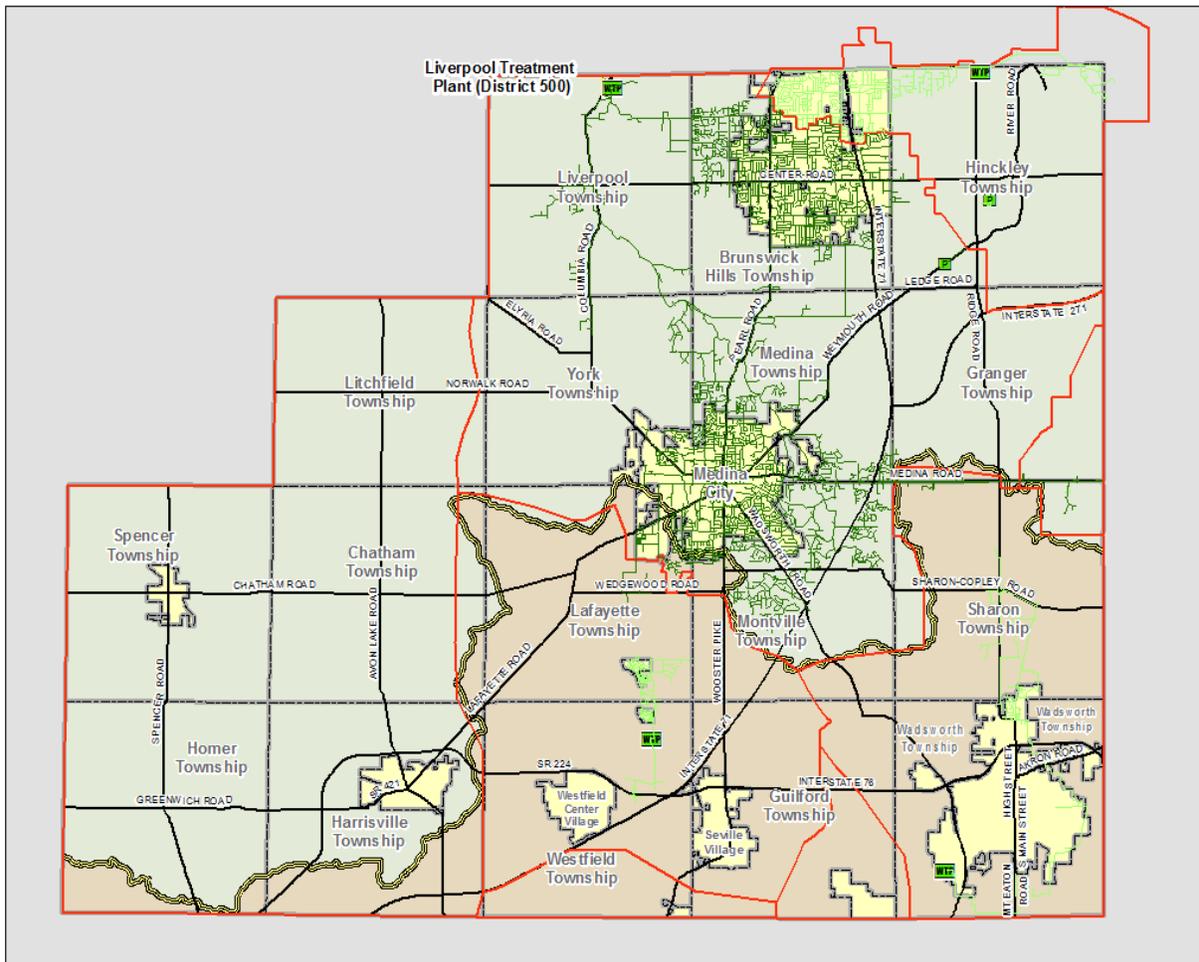


Figure 1: Overall map of the MCSE service area

In 2009 a portion of the trunk sewer collapsed causing an SSO to occur which Medina County was not completely prepared for. From this event it was clear that a better management plan was needed to handle the occurrence of an SSO, primarily an occurrence on the trunk sewer. After the event, considerable effort was put into analyzing the condition of the sanitary trunk sewer and measures to repair and monitor problem areas. Routine maintenance and inspections are administered on a regular basis along the trunk sewer but more so in trouble areas.

Although Medina County was proactive in trying to prevent an SSO from occurring, they felt it would be beneficial to have an emergency response plan in place in the event of an SSO. With a GIS already established for the MCSE office, the idea was presented to develop an application through the GIS to assist with the response to an SSO. Most personnel are currently trained and familiar with the online web map so it would be fitting to have the application run through the web map. This application would also be able to be used to further improve the guidelines in the SSO emergency response procedures.

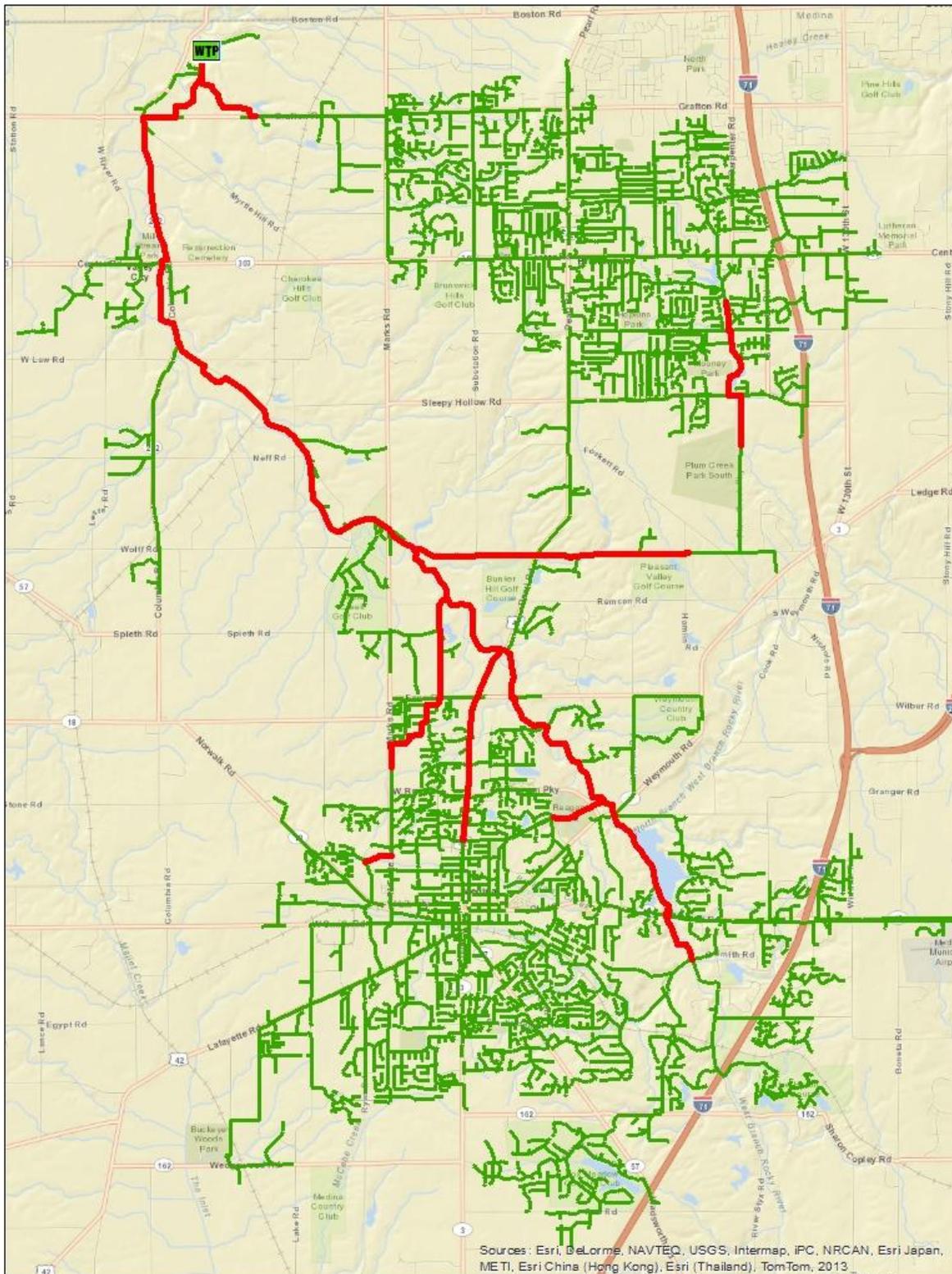


Figure 2: Map of the 500 sewer district with the trunk sewers highlighted

Project Goals

The project has two main parts with a set of goals for each part. The first part of the project was to design a detailed report that will assist a sanitary sewer entity to respond to an emergency SSO. The second part of the project was to create a tool within a web mapping application to generate the designed report based on the site of the SSO. Each of these parts had a subset of goals which were met to complete the project. Listed below is the initial outline of the project goals.

