Geospatial Workflows and Trust: a Use Case for Provenance

Rachel Frances Linck
University of Arkansas, Fayetteville

Geospatial Workflows and Trust: a Use Case for Provenance

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Geography

by

Rachel Frances Linck
University of Central Arkansas
Bachelor of Arts in International Relations, 2011
University of Arkansas
Bachelor of Arts in Geography, 2013

December 2015
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

Dr. Jason Tullis
Thesis Director

Dr. Jackson Cothren
Committee Member

Dr. Xuan Shi
Committee Member

Abstract

At first glance *the Astronomer* by Vermeer, Tutankhamun’s burial mask, and a geospatial workflow may appear to have nothing in common. However, a commonality exists; each of these items can have a record of provenance detailing their history. Provenance is a record that shows who did what to an object, where this happened, and how and why these actions took place. In relation to the geospatial domain, provenance can be used to track and analyze the changes data has undergone in a workflow, and can facilitate scientific reproducibility. Collecting provenance from geospatial workflows and finding effective ways to use this provenance is an important application. When using geospatial data in a workflow it is important to determine if the data and workflow used are trustworthy. This study examines whether provenance can be collected from a geospatial workflow. Each workflow examined is a use case for a specific type of geospatial problem. In addition to this, the collected provenance is then used to determine workflow trust and content trust for each of the workflows examined in this study. The results of this study determined that provenance can be collected from a geospatial workflow in such a way as to be of use to additional applications, such as provenance interchange. From this collected provenance, content trust and workflow trust can be estimated. The simple workflow had a content trust value of .83 (trustworthy) and a workflow trust value of .44 (untrustworthy). Two additional workflows were examined for content trust and workflow trust. The methods used to calculate content trust and workflow trust could also be expanded to other types of geospatial data and workflows. Future research could include complete automation of the provenance collection and trust calculations, as well as examining additional techniques for deciding trust in relation to workflows.

Acknowledgments

I would like to take the time to thank everyone that has helped me throughout the course of this thesis. I could not have completed it without the help of you all!

I would like to thank my advisor, Dr. Tullis, for introducing me to this topic. All of his help and guidance with developing my topic, as well as his patience and encouragement on completing this manuscript have been invaluable and greatly appreciated. Thanks to Dr. Shi for teaching me about programming and for all the patience he’s shown while doing so. Without his classes, I would not have had the skills to complete this thesis. Thanks to Dr. Cothren for taking the time to be on my thesis committee and for all the help and advice he has given.

Thank you to my parents for always supporting me and encouraging me to pursue my education. I’d also like to thank Preston, Shelby, and Lakai, for their encouragement and support through this process.

Table of Contents

1. Introduction

Geospatial workflows are often used in geographic information systems (GIS) as a way to automate a specific task. These workflows can often integrate large amounts of data from a variety of sources. Knowledge of data sources, the creators of the data, as well as what tools or changes were used on the data can be difficult to ascertain by just viewing the workflow. The collection of provenance is important in that it can provide a solution to managing geospatial data in workflows. Defining provenance is essential to being able to adequately collect it.

Provenance is data that is collected from recording the lineage of a specific object. Provenance has been often used in a wide range of fields such as art and computer science. Provenance data can also be recorded about the processes composing a workflow. A workflow is a chain of services or tools that can describe an overall procedure and when executed, produce intermediate and final products for scientific analysis (Hey et al., 2009). When considering provenance in relation to the geospatial domain, provenance can be thought of as the process history of geoprocesses used within a workflow or a study (Tullis et al., 2015).

Through the collection of provenance for a workflow, it is important to understand if a particular workflow can be trusted. The definitions of trust are varied and often depend on the context of the situation. Since using provenance with geospatial workflows is relatively new, it is desirable to test if trust can be determined by using this collected geospatial provenance. Statistical methods, such as Hidden Markov models have been used in non-geospatial applications to predict a trust score for a given workflow. This study examined if certain methods are appropriate for use with geospatial workflows. Due to the burgeoning interest of capturing and storing geospatial provenance, it is important to examine ways in which it can be used.

Over the past decade, the geospatial community has taken a renewed interest in provenance and its applications (Di, 2013; Yue, 2009). Collection of provenance has been an early and continued focus in the geospatial community (Lanter, 1990; He et al., 2015). However studies focusing on ways in which collected geospatial provenance can be used, are minimal. It is important to note than in this study the terms provenance and lineage will be used interchangeably.

1.1.1. Statement of Problem

Much research has been done in developing provenance systems that record geospatial provenance; however research on how to effectively use this provenance is minimal. The disciplines of computer science and information systems have developed applications in conjunction with provenance data which allow for a quantifiable measure of quality or trust to be produced, or a probability that a workflow is trustworthy at a given state (Rajbhandari et. al., 2006; Naseri & Ludwig, 2013). Using geospatial provenance to quantify quality has been approached by Malaverri, however very few articles currently explore this topic in depth for trust.

The expansion of trust to geospatial provenance poses two unique challenges. The first challenge is that very little exploration has been done on methods or benefits for evaluating trust using geospatial provenance. Therefore, other disciplines must be heavily consulted for applicable techniques of determining trust of geospatial data. The second challenge is the unique nature of geospatial data. Special attention must be paid to spatial and temporal aspects of the data and the way in which these can be handled when computing trust. In particular, which statistical measures can be applied to non-static data and which techniques can be used to handle evolving data within a workflow.

1.2. Research Questions

Provenance data has been used to generate a measurement of quality for geospatial data (Malaverri et al., 2012; Malaverri et al., 2013). However, the use of provenance to determine trust for geospatial workflows is an area where research is needed. An overall goal of this study is to determine if provenance collected from a geospatial workflow can serve as an indicator of trust and if a Hidden Markov model can be applied to predict a trust value. The goal can be broken down further into three research questions that must be answered.

The first research question is: can provenance be collected from a geospatial workflow in such a way that it is useful to additional applications? In this particular study, the application is to facilitate the calculation of determining content trust and workflow trust for a given workflow. Provenance data will be collected and stored following the PROV data model. Open source or free trial versions of software will be used if possible.

The second question is: can a measure of trust be calculated for a given workflow based on parameters that are representative of trust, which stem from both the metadata and provenance of the workflow? This thesis will examine which factors can be used to estimate trust for a geospatial workflow. Careful attention will be made to the nature of geospatial data. These parameters will be obtained from the metadata, provenance, and user needs of a workflow.

The third question to be addressed is: can a Hidden Markov Model be used to predict the probability of a level of trust for a geospatial workflow. Naseri's dissertation explored the use of stationary and non-stationary Hidden Markov models for non-geographic data; however this technique has yet to be applied to geospatial data provenance. The goal of the third question is to see if this method is appropriate for analyzing geospatial workflow trust. A thorough review of the literature on provenance, trust, and Hidden Markov models provides more understanding into the three research questions for this study.

2. Literature Review

Research in geospatial provenance has a broad focus. Understanding the need for provenance, provenance collection system design, and provenance data quality are some of the topics analyzed in the literature. Portions of this section will appear in Tullis et al., 2015, which is currently in publication.

2.1. Provenance

Provenance from the French word provenience, to originate, comes from a concept started in the art community (International Foundation for Art Research, 2013). Provenance in art is focused on recording a piece's ownership history. This record would contain a history of a particular piece's owners, transfers, date of these occurrences, and locations (Yeide, 2001). Provenance has since been used in a wide variety of disciplines, such as archaeology, computer science, geospatial analysis, and remote sensing. A more modern definition of provenance can be thought of as a description of objects and their production transformations which can serve as a method of reproduction, authentication, and data trust (Groth & Moreau, 2010).

Although each discipline's provenance may have unique requirements or variations in semantics, the overall concept can be summarized by the following definition provided by the World Wide Web Consortium (W3C): "Provenance of a resource is a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource." (Gil et al., 2010). For the purpose of this paper, W3C's Provenance Working Group's definition and recommendations for provenance will be followed.

The Provenance Working Group is an effort by the W3C to create interoperable guidelines for provenance on the web. The W3C's PROV-O recommendation defines three main classes for provenance as entity, activity, and agent. An agent (actor) can be associated with a particular action towards a specified item, an entity is an item that can be manipulated, and an activity is the particular action towards an entity associated with an agent (Groth & Moreau, 2013). In addition to PROV-O and PROV-XML which are designed for machine consumption, there is another format for how provenance can be represented. PROV-N can be used to display provenance in a format that is easy to read. This format is designed to make provenance easily accessible for humans and is not designed with other applications in mind (Moreau & Missier, 2013). Figure 1 shows an example of PROV-N notation for the derivation of a photograph.

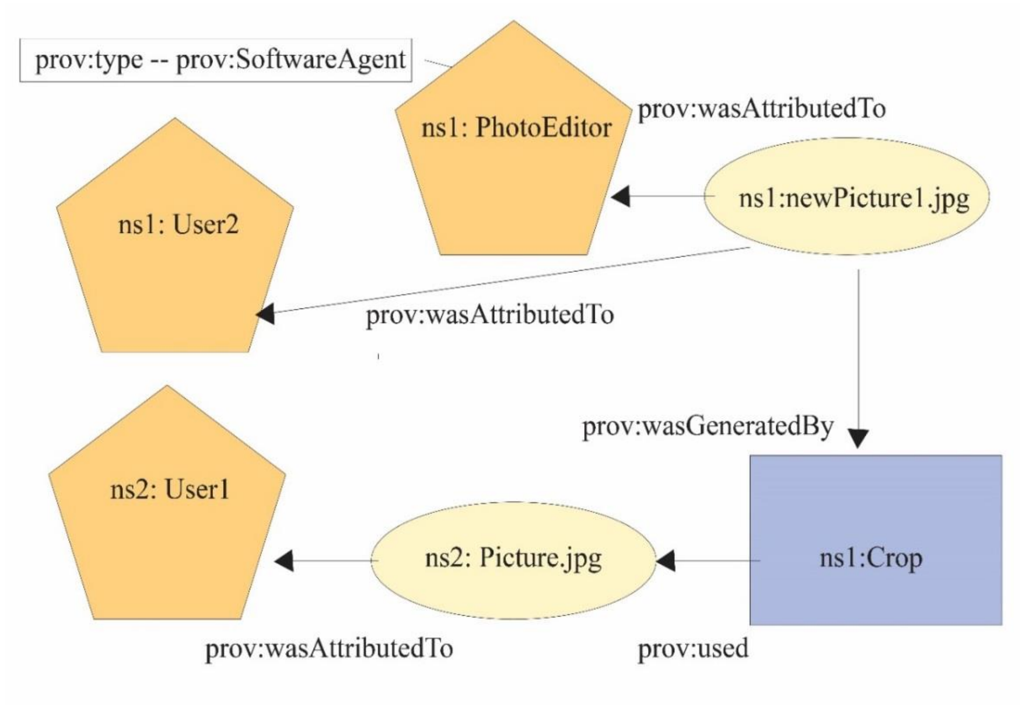


Figure 1. Graphical representation of the PROV-N model of a photograph.

The provenance pictured in Figure 1 in analogous to the type of provenance that might be included with a remotely sensed image. In addition to the discipline's requirements, the data captured will be a reflection of the type or level of provenance being recorded.

2.1.1. Types of Provenance

Different types and granularities of provenance can be captured based on the user's needs. Two types of provenance are *why* and *where* provenance. *Where* provenance describes the location in which the data is stored, while *why* provenance describes parent-child relations between datasets (Buneman, 2001). What this means is that *where* provenance is concerned with file paths for a particular dataset or the location of a tuple in a database that contains provenance data. *Why* provenance on the other hand is concerned with the data itself.

For example, a researcher has a dataset called D1. All of the datasets that make up dataset D1 are considered D1's parent relations. All of the datasets that have been produced using D1 as an input are considered D1's child relations. In addition to provenance type, provenance granularity must also be considered. *Coarse grain* provenance records the processing steps of as system, such as provenance at the workflow level, whereas *fine grain* provenance records information at the data level, such as data in a tuple or pixel size (Tan, 2007; Woodruff, 1997).

Depending on if provenance is *where* or *why*, the method of storage may differ. For example, recording *eager* provenance deals with the storing of where-type lineage, while *lazy* provenance deals with the storage of why-type lineage. *Eager* capturing records *where* provenance at a *fine grain* level and will immediately log changes made, while *lazy* capturing records at a *coarse grain* level (Ikeda, 2009). Provenance collection is not limited to only collecting one type or granularity of provenance. A system can collect multiple types or granularities of provenance and still be a valid working model, as long as the data sets involved supports this (Yue, 2009). Regardless of which type of provenance is recorded, there must be a system in place where the provenance is managed.

2.1.2. Geospatial Provenance Management

In order to analyze provenance, it must be successfully captured and stored. Although multiple techniques have been developed in computer science to collect provenance data, the ones that are discussed have been used to specifically capture geospatial provenance. Inversion, service chaining, and ontology expansion will be discussed below.

Inversion can be used to capture lazy provenance for *fine-grain* data lineage (Woodruff, 1997). Inversion takes place when examining database transaction history and working backwards towards the source (Tullis et al., 2015). Woodruff uses inversion at the tuple level by registering attributes, elements, and algorithms, then performing inversions on these items leading to a lineage trail between data transformations. The data is displayed as a function, which is then weakly inverted. Using an object-relational database and a database visualizer to store the inversion functions, provenance can be effectively stored and queried. The data Woodruff analyzes is based on cyclone tracks, but this method could be applied to various other types of geospatial data. Woodruff's work is significant in that it was one of the first efforts to expand geospatial provenance to the *fine-grain* level of collection and successfully capture the desired transformations of the data.

Yue models a workflow to allow geospatial programs to be in a service chain, allowing for provenance capture (Yue, 2010). A service chain allows for the architecture of a program to be composed of various services that when used together form the overall design of the program. The provenance is registered in a catalogue service and complies with the interoperable specifications of web ontology language (OWL). Yue's work is important in that it provides the start for expanding geospatial provenance for use on the Semantic Web.

Yue expands further on provenance techniques by proposing an ontology be extended to cover the geospatial field, allowing for a catalogue service to be extended covering geospatial provenance (Yue, 2010). A catalogue service makes use of (Hypertext Transfer Protocol) HTTP to allow geospatial data or records to be discoverable over the web. The Open Geospatial Consortium's guidelines (OGC) and the International Standards Organization's ISO-19115:2003 are used to expand an ebRIM catalogue model to extend to geospatial provenance storage. Yue further expands on implementation of this technique by extending provenance relations and developing a more service oriented architecture, using extensible markup language (XML) (Yue, 2011). The significance of this work is that Yue recognizes that in order for provenance to be successfully adopted and used on the Semantic Web a geographic ontology must be extended to incorporate geospatial provenance.

2.1.3. Provenance Standards

The earliest geospatial provenance standard was developed by the American Congress of Surveying and Mapping (ACMS) for the National Committee for Digital Cartographic Data Standards (NCDCDS, 1988). The current standard in use for the geospatial domain is the International Standards Organization’s ISO 19115-2, which has been endorsed by the Federal Geographic Data Committee (ISO, 2009). One major difference between the two standards is the ISO couples lineage with metadata, whereas the NCDCDS did not. As pointed out by Di, geospatial provenance needs a way to conform to standards through ISO 11915:2003 (Di, 2013).

Much work has been done recently on how to achieve compliance with various provenance standards. Di has developed a method for extending a service oriented architecture (SOA) provenance system’s output to comply with ISO 11915-2 and ISO 19115:2003 standards (Di, 2013). Feng has taken this a step further and used the Open Provenance Model (OPM) standards to handle metadata complying with ISO 11915-2. Feng achieves this by adding four new categories to the OPM model that can handle geospatial provenance and by using Java to control the document object model libraries to handle the provenance data (Feng, 2013). Another attempt at fashioning geospatial provenance to an international standard is turning geospatial provenance into metadata as defined by the Dublin Core metadata standard in order to become linked data for web browsing (Yuan, 2013). These standards define the shaping of current and future provenance collection, however to understand how these standards came about it is important to examine early attempts at the concept of provenance and its collection.

2.1.4. Formulating Ideas

The earliest work in geospatial data provenance was spurred by the formation of the National Committee for Digital Cartographic Data Standards (NCDCDS) by the American Congress of Surveying and Mapping in 1982 (Bossler, 2010). Provenance, also known as lineage, was incorporated by the NCDCDS as one of the five fundamental components to assessing data quality (NCDCDS, 1988). With transitions from static to computerized mapping, several things about data quality and data processing were realized. Geospatial data processes need to be tracked from their origins, through revisions to the data, and finally to the output (Moore, 1983). Chrisman brings up the point that data quality and the tracking of data lineage are necessary in order to see if the data is being suitably used within a geographic information system (GIS). In order to judge suitability the GIS lineage could possibly be captured and viewed through an overlay method on a map (Chrisman, 1984).

Other preliminary ideas on the use of lineage focused on other aspects of geospatial data. The temporal aspect of data must be considered as it can greatly impact the quality of data used in a GIS (Langran, 1988). In order to better compliment data transferring, lineage could be used in developing systems to have an understanding of data quality (Nyerges, 1987). Grady also supports lineage as a measure of data quality which can be used to record societal mandates (Grady, 1988). These scientists focused on the theory and possible implications of provenance, however other scientists began experimenting with how to actually record and collect geospatial provenance.

