

# AUTOMATING GEOPROCESSING TASKS WITH MODEL BUILDER AND PYTHON

SMALL AREA HEALTH INSURANCE ESTIMATES PROJECT  
(SAHIE) 2014

**MEHRSHAD NOURANI**  
**SENIOR GIS ANALYST/DEVELOPER**  
[merhrshad.nourani@census.gov](mailto:merhrshad.nourani@census.gov)

**U.S. CENSUS BUREAU, 4600 SILVER HILL ROAD,  
WASHINGTON DC, 20233**

**THE COLEMAN GROUP, INC.  
CORPORATE HEADQUARTERS  
10210 GREENBELT ROAD, SUITE 720  
GREENBELT, MARYLAND 20706**

**ATLANTA OFFICE  
1170 PEACHTREE STREET, SUITE 1200  
ATLANTA, GEORGIA 30309**



ESRI International User Conference  
July 14-18 | San Diego

# Small Area Health Insurance Estimates (SAHIE)

Automate the process by which the mapping services are consumed by SAHIE interactive mapping tool (API)



# Who we are and what we do

## Census Bureau

- Demographic Internet Staff (DIS)
- Dissemination of Public Products

## The Coleman Group, Inc.

- IT/Cyber Security Service provider to Federal Agencies

# Background

## Our Goal

To automate GIS workflow in order to process data, create mapping documents, and publish mapping services for SAHIE interactive web mapping tool (API)

## Our (geo)Process

The process of creating map services had been in place for a few years. It involves data edits and transformation, and several geoprocessing tasks, including creating map services. All tasks are currently performed manually, and last nearly 2 days to complete. The lengthy and repetitive nature of geoprocessing tasks compelled us to automate our workflow so that we can increase process efficiency and reduce human errors.

## Our Approach

1. Use ModelBuilder to test and document our workflow; and
2. Use Python scripting to complete automating the process.

# Work Environment

1. SAS native data format
2. Excel(tab delimited .txt) – MS 2010
3. ArcGIS for desktop 10.1
4. ArcGIS for server 10.1
5. Python 2.7.2
6. Citrix XenApps
7. MS Windows 7

# Tools

1. ArcGIS 10.1, ModelBuilder
2. Python scripting

# Why ModelBuilder, or Python script?

## ModelBuilder

1. Process simplification – No guess work!
2. Visualizing workflow
3. Standardizing workflow
4. Increase efficiency and reduce human error
5. Proof of concept/Test hypotheses
6. Maintaining institutional memory
7. Interacting with our clients – showing ModelBuilder instead of codes
8. Starting a script with a model – need editing
9. Documentation: the ModelBuilder can be used to document and publish our process

# Why ModelBuilder, or Python script?

## Python Script

- Scripting is more efficient
  1. No need to run ArcGIS: use stand-alone script
  2. Simple edits
  3. Easy QA/QC
  4. Do more with scripting
  5. Support community