(1)Emergency Response Report

- *Printer friendly report of necessary actions for SSO*
- *Detailed information in an easy to read and understand format*
- *Information includes:*
 - *System factors based on location of SSO (flow, access, etc.)*
 - *Field response needs (equipment, personnel contacts, etc.)*
 - *Professional & public notification needs (environmental concerns, customer backup potential)*
 - *Additional information related to incident*
- *Location map included on the report*

(2)Mapping Application Tool

- *Run as a geoprocessing service*
- *Generates the HTML report based on SSO location selected*
- *Location map to assist in response to the SSO*
- *Easily customizable to adapt to changes in GIS systems and necessary layers*
- *Packaged for distribution*

Literature Review

The focus of the literature review is to further define what an SSO is and the major implications of when they occur. Seeing how this project is a tool in the response to an SSO, it's appropriate to further understand the event and also look at case studies of how to respond to the event. By looking at guidelines and examples of response plans to SSOs, additional guidelines of what to include in the project plan were developed.

An SSO is defined as a discharge of untreated sanitary sewage from the result of a broken pipe, equipment failure, or overload on the system [1]. The untreated sewage can be directly discharged into streets, and other public areas as well as streams, lakes and other surface waters. The impacts of a SSO can have serious effects on the environment, health of the nearby population, and potential contamination of drinking water sources. Due to the attributes of untreated sewage, the threat of disease can be very high with contaminated areas needing to be thoroughly cleaned and disinfected.

The majority of reported SSOs identify a traceable condition which happened immediately such as a pipe break. Although this may be the immediate problem, the SSO is often caused by an underlying problem that may cause future SSOs [2]. Some of these underlying problems may be age of the pipe, poor operations and maintenance, lack of maintenance, or inadequate flow capacity. Although the immediate action to an SSO is to discover the problem and fix the issue in order to prevent additional discharge and get the system back into normal operation, additional evaluation of the system may be necessary after an occurrence.

Having an effective emergency response plan to an SSO will help define how to respond, control, report and mitigate the event. The benefits of an effective emergency response plan are outlined below as described by the American Public Works Association (APWA) and the American Society of Civil Engineers (ASCE) [3].

- Enhance the protection of public health and the environment.
- Provide compliance with regulations and permits.
- Maintain trust with the public, the regulatory agencies and the non-governmental organizations.
- Minimize the wastewater agency's exposure and liability from claims, enforcement, or litigation.

In addition to the benefits of an effective SSO emergency response plan, the APWA and ASCE also provide elements that should be included in response planning. Also essential resources are included as these should be at the agencies readiness in case of a necessary response. This literature resource is very beneficial in identifying key concepts in the development of this project.

Some case studies related to management plans are also beneficial in further understanding and planning for an SSO emergency response. The City of Delmar, California, presents a sanitary management plan [4] which provides a plan for monitoring the system to prevent SSOs as well as how to respond in the case of an emergency SSO. Greenwood County, South Carolina, also has a beneficial case study published which explains their Comprehensive Management, Operations, and Maintenance Plan [5] to present a way to prevent SSOs from occurring. Having a management plan in place brings community leaders together and alerts the public of a proactive approach to SSOs.

The County of DeKalb, Georgia, presented a SSO Contingency and Emergency Response Plan [6] as part of their Capacity, Management, Operations, and Maintenance Program. This provides a complete example of responding to an SSO including field response activities, public notifications, reporting, investigating and training.

These literature works provided great reference in developing a plan for this project. Although this project is a tool to be used in the case of a particular SSO, the background is necessary to understand. Understanding where other communities have successfully implemented plans will attribute to the success of this project.

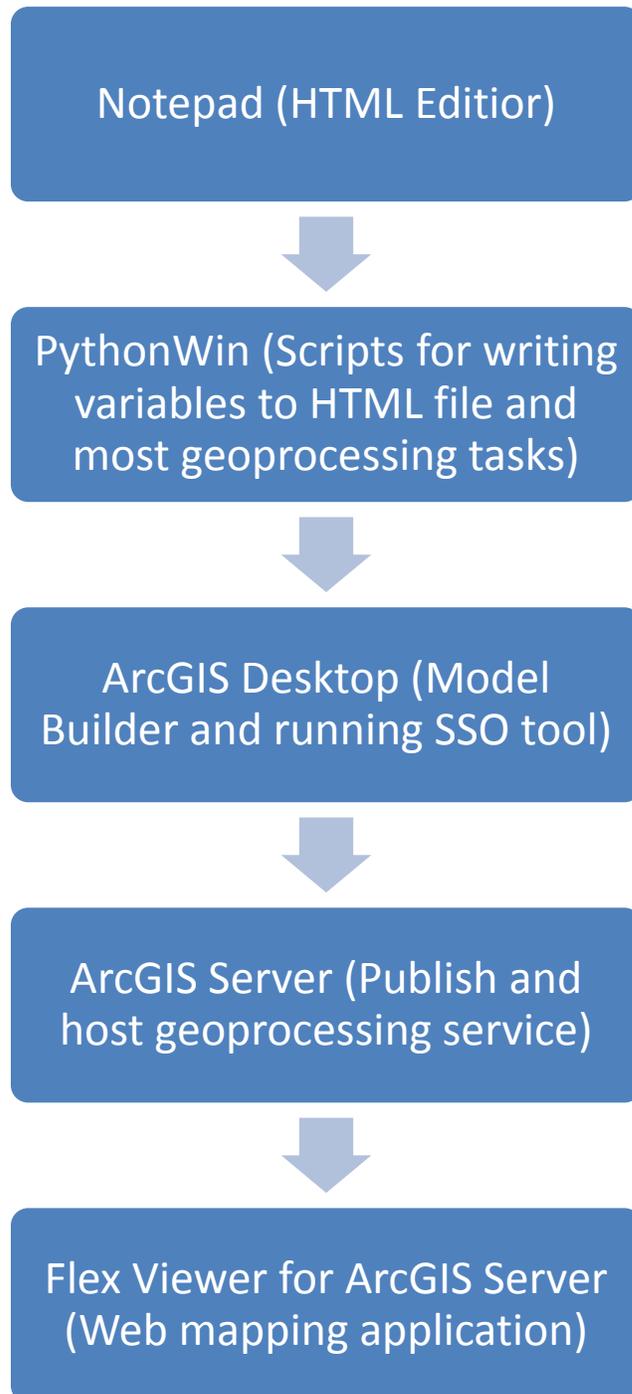
The American Society of Civil Engineers in cooperation with the EPA developed a report to assist in identifying solutions to SSOs [7]. This report covers solutions on methods to avoid SSOs but also provides information in what to include in an overflow response plan. Although there is a great deal of information included, the focus was on the elements of the literature on the overflow response plan.

An additional article by the American Society of Civil Engineers discusses the protocols for identifying an SSO [8]. The focus of this document is on the identification of SSOs to provide a comprehensive set of protocols to develop preventive measures for SSOs. Although the focus of this project is on SSO response as opposed to prevention, the identification of an SSO is necessary to effectively respond to the event.

Dana Abu Sier published an article in 2005 where a hydraulic model was used to determine ideal locations for meters to monitor potential leaks on a sanitary system [9]. The paper discusses a methodology of using a hydraulic model to monitor the system during normal conditions but then also placing blocks at determined locations and times to simulate an SSO. This methodology provided some insight in the development and testing of the network tracing component of the project.

Components of Project

The project uses several different applications together to produce the end-product. The diagram below lists the different products and what their purpose is within the project. Each part is described further in the corresponding sections of the report.



HTML Report

The HTML report represents the end product to be made available to the end user after the application is run. This report was designed using basic HTML and formatted for printing on a letter size sheet of paper. The original HTML document was developed using notepad and eventually implemented into a Python script. The Python script allowed for variables to be easily plugged into necessary areas within the HTML file.