2.1.5. Provenance Prototypes

Several provenance systems were designed to record geospatial data lineage. One of the earliest was the Map Librarian's catalog system in ARC/INFO. The catalog served the function of recording a map document's history. This history showed which tiles contained which layers, locations, and updates (Aronson, 1983). This system falls short of the goal of lineage, as it replaces each layer once it is updated with the newest layer in the catalog. MARKII, a system developed by the United States Geological Society's National Mapping Division, allowed for a database's current dataset to be tracked. However, spatial data requirements of large file sizes and complex geometries made the database systems of the time unable to properly perform (Guptil, 1987). A solution to this problem was proposed by Egenhofer, advocating for the object-oriented principles of propagation and inheritance to be utilized to overcome geospatial database issues (Egenhofer, 1992).

GIS databases were unable to capture the complete lineage of a map document as they were incapable of tracking temporal versions of the data. Langran identified the problem of version overwriting and expressed the shortcomings of the snapshot and log methods for storing lineage (Langran, 1988). The methods of using overlays to capture time changes to the data and the polygon intersect method, were proposed by Langran and Chrisman as possible solutions to the problem of tracking temporal changes in geospatial data. The use of the overlay method or the polygon intersect method was only considered feasible if software capabilities were improved (Langran, 1988). These early prototype systems identified shortcomings of the geospatial provenance collection and the types of problems future systems would need to solve.

2.2. Early Provenance Systems

The results of the provenance prototype systems identified two important issues that would needed to be addressed if geospatial provenance collection was to be successful. Changes in datasets would need to be tracked, not just replaced, and the time these changes occurred must be logged as well. Early provenance systems were focused on taking the idea for the need to capture provenance and turning this idea into a reality. These early systems also addressed the two problems identified by the provenance prototype systems. Audit trails and version control were the main contributions of these systems. Provenance systems such as Geolineus and Geo-Opera, as well as additional systems will be examined in more detail below.

2.2.1. Geolineus

Created by David Lanter, Geolineus is a single tiered architecture provenance management system. This system divided geospatial layers into three types: source layer, intermediate layer, or product category layer (Lanter, 1991). A source layer acts as a parent node that derives intermediate layers. Transformation and manipulation occur on the intermediate layers, which serves as a middle stage between source and product category. Product category layers are derived layers of the provenance system, and are also child nodes. Links are used to connect layers, showing the flow of the transformations in the data.

Software architecture for Geolineus is composed of: frames, a lineage information program comprised of a knowledge representation program, and knowledge representation interrogator. Frames are used to record the attributes of the various nodes and transformations. Three types of frames are used: command frame for transformations, source frame for source layer attributes, and product frame for product layer attributes (Lanter, 1991). The lineage information program (LIP) serves as the application layer for the user. User commands are entered into the LIP, parsed and then executed by the GIS system. A knowledge representation is created, representing relationships among nodes. From this representation, the knowledge interrogator can query the provenance information stored within a database. Geolineus has the ability to update and avoid redundancy. Source path names and attributes are checked for uniqueness. If a duplicate is found, it can be merged as one view within the database. Derived layers can be checked for uniqueness by tracing their links for similar ancestors. To update data, time stamps are checked on each version. For the newest version, parameters are taken, the source is updated, and the transformations are re-applied (Lanter, 1991).

Lanter named his program GEOLINEUS and during test runs came to the conclusion that when source information is lacking, a lineage system's ability to determine data quality can only go so far and, to remedy this, a way to automatically capture information must be incorporated (Lanter, 1994). Automatic updating of data layers can be achieved by following the parent child links of each node, comparing the creation dates, and updating the selected layer through intersection and union (Lanter, 1994). Geolineus is unique in that it is one of the few provenance systems that made it out of the testing stage, into the fully fledged production stage for use by clients. The disadvantage of this system now is that it is not widely available and as technology has changed, aspects have become obsolete.

2.2.2. Geo-Opera

Also incorporating geospatial provenance into its design is Geo-Opera. Geo-Opera allowed for interoperability, data recovery, the ability to log history, and a system for monitoring data versions (Alonso, 1997). Geo-Opera is based on a modular 3-tiered architecture allowing for easy updating (Alonso, 1997). The application service is composed of an internal and external user interface. The internal interface establishes communication protocols to the other layers. The external user interface resides on the client's machine. Geo-Opera's processing service consists of a dispatcher, navigator, object manager, and query manager (Alonso,1997). The dispatcher locates available machines within a network to be used for processing and manages their communication. The navigator is used to monitor the processes and their updates. The object manager updates information for externally registered objects, and the query manager serves as the interface for querying information within the database.

The database layer is composed of five spaces: template, history, instance, object, and configuration to record the provenance information (Alonso, 1997). The spaces allow portions of the database to be stored within different machines. Geo-Opera uses its own modeling language for processes and the Opera Canonical Representation language is used to identify various entities within the provenance system. External objects not represented in Geo-Opera must be registered by the user, and the object's attributes must be defined. When this is done, the external object becomes a black box within Geo-Opera and database functions can then be applied to it (Alonso, 1997). Version attributes are logged, which further contributes to lineage collection. Updates are performed by flagging all related objects to the updated object. The lineage is computed and the source object is re-run using the transformations stored in the database (Alonso, 1994). Geo-Opera is important because if fulfills some of the requirements of a provenance system such as logging changes and tracking versions. Geo-Opera is not considered a complete provenance system in that it lacks the abilities to store and retrieve spatial data.

2.3. Provenance Systems

The following section outlines selected provenance systems that have been used with geospatial data. Although some systems were not designed with capturing geospatial data provenance as their goal, all mentioned systems have been used for this. These systems are more recent than Geolineus or Geo-Opera and therefore have different considerations and designs.

2.3.1. Advanced Microwave Scanning Radiometer-Earth Observing (AMSR-E) System

The AMSR-E Legacy data system focuses on capturing geospatial provenance information from a no longer used AMSR-E satellite. This system not only captures generated provenance information, but uses manual forms to capture provenance information in context (Conover, 2013). AMSR-E uses a two-tiered framework based on the biological provenance system Taverna and the geospatial provenance system Karma (Conover, 2013). Manually entered contextual provenance is given a digital object identifier (DOI) and a Uniform Resource Number (URN) to help query the provenance information. The Earth Science Library for Processing History (ELPH) is used to browse the provenance with XML and is based off of Karma's browser (Conover, 2013). As pointed out by Conover, this and other types of geospatial provenance systems have difficulty being run without using workflows. A solution to this is to run a Linux operating system that allows data logging (Conover, 2013). Since this system is based off of Karma, it has a sound foundation, however centering a provenance system on technology no longer in use may possibly hinder this system moving forward.

2.3.2. Data Quality Provenance System

Taking into account a source’s trustworthiness and the data’s age, Malaverri has created a provenance system that allows a quality index to be assigned (Malaverri, 2012). Malaverri’s provenance model is based on the OPM provenance model and follows ISO-19115 metadata standards. Other criteria for the quality index include: granularity, accuracy of attributes descriptions, completeness of the data, a logical measure of the data, and spatial positional accuracy (Malaverri, 2012). Malaverri’s work is unique in that it attempts to quantify data quality by using provenance.

2.3.3. Earth System Science Workbench

Earth Science System Workbench (ESSW) is n-tiered provenance system architecture. ESSW uses scripting techniques to collect geospatial provenance information. ESSW is composed of two main components: the Lab Notebook and Labware (Frew, 2001). Lab Notebook acts as the server to the system. It essentially serves as a metadata registry and repository (Frew, 2001). Lab Notebook is a Java-based system that gathers lineage and metadata information. Lab Notebook converts the parameters gathered into XML and stores these within a relational database after being parsed. A user accesses Lab Notebook by using Perl scripting, which invokes a call to the application programming interface (API). A daemon runs in the background in order to listen for event calls from the client. When one is obtained, IBM's XML parser for Java is used to convert client input from Perl to XML. Java API then converts the XML into structured query language (SQL) to query the relational database (Frew, 2001). The database is composed of tables with root elements containing document-type definitions. Every experiment ran generates a new table. All tables inherit attributes from a base table, as well as their own attributes that are added after a process is run. A unique ID is assigned to each table in order to be able to query for lineage information, and to prevent reduplication (Frew, 2001).

Earth Science Server monitors the execution of ESSW, and runs automatically in the background. It is composed of a logger and a transmitter. The logger monitors and records program specifications during run-time (Frew, 2008). A plug in can be created in order to allow the logger to run better with specific applications. The transmitter allows a unique ID to be generated for each object in the log file. It also transforms plug-in log files into files that ESSW can read, which are subsequently parsed into XML (Frew, 2008).

Both systems run on the Linux operating system. System processes are able to be logged using the *strace* function (Frew, 2008). Time can be identified by accessing the time on a client's operating system for lineage traces (Frew, 2008). The weakness in this system is that it was created before standards could be applied, making its interoperability questionable.

2.3.4. GeoPWProv

GeoPWProv is a geospatial provenance system designed to move away from displaying provenance as workflows and instead displaying it as a visual graph. GeoPWProv has four parts: a geospatial provenance recorder, provenance storage system, provenance finder, and provenance exhibitor. Geographic Markup Language (GML) is used to interact between a web browser and Open Layers map display (Sun, 2013). Sun’s work is unique in that it displays geospatial provenance information in a way that differs from other systems. Placing emphasis on the visual aspect of provenance, such as the connection between data layers, allows for a clearer understanding of the changes taking place between a workflow’s layers.

2.3.5. HiTempo

Although HiTempo is not a system based on provenance collection it plans to include provenance collection as a component (Van Den Bergh, 2012). HiTempo will deal with MODIS, SPOT, and Advanced Very High Resolution Radiometer (AVHRR) data (Van Den Bergh, 2012). As provenance is not the focus of HiTempo, specifics of how this will be structured into the system is omitted. The lack of current provenance implementation details for HiTempo is disappointing, however if provenance can be successfully collected, the implications for this for geospatial data are considerable. Many studies often make use of MODIS, SPOT, and AVHRR data, and having a way to track the provenance of the data transformation process will be extremely useful.

2.3.6. Karma

Plale makes use of the Karma system designed by Simmhan to collect provenance data on AMSR-E (Plale, 2011). One of the biggest benefits of Karma is its modular architecture. This allows Karma to be interoperable with Java and other web services (Plale, 2011). Karma's architecture for this application consists of an application layer, web service layer, core service layer, and a database layer (Plale, 2011). Open Provenance Model (OPM) specifications and XML are included, thus making its interoperability extend further (Moreau, 2011). Karma is unique in that it is a provenance system that has shown success in a variety of use cases. Its interoperability can allow possible extension for a variety of provenance collection scenarios if so desired.

2.3.7. MyGrid

MyGrid is an n-tiered provenance system consisting of four layers: services, workflows, provenance, and middleware (Zhao, 2003). MyGrid uses an ontology based on DAML-S OWL semantics allowing for interoperability (Zhao, 2003). Freefluo is used to handle the workflows and is capable of running web service definition language. Also in the workflow layer, xScufl is used to extend the Java language for workflow definition and Ws-info doc is used to define workflow parameters .Both languages are based in XML (Zhao, 2003). Freefluo also acts as middleware, parsing and storing provenance information within the provenance repository (Zhao, 2003). MyGrid has a good structure to base a methodology on, however it was built before ISO standards could be actively applied to lineage.

2.3.8. Science Data Processing System (SDPS)

SDPS makes use of the MODIS Operations Data Processing System (MODAPS) and the Ozone Monitoring Instrument Data Processing System (OMIDAPS) for satellite data and algorithms to query provenance (Tilmes, 2008). SDPS uses a scripting process to track changes in algorithm versions. Using this, every version is not stored, but enough information is retained that an older version of data can be recreated. Tilme's data recreation is unique and is something that could be useful in other geospatial provenance systems. The focus on algorithm changes is useful in that it allows one to see exactly how the end data product was created.

2.3.9. UV-CDAT

Santo's UV-CDAT is a relatively new provenance system built for handling large amounts of climate based data (Santos, 2012). UV-CDAT uses a data viewer interface (DV3D), CDAT core, and uses VTK/paraview as an integrated infrastructure with outside components (Santos, 2012). UV-CDAT allows the use of multiple scripts to be used and has a graphic interface for displaying workflows. UV-CDAT is unique in that it has been distributed and is in use by other scientist throughout the climate field. UV-CDAT has a helpful documentation and installation section for those who which to use it in their research (UV-CDAT, 2015). This is different from most of the other systems mentioned, as the actual use of them with everyday projects is obtuse. UV-CDAT's creators have focused on making this system easily adoptable.

2.3.10 Future Research

Future research for geospatial provenance systems that lies outside the scope of this paper's research can be identified in several areas. Research can be done to expand provenance systems to handle more data heavy applications (Wang, 2009). Research can also be done to understand why provenance is not fully incorporated and what can be done to make a change in the mindset to include provenance from now on (Tilmes, 2008). This is especially important in the geospatial domain as what is often seen is the end data product. Without knowing how this product was created or what datasets went into its production, making an informed decision about this product will be difficult. Research has also only been done on provenance in a scientific setting, no articles have been found on attempts to utilize it in the general public. Research can also be done on how to insure provenance is interoperable without a geographic ontology (Jones, 2003). A lineage standard has been adopted through ISO 19115-2 and endorsed by the FGDC, however this lacks much that an ontology could help contribute to (ISO, 2009). While these ideas are not touched upon in this paper, this paper will cover geospatial provenance and its use for evaluating trust in regards to geospatial data and workflows.

2.4. Trust and provenance

Before delving into the properties of trust, it is important to identify how provenance and trust are related. Provenance can be used to help make a more informed decision on whether or not to trust a particular item. In the case of workflow provenance, provenance provides additional data regarding the processes that occurred in a workflow, as well as how they were carried out, and by whom, which can then be used to evaluate trust (Rajbhandari et.al, 2006). Additionally, provenance may also be used to provide more document or data layer metadata and if this provenance is trustworthy, that can help provide validity for trusting a content resource (Gil & Artz, 2007). In order to provide a clearer understanding of what constitutes trust, a more detailed explanation is given below.

2.4.1 Trust

Trust is an integral part of decision making. The decision to trust a piece of information or a particular dataset can result in its adoption for use and while the decision not to trust can lead to the item be refuted and no longer considered for use. For example a researcher may have a workflow to determine the NDVI index for a geographic location that they downloaded from the internet. Upon closer examination of the tools used in the workflow and the parameters evaluated, the researcher notices that the algorithm used to calculate NDVI is incorrect. This leads the researcher to not trust the outputs of the workflow and avoiding using it.

When analyzing a workflow, it is important to determine if its components, such as the inputs, intermediate layers, and outputs, can be trusted. In addition to this, the tools or services used in the workflow, along with the algorithms involved are also critical in determining trust. There are a significant number of definitions for trust, which vary based on the context of a situation. A broad definition of trust as defined by Xin Liu is “the relationship between two entities, where one entity (trustor) is willing to rely on the (expected) actions performed by another entity (trustee)" (Liu et al., 2014). As the context is determined, the definition of trust becomes more specific. A brief overview of the different types of trust relatable to geospatial workflows and their use case domains are given below.

2.4.2. Computer Science

Multiple types of trust are found in computer science literature. Rajbhandari et al., define three types of trust measurements: process trust, service trust, and data trust. Process trust is a subjective decision of trust based on the user's evaluation on the results of a workflow, while service trust is objective and based on observing the past behavior of a workflow's components (Rajbhandari et al., 2006). Data trust is composed of both objective and subjective components and is used on the intermediate workflow processes. A decision tree model is used to determine a trust measure for a given workflow.

Naseri and Ludwig use Hidden Markov Modeling (HMM) to model workflow trust. The parameters execution status, reliability, and availability of a workflow are used to determine the quality of each service in the workflow (Naseri & Ludwig, 2012). These are then classified into three states, which in turn are given a level of trust. The HMM is then used to examine at a given state of the workflow, what is the probability that this state will be in each level of trust.

Artz & Gil give a review of the types of trust that have been focused on in scholarly research, characterized by four main categories: policy based, reputation based, general trust models, and informational trust. Policy based trust is concerned with using credentials such as user name and password to verify trust, while reputation based trust uses factors such as ranking and Eigen trust or performance history (Gil & Artz, 2007). General trust refers to determining which factors influence trust in a particular domain and how trust can be determined in that context. The final category, information trust focuses on the trust of resources based on the internet and their reliability and quality (Gil & Artz, 2007). Content trust also focuses on the trust of a given resource.

Content trust is an evaluation of trust based on the resource or data item itself (Gil & Artz, 2007). Gil and Artz further examine content trust by determining the factors that most influence trust of a resource. Nineteen factors are considered with six as being the most valuable for determining content trust. These six factors are: the source of the resource ("Authority"), similar resources to the resource being examined ("Related Resources"), the provenance of the resource ("Provenance"), bias of the resource's source ("Bias"), what reasons does the resource have to be accurate or unbiased ("Incentive"), and the estimate that the resource is misleading ("Deception") (Gil & Artz, 2007). An example of a system to evaluate and store content trust is TRELLIS. TRELLIS allows users to manually enter their view on a given resources content trust, by allowing the user to select annotations to describe the resource, which is then given an overall rating (Gil & Ratnakar, 2002).