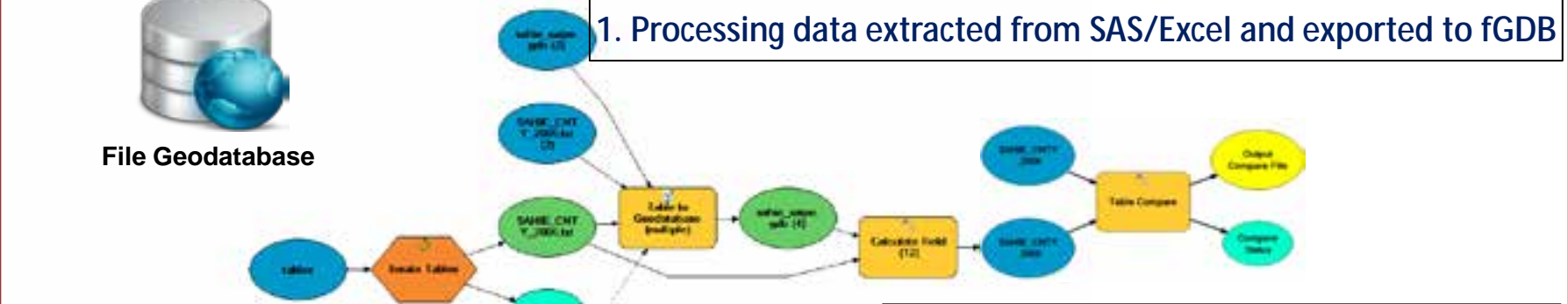
# Our Workflow

# Workflow Model



File Geodatabase

1. Processing data extracted from SAS/Excel and exported to fGDB



2. Verifying and change detection – QA/QC



3. Geoprocessing feature classes

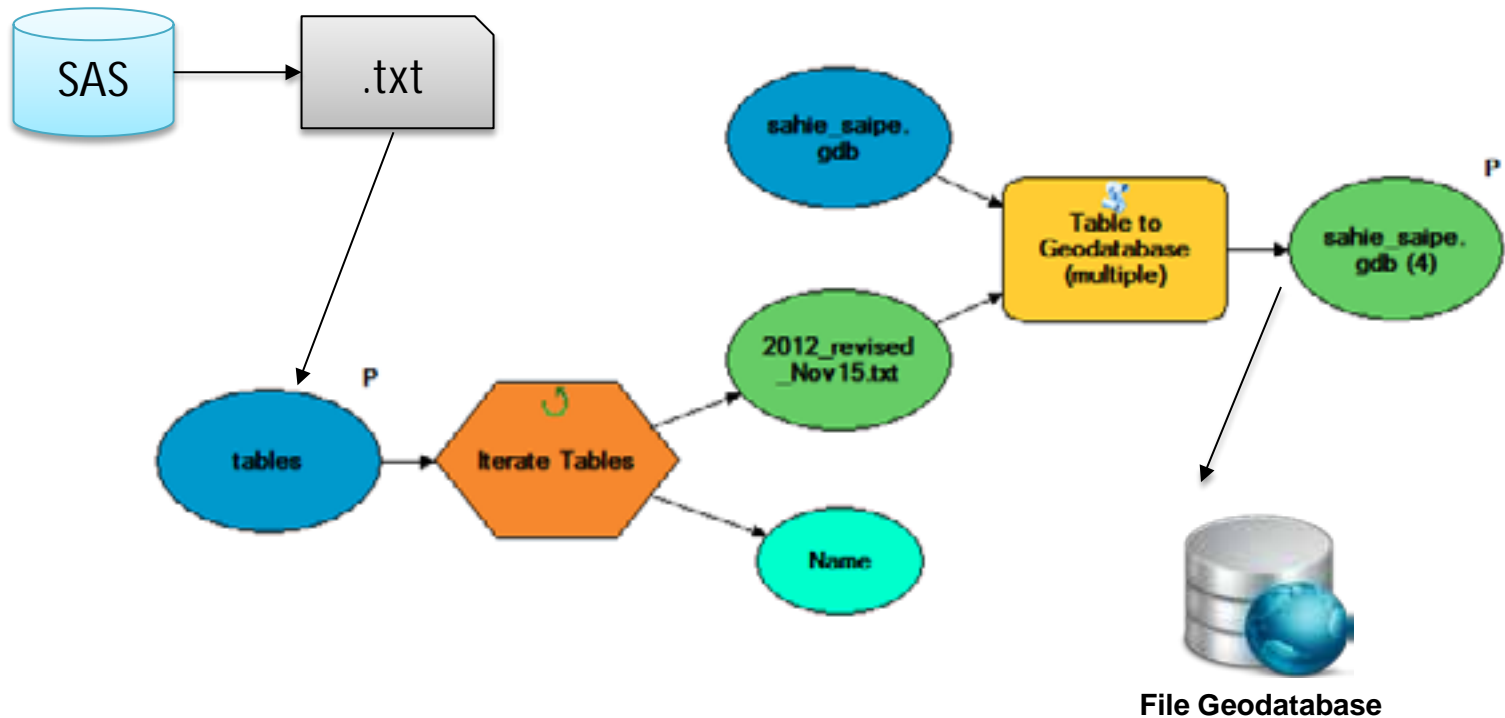


4. Creating, analyzing, and publishing 7 map services to ArcGIS Server à Flex API



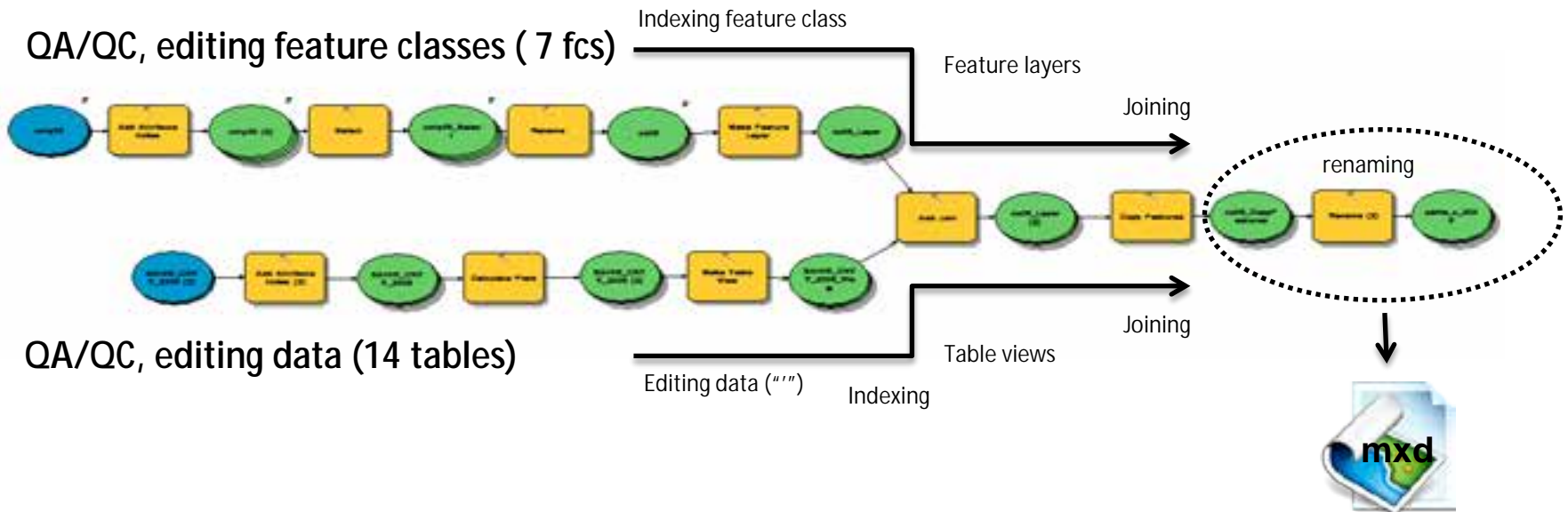
# 1. Extracting, Transforming, Loading (ETL)