With the use of the basic Arcpy command of 'GetParametersAsText', variables were created to allow for derived values to go into the HTML. The HTML is written to a text file through the Python script. Figures 3, 4, and 5 show some screenshots of the HTML being derived through a Python script.

```
27 bufferRadius = arcpy.GetParameterAsText(16)
28 nearStreams = arcpy.GetParameterAsText(17)
29 nearLakes = arcpy.GetParameterAsText(18)
30 nearComm = arcpy.GetParameterAsText(19)
31 easementWidth = arcpy.GetParameterAsText(20)
32 htmlFile = arcpy.GetParameterAsText(21)
33
34 html_f = open(htmlFile, "w")
35
36 html_f.write("""
37 <html>
38 <body>
39
```

Figure 3: Screenshot of setting variables for the HTML report and also opening up an HTML document to write to.

```
45
46 <tr valign="top">
47 <td style="background-color:lightgray;width:600px;height:175px;text-align:left;border-width:2;border-color:black;">
48 <b>Reporting Information</b><br><br>
49 """)
50
51 html_f.write("Date: <b>" + str(datetime.date.today()) + "</b><br>\n")
52 html_f.write("Current Weather Conditions: <b>" + weatherDesc + "</b><br>\n")
53 html_f.write("Specific Location: <b>" + locDesc + "</b><br>\n")
54 html_f.write("Description of SSO: <b>" + ssoDesc + "</b><br>\n")
55 html_f.write("Reported By: <b>" + reportBy + "</b><br>\n")
56 html_f.write("Contact Number: <b>" + contactNum + "</b><br>\n</td>\n</tr>\n")
57
```

Figure 4: Screenshot of inserting HTML text into the document. This also includes using a function to set the date and time of the report being generated.

```
67 html_f.write("""
68 <tr valign="top">
69 <td colspan="2" style="background-color:lightgray;width:1000px;height:200px;text-align:left;border-width:2;border-color:black;">
70 <b>Sanitary Sewer System Information</b><br><br>
71 """)
72
73 html_f.write("In Easement: <b>" + easement + "</b> Easement Width: <b>" + easementWidth + "</b><br>\n")
74 html_f.write("Pipe Diameter: <b>" + pipeDiam + "</b><br>\n")
75 html_f.write("Manhole Casting Grade: <b>" + mhGrade + "</b><br>\n")
76 html_f.write("Total ERU's: <b>" + avgFlow + "</b><br>\n")
77 html_f.write("Customers Upstream: <b>" + custUpstream + "</b><br>\n")
78 html_f.write("Sewerline Upstream Below Overflow: <b>" + mhOverflow + "</b><br>\n")
79 html_f.write("Customers Below Overflow: <b>" + custBasement + "</b><br>\n</td>\n</tr>\n")
80
```

Figure 5: Screenshot of plugging in variables defined through the geoprocessing tool which get plugged into the HTML file at the appropriate locations.

The final derived report is divided on the page into organized, easy to interpret categories to provide the end-user quick analysis of the situation. On the example HTML page (Figure 6) the categories are broken down in what they represent. Some of the information is user input derived from the reported call. Once this information is collected and the SSO location is identified, the sewer tributary trace is completed to collect additional information (this will be discussed in much more detail in the following geoprocessing workflow section). Finally some information is standard to the report and also should be included in an SSO emergency response plan.

When an SSO is called in, pertinent information will be collected from the caller to potentially assist in responding to the SSO. Also it is good to have a record of all parties involved in case follow up information is needed or a follow up response is necessary. This information will be entered through the geoprocessing tool to be included on the report. In reference to the following example, this information consumes the 'Reporting Information' and 'MCSE Details' portions of the report.

The next section ('Sanitary Sewer System Information') will be derived from information collected in the sanitary tributary trace. Based on the SSO location, sewer network information is collected along with details of what is in the immediate vicinity ('Environmental Concerns'). This relates to water features and population density which could be affected.

In addition to the derived information, there is also standard information related to an SSO ('Factors to be Considered' and 'Recommended Response Steps') available for field personnel. This information is used to provide assistance in response planning along with steps to follow in responding to the SSO. This data was derived mostly from the literature review and also information collected from the MCSE office. Most of this information should be included in an SSO plan and personnel should be trained on these steps ahead of time. In the time of emergency it will be beneficial to provide the information to prevent anything being missed.

The report also includes contact information ('Contacts') for important local entities and necessary contractors and equipment suppliers. As the goal of this report is to assist in the quick response to an SSO, this information is only fitting as it alleviates the need to look up contact information. Also this prevents the potential overlook of contacting an entity such as the EPA in the event of an environmental hazard.

The location map provides a visual reference to personnel in determining the location of the SSO. This also provides information of what is in the immediate vicinity to assist in planning for equipment or material needs. With most systems, many miles of sewer line are maintained which makes it impossible for personnel to fully know the dynamics of the system. This small map provides much more beneficial information than the accompanying address and manhole number included in the report.

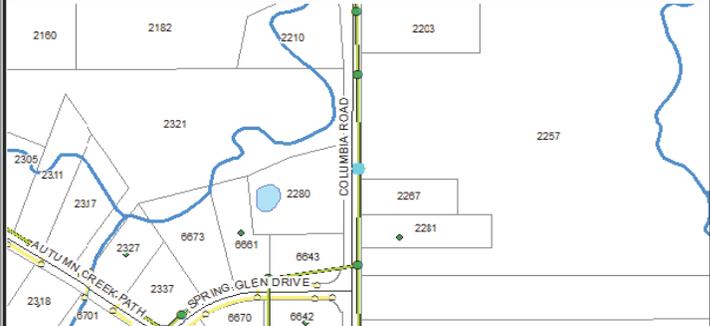
<h2>SSO Response Plan</h2>	
<p>Reporting Information</p> <p>Date: 2014-04-26 Current Weather Conditions: Rain Specific Location: Columbia Rd Description of SSO: Sewage leaking from manhole Reported By: John Smith Contact Number: 323-444-5656</p>	<p>MCSE Details</p> <p>Call Taken By: Bill Manhole ID at SSO: 3719 Personnel Reported To: Jim and Dave Resident complaining of sewage leaking from manhole.</p>
<p>Sanitary Sewer System Information</p> <p>In Easement: No Easement Width: NA Pipe Diameter: 12 Manhole Casting Grade: 892.61999511 Total ERU's: 265 Customers Upstream: 139 Sewerline Upstream Below Overflow: 4065.40005892477 Customers Below Overflow: 54</p>	
<p>Factors to Consider for Equipment Needs:</p> <ul style="list-style-type: none"> SSO location Stream crossing and slope Manhole casting grade Track hoe or mini track hoe Diversion pump based on expected flow Cleanup supplies 	<p>Environmental Concerns</p> <p>Buffer Radius: 4000 Feet Is there a stream(s) nearby: Yes Is there a lake(s) nearby: Yes How many addresses are nearby: 270</p>
<p>Recommended Response Steps</p> <ul style="list-style-type: none"> Perform initial investigation and determine cause of SSO Develop plan and fix the cause of the SSO Contain the SSO Perform any necessary clean-up Provide notification and report incident Complete documentation Further investigative work to prevent future occurrences 	<p>Contacts</p> <p>Environmental Ohio EPA: 1-800-282-9378</p> <p>Contractors & Suppliers Contractor A: XXX-XXX-XXXX Contractor B: XXX-XXX-XXXX Supplier A: XXX-XXX-XXXX</p> <p>Local Government Entities Medina County Engineer: 330-723-9561 Medina County Health Department: 330-723-9523</p>
	<p>(Enter Field Notes Here)</p>

Figure 6: Sample HTML report generated from the SSO geoprocessing tool.

Geoprocessing Workflow

Overview

As this project is designed to be used as a geoprocessing service through a GIS web map, a functional geoprocessing workflow was necessary. Everything needed to be connected together to be included in a complete geoprocessing service which could be run as a widget through an application such as Flex Viewer.

Most of the time put into the project focused around the geoprocessing workflow. The HTML file was the first part of the project and completed before much work was put into the rest of the workflow. This allowed for more structure in putting together the different processes. The workflow consisted of migrating HTML, Python and ArcGIS Model Builder together to produce a complete tool. Although many of the processes could have been completed in Model Builder alone, Python provided more in depth customization for certain geoprocessing tasks.