Jung et al., define two types of trust in relation to grid computing, domain based trust reasoning and property based trust reasoning. Domain based trust reasoning allows for the data versions, provenance, and the semantic origins of the data to be verified, while property based trust reasoning only focuses on the artifacts of interest to the user (Jung et al., 2011). These are then combined to form a multi-layer trust reasoning for use with the Open Provenance Model. Although these definitions of trust are defined in computer science literature for that particular discipline, they could be applied to GIS, as GIS can fall within the domain of computer science. More specific definitions of trust applicable to GIS or geography are discussed below.

2.4.3. Geography

Trust in relation to geographic data is examined in a number of articles. Keßler and Groot examine informational trust in the context of volunteered geographic information datasets in OpenStreetMap. Informational trust indicates how much a particular feature can be trusted based on its editing history and assessment of quality (Keßler& de Groot, 2013). They consider the following parameters as indicative of trust: version history, amount of users editing a feature, edits made nearby the feature in question, corrections to tags, and number of times a feature has been reverted to a previous version. A ranking of the parameters is then used to assign trust.

Garijo et. al., identify that it is important to have geospatial provenance in order to determine trust about geographic data. They specifically focus on the context of content trust. A set of questions is developed in regards to geospatial provenance, that the user can ask his or herself in order to evaluate the provenance data and decide on trust (Garijo et al., 2014). Malaverri examines trust in a geospatial context in relation to quality. Trust in a source and trust in the temporality of the data are given a normalized range [0,1] in relation to an Agent and Artifact (Malaverri, 2013). Provenance of geospatial data is used to help determine the quality of the data. The trust scores used to help determine quality are subjectively assigned by experts based on their opinions of the data (Malaverri, 2012). Special consideration is given to aspects of geospatial quality in Malaverri’s model, incorporating aspects from FGDC metadata to help determine this.

Trust in geospatial linked data is also an area of interest. Harth and Gil propose that because geospatial data can be integrated into one object from different sources, that the provenance of each data item is necessary to determine content trust. Granularity of the provenance recorded and the temporal aspect of geospatial data are of particular need in evaluating trust of a given dataset (Harth & Gil, 2014). From the following articles, it can be gathered that provenance is an integral part to determining trust. Provenance has not only proved valuable to determining trust of content on the web, but also offline as well (Moreau, 2013). After provenance has been collected and trust defined, the next step is to evaluate or estimate the trust for a particular workflow or data item. One of the ways in which this has been done outside of the geospatial context is to use Hidden Markov Models.

2.5. Hidden Markov Model

Hidden Markov models (HMM) have been used in a variety of applications such as speech recognition, finance, engineering, and computer science. A Hidden Markov model is a Markov chain model with a sequence of unobservable states. The Markov property must be met in order for this to be used. The Markov property can be defined as a stochastic process that can be thought of as memoryless. Another way to phrase this is that the future state of the model depends only on the present state of the model (Ramachandran & Tsokos, 2015). For example, equation (1) represents a first order Markov chain as:

$$P(q_t=S_j|q_{t-1}=Si,q_{t-2}=S_k\ldots) = P(q_t=S_j|q_{t-1}=S_i)$$

Equation 1: Markov property (Rabiner, 1989).

Where q_t is the current state at time t, and S is a set of discrete finite states $\{S_j,Sk,Si\}$. A Markov chain can represent a discrete or continuous sample space and can also represent a discrete or continuous time series. Which technique is selected is dependent upon the subject being modeled. In addition to determining discrete or continuous, the class of model must be chosen. Ergodic Markov chains are Markov chains in which every state can be reached from all states in the model, which will most likely contain a small number of states if modeling a finite state space (Cappé et al., 2005). An example of an Ergodic Markov chain is shown in Figure 2.

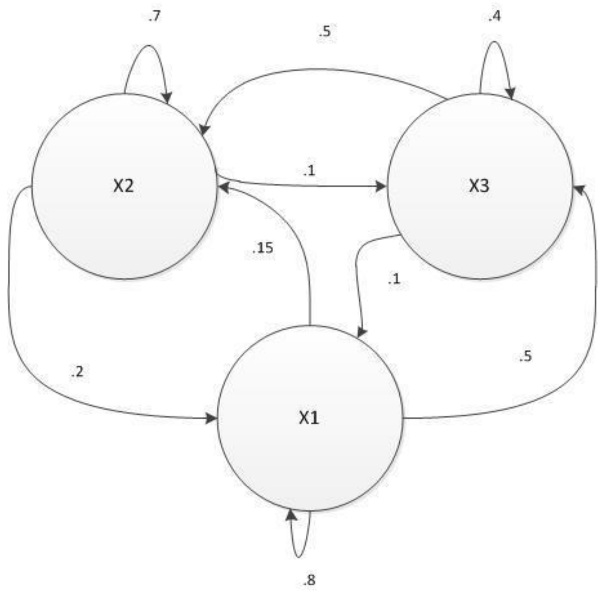


Figure 2: An example of an ergodic Markov chain

A regular Markov chain can be an ergodic Markov chain however not all ergodic Markov chains are regular. A regular Markov chain has a transition matrix, A, which when multiplied to some power *n*, yields a transition matrix with all positive entries (Elizade, 2006). The other major class of model is a left-to-right Markov chain. A left-to-right Markov chain begins in an initial state, moves through the intermediate states to the right based on its transition matrix, however it cannot go backwards towards a previous state, and ends in an absorbing final state (Cappé et al., 2005).

The previously mentioned properties for Markov models also apply to HMMs. There are additional parameters that are required for a HMM. HMM have two states and three probability distributions that must be specified. There is the physically observable sequence which can be thought of as the data that is the output of the hidden state. For a discrete space HMM, this observable sequence can be described as V={v_1,v_2,v_3…v_M}. Where V is the observation at time t, and M is the number of observations for the model (Rabiner, 1989). The series of states that produce the observations for the model are hidden and therefore unobservable, except for the output observations V. The set of hidden states can be modeled by the number of states the model is composed of such as S={s_1…s_N}, with N being the total number of states (Rabiner, 1989). The three probability distributions that must be calculated are the initial probability distribution, the transition probability distribution, and the emission probability distribution which are in the form of matrices.

The initial probability distribution is the probability that the model starts in a specific hidden state. This can be modeled by equation (2).

$$\pi_i=P(q_{1} = S_i) \quad 1 \leq i \leq N$$

Equation 2: Probability of initial starting states (Rabiner, 1989).

Where q_1 is the initialization time and S_i is a given state from S. The transitional probability distribution is the probability one state will transition to another state or state in its current state, based on the HMM used. The size of this matrix is dependent upon the number of states in the model. The transition probability distribution A is shown in equation (3):

$$A = \{a_{ij}\} = a_{ij} = P(q_{t+1}=S_j|q_t=S_i) \quad 1\leq i,j\leq N$$

Equation 3: State transition matrix (Rabiner, 1989).

Therefore the probability of a state transitioning to another state is the conditional probability of its current state S_i transitioning into state S_j. If the transition matrix remains constant for each time state t, then the HMM is stationary. The emission probability matrix represents the probabilities of, given the current observable state, the probability of the hidden state. This can be represented by equation (4).

$$B_j(k) = P(v_{kt}|q_t=S_j) \quad 1 \leq j \leq N,\ 1\leq k\leq M$$

Equation 4: Emission probability matrix (Rabiner, 1989).

Where the emission probability of observing V_k at time t, is conditional on the probability of hidden state S_i at time q_t. This study will make use of various aspects discussed in the Hidden Markov model section, the trust section, and the provenance section. Details on the methods used in this study are given below.

3. Methods

The following section discusses the methods used in this study in detail. Open source, freely available, or trial versions of data and software are used in this study in order to make its replication more widely available to users. Techniques used are discussed in detail within the body of this work. Code used as well as additional data are available in Appendices A-F.

3.1. Software & Hardware

ESRI's ArcGIS ArcMap 10.2.2 software was selected to complete the geoprocessing tasks. ESRI provides a student trial edition of their ArcMap software. At the time of this study, a 60 day trial version is available to the public (ESRI, 2015). To compose the workflows and generate the initial Python script ArcGIS ModelBuilder10.2.2 was used. Notepad ++ v 6.6.8 was used as the primary text editor for making changes to the python scripts. Python 2.7.5 was the programming language and version used, and the scripts are executed through Python’s built in integrated development environment (IDE). It is important to note that the scripts used in this study are not compatible with Python version 3 or greater. LXML Python package was used to generate the extensible markup language (XML) used to store workflow provenance. RDFLib is a Python package that was used to store workflow provenance as resource description framework (RDF). MySQL 5.6 was used to store and query each workflow’s provenance. R version 3.13 was used for the statistical calculations. The R package HMM was used to run the Hidden Markov models used in this study. The software was run on Microsoft’s Windows 8 operating system. The hardware used in this study was an Asus X551 laptop with an Intel I3-3217U CPU at 1.80 GHZ, 4 GB of RAM, and 500 gigabytes of disk space.

3.2. Initial Workspace

Each workflow was given its own folders and .mxd document for use. For each workflow a file geodatabase was created containing input, intermediate, and output folders. Within the geodatabase, a toolbox and model were initialized. All available extensions within ArcMap were turned on in order to access the full range of ArcMap’s functionality. ArcMap’s log file for geoprocessing was also enabled at this point. However, this log file was not actually used in the study, beyond the initial comparison of determining if it was adequate for provenance collection. Once that determination was made, the log file was left enabled as a backup log, in case provenance collection failed.

3.2.1. Data

Geospatial data is composed of two main types, raster and vector data. A raster dataset is composed of pixels, with each pixel representing a spatial dimension and containing a value representing real world objects detected by the sensor (Faust, 2008). Vector data is composed of lines, points, and polygons representing real world objects (Fause, 2008).

Workflows used in this study contained a mixture of raster and vector data layers. Imagery was obtained from Landsat 8 via the USGS Earth Explorer. Digital elevation model (DEM) data was obtained from the National Elevation Dataset from the USGS. Vector data was downloaded from the Arkansas GIS Office’s, formerly GeoStor, data repository. In addition to these data sources, additional data was used from the ArcGIS ArcTutor Spatial Analyst tutorial. This tutorial data included both raster and vector layers. Related data was stored in its own directory, under a file geodatabase in ArcMap using ArcCatalog.

3.3. Workflow Creation

Three example workflows were created for this study. These workflows represented real world use cases of geospatial data and the methods used to analyze it. Each workflow used was defined as simple, intermediate, or advanced. These terms do not indicate complexity of the workflows goal, the users experience level, or the computational power required to execute each tool, rather they refer to number of processes that were executed in each workflow. Model Builder's GUI was used to drag and drop layers and tools to create the initial workflows. The workflow was created by initiating a new ToolBox, and within that ToolBox creating a new model in the same directory that the map document was stored in.

The simple workflow contained four processes. The goal of this workflow was to calculate NDVI for April 23, 2014 in Damascus, Arkansas. This workflow can be observed in Figure 3.

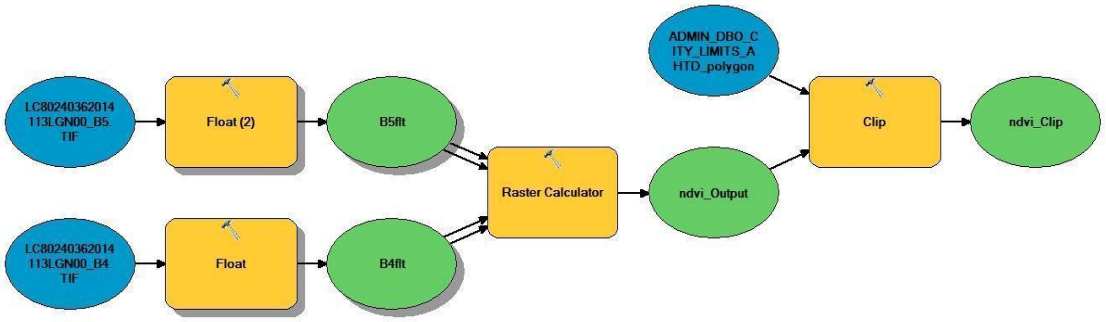

LC80240362014113LGN00_B5.TIF
Float (2)
B5flt
LC80240362014113LGN00_B4.TIF
Float
B4flt
Raster Calculator
ndvi_Output
ADMIN_DBO_CITY_LIMITS_AHTD_polygon
Clip
ndvi_Clip

Figure 3: Simple Workfllow - NDVI workflow composed in ModelBuilder.

The final output of the simple workflow is shown below in Figure 4.

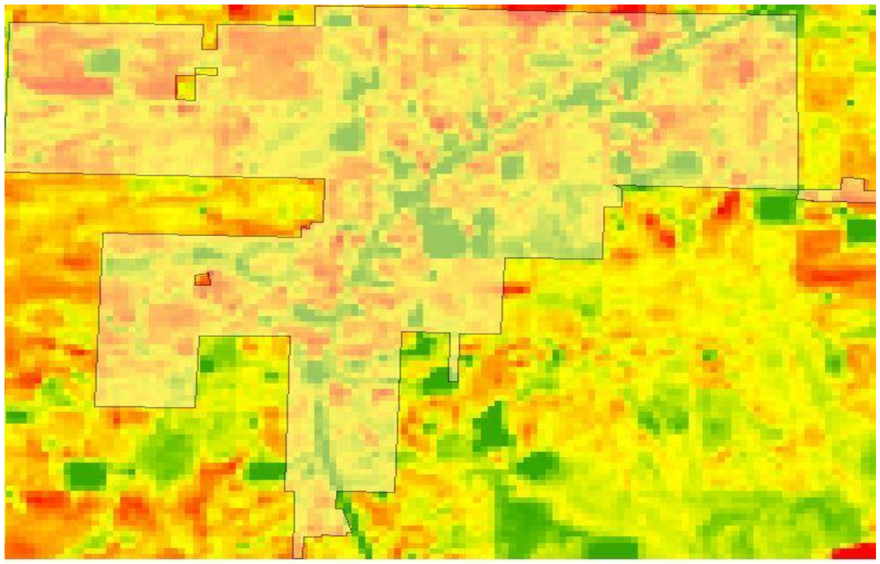


Figure 4: Simple workflow's final output showing calculated NDVI for Damascus, Arkansas and neighboring area.

The intermediate workflow was developed for exploring the hydrology of the area of Van Buren County Arkansas. This workflow was composed of eight processes. Each process is a hydrologic tool in ArcMap. The intermediate workflow can be observed in Figure 5.

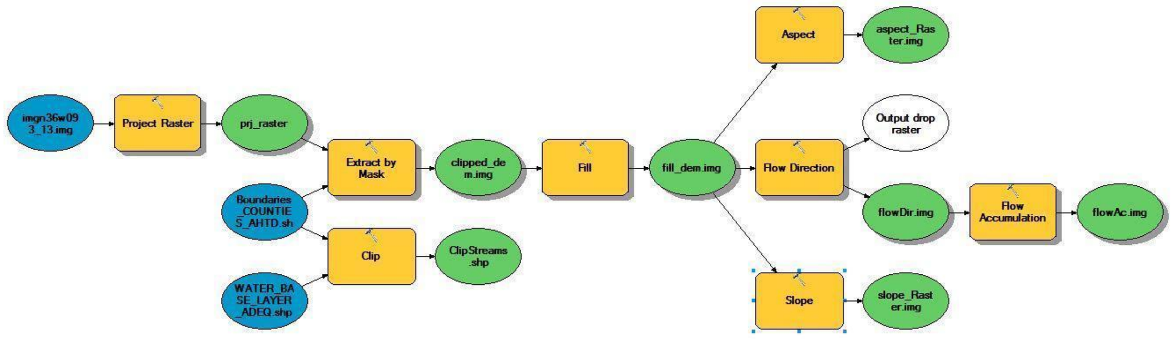


Figure 5: Intermediate Workflow – Hydrologic Modeling composed in ModelBuilder.

Examples of the layers generated from the intermediate workflow are shown below in Figure 6. Slope, aspect, flow direction, and flow accumulation were all calculated.

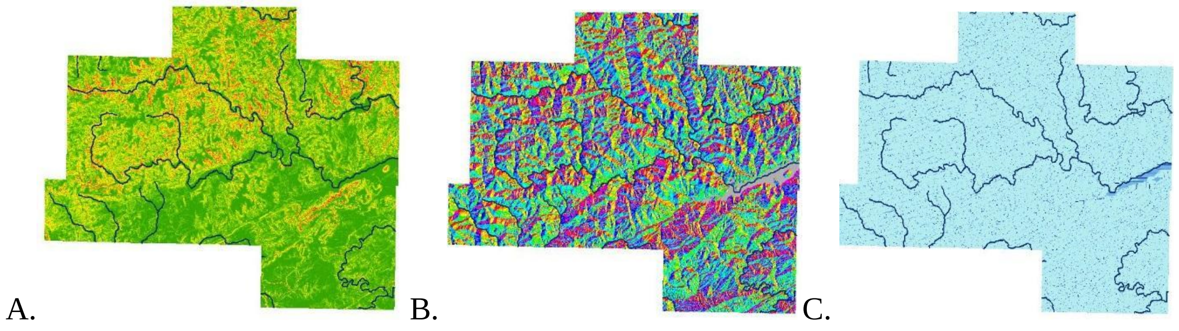


Figure 6: A) Derived and clipped slope layer from intermediate workflow including streams. B) Derived and clipped aspect layer from intermediate workflow including streams. C). Derived and clipped flow accumulation layer from intermediate workflow including streams.