1. Importing tabular data (SAS → .txt → fGDB)
2. QA/QC: verifying number and fields, rows, cells; text vs. integer;
3. Exporting to fGDB



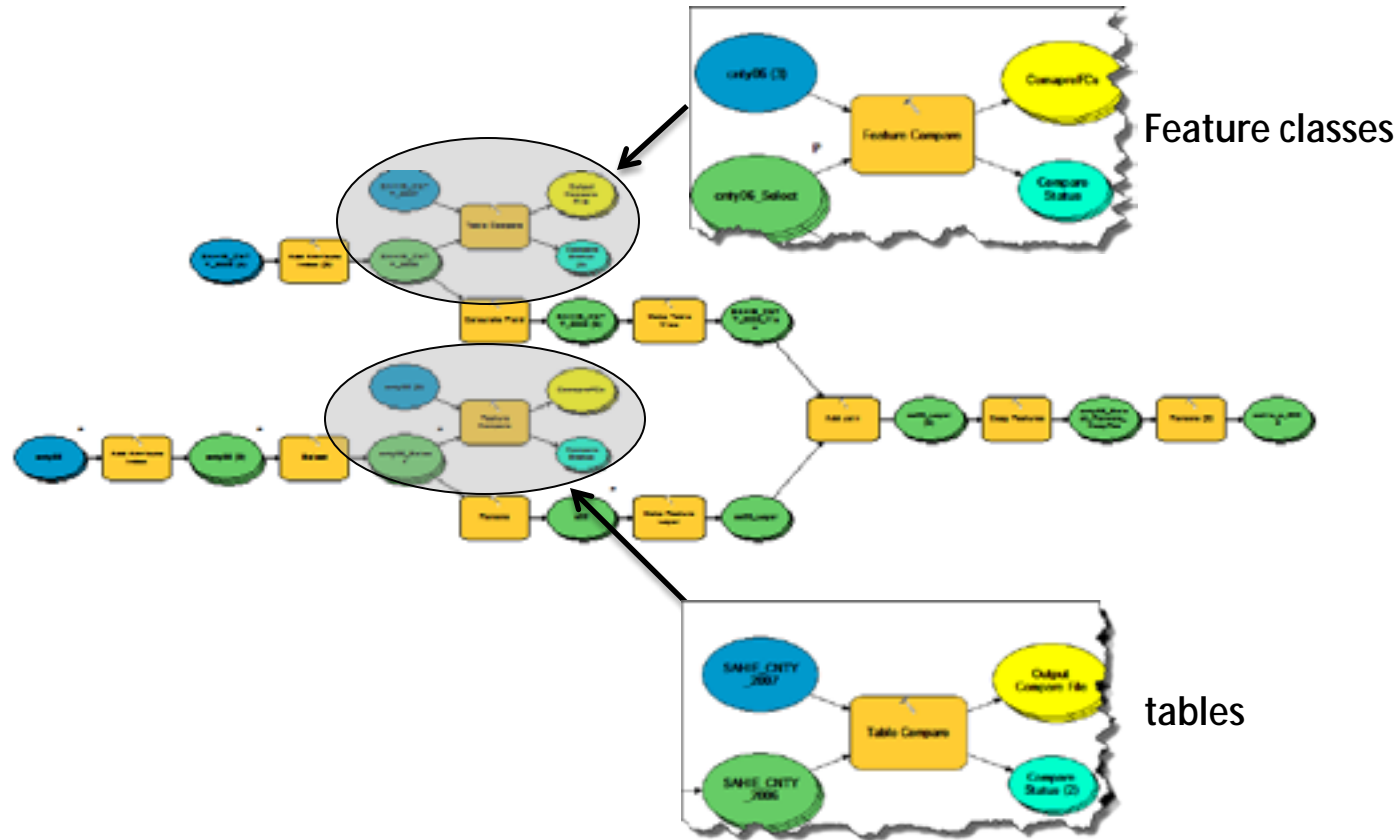
# 2. Editing data and Feature Classes

1. Reformatting fields in tables : SASà .txtà fGDB (text to integer!!)=> add "quotation marks"
2. Indexing tables and feature classes for faster processing
3. Creating feature layers and table views
4. AddJoinà create new fcs
5. Rename fcs = must match Flex API config
6. Creating mxds



# 3. Verifying and change-detection

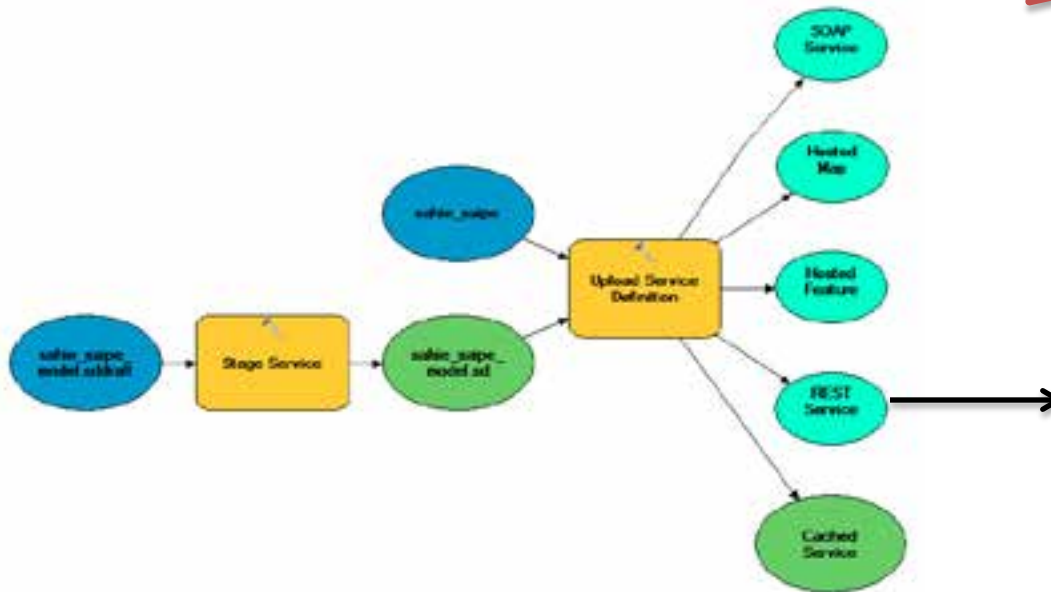
1. Comparing feature classes by year and determining changes in geography
2. Comparing annual data tables and determining changes in number of fields, etc.



# 4. Creating and publishing map services

Publishing mxds as map services for consumption by Flex API through REST services

**Publishing mxds as map services (SD):  
all layers must be in the order the Flex API  
reads and accesses data**



ArcGIS REST Services Directory  
Home > Services > demo > safrs\_safrs (MapServer)  
demo/safrs\_safrs (MapServer)  