Within the diagram in Figure 7, the geoprocessing workflow is outlined. The first step of the process is to run a geometric network trace of the SSO location. From the trace pertinent information is collected and used to gather information related to the sewer system necessary for the report. This information is then pushed to the HTML report mentioned previously to be used during the SSO. Although many processes are going on in the background, the overall goal of the report is to present all necessary information in a compact, easy to comprehend HTML report.

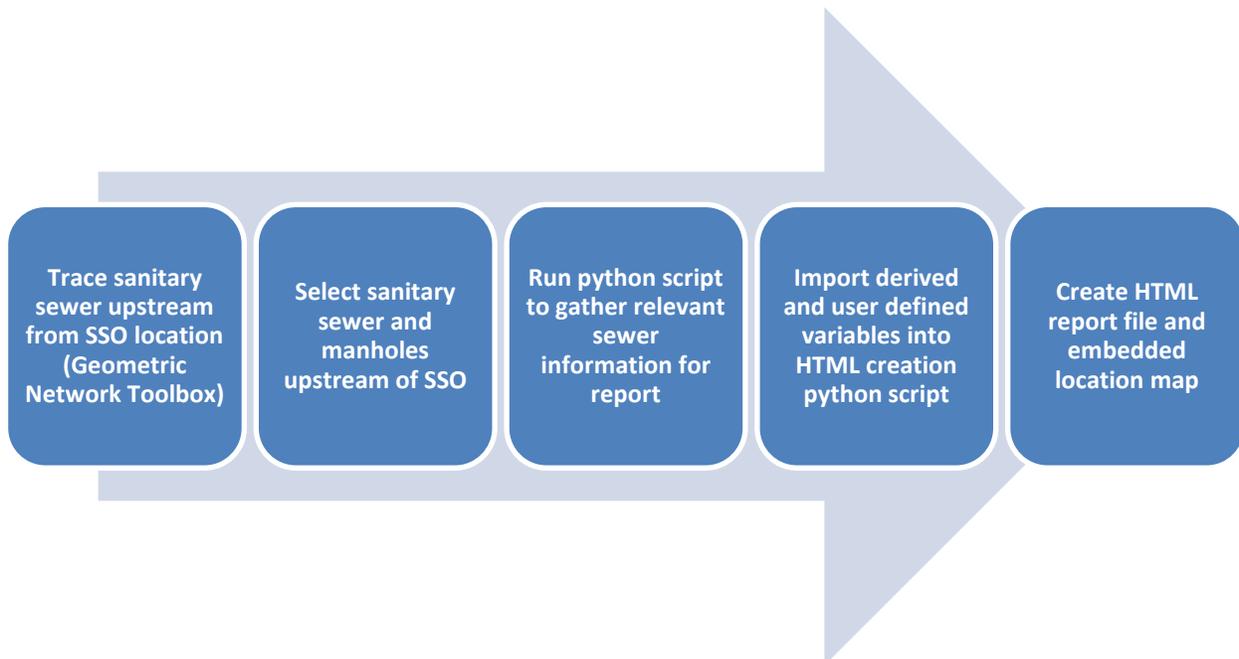
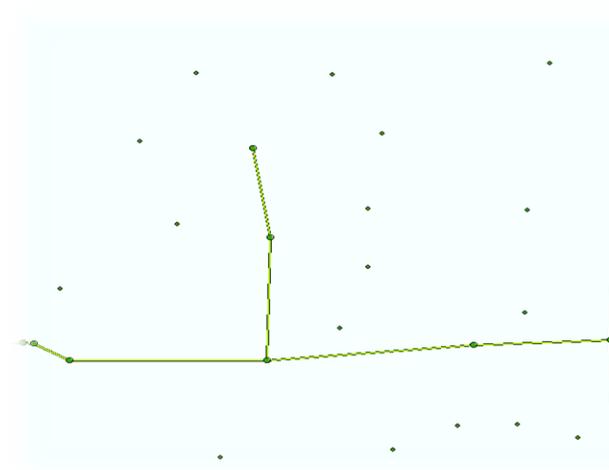


Figure 7: Basic workflow of the SSO report.

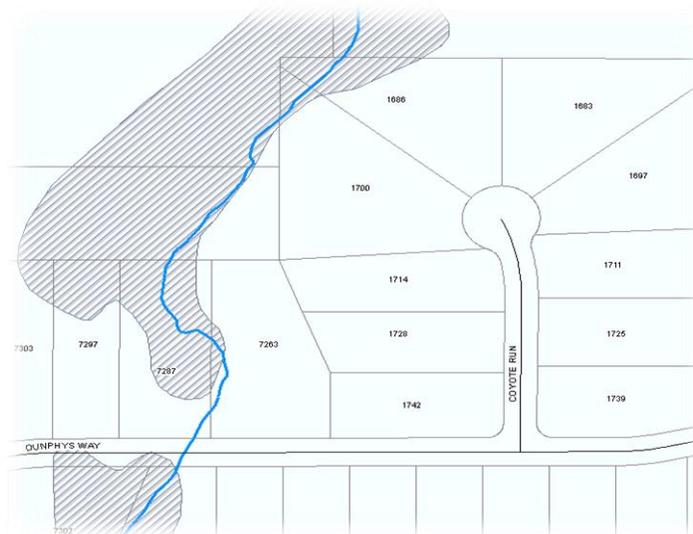
Primary GIS Layers

In order for the workflow to run, several GIS layers are necessary. In the packaged data discussed later, sample layers are included representing these layers.

- Sanitary Layers
 - Sanitary Sewer Lines
 - Sanitary Manholes
 - Sewer Customer Points



- Base Layers (Used in Script)
 - House Points
 - Streams
 - Lakes
 - Easements



Workflow Model

Although notepad and Python were used to create most of the processes in the application, ArcGIS Model Builder is where everything is brought together. In Figure 8, the entire geoprocessing workflow is displayed with groupings highlighted. The initial network trace and selection of upstream sewer lines and manholes are completed directly in the model. The model is also responsible for identifying the nearest manhole to the SSO for determining the sewer system network location of the SSO.

Once the tributary network is identified, the information is pushed to a Python script to gather additional information related to sewer system and immediate area. This information is then accessed as text variables to plug into the HTML script discussed previously and exported as a report.