The advanced workflow was composed of sixteen processes. The goal of this workflow was to select the optimal site for a new school. The advanced workflow can be viewed in Figure 7.

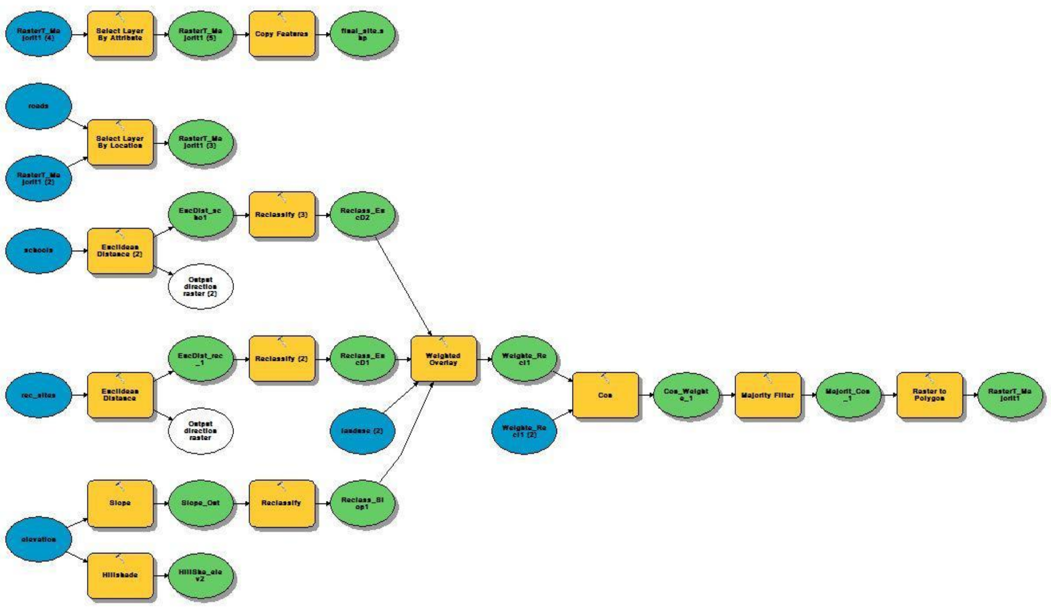


Figure 7: Advanced Workflow – Optimal site selection composed in ModelBuilder

The original model only contained fourteen processes, however when the model was exported as a Python script an additional two processes were added in order for the script to work correctly. Both added processes were the Make Feature Layer tool. The advanced workflow was based off of the spatial analyst tutorial from ArcTutor (ESRI, 2010). The final output of the executed model is shown below in Figure 8.

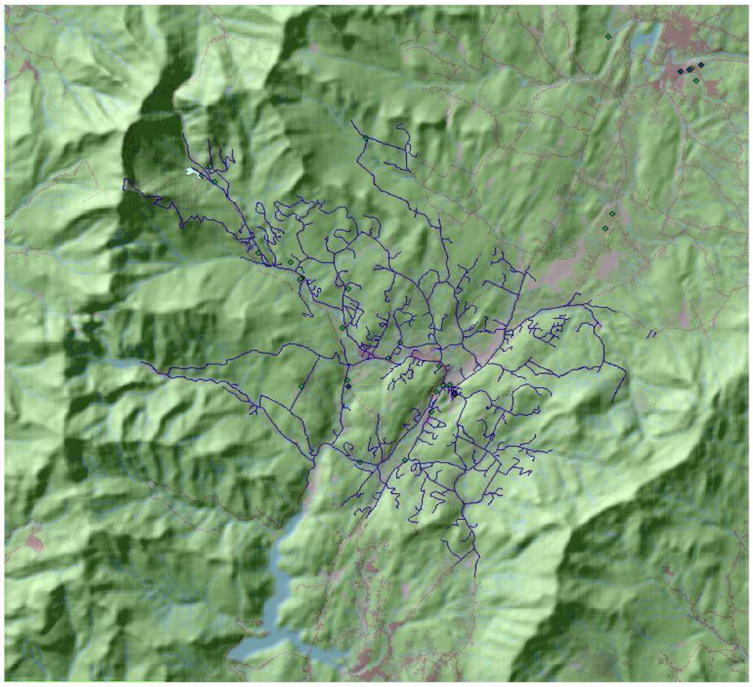
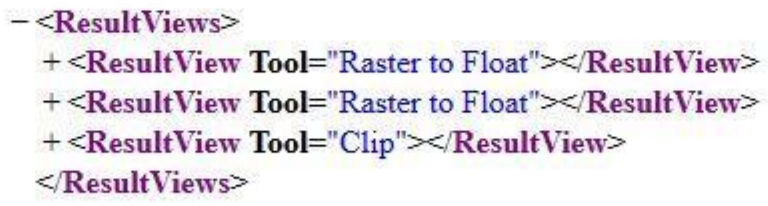


Figure 8: Final output of the advanced model showing the optimal site as light blue.

There are several options for running a model produced in ModelBuilder. The model may be executed by running the entire model at once, running each process individually by clicking, or by a python script. When the entire model is run at once, an XML log file is generated and stored in the ArcToolbox/History directory on the local machine. This log file contains execution history for the model. At first glance, this automatically generated log file may look like enough to collect provenance from, however there is at least one issue with this approach. The Raster Calculator (ArcGIS tool) operations are not stored in this geoprocessing log. If each process is run individually, a separate folder will be created within the working directory, holding the geoprocessing log for each object. Raster calculator does produce a geoprocessing history if executed in this way, however the log is incomplete.

The different locations for the XML processing logs and the lack of completeness, do not make either of these options ideal for collection provenance. An example of a log file that is missing Raster Calculator is shown below in Figure 9.



```
− <ResultViews>
  + <ResultView Tool="Raster to Float"></ResultView>
  + <ResultView Tool="Raster to Float"></ResultView>
  + <ResultView Tool="Clip"></ResultView>
  </ResultViews>
```

Figure 9: Condensed ArcGIS log file showing missing Raster Calculator/Map Algebra

A Python script of each model used can be exported from ModelBuilder. This exported Python script when run can generate the same processing log as if the model was run by manually clicking “Run Entire Model” within ModelBuilder. Special consideration must be given to the generated script in order to make sure it runs correctly, and in customization of the Python script to collect provenance information. The Python scripting option is chosen for this study due to its flexibility and extended functionality through Python libraries.

3.4. Python Script

ModelBuilder's export function was used to generate a Python script. Unfortunately some tools and features that work in ModelBuilder do not automatically work in Python. Examination of the exported script to ensure it is correct must occur. For example, raster calculator is not supported in Python and Map Algebra must be used in its place (ESRI, 2014). Geoprocesses must be stored and added to the Python script in order to generate the same log file as if the workflow is run as a model in ModelBuilder. After a script has successfully run, a detailed processing log can be found in ArcToolbox's history folder. Not every process will be logged however. The generated XML file must be carefully checked to make sure a process has not been left off. For example, Map Algebra will not log as a tool in a python workflow and will be missing from the XML. Due to this, each workflow’s Python script was modified to collect its own provenance.

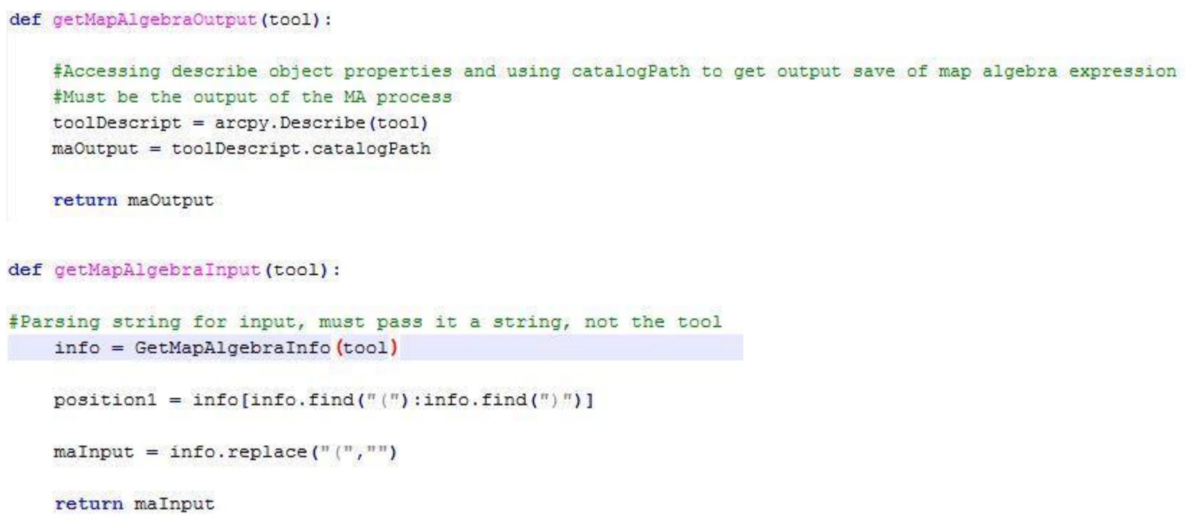
3.5. Provenance Collection

Provenance collected in this study was at the coarse-grained or workflow level. The following were the desired data to be collected: user, tool, input, output, time begun, time ended, source, and unique id. The provenance from the workflow was collected and serialized using XML. In addition to this serialization, examples are shown on how the geospatial provenance collected can also be serialized using XML and following the PROV-XML model, as well as using RDFLib to be serialized into PROV-O. The mapping into PROV-O can be found in Table 1 in Appendix A. Provenance gathered in this study was collected at the semi-automatic level. This means that for certain portions of the script user input or user manipulation is required.

The initial provenance collection began in the ModelBuilder generated Python script. Two classes of ArcPy objects were used, the Describe object and the Result object. The Result object was created by the execution of a geoprocessing tool and contained data about that particular geoprocess (ESRI, 2014). ArcPy’s Describe object contains functions that are useful in determining the file path or type of element used in ArcPy (ESRI, 2014). The idea to use the Describe and Result objects came from Korose’s thesis manuscript (Korose, 2010).

Korose used the Describe and Result objects to collect provenance from a carbon sequestration workflow created in ArcGIS, by passing the collected provenance to a MySQL RDF store using RDFLib. Whereas Korose used RDF and the Open Provenance Model, this study used the XML serialization of PROV-XML recommendation. In addition, PROV-O was used as an example of how to show provenance in an RDF serialization. However, most of the actual processing of the data was done using the PROV-XML format. This was done because the LXML Python library for use with XML was easy to learn and implement.

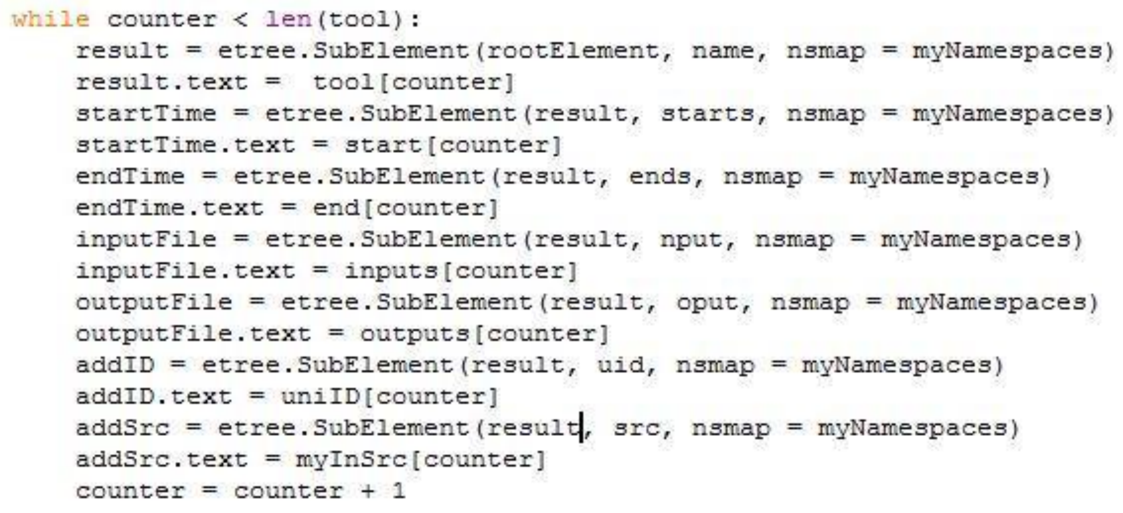
Functions were created in Python to parse the Result and Describe objects generated by the execution of the Python scripts. In order to parse each tool, the line of code was converted into a string in the next line. If a tool was not logged as a geoprocessing object, i.e. Map Algebra, its provenance was collected by using functions specific to that tool. This was a semi-automatic process done by calling the function before each tool in the workflow.



```
def getMapAlgebraOutput(tool):

    #Accessing describe object properties and using catalogPath to get output save of map algebra expression
    #Must be the output of the MA process
    toolDescript = arcpy.Describe(tool)
    maOutput = toolDescript.catalogPath

    return maOutput

def getMapAlgebraInput(tool):

#Parsing string for input, must pass it a string, not the tool
    info = GetMapAlgebraInfo(tool)

    position1 = info[info.find("("):info.find(")")]

    maInput = info.replace("(","")

    return maInput
```

Figure 10: Functions written to collect provenance for non-geoprocessing tasks in ArcPy. The item returned by the function was then stored in a list which corresponded to the type of provenance it contained, i.e. input, output, start time, and so on.

Once the provenance data was stored in its corresponding list, three functions may be called depending on the desired serialization of the workflow’s provenance; generateXML, generateProvXML, and generateRDF. The generateXML function serialized provenance into XML, similar to ArcGIS’s log file. The generateProvXML serialized provenance into XML with defined namespaces.



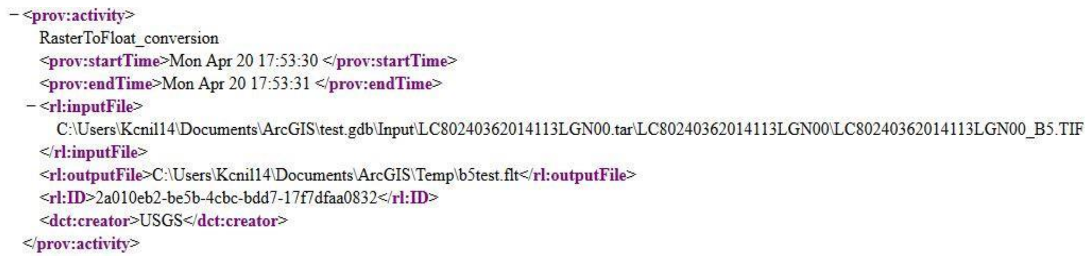
```
while counter < len(tool):
    result = etree.SubElement(rootElement, name, nsmap = myNamespaces)
    result.text =  tool[counter]
    startTime = etree.SubElement(result, starts, nsmap = myNamespaces)
    startTime.text = start[counter]
    endTime = etree.SubElement(result, ends, nsmap = myNamespaces)
    endTime.text = end[counter]
    inputFile = etree.SubElement(result, nput, nsmap = myNamespaces)
    inputFile.text = inputs[counter]
    outputFile = etree.SubElement(result, oput, nsmap = myNamespaces)
    outputFile.text = outputs[counter]
    addID = etree.SubElement(result, uid, nsmap = myNamespaces)
    addID.text = uniID[counter]
    addSrc = etree.SubElement(result, src, nsmap = myNamespaces)
    addSrc.text = myInSrc[counter]
    counter = counter + 1
```

Figure 11: A portion of the function generateProvXML().

The LXML library allowed for the use of Python to generate or parse XML, HTML, and XSLT (lxml, 2015). The ElementTree module was imported in the Python script. The Element object contained the root node for the XML document and the SubElement object contained subnodes. A while loop was created to iterate over each item in the tools list, thus creating a subelement which contained the provenance for each process. The datetime module was imported and used to name each XML file. This ensured that each provenance file had a unique file name that also refers to the date and time of its creation. The file was then written to the working directory as XML.

3.6. Provenance as PROV-XML and PROV-O

The function for generating PROV-XML is similar to the function generateXML. The generateProvXML function included the addition of a dictionary to hold the uniform resource identifiers (URI). A URI was used to uniquely identify the location or name of a resource (W3C, 2001). The QName class in LXML was used to create a qualified XML namespace (lxml, 2015). The namespace ‘RL’ was defined to hold terms specific to GIS, such as input layer and output layer. This URI is not resolvable on the web, however if so desired could be using HTTP. Dublin Core vocabulary is also used to describe the provenance data created.