View By: ArcGIS JavaScript ArcGIS LayerView Desktop Earth ArcSWF Explorer  
View Footprint By: ArcGIS.com Map  
Service Description  
Map Name: safrs\_safrs  
Legend  
All Layers and Tables  
Dynamic Legend  
Dynamic All Layers  
Layers:  
• State\_06 (0)  
• State\_07 (1)  
• State\_08 (2)  
• State\_09 (3)  
• State\_10 (4)  
• State\_11 (5)  
• County\_06 (6)  
• County\_07 (7)  
• County\_08 (8)  
• County\_09 (9)  
• County\_10 (10)  
• County\_11 (11)  
• ElementaryUnifedSchoolDistrict\_09 (12)  
• ElementaryUnifedSchoolDistrict\_10 (13)  
• ElementaryUnifedSchoolDistrict\_11 (14)  
• SecondaryUnifedSchoolDistrict\_09 (15)  
• SecondaryUnifedSchoolDistrict\_10 (16)  
• SecondaryUnifedSchoolDistrict\_11 (17)  
Tables:  
• safrs\_s\_09 (18)  
• safrs\_s\_10 (19)  
• safrs\_s\_11 (20)  
• safrs\_c\_09 (21)  
• safrs\_c\_10 (22)  
• safrs\_c\_11 (23)  
• safrs\_safd\_09 (24)  
• safrs\_safd\_10 (25)  
• safrs\_safd\_11 (26)  
• safrs\_safd\_09 (27)  
• safrs\_safd\_10 (28)  
• safrs\_safd\_11 (29)  
• safrs\_s\_09 (30)  
• safrs\_s\_10 (31)  
• safrs\_s\_11 (32)  
• safrs\_s\_09 (33)  
• safrs\_s\_10 (34)  
• safrs\_s\_11 (35)  
• safrs\_c\_06 (36)