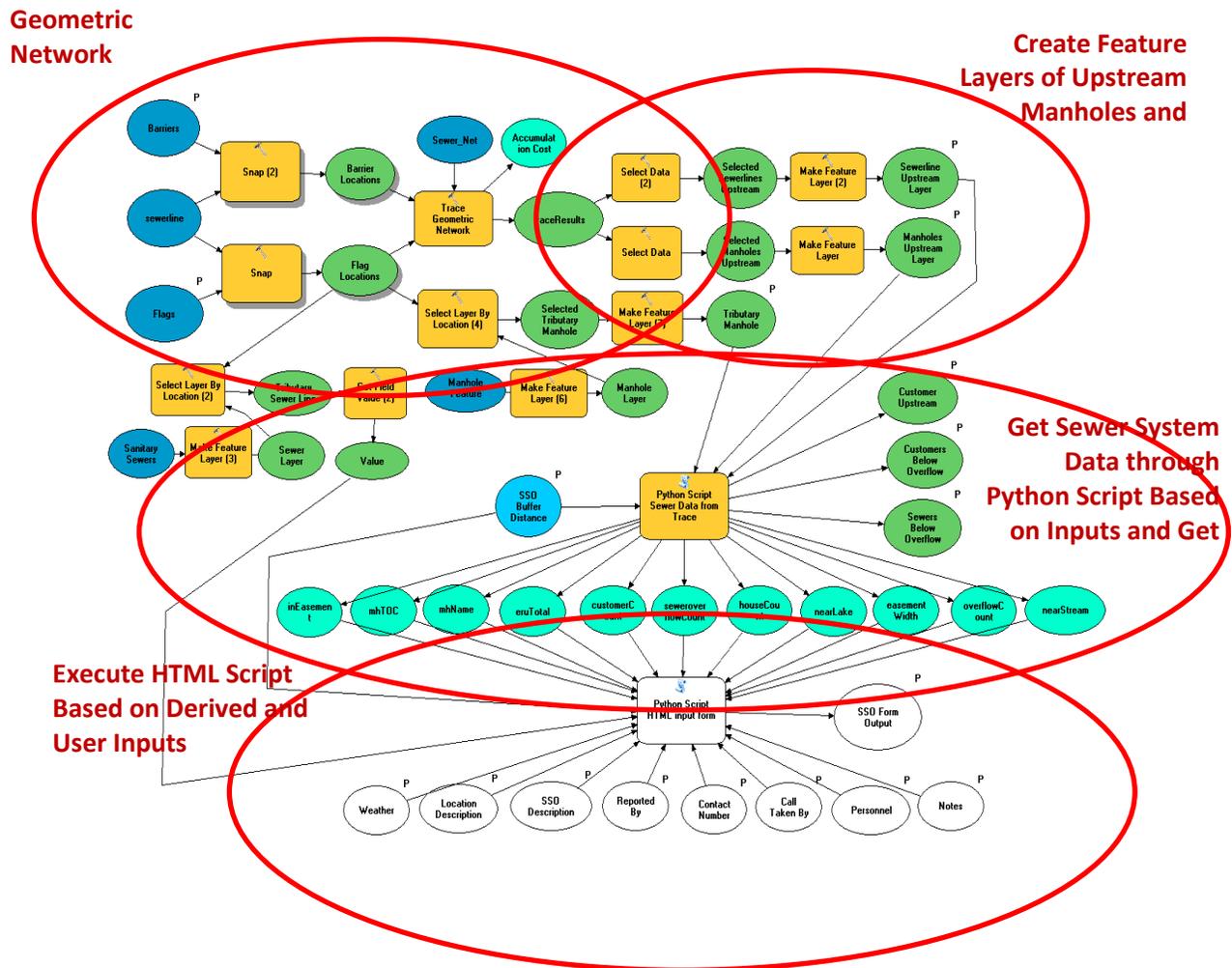


Figure 8: Image of Model Builder workflow for the SSO geoprocessing tool.

Python Geoprocessing Script

As described in the previous image, the center of the model focuses on getting sewer system data through a Python script. This is where most of the work and processing takes place in running the tool. By using the layers created in the tributary trace, much more information is gathered related to the system and surrounding area.

In the screenshots included below a couple sections of the script are displayed. In the first screenshot (Figure 9), the variables are set to be used in the script from the Model Builder geometric trace operations. Once these are set, one of the first actions needed is to identify all customers upstream of the SSO. Due to the current state of the MCSE GIS system, customer points have not been attached to the mains by laterals to complete the network. With the number of customers and amount of corresponding work necessary to complete this, a workaround was used to locate customer points within a distance of the selected sewer lines. Through testing, it was determined that the majority of customer points were within 300' of the sanitary sewer lines. By making this the standard buffer in the selection criteria, the majority of customers were selected to make the information reliable. Once the customer points are networked to the sewer system, the process can be written to collect this information directly from the network trace as opposed to using the 'Select Layer by Location' tool.

The code in the second screenshot (Figure 10) takes the customer layer created in the first screenshot and collects what the ERU total is of all customers. This is a value used by the MCSE office to estimate sanitary usage for billing purposes. One ERU represents a single family household's usage and is the lowest possible value a customer can have. For high volume users several ERU's may be assigned to the account for billing purposes. For the benefit of the report, the ERU's are added together and can be used to determine how much flow should be expected at any location of the sewer system. This information can be analyzed for multiple benefits in the response from diversion pump/equipment needs or environmental concerns by volume of overflow.

```
8 #Parameter Variables
9 selectedManholes = arcpy.GetParameterAsText(0)
10 selectedSewers = arcpy.GetParameterAsText(1)
11 traceManhole = arcpy.GetParameterAsText(2)
12 customerLayer = arcpy.GetParameterAsText(3)
13 overflowCustLayer = arcpy.GetParameterAsText(4)
14 overflowSewersLayer = arcpy.GetParameterAsText(11)
15 bufferDistance = arcpy.GetParameterAsText(12)
16
17 #Select Upstream Customers
18 arcpy.MakeFeatureLayer_management("Sewer_System_Sanitary_Sewer_Customers", "selectedCustomers")
19 arcpy.SelectLayerByLocation_management("selectedCustomers", "WITHIN_A_DISTANCE", selectedSewers, "300 Feet")
20 arcpy.MakeFeatureLayer_management("selectedCustomers", customerLayer)
21
22 #Get Manhole ID and Elevation Attributes of Manhole where Trace Originated
23 cursor = arcpy.SearchCursor(traceManhole)
24 -for row in cursor:
25     mhName = row.getValue("PT_ID")
26     mhTOC = row.getValue("ELEV_NUM")
27
```

Figure 9: Screenshot of retrieving variables from model and selecting sewer customers upstream of the SSO.

```

86 #Calculate ERU's Upstream
87 customerCursor = arcpy.SearchCursor(customerLayer)
88 customerRow = customerCursor.next()
89
90 #Make count and ERU Counter
91 count = 0
92 eruCount = 0
93
94 #Loop through customers and get count and eru's
95 -while customerRow:
96     eruValue = customerRow.getValue("ERU_NUM")
97     count = count + 1
98     -    if eruValue > 0:
99         eruCount = eruCount + eruValue
100        customerRow = customerCursor.next()
101     -    else:
102        customerRow = customerCursor.next()
103
104 #Push values to Parameters
105 eruTotal = eruCount / 100
106 customerCount = count
107

```

Figure 10: Screenshot of calculating number of ERU's upstream of SSO.

In addition to the customer and ERU examples, the Python geoprocessing workflow also calculates what sewer line and customers are upstream and below the SSO overflow elevation. This is completed by identifying the manhole rim elevation of the closest manhole to the SSO. This value is then used to select all sewer lines upstream with an invert lower than this overflow value. This is important to help identify customers upstream who may be susceptible to sewage in their basements.

Many of the sewer lines which would be susceptible to a major SSO are located out of road right-of-ways and require off road travel to access. Most of the sanitary sewer easements have been digitized into a GIS layer for reference by the MCSE. The script will identify if the SSO location falls within an easement and the size of the easement. This helps to determine how to access the SSO and also verify that the MCSE personnel are in the legal boundaries of accessing the SSO site.

The final portion of the Python geoprocessing script is to identify pertinent features nearby of the SSO. Using a 'Select by Location' a buffer is used (adjustable through a variable) that will identify nearby streams and lakes. This buffer will also identify nearby households to determine how densely populated the area is. For the streams and lakes, the app only looks to see if either exist within the buffer but for households the actual count is given on the report.

Running the SSO Tool

With everything working together through the ArcGIS model, the geoprocessing tool is run through ArcMap. A tool window will open as in Figure 11 which will allow for parameters to be set and the SSO location to be identified on the map. All values except for barriers are required to be entered. This will prevent users from not entering or forgetting to enter pertinent information. The tool will also allow for the user to decide where the HTML file should be saved for retrieval.