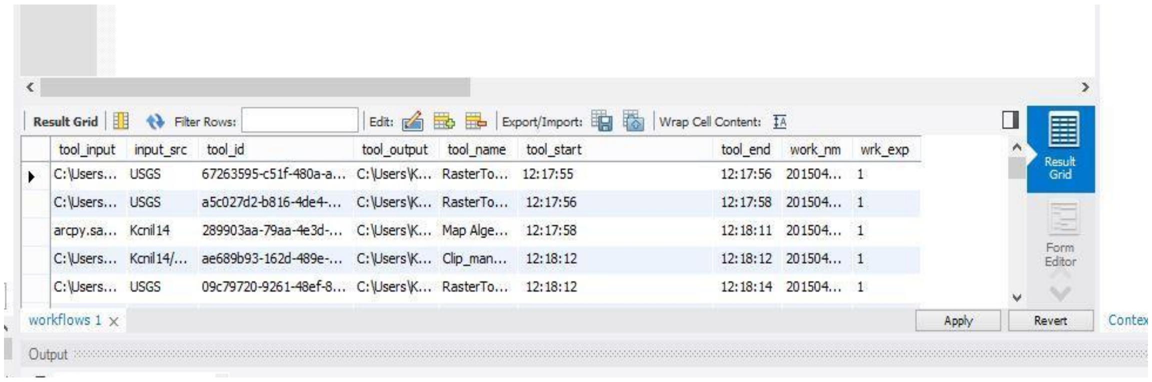
```
- <prov:activity>
    RasterToFloat_conversion
    <prov:startTime>Mon Apr 20 17:53:30 </prov:startTime>
    <prov:endTime>Mon Apr 20 17:53:31 </prov:endTime>
  - <rl:inputFile>
      C:\Users\Kcnil14\Documents\ArcGIS\test.gdb\Input\LC80240362014113LGN00.tar\LC80240362014113LGN00\LC80240362014113LGN00_B5.TIF
    </rl:inputFile>
    <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Temp\b5test.flt</rl:outputFile>
    <rl:ID>2a010eb2-be5b-4cbc-bdd7-17f7dfaa0832</rl:ID>
    <dct:creator>USGS</dct:creator>
  </prov:activity>
```

Figure 12: XML document following PROV-XML specifications.

RDF/XML was created by adding the rdf:RDF namespace tag and the rdf:Description tag for each subelement. For this study, the change was done manually however it could easily be accomplished using LXML or RDFLib. When the RDF tags have been added, RDFLib can then be used to parse and serialize the RDF/XML (PROV-XML) into other serializations of RDF such as N3 or Turtle (PROV-O).

3.7. Provenance Storage

During the generation of provenance as an XML file from the generateXML function, a .csv file was also generated with the csvBatch() function. This function collected the provenance as a comma separated values file that was then uploaded to Excel for data cleanup. Once the data was cleaned up in Excel it was batch loaded into a MySQL database for query. An additional column value was added for the database called wrk_exp. This category recorded the workflow level each process belonged to. The simple workflow corresponded to 1, the intermediate to 2, and the advanced to 3.



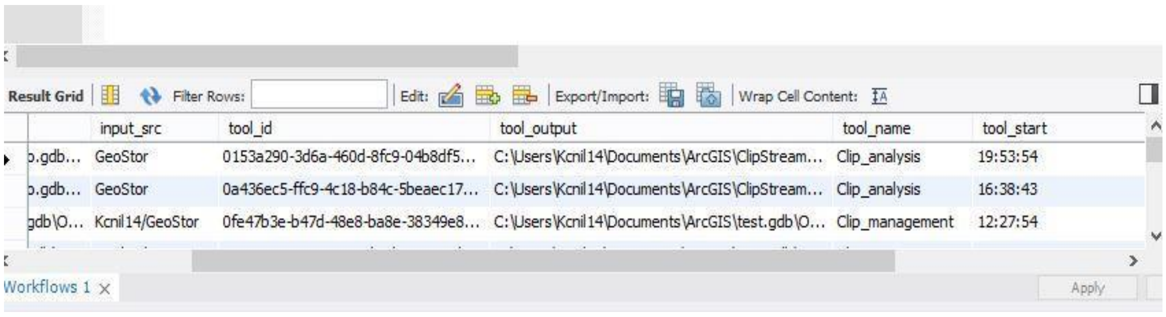
tool_input	input_src	tool_id	tool_output	tool_name	tool_start	tool_end	work_nm	wrk_exp
C:\Users...	USGS	67263595-c51f-480a-a...	C:\Users\K...	RasterTo...	12:17:55	12:17:56	201504...	1
C:\Users...	USGS	a5c027d2-b816-4de4-...	C:\Users\K...	RasterTo...	12:17:56	12:17:58	201504...	1
arcpy.sa...	Kcnil14	289903aa-79aa-4e3d-...	C:\Users\K...	Map Alge...	12:17:58	12:18:11	201504...	1
C:\Users...	Kcnil14/...	ae689b93-162d-489e-...	C:\Users\K...	Clip_man...	12:18:12	12:18:12	201504...	1
C:\Users...	USGS	09c79720-9261-48ef-8...	C:\Users\K...	RasterTo...	12:18:12	12:18:14	201504...	1

Figure 13: MySQL database for querying provenance

The provenance was queried by using SQL. For example if a user wanted to discover which workflows used the Clip tool, this could be done with the following statement:

```
SELECT * FROM Workflows
WHERE tool_name LIKE '%clip%'
```

This statement returns every instance of a tool name that contains clip. In the case of this database it returns both clip analysis and clip management.



	input_src	tool_id	tool_output	tool_name	tool_start
).gdb...	GeoStor	0153a290-3d6a-460d-8fc9-04b8df5...	C:\Users\Kcnil14\Documents\ArcGIS\ClipStream...	Clip_analysis	19:53:54
).gdb...	GeoStor	0a436ec5-ffc9-4c18-b84c-5beaec17...	C:\Users\Kcnil14\Documents\ArcGIS\ClipStream...	Clip_analysis	16:38:43
gdb\O...	Kcnil14/GeoStor	0fe47b3e-b47d-48e8-ba8e-38349e8...	C:\Users\Kcnil14\Documents\ArcGIS\test.gdb\O...	Clip_management	12:27:54

Figure 14: SQL query results from the provenance database

Additional tables could then be created to store data that can be used to determine a workflow's trust.

3.8. Trust Score Assignment

The second part of this thesis used the provenance collected to calculate a trust measurement from each example workflow. When examining trust it is important to define which type of trust will be used. This study focused on measuring two types of trust relevant to workflows, content trust and workflow trust. Content trust is the trust in the information or content of a given resource that is located either online mapped by a URI or offline in possession of the user (Gil & Ratnakar, 2002). A ModelBuilder workflow's content can come from online via HTTP or FTP download or offline, via user generated data. For each step in the workflow, the shapefiles and imagery used as inputs were evaluated. In addition to shapefiles and imagery, feature layers, tables, intermediate layers, or anything used as an input for a particular service in a workflow was evaluated for content trust. Overall content trust for a workflow was also evaluated. In the case of content trust, the performance of the tools comprising a workflow is not evaluated, only the data used in a workflow and the selection of tools used in the workflow. The measure of content trust is therefore more subjective in nature. Content trust is determined based upon a user's perceptions of the content. Therefore, different users could evaluate the same data item and obtain two very different content trust scores.

Workflow trust was evaluated differently compared to content trust. Workflow trust can be defined as a measure of trust for each tool or service composing the workflow, approximated by the execution status, availability, and reliability of each tool (Naseri, 2013). Workflow trust in this aspect was not focused on the content comprising the workflow, but rather was concerned with the trustworthiness that the workflow will run in an efficient and predictable manner. Workflow trust does not take into account the content of a workflow. It is only concerned with how the tools or services in a workflow perform. The tools used in one workflow can also be used in additional workflows. The workflow trust score takes into account all workflows that use a particular tool or service. Workflow trust is more objective in its measure. It is based on the performance of the tools in a workflow and does not consider users' perceptions in its evaluation.

3.8.1. Results of Content Trust Evaluation

When evaluating content trust, a metric must be created to evaluate the workflow. Authority, provenance, bias, and related resources have been used to evaluate content trust for documents available on the web accessed via a search engine (Gil & Artz, 2007). Authority can be defined as the prominence of the workflow or data item's creator. For example, a data layer published by the USGS would have a higher authority than one published by a first year student of GIS. Bias can be defined as the degree in which a workflow or data item has an underlying agenda or skews content to achieve a desired result. Bias requires a knowledge of the subject matter to detect properly (Gil & Artz, 2007). Related resources were not used in this study, as it is difficult to determine related resources to geographic data. Provenance was also not explicitly included as an evaluation factor, as every workflow instance included provenance in this study. However, factors such as authority and bias were determined from the provenance for the content trust of the workflow steps. This study also differed in that the content trust was only based on the evaluation of one user's perceptions, while Gil & Artz examined content trust from the perception of multiple users. In addition to authority and bias, which can be used for content of any type, factors to evaluate the unique nature of geospatial data were considered. Geospatial data is unique in that it has spatial and temporal components. In order to evaluate this, the quality of geospatial data was also used to evaluate content trust.

Malaverri identified several factors in her dissertation that can be used to evaluate quality for geographic data based on FDGC standards in conjunction with provenance. Of these, the inclusion of metadata, spatial accuracy of the data and completeness of the data were used to approximate quality in this study. To account for the temporality of geospatial data, it was assumed that the data were of accurate temporal resolution for the user's needs, as it was included in the workflow. The quality factors used in this study can be seen in Table 2. In order to simplify the evaluation of content trust, only two outcomes were considered, trustworthy or untrustworthy. For the content trust of a workflow to be trustworthy it was required to be in the range from [.5,1] which is categorized as the discrete category (*T*), and from [0,.49] for untrustworthy represented by the discrete category (*U*). The factors that determine content trust and their criteria for trustworthy and untrustworthy can be viewed in Table 1.

Content Trust Level	**Bias**	**Quality**	**Authority**
Trustworthy (T)	Limited to no bias is perceived in the content of the data.	Quality of the data is included and meets user needs.	Source of the data is well respected and authoritative in the field.
Untrustworthy (U)	Content of data is clearly perceived as bias.	Quality of the data is not sufficient for user needs or no quality information is included.	No source information included with data or source is not well respected or known.

Table 1: Parameters for the evaluation of content trust

Since quality was determined from several factors, the mean of all of the factors classified as representing quality was used to determine the value for overall quality.

This can be shown in equation (4).

$$\bar{x} = \frac{\sum_{i=1}^{n} xi}{N}$$

Equation 4: Equation for finding the mean

Where x̄ was the mean score for the quality parameters, $\sum_{i=1}^{n} xi$ was the sum of each quality parameter and *N* was the total number of quality parameters. If x̄ was between [.5,1], then that component in the workflow was classified as high quality (*H*) , if x̄ was between [0,.49], then the step was classified as low quality (*L*).

Quality Level	**Metadata**	**Spatial Accuracy**	**Completeness**	**Workflow**
Low quality	No metadata is included or there is only limited metadata.	No spatial accuracy is included or data has low spatial accuracy.	Data is incomplete.	Incorrect analysis used to obtain derived product.
High quality	Metadata includes standard data or meets FDGC standards.	Data has medium to high spatial accuracy.	Data is inclusive of all necessary components.	Traditional analysis methods used to obtain derived product.

Table 2: Parameters for the evaluation of quality.

In the case of the spatial accuracy parameter, if the resource being evaluated did not contain a spatial component, for example the resource was a joined Excel table, the resource could be evaluated for general accuracy instead. One instance of each workflow was evaluated for each example and a content trust score was assigned. The quality parameter for workflow is not evaluated for every data item used in the workflow. It is only evaluated once based on the overall workflow, as can be seen in Equation 6. The workflow quality parameter is also not a reflection of workflow trust. Gil used a weight and normalizing value to integrate the annotations and sources for each user in her study. Content trust was allowed to have a negative value up to -1, which was reflected in the equation she used to compute content trust. The content trust score for this study did not include negative values, therefore the mean equation was used to determine overall content trust for each workflow.

The high and low categories were mapped to the function *f(c)*, where *c* was the parameter being examined. This is shown in equation (5).

$$\mathrm{f(c)} = \{high = 1\ low = 0\ \}$$

Equation 5: Mapping of high and low categories

If a resource was placed in the low category it was given a null value in the overall count. If it was placed in the high category it was assigned the value of 1.

The following example shows how content trust was evaluated for the simple workflow. The workflow instance, 20150425122520 hereafter called W1, was chosen at random by using the sample command in R. This workflow's name was in the format year, month, day, hour, minute, second. W1 was composed of data from two sources, the USGS and GeoStor (which is now known as Arkansas GIS Office's data portal). The first step was to examine the data layers used, which were band 4 and band 5 from Landsat 8, as well as a vector layer of a county from GeoStor. The following table shows the evaluation of the data resources used in W1.

Resource	**Metadata**	**Spatial Accuracy**	**Completeness**
LC80240362014113LGN00_B5.TIF	High	High	High
LC80240362014113LGN00_B4.TIF	High	High	High
ADMIN_DBO_CITY_LIMITS_AHTD	Low	Low	High

Table 3: Quality evaluation for data layers

Although GeoStor is a trustworthy provider of GIS data for Arkansas, this particular resource did not have associated metadata included with it as a download or online. Spatial accuracy was not included either for the vector layer file. The two raster data layers are from a calibrated data product which has been corrected for distortion using digital elevation models (DEM) and ground control points (NASA, 2011; USGS, 2014). The raster .TIFF downloaded for the Landsat data included metadata. It was assumed that the data products were complete as all products downloaded successfully and were operational in ArcMap. The workflow used traditional methods for calculating NDVI which resulted in the high category assignment. The provenance data was used to view the method for calculating NDVI. To calculate the quality score for W1, the mean score was calculated yielding .83 as seen in equation (6), which would place the quality in the high category.

$$q = (2/3+2/3+3/3+1/1)/4 = 3.33/4=.83$$

Equation 6: Arithmetic mean being calculated for quality of workflow

Once the quality of a workflow was calculated, the other portions of content trust were also determined. The bias of the workflow was determined based on the perceived bias of the data layers used as well as the perceived bias of the workflow's constructor. Table 4 shows the evaluation for the workflow's bias.

Resource	Bias
LC80240362014113LGN00_B5.TIF	No perception of bias
LC80240362014113LGN00_B4.TIF	No perception of bias
ADMIN_DBO_CITY_LIMITS_AHTD_polygon.shp	No perception of bias
W1	No perception of bias

Table 4: Evaluation of bias for the content comprising the simple workflow.

From examining the workflow, metadata and the data layers themselves, there was no indication that the workflow was biased. An evaluation of W1 was also included for bias in order to account for the possibility that it was introduced by parameters or environment variables set by the user. To calculate the bias, b, the arithmetic mean was used for each resource being evaluated as shown in equation (7).

$$b=(1+1+1+1)/4=1$$

Equation 7: Bias calculation for simple workflow.

This placed W1 in the high category, meaning that there was little to no perceived bias in the data. Authority was evaluated using the same method as quality and bias. Authority was based on the data resources' provider. In this case, there were three data source providers: USGS, GeoStor, and the workflow's creator. The data providers and their corresponding authority score can be seen in Table 5.

Source	Authority
USGS	High
GeoStor	High
Workflow Creator	Low

Table 5: Evaluation of authority for workflow.

USGS and GeoStor are ranked high in authority, as they are both well-known and reputable providers of geospatial data. The workflow creator was given a low score. Although the workflow included provenance, the workflow creator was not well known, therefore was scored in the low category. To derive the score for authority, a, equation (8) was used:

$$a = (1+1+0)/3 = .67$$

Equation 8: Calculation of authority for the simple workflow.

Although this authority score appears relatively low, it was still enough to be classified as high authority. To derive the overall content trust score, equation (9) was used:

$$t = (a + b + q)/n$$

Equation 9: Overall content trust.

Where *t* was the content trust score for the workflow, *a* was the authority value, *b* was the bias value, *q* the quality value, and *n* the number of parameters that were evaluated. To calculate the overall content trust, equation (10) was used.

$$t = (.67 + 1 + .83)/3 = 2.5/3 = .83$$

Equation 10: Overall content trust for simple workflow.

A score of .83 placed the workflow’s content as trustworthy. Workflows of the intermediate and advanced level were evaluated the same way as this example. The above method provided a way to calculate content trust for a geospatial workflow. This method could be improved by implementing an automated way to calculate a workflow’s content trust, instead of the manual calculation of this example. Content trust for a workflow was dependent upon the data layers used in its creation. Therefore, a workflow using the same four tools as the simple workflow, but using different data as inputs, might receive a completely different content trust score. Further extension of geospatial workflow content trust could include additional quality parameters, as well as examining the correlation between differing input layers and user evaluations for a workflow. Content trust evaluation for the intermediate and advanced workflows can be found in Table 1 of Appendix C.

3.8.2. Results of Workflow Trust Evaluation

In addition to content trust, workflow trust was also examined for a geospatial workflow. Workflow trust measures the degree of trust that can be placed in the workflow's tools or services executing in a successful and timely manner. In order to approximate a measure of workflow trust, parameters such as reputation, execution time, cost, and availability may be used to derive the quality of each service comprising the workflow (Naseri, 2013).