# The First Step



# From ModelBuilder to Python

1. Export your model from ModelBuilder to Python script
2. Review Python script in text editor
3. Can you run this script?



```
ahieModelBuilder.py - C:\Workspace\ahie_modelbuilder_script\ahie_modelbuilder\ahieModelBuilder.py
File Edit Format Run Options Windows Help
-----
# ahieModel.py
# Created on: 2014-04-11 09:12:10.00000
# (generated by ArcGIS/ModelBuilder)
# Usage: ahieModel
# Description:
-----

# Import arcpy module
import arcpy

# Load required toolboxes
arcpy.ImportToolbox("H:\WorkSpace\ahie\ahie\ahie.gdb\Tool")

# Design arguments
pt_2_ = arcpy.GetParameterAsText(0)
if pt_2_ == "?" or not pt_2_:
    pt_2_ = "H:\WorkSpace\ahie\ahie.gdb\Tool" # provide a default value if unspecified

pt_3_ = arcpy.GetParameterAsText(1)
if pt_3_ == "?" or not pt_3_:
    pt_3_ = "H:\WorkSpace\ahie\ahie.gdb\Tool" # provide a default value if unspecified

pt_5_ = arcpy.GetParameterAsText(2)
if pt_5_ == "?" or not pt_5_:
    pt_5_ = "H:\Memory\1\42111F7-1220-430C-8064-E10A430081" # provide a default value if unspecified

pt_4_ = arcpy.GetParameterAsText(3)
if pt_4_ == "?" or not pt_4_:
    pt_4_ = "H:\Memory\1\42111F7-1220-430C-8064-E10A430081" # provide a default value if unspecified

pt_6_ = arcpy.GetParameterAsText(4)
if pt_6_ == "?" or not pt_6_:
    pt_6_ = "H:\Memory\1\42111F7-1220-430C-8064-E10A430081" # provide a default value if unspecified

pt_7_ = arcpy.GetParameterAsText(5)
if pt_7_ == "?" or not pt_7_:
    pt_7_ = "H:\Memory\1\42111F7-1220-430C-8064-E10A430081" # provide a default value if unspecified
```



# Review your script in text editor...

Before being able to run your script, you may have to make some modifications to the script that is generated through ModelBuilder. For example:

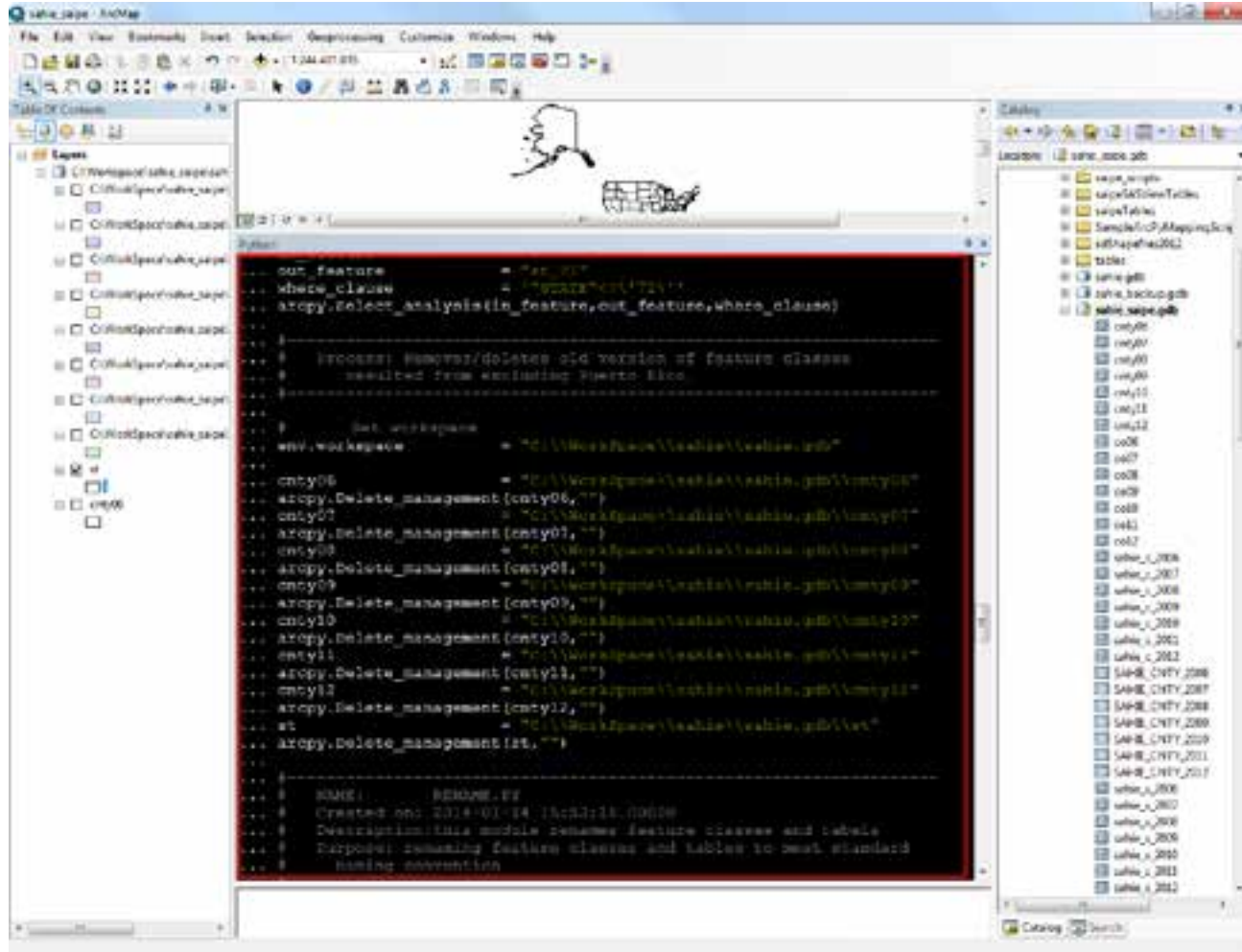
1. If you are processing a series of fcs using several tools, the ModelBuilder repeats the same script individually for all the fcs – 450 lines of code were generated by the ModelBuilder. (runs)
2. “iterators” do not work as Python script (from ModelBuilder) (modify)
3. If you have imported tools from Toolbox, the script may have difficulty locating and running it! (modify)
4. Change the gp name to default toolbox name
5. Check the paths to your workspace (gdb, etc.)



```
13 .....  
14  
15 from arcpy import env  
16 env.workspace = "C:\\WorkSpace\\leahie\\leahie.gdb"  
17  
18 .....  
19 # Process: rbl.py extracts, transforms and loads tab delimited text  
20 # tables from XRM to geodatabase tables.  
21 .....  
22 # Extract Script arguments  
23 SARIS_CNTY_2006_txt = arcpy.GetParameterAsText(0)  
24 if SARIS_CNTY_2006_txt == "" or not SARIS_CNTY_2006_txt:  
25     SARIS_CNTY_2006_txt = "C:\\WorkSpace\\leahie\\tables\\leahie\\2006-2010_saris\\SARIS_CNTY_2006.txt"  
26     # provide a default value if unspecified  
27  
28 SARIS_CNTY_2007_txt = arcpy.GetParameterAsText(1)  
29 if SARIS_CNTY_2007_txt == "" or not SARIS_CNTY_2007_txt:  
30     SARIS_CNTY_2007_txt = "C:\\WorkSpace\\leahie\\tables\\leahie\\2007-2010_saris\\SARIS_CNTY_2007.txt"  
31     # provide a default value if unspecified  
32  
33 SARIS_CNTY_2008_txt = arcpy.GetParameterAsText(2)  
34 if SARIS_CNTY_2008_txt == "" or not SARIS_CNTY_2008_txt:  
35     SARIS_CNTY_2008_txt = "C:\\WorkSpace\\leahie\\tables\\leahie\\2008-2010_saris\\SARIS_CNTY_2008.txt"  
36     # provide a default value if unspecified  
37  
38 SARIS_CNTY_2009_txt = arcpy.GetParameterAsText(3)  
39 if SARIS_CNTY_2009_txt == "" or not SARIS_CNTY_2009_txt:  
40     SARIS_CNTY_2009_txt = "C:\\WorkSpace\\leahie\\tables\\leahie\\2009-2010_saris\\SARIS_CNTY_2009.txt"  
41     # provide a default value if unspecified  
42  
43 SARIS_CNTY_2010_txt = arcpy.GetParameterAsText(4)  
44 if SARIS_CNTY_2010_txt == "" or not SARIS_CNTY_2010_txt:  
45     SARIS_CNTY_2010_txt = "C:\\WorkSpace\\leahie\\tables\\leahie\\2010_saris\\SARIS_CNTY_2010.txt"  
46     # provide a default value if unspecified  
47  
48 SARIS_CNTY_2011_txt = arcpy.GetParameterAsText(5)
```

The screenshot shows a Python script with several lines of code. Red circles are drawn around the following lines: 1, 2, 3, 4, 5, and ...14. These circles correspond to the list items on the left side of the slide.