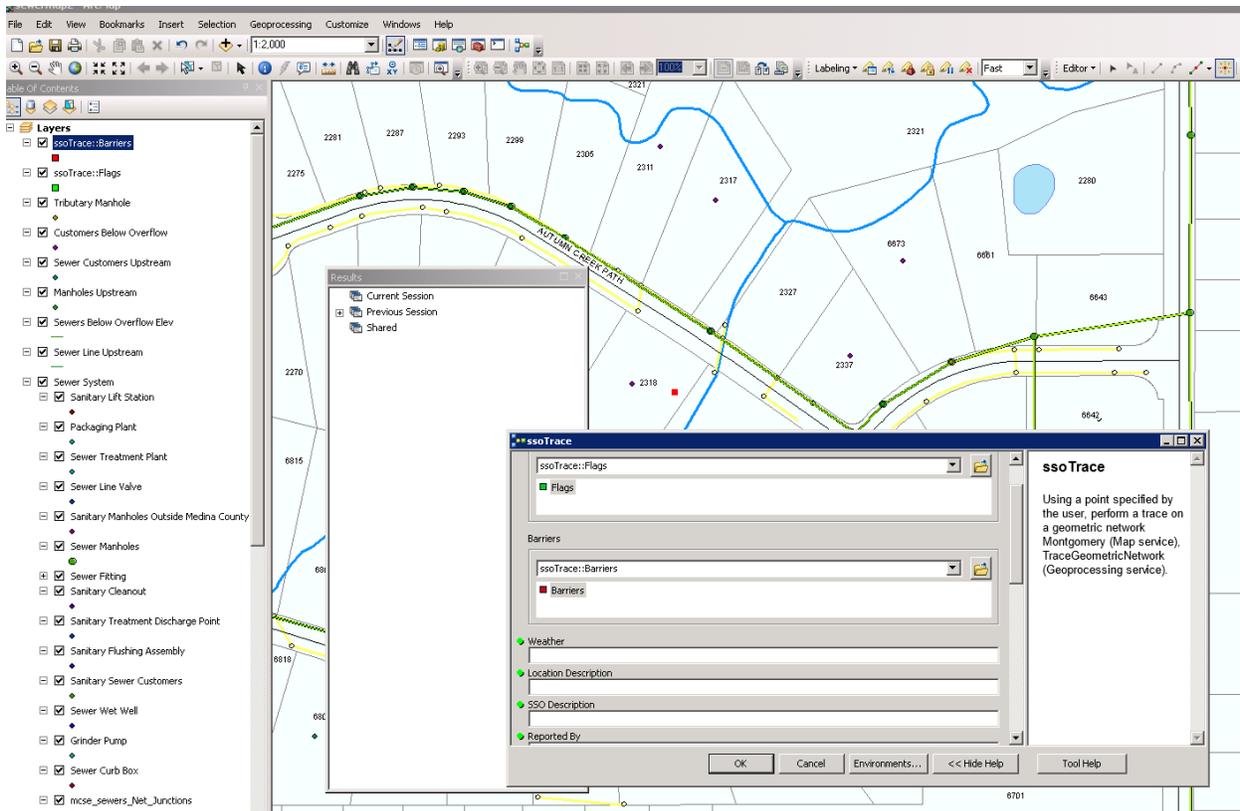


Figure 11: Screenshot of the geoprocessing tool within ArcMap.

The geoprocessing tool is where the project technically stops in terms of supplied data. The app is designed however to be run as a geoprocessing service through a web mapping application. ArcGIS Server is required to move forward to this part of the application which will be discussed in the following section. If an organization does not have ArcGIS Server or chooses to not use it for this tool, all functionality is still available to produce the report through ArcMap.

ArcGIS Server Geoprocessing Service

To fully utilize the design of the SSO tool, the tool can be published as a geoprocessing service using ArcGIS server. This will allow for users to run the tool without access to ArcGIS desktop or the knowledge to run the desktop application. In considering the MCSE office, most potential users have been trained how to use the existing web mapping application. The service will be inserted into the existing web map as a widget customized to existing map services. Training should be straightforward as most users are already familiar with running geoprocessing services through the web map.

The first step in making this application available as a geoprocessing service is to run the application through ArcGIS desktop. Using the catalog window and navigating to the toolbox, the user will run the SSO tool. The data package includes sample data that needs to be added to the map either by opening the included MXD or adding the layers individually. Once the data is added and the tool is opened, a box will appear allowing for the parameters to be defined. For the purposes of publishing to ArcGIS server, any values can be added to meet the 'value required' parameter and run the tool.

Once the tool is run successfully, the geoprocessing results window will need to be opened and the current results from the tool will be displayed under current results. By right clicking on this result, the option to 'Share as a Geoprocessing Service' will be listed (Figure 12). Select this option and follow the wizard to publish to server. It is expected that the user has ArcGIS server experience to complete this part of the process.

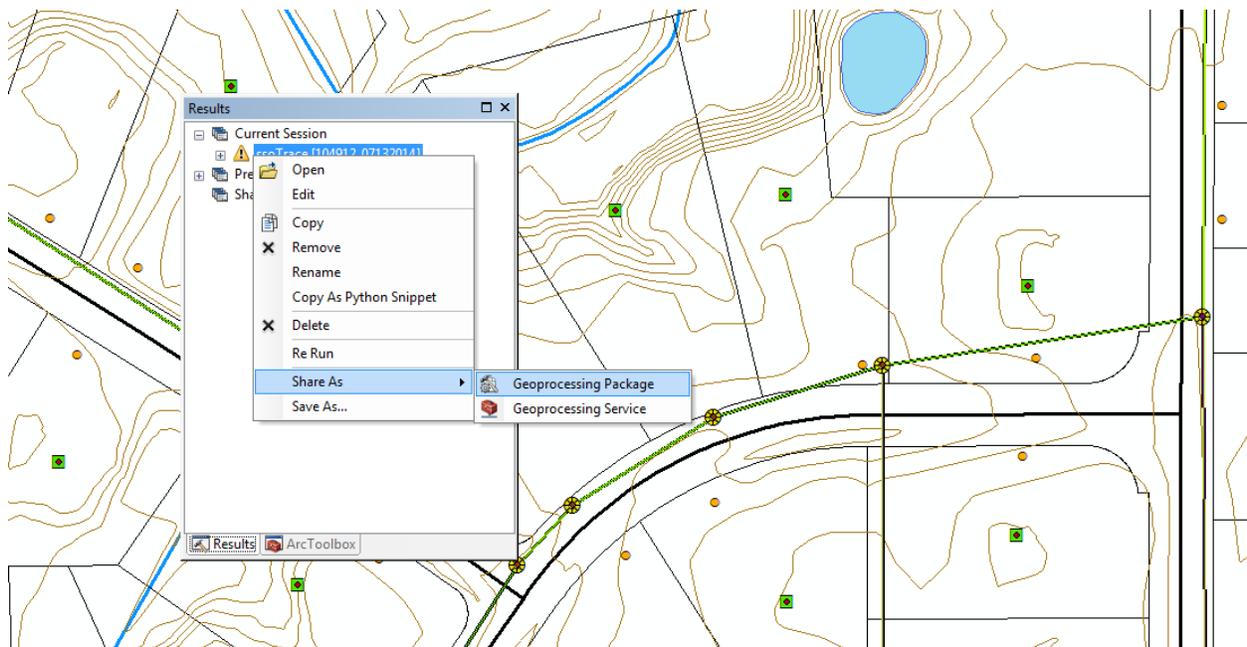


Figure 12: Screenshot of accessing the geoprocessing results window and selecting to share as a geoprocessing service.

With the tool now published to ArcGIS server as a geoprocessing service, the service can be inserted into an existing web mapping application. For the purposes of this project, flex viewer for ArcGIS was used to build the web mapping application. Using the application builder that comes with the flex viewer application, the geoprocessing service can be added to the web map as a widget. Figure 13 shows the editing interface of the widget once it is added to the application builder. Within this interface, symbols for the flags, barriers, and selection sets can be set for display on the map. Also default values can be set for any of the text parameters. Due to the fact that all of the fields are required, values had to be entered into the fields for the tool to run which were published as default values. It would be a good idea to go into each parameter and remove the default value. The user would then be required to enter the values and not potentially skip over a field.