This study made use of the quality parameters of execution time, execution status, and availability. Execution time was the amount of time a tool in the workflow took to execute. Execution status represented if each tool in the workflow successfully completed its function. Both can serve as an approximation of the reliability of each tool (Naseri, 2013). Availability referred to the availability of the data processed during the workflow's execution. It can also refer to the availability of each tool the workflow was composed of, however since all tools were available locally, this aspect was not considered. Execution status took the values of 0 or 1. 0 was given for a tool that did not finish executing, while 1 was assigned for a tool that was successful in execution.

Reliability was based on the time it took each tool in the workflow to finish running. For each tool in the workflow the mean value was calculated using the summary command in R, for the elapsed time it took the tool to finish processing. Values less than or equal to the mean time were mapped to trustworthy, while values greater than the mean time were mapped to the value untrustworthy. Availability of the data and execution time values ranged from [0,1] which were then mapped to the intervals low quality for values between [0,.49] and high quality for values between [.50,1]. The quality parameters were then mapped to untrustworthy (*U)* for low quality and unsuccessful execution or trustworthy (*T*) for high quality and successful execution. Table 6 shows the estimated corresponding trust decision for each.

Trust Level	**Execution Status**	**Reliability**	**Availability**
U	0	L	L
U	1	L	L
U	0	L	H
U	1	L	H
U	0	H	L
U	1	H	L
U	0	H	H
T	1	H	H

Table 6: Quality parameters and their corresponding level of trust.

While Naserri divided his study into three trust levels; low, medium, and high, this study only made use of two levels or states of trust, untrustworthy and trustworthy. This was done to reduce the complexity of the model. Since there was no medium level of trust, a value of trustworthy was only assigned to those workflows with all quality parameters achieving the highest level. The assignment of trust levels based on execution status, reliability, and availability for each workflow can be found in Appendix D.

4. Implementing the Hidden Markov Model

The third part of this study was to determine if a Hidden Markov Model could be used to estimate workflow trust for a given workflow. The first step was to define the components of the HMM. A HMM has five parts: the hidden states, the observations, the initial state probabilities, the state transition matrix, and the observation matrix (Rabiner, 1989).

The model for this study had two hidden states, untrustworthy and trustworthy, where S = $\{S_{1,}S_{2\ldots}S_t\}$ with the state at a given time *t* represented by q_t. The observations, represented by *O* = $\{O_1,O_2,\ldots O_t\}$, corresponded to the tool composing the workflow that was observed at time t. Both the hidden states and the observation states were discrete, with the set of *S* equal {T,U} and the set of *O* equal to {raster to float,clip,map algebra}for the simple workflow. For the intermediate and advanced workflows the set of *O* corresponded to the tools used in that particular workflow. The probability matrices can be solved using several methods such as estimation from collected data (Naseri, 2013), maximum likelihood (ML), or expectation maximization (EM) using the Baum-Welch algorithm (Cappé et al., 2005). Estimation from collected data was used in this study. The initial state probability matrix was set as π = [.5,.5] giving an equally likely chance for the workflow to start in either hidden state.

4.1. Estimation of State Transition Matrix Using Provenance

The state transition probability matrix, A, gives the probabilities for all of the possible transitions between hidden states. This can be represented in equation (15).

$$A = \{a_{ij\}} = P[q_{t+1} = S_j|q_t = S_i],\ 1 \leq i,j \leq N \text{ (Rabiner, 1989).}$$

Equation 15: Computing the transition from S_i at time t, to S_j at time t+1

Solving the state transition probability matrix yields a stochastic matrix whose row probabilities sum to 1. For this study the state transition matrix took the form of a 2x2 matrix as follows:

$$a_{ij} = \begin{array}{c c} & \begin{array}{cc} T & U \end{array} \\ \begin{array}{c} T \\ U \end{array} & \begin{bmatrix} P(S1) & 1 - P(S1) \\ 1 - P(S2) & P(S2) \end{bmatrix} \end{array}$$

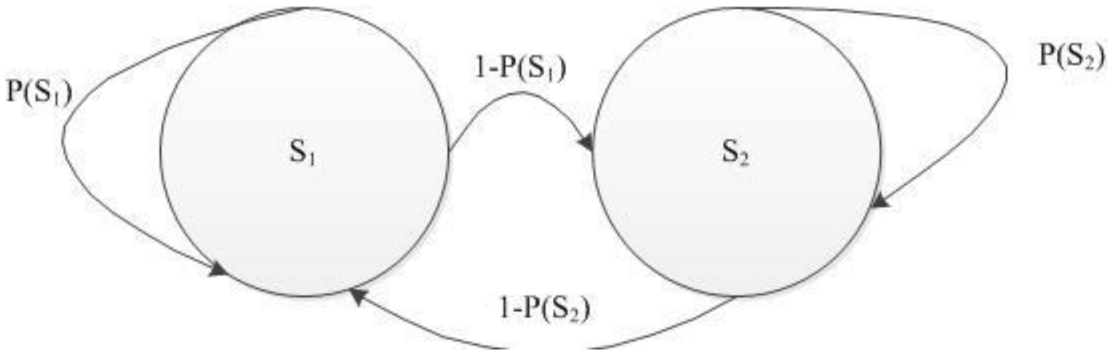


Figure 15: State transition in matrix and DAG form for study.

In order to estimate the state transition matrix, the quality parameters were mapped to the corresponding trust level in Table 6. This was done for each step in the workflow. Each workflow was executed a total of 50 times. To estimate the transition probabilities, the states' conditional probability distribution was used as shown in equation (17). In order to find the conditional probability of transitioning to state *j* at time *t*, given that at time *t-1* the model was in state *i* from the provenance data was be found by calculating for each step in the workflow, the transitions from S_i to S_j divided by the total number of transitions from S_i for the entire model. This can be represented in Equation 16.

$$\frac{\mathrm{nSiSj}}{\mathrm{nSi}}$$

Equation 16: Estimation of state transition matrix (Naseri, 2013).

When calculating this for the simple workflow, the transition counts can be shown in Table 7.

	(S_1,S_2)	**(S_2,S_3)**	**(S_3,S_4)**	**Total**
T to U	0	18	21	39
U to T	37	1	4	42
T to T	12	31	11	54
U to U	1	0	14	15
Total	50	50	50	150

Table 7: Transition counts for simple workflow

In addition to finding the transition counts, the total amount of time each step in the workflow was either *T* or *U* was also counted. This is shown in the Table 8 below.

	O_1	**O_2**	**O_3**	**Total**
T	61	32	15	108
U	39	18	35	92
Total	100	50	50	200

Table 8: Table displaying hidden state counts at a particular observation for simple workflow.

Using the above tables the transition probabilities can be calculated easily. For example, to find the probability of transitioning from state *T* to *U*, the following calculation was performed in equation (17).

$$P[q_t = S_j|q_t\text{-}1= S_i] = ((0/.24)+(.36/.98)+(.42/.64))/3 = .34$$

Equation 17: Calculation of transition from state T to state U.

S_i was the hidden state *T* transitioning to the hidden state *U*, S_j. After using equation 17 to calculate for each transition, the state transition matrix (*A*) was produced.

$$A = \begin{array}{cc} & \begin{array}{cc} T & U \end{array} \\ \begin{array}{c} T \\ U \end{array} & \begin{bmatrix} .66 & .34 \\ .73 & .27 \end{bmatrix} \end{array}$$

Figure 16: State transition matrix estimate from geospatial provenance.

Once the state transition matrix was estimated, the observation matrix (*B*) was calculated in order to gather all of the parameters necessary for the Hidden Markov model.

4.2. Estimation of Observation Matrix Using Provenance

The observation matrix, *B*, is a stochastic matrix that gives the conditional probability of observing observation O_k at a particular time t, given that the model is in hidden state S_j. This can be shown as equation (18).

$$B = \{b_j(k)\} = P[O_{kt}|q_t=S_j],\ 1 \leq j \leq N;\ 1 \leq k \leq M \text{ (Rabiner, 1989).}$$

Equation 18: Conditional probability for an observation while in a particular hidden state.

Given a series of observations, in this case workflows, the observation matrix was estimated using the collected provenance. First, the total number of transitions from hidden state T to hidden state U, and hidden state U to T was counted. Next, the transition from each step in the workflow, for example (S_1,S_2), that transitions to T were calculated. This was done for each step in the workflow. The observation probability was calculated by using equation (19), where n_{stj} was the number of transitions to S_j for the step in the workflow being examined, and n_j was the total number of times state $q_t{=}_j$ occurred.

$$\frac{nstj}{nj}$$

Equation 19: Equation for finding $P(O_t=o_t|q_t=S_j)$, (Nasserri, 2013).

Using Table 8 this was easily calculated. For example, to calculate the probability of being in hidden state T and observing the observations produced at o_1 at a given time $_t$, the following calculation in equation (20) can be performed:

$$P[o_{kt} \mid q_t = S_i = \frac{61}{108} \approx .56$$

Equation 20: Observation probability calculation of observing a particular observation o_k at time $_t$ and being in the hidden state T.

For the simple workflow, the following observation matrix was produced:

$$\begin{array}{cc} & \begin{array}{ccc} O1 & O2 & O3 \end{array} \\ \begin{array}{c} T \\ U \end{array} & \begin{bmatrix} .56 & .30 & .14 \\ .42 & .20 & .38 \end{bmatrix} \end{array}$$

Figure 17: Observation matrix B, for simple workflow.

The models estimated from the provenance data for the intermediate and high complexity workflows can be seen in Appendix D. Having estimated all three parameters from the provenance data allowed for the Hidden Markov model λ=[A,B, π] to be applied further.

4.3. Application of Hidden Markov Model

Once the parameters for a HMM have been estimated or randomly generated, they can be used to solve three types of problems: the evaluation problem, the state sequence, problem, and the learning optimization problem. This study focused on using HMMs to solve what the current hidden state is given a series of observations (decoding problem). The first problem is solved by using filtering to compute $P(_t|o_1...o_t)$. Filtering can be defined as recursively computing the posterior distribution for a hidden state given all of the observations so far and is shown in equation 21 (Russell & Stuart, 2003).

$$P(q_t+1|o_1...o_{t+1}) \propto \alpha(o_{t+1}|q_{t+1}) \sum P(q_{t+1}|q_t)P(q_t|o_1\ldots o_t) \propto \alpha \text{ Forward}(f_{1..t},o_{t+1})$$

Equation 21: Forward algorithm for solving probability of the hidden states given observations (Russell & Stuart, 2003).

The forward algorithm shown in equation (20) was used to help solve the filtering problem. For this study, the forward algorithm was implemented in R using the HMM package by Himmelman for discrete time and space HMMs (Himmelmann, 2010). The forward algorithm has been used to help solve the filtering problem for workflow trust in previous study (Naseri, 2010). In this case it was tested on workflows in the geospatial context. For the observation sequence of the simple workflow (Raster to Float, Raster to Float, Map Algebra, and Clip Management), the following probabilities shown in Table 9 were generated with the forward algorithm.

	t_1	t_2	t_3	t_4
T	.28	.189336	0.05146029	0.006422829
U	.21	.063798	0.01631994	0.008323095

Table 9: Forward probabilities given λ and an observation set.

The forward probabilities can then be used to estimate $p(x_t|o_t)$. To do this the following equation, equation 21, was used.

$$P(qt = S_i|o1...ot) = \alpha_t(S1)/ \sum_i \alpha_t(i)$$

Equation 21: Conditional probability of observing sequence o_1 through o_t and being in state S_i (Allen, 2003).

Using this equation, the following probabilities were generated and are shown in Table 10.

	o_1t_1	o_2t_2	o_3t_3	o_4t_4
T	.57	.75	.76	.44
U	.43	.25	.24	.56

Table 10: Decoded forward probabilities for the simple workflow.

From the following table it can be interpreted as, if Raster to Float was observed at time t_1, then there was a greater probability that it was trustworthy (.57) compared to untrustworthy (.43). Continuing for each time step, at t_2 there was a greater probability (.75) that at this step in the workflow, the workflow was considered trustworthy. As the model progressed from observation 2 to observation 3, it still retained a higher probability of being trustworthy (.76), compared to untrustworthy (.24). However, as the model moved into observation 4, the probability of it being trustworthy markedly dropped from .76 to .44. This means that as the model progressed it moved through the trust states as: T, T, T, and U. Judging from the table, by the time the workflow ended, the result was no longer trustworthy.

Solving the filtering problem by using the forward probabilities can be thought of as follows. The forward probability, α, serves as the value of the sum of the hidden states during each earlier time in the workflow, this partial probability is then forwarded along as t moves forward in time (Blunson, 2004). Once the forward probabilities for each time step have been calculated, they can then be decoded using equation (21), thereby completing the filtering problem. In addition to the forward probabilities, the backward probabilities may also be calculated and used in solving the filtering problem. However in this case, the backwards probabilities were not used as the conditional probability being computed was $P(qt = S_i|o_1...o_t)$, instead of $P(qt = S_i|o1...O_T)$ (Smola, 2015).

In addition to being used to solve the decoding problem, the forward probabilities can also be used to evaluate the probability that a given Hidden Markov model produced the observations. By taking the sum of α, over each time step, the resulting probability can be used to determine if the model is a good fit for the observations. Using the forward probability in this way, can allow the HMM that best corresponds with the observations to be selected if there are multiple models to choose from (Rabiner, 1989).

5. Discussion and Conclusion

This study examined three research questions: (1) Can geospatial provenance be collected in such a way that it is useful to other applications?, (2) Can a measure of trust be manually calculated for a given workflow?, (3) Can a Hidden Markov model be used to predict workflow trust? Each research question is examined below.

The conclusion for the first question was that it was possible to collect geospatial provenance in a way that was useful to additional applications. The Result and Describe ArcPy objects allowed for their parameters to be used within a provenance record (Korose, 2010). Serializing provenance in a format such as XML or RDF allowed for provenance interchange to easily take place and for the provenance to be used between various applications. In this case XML is sufficient for use, however if the goal of provenance was to use it as linked data, RDF would be the desired format. In addition to using XML, the .csv file allowed for the easy use of transferring the collected geospatial provenance in Excel, as well as R. By following the recommendation of the PROV data model, including PROV-XML and PROV-O, the collected provenance follows the recommendations for use on the semantic web (Moreau, 2013).

The provenance collected for this study did have some shortcomings. There was no automatic collection of the provenance data. Semi-automatic provenance collection occurred during the execution of the Python script. Another improvement could be to create a main function, that way there would only need to be one function call, instead of several. The Python scripting used in this study is not elegant and is rather verbose. By creating a main function and rewriting code for more efficiency, this issue should be resolved. Another issue was the cleanup of the collected geospatial provenance. The provenance generated required cleanup before it could be inserted in MySQL. This cleanup was done for two reasons. The first was to make sure that the value from the collected provenance being inserted into a particular column, matched the data type and size constraints of the column. The second reason was to remove unnecessary characters or strings in the collected data. For example, because of the type of *datetime* module used to collected the start and end time for a given tool, the day of the week, the month, and date, are collected as well. This was redundant due to the month, day, and year already being included in the workflow file name. This could be improved by creating functions to automatically clean the data in such a way that it can be directly inserted into the database. Additional serializations of geospatial provenance could be incorporated depending on user needs.

For the second research question, a measure of content trust can be calculated for a geospatial workflow. By using the provenance collected from the first research question, and the accompanying file metadata or lack of, a content trust score was calculated. The intermediate workflow had a content trust score of .71 and the advanced workflow a content trust score of .64. The content trust score in this study was the average of the authority, bias, and quality for each data item examined in a workflow. These parameters were consistent with parameter used in other studies (Gil & Ratnakar, 2002; Malaverri, 2012; Malaverri, 2013). In addition to the average, the weighted average could also be used to calculate a content trust score, if one category needed to be emphasized more than another (Gil & Ratnakar, 2002). One way to improve the methods used to calculate the content trust score in this study, would be to incorporate automatic calculation. Another improvement would be to allow for more or less categories to determine content trust by, based on the use case domain or the user needs.

The result for the third question is that a Hidden Markov model can be used to estimate a level of workflow trust for a given geospatial workflow. The forward algorithm can be used to see the hidden trust states if it is used in the decoding process, or to see the probability that a given Hidden Markov model produced a sequence of observations if just the forward probabilities are used. The result successful result obtained using the forward algorithm for the probability of being in a trust state at a given time in a workflow, was consistent with the results found by Naseri who used the same technique (Naseri, 2013) . By using the Hidden Markov models created from the provenance data it was shown that the simple workflow had a sequence of T,T,T,U, the intermediate workflow had a sequence of T,T,T,T,T,T,T,T,T and the advanced workflow had a sequence of U,T,T,T,T,T,T,T,T,T,T,T,T,T,T,T,T. These state sequences are consistent with the state sequences generated by the Viterbi algorithm. The simple workflow has a workflow trust value of .44. The intermediate workflow has a workflow trust value of .73 and the advanced workflow has a workflow trust value of .82.

There are several possible expansions from this study in the use of Hidden Markov models with geospatial data trust. The Baum-Welch algorithm or the Expectation-Maximization (EM) algorithm (Rabiner, 1989) could be used to learn the HMM from the provenance data and this model could be compared with the one calculated from the provenance data counts. The Hidden Markov models could also be varied based on the number of states included in them. For example, a third or fourth hidden state could be added to see if that particular model more accurately reflects the data compared to the two state model. The Hidden Markov model could also have the stationary assumption relaxed, and a non-stationary Hidden Markov model could be used (Naseri, 2013).