# ...run your script in ArcGIS/Python



# Do you know Python?

## Rewrite/edit ModelBuilder script

Script imported from ModelBuilder ~ 43 lines

Edited in Python ~ 10 lines

```
from arcpy import env
env.workspace = "C:\\workspace\\tahle\\tahle.gdb"
SARIE_CNTY_2004_txt = arcpy.GetParameterAsText(0)
if SARIE_CNTY_2004_txt == "4" or not SARIE_CNTY_2004_txt:
    SARIE_CNTY_2004_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_CN
SARIE_CNTY_2007_txt = arcpy.GetParameterAsText(1)
if SARIE_CNTY_2007_txt == "4" or not SARIE_CNTY_2007_txt:
    SARIE_CNTY_2007_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_CN
SARIE_CNTY_2009_txt = arcpy.GetParameterAsText(2)
if SARIE_CNTY_2009_txt == "4" or not SARIE_CNTY_2009_txt:
    SARIE_CNTY_2009_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_CN
SARIE_CNTY_2008_txt = arcpy.GetParameterAsText(3)
if SARIE_CNTY_2008_txt == "4" or not SARIE_CNTY_2008_txt:
    SARIE_CNTY_2008_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_CN
SARIE_CNTY_2010_txt = arcpy.GetParameterAsText(4)
if SARIE_CNTY_2010_txt == "4" or not SARIE_CNTY_2010_txt:
    SARIE_CNTY_2010_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_CN
SARIE_CNTY_2011_txt = arcpy.GetParameterAsText(5)
if SARIE_CNTY_2011_txt == "4" or not SARIE_CNTY_2011_txt:
    SARIE_CNTY_2011_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_CN
SARIE_CNTY_2012_txt = arcpy.GetParameterAsText(6)
if SARIE_CNTY_2012_txt == "4" or not SARIE_CNTY_2012_txt:
    SARIE_CNTY_2012_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_CN
SARIE_ST_2004_txt = arcpy.GetParameterAsText(7)
if SARIE_ST_2004_txt == "4" or not SARIE_ST_2004_txt:
    SARIE_ST_2004_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_ST
SARIE_ST_2007_txt = arcpy.GetParameterAsText(8)
if SARIE_ST_2007_txt == "4" or not SARIE_ST_2007_txt:
    SARIE_ST_2007_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_ST
SARIE_ST_2008_txt = arcpy.GetParameterAsText(9)
if SARIE_ST_2008_txt == "4" or not SARIE_ST_2008_txt:
    SARIE_ST_2008_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_ST
SARIE_ST_2009_txt = arcpy.GetParameterAsText(10)
if SARIE_ST_2009_txt == "4" or not SARIE_ST_2009_txt:
    SARIE_ST_2009_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_ST
SARIE_ST_2010_txt = arcpy.GetParameterAsText(11)
if SARIE_ST_2010_txt == "4" or not SARIE_ST_2010_txt:
    SARIE_ST_2010_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_ST
SARIE_ST_2011_txt = arcpy.GetParameterAsText(12)
if SARIE_ST_2011_txt == "4" or not SARIE_ST_2011_txt:
    SARIE_ST_2011_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_ST
SARIE_ST_2012_txt = arcpy.GetParameterAsText(13)
if SARIE_ST_2012_txt == "4" or not SARIE_ST_2012_txt:
    SARIE_ST_2012_txt = "C:\\workspace\\tahle\\tahle\\tahle_2004-2012_siar\\SARIE_ST
```

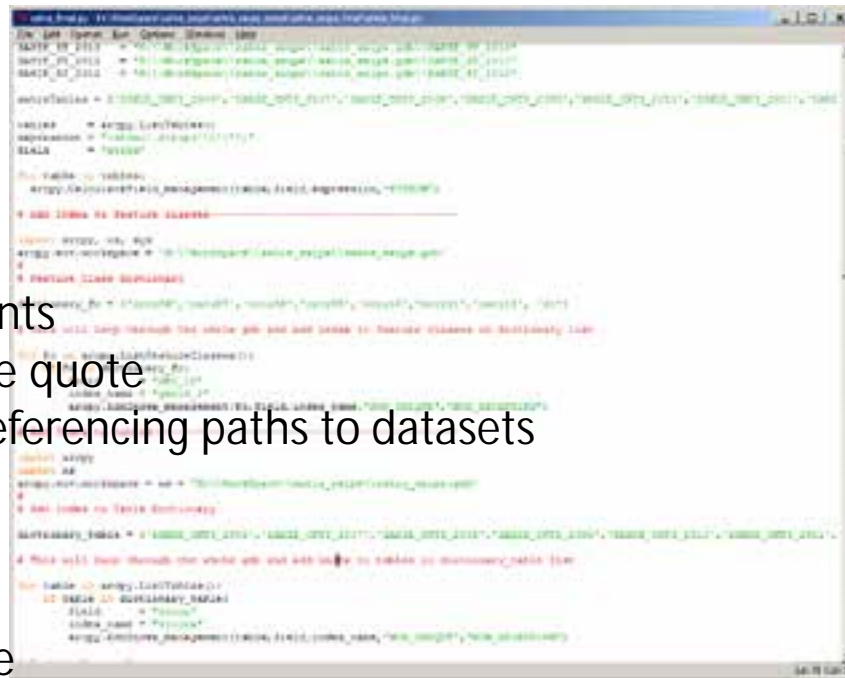


```
import arcpy
import os
from arcpy import env
env.workspace = "C:\\workspace\\tahle\\tahle.gdb"
tables = arcpy.ListTables()
outlocation = "C:\\workspace\\tahle\\tahle\\tahle.gdb"
try:
    print "Importing tables to gdb: " + outlocation
    arcpy.TableToDatabaseConversion(tables, outlocation)
except:
    print arcpy.GetMessage()
```