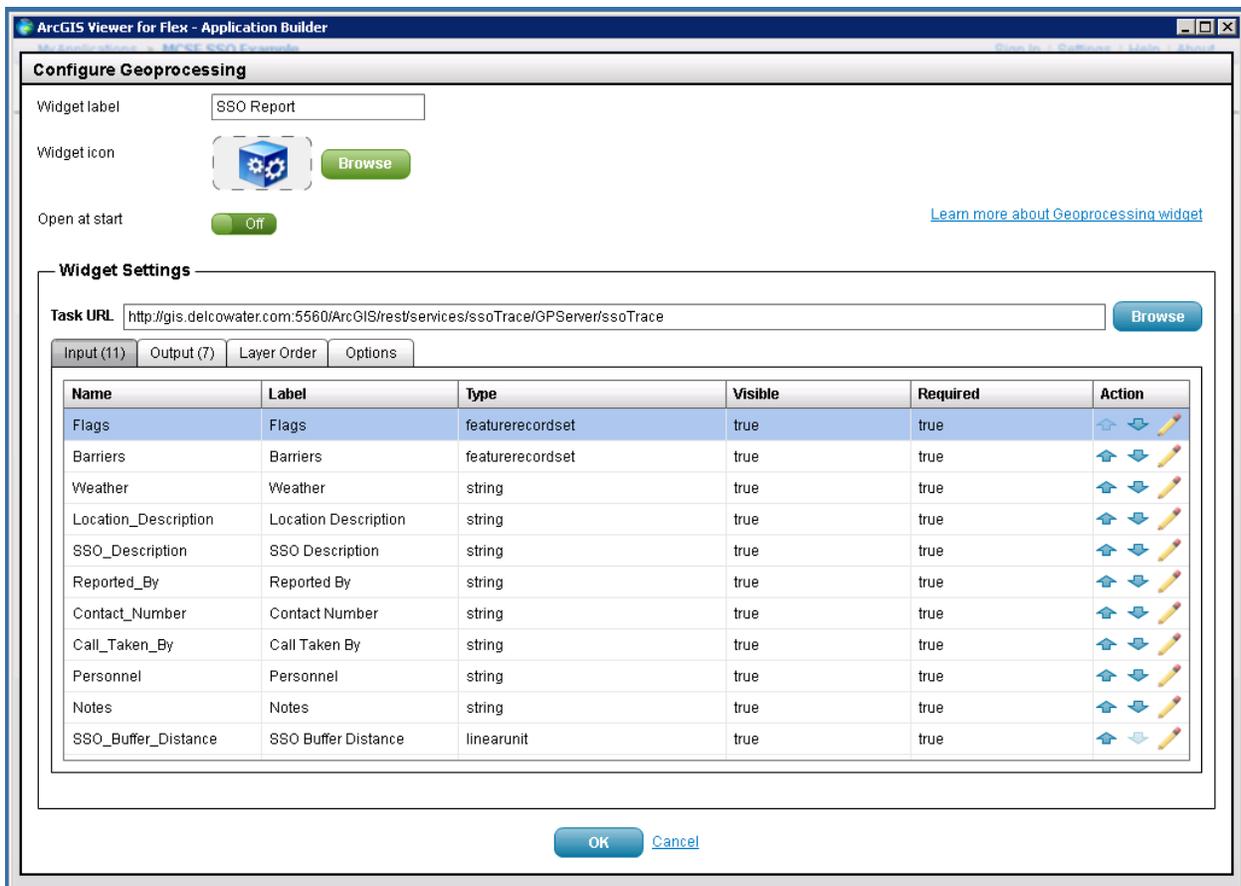


Figure 13: Screenshot of the edit screen for adding the geoprocessing service to the web map using the flex viewer application GUI.

Once the service is included in the web map and saved, the web map can be opened up and the widget should be available on the top menu bar. In Figure 14 the map is zoomed to the location of the sample SSO used throughout the report. The service is activated and the parameter box is opened up. Here the user can enter the values and adjust any default values as necessary. This interface is much more user friendly than what is available through ArcMap and should be easy to train users on.

Once the tool is run, a window will open allowing the user to select where to save the HTML file. Once saved, the user can use windows explorer to navigate to the file to view the information in the report. Due to the design of the HTML, the report can be printed out or printed to a PDF for sending to other necessary personnel.

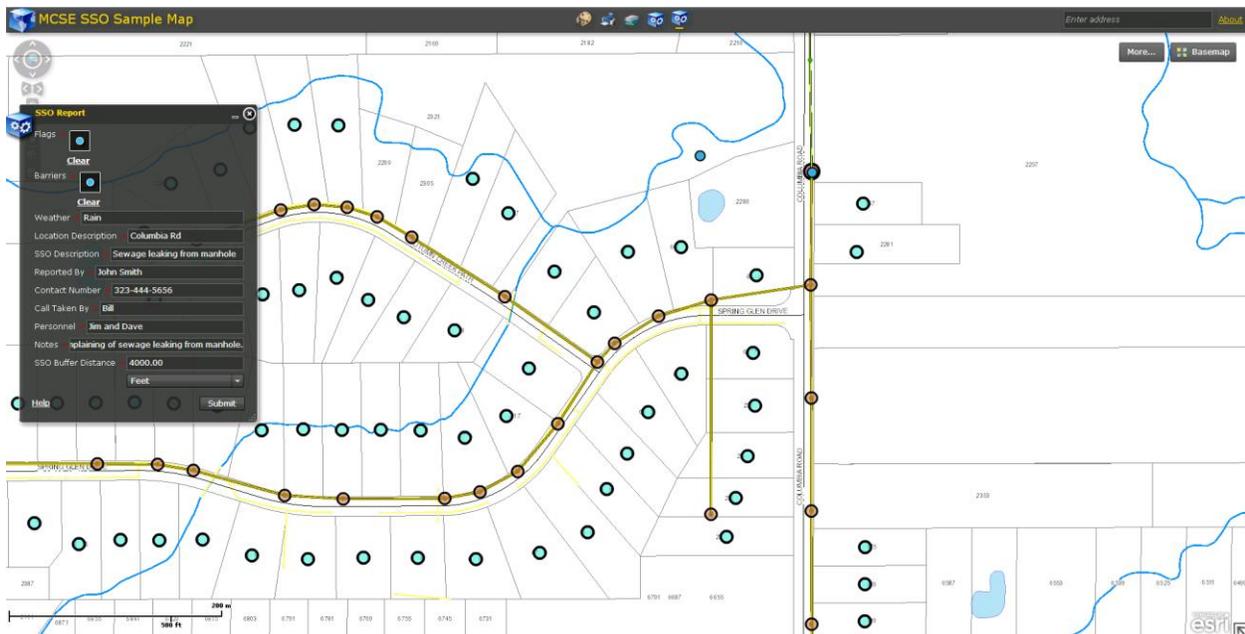
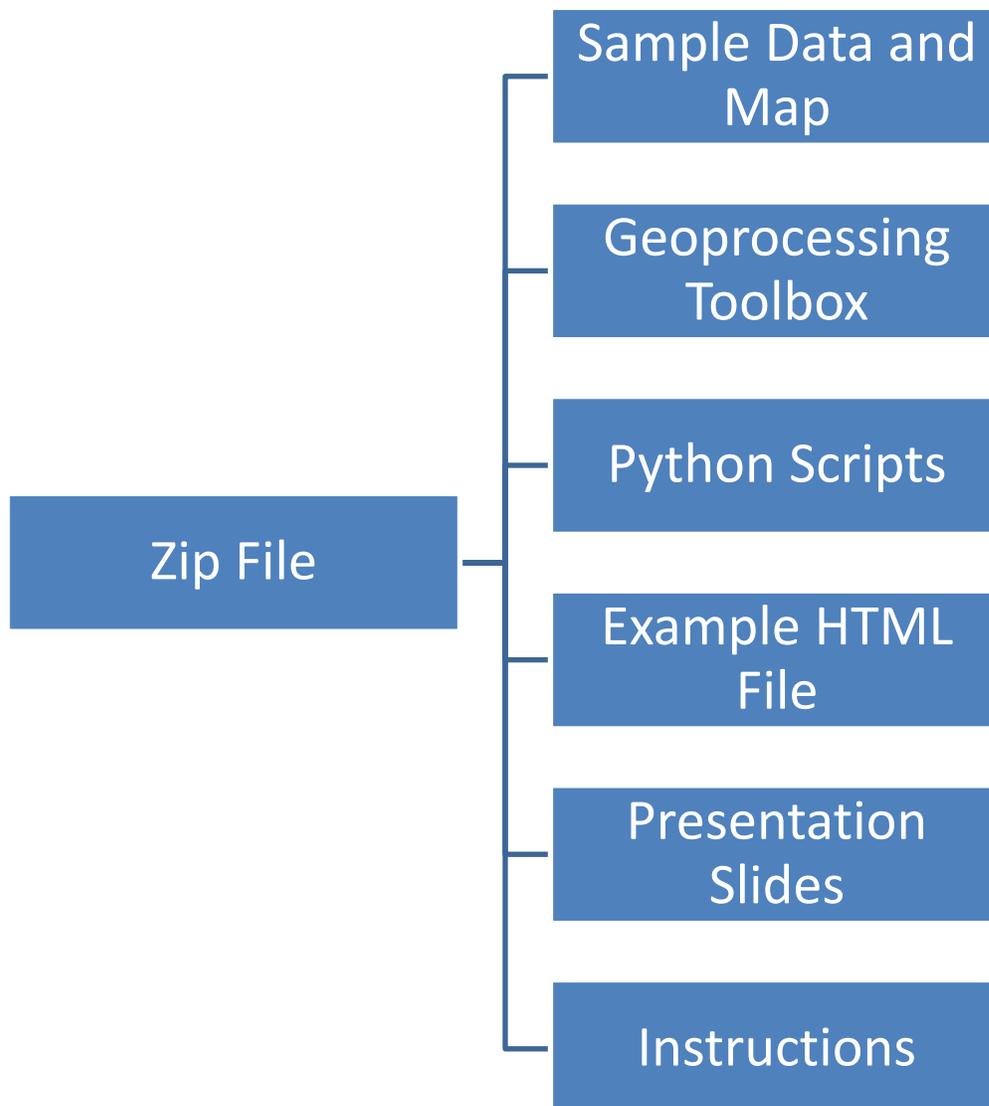


Figure 14: Screenshot of the geoprocessing service used as a widget in a flex viewer application for ArcGIS.