The workflows used in this study used tools only available on the local machine. One way to expand this study is to use tools or data that are hosted online, such as Web Feature Services or Web Mapping Services. For desktop application, if a tool used in a workflow is not considered trustworthy, there may not be any options to replace that given tool in the workflow. However if a workflow is composed of online services available on the web, if the workflow has a low trust level at a given time, that particular service might be replaced with a similar service. An additional expansion of this study is to use a database that incorporates and stores geospatial data. Instead of using a MySQL database, a PostGIS database could be used. By using a spatial database, the geographic data the provenance is based on could be displayed to the user. This is similar to the emphasis placed on visualization in provenance stores such as GeoPWProv (Sun, 2013).

An important point of this study was that geospatial provenance, content trust, and workflow trust can be used in conjunction with one another. Content trust deals directly with the data a workflow is composed of, while workflow trust deals with the efficiency and success of a given workflow executing. For example, the advanced workflow had a content trust score of .64, which is on the low end of trustworthy. It had a .82 probability of being in state trustworthy at the end of execution. The workflow consistently had probabilities in the high 90s for time steps t_2 through t_{15} of the workflow. Although the state was still trustworthy, this drop may signal that something unusual is happening. Re-examining the content trust quantities for the output at time t_{16} shows low quality for all three criteria. When the geospatial provenance was examined, it showed a warning automatically generated by ArcGIS stating that an empty output was generated. If the layer was added to ArcGIS, it can be seen in the table of contents, however nothing is displayed in the data frame. This might be something a user would notice if they were manually adding this layer, however if the layer is used automatically within a workflow, the workflow could allow processing to continue. Geospatial provenance, content trust, and workflow trust could be used to prevent instances like what occurred in this use case from happening. Future research could examine more in-depth the relationship between content trust and workflow trust in geospatial workflows.

Finally, it is important to consider the implications of the categorizations of the workflows as simple, intermediate, and advanced. In this study, only one instance of each was tested for both workflow and content trust. To determine if it is proper to evaluate workflow and content trust based on the number of tools or services in the workflow, more instances of workflows classified in this way need to be evaluated. The low workflow trust score of the simple workflow may not be indicative of a general pattern of content trust and workflow trust scores for simple workflows. The same holds true for intermediate and advanced workflows. Therefore, it is important to reiterate that the trust scores in this study are only reflective of the three use cases. Future research in this area can test to see if the workflow and content trust scores of additional workflows reflect the scores obtained in this study. Additional research could also be done in using a different metric to categorize the workflows instead of the number of tools used.

It is useful to not only talk about how geospatial provenance can be collected, but on how it could be applied to various real life use cases. Once collected, geospatial provenance can be used to estimate a trust level for both workflow and content trust. As this process becomes more automated, it could be integrated in several ways. Businesses could possibly use this as a way to check and see if the workflows they are using are running efficiently and producing trustworthy data products. Researchers could also possibly use these techniques to ensure that the software they are using has not introduced any random errors into the processing of their geospatial data, which might cause an untrustworthy output. Using geospatial provenance, content trust, and workflow trust in conjunction with one another allows for the user of a workflow to gain a better understanding of the mechanisms going on behind the scenes in a workflow, as well as quantify a trust levels related to both content and performance.

References

Allen, J. 2003. Lecture 11: efficient methods for training HMMs. CSC 248 lecture notes, University of Rochester, Rochester, NY. http://www.cs.rochester.edu/u/james/CSC248/Lec11.pdf.

Alonso, G. and C. Hagen. 1997. Geo-Opera: workflow concepts for spatial process. *5th International symposium, SSD, Berlin Germany*. 238-258.

Aronson, P. and S. Morehouse. 1983. The ARC/INFO map library; a design for a digital geographic database". *Auto-Carto six; proceedings of the sixth international symposium on automated cartography*. 1:372-382.

Blunson, P. 2004. Hidden Markov Models. http://digital.cs.usu.edu/~cyan/CS7960/hmm-tutorial.pdf.

Bossler, J.D., J.B. Campbell, R.B. McMaster, and C. Rizos. 2010. *Manual of geospatial science and technology*. 2nd ed. CRC Press. Taylor & Francis Group. 597-598.

Buneman, P. and S. Khanna. 2001. Why and where: a characterization of data provenance. Proceedings of 8th *International Conference on Database Theory*, London,UK, January 4-6, 2001. 316-330.

Cappé, O., E. Moulines, and T. Ryden. 2005. *Inference in Hidden Markov Models*. New York: Springer.

Conover, H., R. Ramachandran, B. Beaumont, A. Kulkarni, M. McEniry, K. Regner, and S. Graves. 2013. Introducing provenance capture into a legacy data system. *IEEE transactions on geoscience and remote sensing* 51 (l1): 5098-5104.

Chrisman, N. R. 1984. The role of quality information in the long-term functioning of a geographic information system. *Proceedings of the sixth international symposium on automated cartography* 1: 302-312.

Di, L., P. Yue., H.K. Ramapriyan, and R.L. King. 2013. Geospatial data provenance: an overview. *IEEE transactions on geoscience and remote sensing*. 1-8. doi 10.1109/TGRS.2013.2242478.

Di, L., Y. Shao, and L. Kang. 2013. Implementation of geospatial data provenance in a web service workflow environment with ISO 19115 and ISO 19115-2 lineage model. *IEEE transactions on geoscience and remote sensing* 51(11): 5082-5089.

Donovan, A. and Y. Gil. 2007. A survey of trust in computer science and the semantic web. *Journal of web semantics: science, services and agents on the world wide web.*

Egenhofer, F. 1992. Object oriented modeling in GIS: inheritance and propagation. *URISA Journal* 4(2): 3-19.

Elizade, S. 2006. Ergodic markov chains. Math 20 discrete probability lecture 15 notes. Hanover, New Hampshire: Dartmouth University. https://math.dartmouth.edu/archive/m20x06/public_html/.

ESRI. 2015. ArcGIS for desktop. http://www.esri.com/software/arcgis/arcgis-for-desktop/free-trial.

ESRI. 2014. Describe (arcpy). ArcGIS help 10.2, 10.2.1, and 10.2.2. ArcGIS resources. http://resources.arcgis.com/en/help/main/10.1/index.html#//018v00000013000000.

ESRI. 2014. Result (arcpy). ArcGIS help 10.2, 10.2.1, and 10.2.2. ArcGIS resources. http://resources.arcgis.com/en/help/main/10.1/index.html#//018z00000046000000.

ESRI. 2014. Raster calculator (spatial analyst). ArcGIS help 10.2, 10.2.1, and 10.2.2. ArcGIS resources. http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//009z000000z70 00000.htm.

ESRI. 2010. Spatial analyst tutorial. http://help.arcgis.com/en/arcgisdesktop/10.0/pdf/spatial-analyst-tutorial.pdf.

Faust, N. 2008. Raster. In *Encyclopedia of geographic information science,* ed. K. Kemp, 361- 364. Thousand Oaks, CA: SAGE Publications, Inc. doi: http://0-dx.doi.org.library.uark.edu/10.4135/9781412953962.n169.

Faust, N. 2008. Geometric primitives . In *Encyclopedia of geographic information science,* ed. K. Kemp, 361-364. Thousand Oaks, CA: SAGE Publications, Inc. doi: http://0-dx.doi.org.library.uark.edu/10.4135/9781412953962.n169.

Feng, C. 2013. Mapping geospatial metadata to open provenance model. *IEEE transactions on geoscience and remote sensing* 51(11): 5073-5081.

Frew, J. 2008. Automatic capture and reconstruction of computational provenance. *Concurrency and computation: practice and experience* 20: 485- 496.

Frew, J. and R. Bose. 2001. Earth system science workbench: a data management infrastructure for earth science products. *Scientific and Statistical Database Managemen*t: 180-189.

Garijo, D., Y. Gil, and A. Harth. 2014. Challenges in modeling geospatial provenance. *Proceedings of the fifth international provenance and annotation Workshop(IPAW), Cologne, Germany, June 9-13, 2014.*

Gil, Y. and V. Ratnakar. 2002. Trusting information sources one citizen at a time. *Proceedings of the First International Semantic Web Conference (ISWC), Sardinia, Italy, June 9 -12,2002.* http://www.isi.edu/expect/web/semanticweb/iswc02_trellis.pdf.

Gil, Y. and D. Artz. 2007. Towards content trust of web resources. *Web semantics: science, services and agents on the world wide web* 5(4): 227-239. http://www.isi.edu/~gil/papers/gil-artz-jws07.pdf.

Gil, Y., J. Cheney, P. Groth, O. Hartig, S. Miles, L. Moreau, and P. Pinheiro da Sila. 2010. "Provenance XG final report". W3C Incubator Group Report. Accessed On October 07, 2013. http://www.w3.org/2005/Incubator/prov/XGR-prov-20101214/.

Grady, R. K. 1988. The lineage of data in land and geographic information systems. *Proceedings of GIS/LIS '88 American congress on surveying and mapping: data lineage in land and geographic information systems* 2: 722-730.

Groth, P. and L. Moreau. 2013. PROV-O: The PROV ontology. http://www.w3schools.com/xml/default.asp Accessed on August 28, 2014.

Guptill, S. C. 1987. Techniques for managing digital cartographic data. *Proceedings of the 13th international cartographic conference, Morelia, Mich, Mexico* 4(16): 221-226.

Harth, A. and Y. Gil. 2014. Geospatial data integration with linked data and provenance tracking. *Proceedings of linking geospatial data, London, W3C/OGC May 3, 2014.*

He, L. P., Yue, L. Di, M. Zhang, and L. Hu. 2015. Adding geospatial data provenance into SDI – a service oriented approach. *IEEE geoscience & remote sensing society selected topics in applied earth observations and remote sensing* 8(2): 926-936.

Hey, A.J.G., S. Tansley, and K.M. Tolle. 2009. *The fourth paradigm data-intensive scientific discovery*. Microsoft Research. Redmond, Washington.137-38. http://research.microsoft.com/en-us/collaboration/fourthparadigm/.

Himmelmann, L. 2010. Package 'HMM'. CRAN. https://cran.r-project.org/web/packages/HMM/HMM.pdf.

Ikeda, R. and J. Widon. 2009. Data lineage: a survey. Stanford University Publications. http://ilpubs.stanford.edu:8090/918/1/lin_final.pdf.

International Foundation for Art Research. 2013. Provenance guide. htpp://www.ifar.org/provenance_guide.php. Accessed August 4, 2013.

International Standards Organization (ISO). 2009. ISO 1915-2. 1-45.

Jones, C., A.I. Abdelmoty, and C. Fu. 2003. Maintaining ontologies for geographical information retrieval on the web. *On the move to meaningful internet systems 2003: CoopIS, DOA, and ODBASE.* 934-951.

Jung, I.Y., H. Eom, and H.Y. Yeom. 2011. Multi-layer trust reasoning on Open Provenance Model for e-Science environment. *IEEE 9th international symposium on parallel and distributed processing with applications (ISPA), May 26-28th, 2011*. 294-299.

Keßler, C., and R.T.A. De Groot. 2013. Trust as a proxy measure for the quality of volunteered geographic information in the case of OpenStreetMap. *Geographic Information Science at the Heart of Europe, Lecture Notes in Geoinformation and Cartography*. 21-37.

Korose, C.P. 2010. *Spatial provenance: a case study on geological carbon sequestration*. University of Illinois at Urbana-Champaign. https://www.ideals.illinois.edu/bitstream/handle/2142/18245/Korose_Christopher.pdf?sequence=1.

Langran, G. 1988. Temporal GIS design tradeoffs. *Proceedings of GIS/LIS '88 American congress on surveying and mapping* 2: 890-899.

Langran, G. and N.R. Chrisman. 1988 A framework for temporal geographic information. *Cartographica* 25(3): 1-14.

Lanter, D. 1990. Lineage in GIS: the problem and a solution. ESRI Campus. http://downloads2.esri.com/campus/uploads/library/pdfs/5819.pdf

Lanter, D. 1991. Design of a lineage-based meta-data base for GIS. *Cartography and Geographic Information Systems* 18(4): 255-261.

Lanter, D. 1994. Metadata analysis of GIS data processing. *Proceedings of the fourteenth annual ESRI user conference May, 1994.* 1007-1014. http://downloads2.esri.com/campus/uploads/library/pdfs/140134.pdf.

Lanter, D. 1994. A lineage metadata approach to removing redundancy and propagating updates in a GIS database. *Cartography and geographic information systems* 21(2): 91-98.

Liu, X., A. Datta, and E. Lim. 2015. *Computational trust models and machine learning*. CRC Press. Boca Raton, FL. 3.

Lxml. 2015. Lxml-XML and HTML with Python. http://lxml.de/.

Lxml. 2015. Class QName. Lxml API. http://lxml.de/api/index.html.

Malaverri, J.E.G., C.B. Medeiros, and R.C. Lamparelli. 2012. A provenance approach to assess the quality of geospatial data. *Proceedings of the 27th annual ACM symposium on applied computing* 2043-2044.

Malaverri, J.E.G. 2013. Supporting data quality assessment in eScience: a provenance based approach. University of Campinas Institute of Computing. 1-83.

Mohri, M. 2012. *Foundations of machine learning*. Cambridge, MA.: MIT Press.

Moore, H. 1983. The impact of computer technology in the mapping environment. *Automated cartography: international perspectives on achievements and challenges* 1(6): 60-68.

Moreau, L. and P. Groth. 2013. Provenance: an introduction to PROV. *In Synthesis Lectures on the Semantic Web: Theory and Technology*. Morgan & Claypool.

Moreau, L. and P. Missier. 2013. PROV-N: The Provenance Notation. W3C. http://www.w3.org/TR/prov-n/ Accessed on October 12, 2015.

Moreau, L. 2013. PROV-O: The PROV Ontology. W3C. http://www.w3.org/TR/prov-o/. Accessed on November 5, 2015.

Moreau, L. 2013. PROV-XML: The PROV XML Schema. W3C. http://www.w3.org/TR/prov-xml/. Accessed on November 5, 2015.

NASA. 2011. Landsat 7 science data users handbook. Greenbelt, Maryland. 114. http://landsathandbook.gsfc.nasa.gov/pdfs/Landsat7_Handbook.pdf.

Naseri, M. 2013. A multi-functional provenance architecture: challenges and solutions. University of Saskatchwen department of computer science. 53 – 80. http://ecommons.usask.ca/bitstream/handle/10388/ETD-2013-12-1419/NASERI-DISSERTATION.pdf?sequence=4.

Naseri, M. and S.A. Ludwig. 2013. Evaluating workflow trust using Hidden Markov modeling and provenance data. *Data provenance and data management in eScience. Studies of Computational Intelligence* (426): 35-58.

NCDCDS. 1988. The proposed standard for digital cartographic data. *The American cartographer* 15(1).

Nyerges, T. 1987. GIS research needs identified during a cartographic standards process: spatial data exchange. *International geographic information systems symposium: the research agenda* 1: 319-330.

Pipino, L. L., Y. W. Yang and R.Y. Wang. 2002. Data quality assessment. *Communications of the ACM*. 45(4): 211-218.

Plale, B., B. Cao,C. Herath, and Y. Sun. 2011. Data provenance for preservation of digital geoscience data. *The Geological Society of America special paper 482*: 125-137. In *Societal challenges and geoinformatics*. Boulder Colorado.

Rabiner, L. 1989. A tutorial on Hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(3): 257-286.

Rajbhandari, S., I. Wooten, A. Shaikh, R. Ali., and F. Omer. 2006. Evaluating provenance-based trust for scientific workflows. *Proceedings of the Sixth IEEE international symposium on Cluster computing and the grid (CCGRID'06). Singapore*. 365-372.

Russell, S., and P. Norvig. 2003. *Artificial Intelligence: a modern approach*. Upper Saddle River, N.J.: Prentice Hall/Pearson Education.

Santos, E., D. Koop, T. Maxwell, C. Doutriaux, T. Ellqvist, G. Potter, J. Freire, D. Williams, and C.T. Silva. 2012. Designing a provenance-based climate data analysis application. *IPAW'12 Proceedings of the 4th international conference on provenance and annotation of data and processes:* 214-219.

Smola, A. 2015. Carnegie Mellon University, Machine Learning 10-701 Homework 11. http://alex.smola.org/teaching/10-701-15/homework/hw11.pdf.

Sun, Z., P. Yue, L. Hu, J. Gong, L. Zhang, and X. Lu. 2013. GeoPWPRov: interleaving map and faceted metadata for provenance visualization and navigation. *IEEE Transactions on Geoscience and Remote Sensing* 51(11): 5131-5136.

Tan, W. 2007. Provenance in databases: past, current, and future. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*:1-10.

Tilmes, C. and A.J.Fleig. 2008. Provenance tracking in an earth science data processing system. *Second International Provenance and Annotation Workshop*: 221-228.

Tullis, J., Cothren, J., Lanter, D., Shi, X., Limp, F., Linck, R., Young, S., Alsumaiti, T. (In Press). Geoprocessing, Workflows, and Provenance. *Remote Sensing Handbook.* 1st ed. Vol. 1. Taylor and Francis.