Example: importing multiple data tables

# Test your Python script

1. Verify your tasks were completed
2. Use geoprocessing results (only if run from ArcGIS environment)
3. Look for empty fcs
4. Task may be completed but no results generated
5. Look for errors
  - a. Python error
    - I. Typos
    - II. Python is case sensitive
    - III. Indentation
    - IV. Colons for compound statements
    - V. Mixing single quote and double quote
    - VI. Using single backslash when referencing paths to datasets
  - b. ArcGIS error
    - I. fc/table does not exist
    - II. Incorrect paths
    - III. Previous task did not complete



```
def table_to_dataframe(table):
    """Convert a table to a pandas DataFrame"""
    # Get the table name and path
    table_name = table.get('name')
    table_path = table.get('path')

    # Check if the table exists
    if not os.path.exists(table_path):
        raise ValueError("Table does not exist")

    # Read the table into a pandas DataFrame
    df = pd.read_csv(table_path)

    # Return the DataFrame
    return df
```





# ModelBuilder tips that worked for me

1. Set your work environment from geoprocessing menu (ArcMap/Catalog)
2. Use relative path
3. Start with smaller models
4. Integrate smaller models to create, or expand to a larger model
5. Use Geoprocessing menu to review and monitor your model
6. Become active in the community and ask questions in forums

# Python tips that worked for me

1. Use ModelBuilder to create your first Python script
2. Rewrite Python by using Toolbox examples in tool's Help:
  - a. Iterating over a list or a string
  - b. for statement
  - c. List
  - d. Dictionary à

```
>>> import arcpy
import os
arcpy.env.workspace = ws = 'C:\\WorkSpace\\sahie_saipc\\sahie_saipc.gdb'
# Add Index to Table Dictionary
dictionary_table = ('SAHIE_CNTY_2006','SAHIE_CNTY_2007','SAHIE_CNTY_2008','
SAHIE_CNTY_2009','SAHIE_CNTY_2010','SAHIE_CNTY_2011','SAHIE_CNTY_2012','SAH
IE_ST_2006','SAHIE_ST_2007','SAHIE_ST_2008','SAHIE_ST_2009','SAHIE_ST_2010'
,'SAHIE_ST_2011','SAHIE_ST_2012')
for table in arcpy.ListTables():
    if table in dictionary_table:
        field      = "stcou"
        index_name = "stcou65"
        arcpy.AddIndex_management(table,field,index_name,"NON_UNIQUE","NON_
ASCENDING")
```

3. Use online Python resources: ArcGIS Resources, GIS StackExchange... enhance your script
4. Join user communities/forums: ArcGIS Resource, GIS StackExchange, Python...

# Lessons learned

## ModelBuilder and Python

1. ModelBuilder = capable – limited efficiency
2. ModelBuilder + Python = capable, efficient, and powerful
3. Best resource: your system tool box = most likely, there is already a script tool for your need
4. Modify existing tools to create new tools for your needs
5. Build and integrate smaller models to create a larger model
  - a. If computing resources are limited, you can run smaller models to complete your tasks
6. Convert your model to an interactive Python tool for future use



# Work environment challenges

**Limitations of building a complex model in Desktop Virtual Environment** - Using a network location to save and process = slow processing time

1. System latency (hours vs minutes)
2. Writing to a common area folder = conflict with other users
3. Desktop alternative

# ArcGIS/Python Learning Resources

1. Learn Python: abundantly free online resources
2. ArcGIS Tools Help examples and modify for your needs
3. ArcGIS Resources: videos, forums, blogs
4. Python programming YouTube videos
5. Python.org

# Questions

MEHRSHAD NOURANI  
SENIOR GIS ANALYST/DEVELOPER  
THE COLEMAN GROUP, INC.  
U.S. CENSUS BUREAU, WASHINGTON DC  
[merhrshad.nourani@census.gov](mailto:merhrshad.nourani@census.gov)