Packaged Data

Although this project was designed to be used for the MCSE sanitary sewer system, the design took into account that other organizations could customize the app for their own needs. This will require some customization but the framework should be sufficient in other applications. To provide the application, the necessary data is packaged together in a zip file for download. This package includes sample data and instructions to run the tool and see the results. Minimal customization is required once the package is downloaded and saved on a local machine or network. Additional information is also provided in the instructions for starting points in implementing different data.

To summarize the packaged data, a diagram of the corresponding contents is displayed below:



Due to the complexity of the application, experience with Python, ArcGIS desktop and HTML is necessary for successfully implementing it into a different dataset. Also it would be beneficial to have ArcGIS server and web mapping experience, but these are not required to set up the application.

The package has been published to ArcGIS Online as a 'Geoprocessing Sample' and is made available for download to a local computer or network. This will allow for much easier access to the application within ArcGIS Online keyword searches. The package is publicly available but the document is designed to be used 'as is'. Some support may be available within reason by using the contact information included with the file.

The following link will take you to the details page of the download page:

<http://www.arcgis.com/home/item.html?id=3e8a54fd43234d0ba1304aaf499c4c95>

Project Conclusions

Through the project several observations were noticed which had an impact on the project or led to additional thoughts on future enhancements and applications. The following observations are a few that really stood out through the project.

Model Builder, Python, or both?

When determining how to structure the application there was the question of whether to build everything in Python, Model Builder or to use both. Each approach has its limitations whether its functionality or ease of use. Seeing how the tool was going to be run as a geoprocessing service, it made sense to incorporate Model Builder into the layout and use a workflow of the processes. Model Builder has its limitations where Python allows for much more customization of geoprocessing tools and looping through data.

It was decided that the most effective design was to incorporate both and take advantage of the benefits of each. Other than the initial sewer tributary trace, Model Builder works more as a workflow design tool to bring different data together. The GUI allows for the developer to view all data and how the different scripts work with the necessary variables.

Python was used to run most of the geoprocessing tools, looping through information and reading attribute data within the GIS layers. The Pythonwin interface also allowed for scripting to take place to write data to an HTML file. This proved to be an easy and effective way to produce the end report.

Different ways to manipulate data

Building off of the previous observation, many design ideas resulted in many different ways to manipulate the data and get the end product. Using the Python and Model Builder example, either application could have been used to produce a valid report in response to an SSO. There is not a right and wrong way to go about the design but depending on the path taken more work or code may be required to get the end result. If having to do it all over again I may have chosen to do certain things differently but in the end I feel the design I decided on is the most practical and customizable to the project.

When looking at the end product of the HTML report, all of the data is listed in short, clean text blocks. From the end-users perspective this needs to look simple and to the point as the method of accessing the data is not a concern as they just need it available and be accurate. In the background many steps are happening to extract the data. For example, to get to the total ERU's for the location of the SSO several processes must happen first before the ERU calculation can happen. Within those processes, there are different tools and methods available that would produce the end result. A secondary goal of this project was to choose an effective workflow that could be understood for future customization.

What else is possible with the same concept?

This project was designed to assist in the response plan for a major SSO. In particular it was designed with the MCSE sewer system in mind for how their organization would need to respond. The first realization is that many other organizations managing sewer systems could integrate this into their operations or use some of the concepts for their own plan. This is why a data package was put together for distribution to other organizations. The accompanying data should provide ideas that could be incorporated in a variety of applications.

Additionally the concepts of this app can be beneficial to other organizations. An obvious organization would be emergency management. Seeing how the SSO is an emergency, many of the same concepts could be used for other emergencies such as some other type of hazardous spill.

Currently my work is in the water industry and I have already looked at ways that this application could be used in some water application. For instance could this be used to assist in the response to a major water leak? Or could some of the concepts be used in the planning of a major water project?

Where to stop?

In working through the project, it became evident that in some cases the cutoff for information provided needed to be set at a certain point. For much of the available data, before deciding what to include in the report, it first needed to be determined if it was necessary and useful in the SSO plan. The idea of the project was to provide a clear, concise report that would not be larger than one page. Filtering through too much information could have negative effects in a emergency response event.

Also much information could be displayed in many different ways. Take the total ERU's as an example of this. For the report purposes, I just listed the total ERU's for the location of the SSO. To go further you could display estimated gallons per minute or per day. A question that comes up next is whether to show this in actual gallons or thousands of gallons. The appropriate way to display the data may be different based on the organizations needs but the report should be customized to meet the user's needs.

A benefit of this application is that additional fields can be added and the processes can be plugged into the scripts. Items could also be removed if they are deemed unnecessary to the report. Over time if an organization implements this app, much customization will take place as needs change based on real world situations. This report may take on a completely different look after more user feedback.

Acknowledgements

There have been several individuals that have been helpful through the different phases of this project. I greatly appreciate their cooperation and time in providing advice and information.

First of all I have to thank the Medina County Sanitary Engineers Office for their cooperation through this project. Within Medina County I need to especially thank Amy Lyon Galvin for providing a need assessment for the tool and answering questions along the way. As I left the Sanitary Engineers office early on in this project to further my career, they were still cooperative in assisting in providing information and feedback.

Second of all I need to thank my current employer, Del-Co Water Co, as they have been very cooperative with the completion of the capstone project. They were also very helpful in assisting in costs for traveling the Pennsylvania State GIS Conference where the project was presented.

I would also like to thank the Penn State faculty and staff involved in the MGIS program for their assistance through the entire program and especially with inquiries with the capstone project. In particular I need to thank Beth King and Doug Miller for their assistance through the project.

Finally I would like to thank my advisor, Dr. Jan Oliver Wallgrün. I am not able to thank him enough for his support and advice. It was a pleasure working with him through the project and the insight provided was very beneficial.

References

- [1] EPA (2001). Managing Sewer Overflows. Retrieved on October 22, 2013 from http://www.epa.gov/safewater/sourcewater/pubs/fs_swpp_ssocso.pdf
- [2] EPA (2001). Why Control Sewer Overflows. Retrieved on October 29, 2013 from http://www.epa.gov/npdes/pubs/sso_casestudy_control.pdf
- [3] APWA (2010). SSO Response Plan. Retrieved on October 22, 2013 from <http://www2.apwa.net/documents/resourcecenter/Final%20Core%20Attributes%20July%202010.pdf>
- [4] Delmar, CA (2010). Sanitary Sewer Management Plan: City of Delmar, CA. Retrieved on October 26, 2013 from http://www.delmar.ca.us/Government/dept/Documents/SSMP_Final2010.pdf
- [5] EPA (ND). CMOM Case Study: Greenwood, SC. Retrieved on October 26, 2013 from http://www.epa.gov/npdes/pubs/sso_casestudy_greenwood.pdf
- [6] Dekalb County, Georgia (ND). Sanitary Sewer Overflow Contingency and Emergency Response Program. Retrieved on October 22, 2013 from http://www.dekalbwatershed.com/PDF/prog_updates_sanitary_sewer_overflow_contingency_&_emergency_response_plan.pdf
- [7] American Society of Civil Engineers (Under Cooperative Agreement with EPA, Project No. CP-828955-01-0) (2004). Solutions for Sanitary Sewer Overflows. Retrieved on November 6, 2013 from <http://epa.ohio.gov/portals/35/permits/sso%20solutions%20asce%20epa%20guidance%202004.pdf>
- [8] American Society of Civil Engineers (Under Cooperative Agreement with EPA) (2000) Protocols for Identifying Sanitary Sewer Overflows (SSOs). Retrieved on November 6, 2013 from <http://scap1.org/Collection%20Reference%20Library/Folder%20contains%20Misc%20EPA%20files%20for%20Collection%20Systems/Protocols%20Identifying%20SSO.pdf>
- [9] Sier, D., & Lansey, K. (2005). Monitoring sewage networks for sanitary sewer overflows. Civil Engineering & Environmental Systems, 22(2), 123-132. Retrieved on November 7, 2013 from <http://ezaccess.libraries.psu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=17473374&site=ehost-live>