Tu, S. Derivation of Baum-Welch algorithm for Hidden Markov Models. http://www.cs.berkeley.edu/~stephentu/writeups/hmm-baum-welch-derivation.pdf.

United States Geological Survey (USGS). 2014a. USGS Earth Explorer. United States Department of Interior. http://earthexplorer.usgs.gov/.

United States Geological Survey (USGS).2014b. Landsat 8. http://landsat.usgs.gov/landsat8.php.

UV-CDAT. 2015. Ultrascale visualization: climate data analysis tools. http://uvcdat.llnl.gov/.

Van Den Bergh, F., K.J. Wessel, S. Miteff, T.L. Van Zyl, A.D. Gazendam, and A.K. Bachoo. 2012. HiTempo: a platform for time-series analysis of remote-sensing satellite data in a high-performance computing environment. *International Journal of Remote Sensing* 33(15): 4720-4740.

W3C. 2001. URIs, URLs, and URNs: clarifications and recommendations 1.0. *Report from the joint W3C/IETF URI planning interest group, September 21, 2001.* http://www.w3.org/TR/uri-clarification/.

Wang, S., A. Padmanabhan, J.D. Myers, W. Tang, and Y. Liu. 2008. Towards provenance-aware geographic information systems. *ACM GIS:* 1-4.

Woodruff, A. and M. Stonebraker. 1997. Supporting fine-grained lineage in a database visualization environment”. *ICDE '97 Proceedings of the Thirteenth International Conference on Data Engineering*: 91-102. http://db.cs.berkeley.edu/papers/icde97-dl.pdf.

Yeide, N., K. Akinsha, and A.L. Walsh. 2001. *The AAM guide to provenance research*: 1-10. Washington D.C.: American Association of Museums.

Yuan, J., P. Yue, J. Gong, and M. Zhang. 2013. A linked data approach for geospatial data provenance". *IEEE Transactions on Geoscience and Remote Sensing* 51(11): 5105-5112.

Yue, P. and L. He. 2009. Geospatial data provenance in cyberinfrastructure. *17th International Conference on Geoinformatics*: 1-4.

Yue, P., J. Gong, L. Di, and Y. Wei. 2010. Semantic provenance registration and discovery using geospatial catalogue service. *Elsevier Computers & Geosciences* 36(3): 270-281.

Yue, P., J. Gong, L. Di. 2010. Augmenting geospatial data provenance through metadata tracking in geospatial service chaining. *Elsevier Computers & Geosciences* 51(11): 5105-5112.

Yue, P., Y. Wei, L. Di, L. He, J. Gong, and L. Zhang. 2011. Sharing geospatial provenance in a service-oriented environment. *Elsevier Computers, Environment and Urban Systems 35:* 333-343.

Zhao, J., C. Goble, M. Greenwood, C. Wroe, and R. Stevens. 2003. Annotating, linking, and browsing provenance logs for e-Science”. *Proceedings of the 2nd. International Semantic Web Conference.*

Appendix A: Collected Provenance to PROV mapping

Collected Data	**PROV-O Mapping**
User	prov:agent
Tool	prov:activity
Input	rl:inputFile
Output	rl:outputFile
Time Began	prov:startTime
Time Ended	prov:endTime
Activated Tool	prov:wasAttributedTo
Parent	prov:wasGeneratedBy
Child	prov:wasDerivedFrom
Inputs tool used	prov:used

Appendix B: Python Scripts

B1. Main.py

```
import ndviProvExp
import sys
import hydroProvExp
import siteSelectionExp

i = 0
while i <51:
  siteSelectionExp.genSiteProv()
  i = i+1

  '''ndviProvExp.genNdviProv()
  hydroProvExp.genHydroProv()
  i = i +1'''
```

B2. ndviProvExp.py

```
def genNdviProv():

  # Import arcpy module
  import arcpy
  from datetime import datetime
  from xmlGenerate import *
  #import prov.model as prov


  #Workspace
  from arcpy import env

  env.workspace = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\test.gdb\\"
  # Check out any necessary licenses
  arcpy.CheckOutExtension("spatial")

  #Allowing rewrite
  arcpy.env.overwriteOutput=True

  #Enabling log of processing
  arcpy.SetLogHistory(True)

  # Local variables:
        Band5 =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\test.gdb\\Input\\LC80240362014113LGN00.tar\\LC8
0240362014113LGN00\\LC80240362014113LGN00_B5.TIF"
        Band4 =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\test.gdb\\Input\\LC80240362014113LGN00.tar\\LC8
0240362014113LGN00\\LC80240362014113LGN00_B4.TIF"
        ADMIN_DBO_CITY_LIMITS_AHTD_polygon =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\test.gdb\\Input\\Damascus_CL\\GeoStor\\ADMIN_D
BO_CITY_LIMITS_AHTD_polygon.shp"
        B5flt = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Temp\\b5test.flt"
        B4flt = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Temp\\b4test.flt"
        ndvi_Output = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\test.gdb\\Output\\ndvi_Output"
        ndvi_Clip="C:\\Users\\Kcnil14\\Documents\\ArcGIS\\test.gdb\\Output\\clipped_NDVI"

        #Lists/Arrays for holding info on layers
        toolList = []
        stList = []
        etList = []
        inputList = []
        outputList = []
        dataSrc = []
        uID = []

      #Process: Raster Calculator DOES NOT WORK AS DIRECT EXPORT MUST BE
MODIFIED.
      #arcpy.gp.RasterCalculator_sa("Float((\"%LC80240362014113LGN00_B5.TIF (3)%\" -
\"%LC80240362014113LGN00_B4.TIF%\"))/Float((\"%LC80240362014113LGN00_B5.TIF
(3)%\" + \"%LC80240362014113LGN00_B4.TIF%\"))", result)

        #Converting first raster band to float
        floatConversion1 = arcpy.RasterToFloat_conversion(Band5, B5flt)

        #Converting to string for parsing tool name
        sFloatConversion1="floatConversion1 = arcpy.RasterToFloat_conversion(Band5, B5flt)"
        tool1=GetToolName(sFloatConversion1)

        #Appending all info to lists to pass to etree
        toolList.append(tool1)

        #Accessing the Descript and Result Objects may not need this tho
        Band5Descript = arcpy.Describe(floatConversion1)

        #Continuing appending lists
        tool1ST = getStartTime(floatConversion1)
        stList.append(tool1ST)
```

```
  tool1ET = getEndTime(floatConversion1)
  etList.append(tool1ET)

  tool1Input = str(getInputs(floatConversion1))
  inputList.append(tool1Input)

  tool1Output = getOutputs(floatConversion1)
  outputList.append(tool1Output)

  uniID1 = str(uuid.uuid4())
  uID.append(uniID1)
  #Updating source information for each file
  #userInputs(dataSrc)

#Convert second raster band to float
  floatConversion2 = arcpy.RasterToFloat_conversion(Band4, B4flt)
  sFloatConversion2 = "floatconversion2 = arcpy.RasterToFloat_conversion(Band4, B4flt)"
  tool2=GetToolName(sFloatConversion2)
  toolList.append(tool2)

  tool2ST = getStartTime(floatConversion2)
  stList.append(tool2ST)

  tool2ET = getEndTime(floatConversion2)
  etList.append(tool2ET)

  tool2Input = str(getInputs(floatConversion2))
  inputList.append(tool2Input)

  tool2Output = getOutputs(floatConversion2)
  outputList.append(tool2Output)

  uniID2 = str(uuid.uuid4())
  uID.append(uniID2)

  #Accessing the Descript and Result Objects

  Band4Descript = arcpy.Describe(floatConversion2)

  #Updating source information for each file
  #userInputs(dataSrc)

  #Full syntax for Map Algebra is used to enable ease of parsing
  #Start time for map algebra process
```

```
        maToolStartTime = datetime.now().strftime('%a' + ' ' + '%b' + ' ' + '%d' + ' ' + '%I' + ":" +
'%M' + ":" + '%S')

        #Starting map algebra
        ndvi_calc=(arcpy.sa.Minus(B5flt,B4flt))/(arcpy.sa.Plus(B5flt,B4flt))
        ndvi_calString="(arcpy.sa.Minus(B5flt,B4flt))/(arcpy.sa.Plus(B5flt,B4flt))"

        #End time for map algebra process
        #The total time elapsed is not recorded, however it could be if needed.
        ndviSave = ndvi_calc.save(ndvi_Output)

        maToolEndTime = datetime.now().strftime('%a' + ' ' + '%b' + ' ' + '%d' + ' ' + '%I' + ":" +
'%M' + ":" + '%S')


        #Getting the runtime output information for the map algebra process.
        ndviOut = getMapAlgebraOutput(ndvi_Output)
        tool3=SetMapAlgebra()
        tool3info = GetMapAlgebraInfo(ndvi_calString)
        toolList.append(tool3)

        tool3Input = getMapAlgebraInput(ndvi_calString)
        inputList.append(tool3Input)
        tool3Output = getMapAlgebraOutput(ndvi_Output)
        outputList.append(tool3Output)
        stList.append(maToolStartTime)
        etList.append(maToolEndTime)

        uniID3 = str(uuid.uuid4())
        uID.append(uniID3)

        #Updating source information for each file
        #userInputs(dataSrc)

        # Process: Clip
        clip = arcpy.Clip_management(ndvi_Output, "551945.600299715 3911928.85510621
555608.114799695 3914858.14920625", ndvi_Clip,
ADMIN_DBO_CITY_LIMITS_AHTD_polygon, "-3.402823e+038", "NONE",
"NO_MAINTAIN_EXTENT")
        sndvi_calc='arcpy.Clip_management(ndvi_Output, "551945.600299715 3911928.85510621
555608.114799695 3914858.14920625", ndvi_Clip,
ADMIN_DBO_CITY_LIMITS_AHTD_polygon, "-3.402823e+038", "NONE",
"NO_MAINTAIN_EXTENT")'
        tool4 = GetToolName(sndvi_calc)
```

```
  toolList.append(tool4)

  tool4ST = getStartTime(clip)
  stList.append(tool4ST)
  tool4ET = getEndTime(clip)
  etList.append(tool4ET)
  tool4Inputs = str(getInputs(clip))
  inputList.append(tool4Inputs)
  tool4Outputs = getOutputs(clip)
  outputList.append(tool4Outputs)

  uniID4 = str(uuid.uuid4())
  uID.append(uniID4)

  #Updating source information for each file
  #userInputs(dataSrc)//turn this back on when fixed for hydro exp.

  #generateXML(toolList,stList,etList,inputList,outputList)
  generateProvXML(toolList,stList,etList,inputList,outputList,uID)

  #generateProv(toolList,stList,etList,inputList,outputList)

  print 'done'
```

B3. genHydroProv.py

```
def genHydroProv():
# Import arcpy module
  import arcpy
  from datetime import datetime
  from xmlGenerate import *

#Workspace
  from arcpy import env

  env.workspace = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\"
# Check out any necessary licenses
  arcpy.CheckOutExtension("spatial")

#Allowing rewrite
  arcpy.env.overwriteOutput=True

#Enabling log of processing
```

```
    arcpy.SetLogHistory(True)

    # Check out any necessary licenses
    arcpy.CheckOutExtension("spatial")

    # Local variables:
    imgn36w093_13_img =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Input\\n36w093\\imgn36w093_13.img"
    Boundaries_COUNTIES_AHTD_shp =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Input\\FME_011F5D59_1422298114857
_26956\\GeoStor\\Boundaries_COUNTIES_AHTD.shp"
    WATER_BASE_LAYER_ADEQ_shp =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Input\\WATER_BASE_LAYER_ADEQ\
\WATER_BASE_LAYER_ADEQ.shp"
    prj_raster = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\prj_raster"
    clipped_dem_img =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\clipped_dem.img"
    ClipStreams_shp = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\ClipStreams.shp"
    slope_Raster_img =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\slope_Raster.img"
    aspect_Raster_img =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\aspect_Raster.img"
    fill_dem_img =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\fill_dem.img"
    flowDir_img =
"C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\flowDir.img"
    Output_drop_raster = ""
    flowAc_img = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\flowAc.img"


    #Lists/Arrays for holding info on layers
    toolList = []
    stList = []
    etList = []
    inputList = []
    outputList = []
    dataSrc = []
    uID = []

    # Process: Clip
    clip = arcpy.Clip_analysis(WATER_BASE_LAYER_ADEQ_shp,
Boundaries_COUNTIES_AHTD_shp, ClipStreams_shp, "")
    strClip = 'arcpy.Clip_analysis(WATER_BASE_LAYER_ADEQ_shp,
Boundaries_COUNTIES_AHTD_shp, ClipStreams_shp, "")'
    tool1 = GetToolName(strClip)
```

```
        toolList.append(tool1)

        tool1ST = getStartTime(clip)
        stList.append(tool1ST)

        tool1ET = getEndTime(clip)
        etList.append(tool1ET)

        tool1Inputs = str(getInputs(clip))
        inputList.append(tool1Inputs)

        tool1Outputs = getOutputs(clip)
        outputList.append(tool1Outputs)

        uniID1 = str(uuid.uuid4())
        uID.append(uniID1)
      # Process: Project Raster
        prjRast = arcpy.ProjectRaster_management(imgn36w093_13_img, prj_raster,
"PROJCS['NAD_1983_UTM_Zone_15N',GEOGCS['GCS_North_American_1983',DATUM['D
_North_American_1983',SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Gree
nwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PAR
AMETER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Centr
al_Meridian',-
93.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Me
ter',1.0]]", "NEAREST", "9.33429600149082 9.33429600149075", "", "",
"GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROID['GR
S_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.01745329251
99433]],VERTCS['Unknown
VCS',VDATUM['Unknown'],PARAMETER['Vertical_Shift',0.0],PARAMETER['Direction',1.0]
,UNIT['Meter',1.0]]")
        strPrjRast = 'arcpy.ProjectRaster_management(imgn36w093_13_img, prj_raster,
"PROJCS["NAD_1983_UTM_Zone_15N",GEOGCS["GCS_North_American_1983",DATUM[
"D_North_American_1983",SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["
Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Transverse_Mercator"
],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],PARAMETE
R["Central_Meridian",-
93.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_Of_Origin",0.0],UNIT["
Meter",1.0]]", "NEAREST", "9.33429600149082 9.33429600149075", "", "",
"GEOGCS["GCS_North_American_1983",DATUM["D_North_American_1983",SPHEROID["
GRS_1980",6378137.0,298.257222101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.017453
2925199433]],VERTCS["Unknown
VCS",VDATUM["Unknown"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",
1.0],UNIT["Meter",1.0]]")'
        tool2 = GetToolName(strPrjRast)
        toolList.append(tool2)
        tool2ST = getStartTime(prjRast)
        stList.append(tool2ST)

        tool2ET = getEndTime(prjRast)
        etList.append(tool2ST)

        tool2Input = str(getInputs(prjRast))
        inputList.append(tool2Input)

        tool2Output = getOutputs(prjRast)
        outputList.append(tool2Output)

        uniID2 = str(uuid.uuid4())
        uID.append(uniID2)
      # Process: Extract by Mask
        exMask = arcpy.gp.ExtractByMask_sa(prj_raster, Boundaries_COUNTIES_AHTD_shp,
clipped_dem_img)
        strExMask = 'arcpy.gp.ExtractByMask_sa(prj_raster, Boundaries_COUNTIES_AHTD_shp,
clipped_dem_img)'

        tool3 = GetToolName(strExMask)
        toolList.append(tool3)

        tool3ST = getStartTime(exMask)
        stList.append(tool3ST)

        tool3ET = getEndTime(exMask)
        etList.append(tool3ET)

        tool3Input = str(getInputs(exMask))
        inputList.append(tool3Input)

        tool3Output = getOutputs(exMask)
        outputList.append(tool3Output)

        uniID3 = str(uuid.uuid4())
        uID.append(uniID3)

        # Process: Fill
        fill = arcpy.gp.Fill_sa(clipped_dem_img, fill_dem_img, "")
        strFill = 'arcpy.gp.Fill_sa(clipped_dem_img, fill_dem_img, "")'
        tool4 = GetToolName(strFill)
        toolList.append(tool4)
```

```
tool4ST = getStartTime(fill)
stList.append(tool4ST)

tool4ET = getEndTime(fill)
etList.append(tool4ET)

tool4Input = str(getInputs(fill))
inputList.append(tool4Input)

tool4Output = getOutputs(fill)
outputList.append(tool4Output)

uniID4 = str(uuid.uuid4())
uID.append(uniID4)
# Process: Slope
slope = arcpy.gp.Slope_sa(fill_dem_img, slope_Raster_img, "DEGREE", "1")
strSlope = 'arcpy.gp.Slope_sa(fill_dem_img, slope_Raster_img, "DEGREE", "1")'
tool5 = GetToolName(strSlope)
toolList.append(tool5)

tool5ST = getStartTime(slope)
stList.append(tool5ST)

tool5ET = getEndTime(slope)
etList.append(tool5ET)

tool5Input = str(getInputs(slope))
inputList.append(tool5Input)

tool5Output = getOutputs(slope)
outputList.append(tool5Output)

uniID5 = str(uuid.uuid4())
uID.append(uniID5)
# Process: Aspect
aspect = arcpy.gp.Aspect_sa(fill_dem_img, aspect_Raster_img)
strAspect = 'arcpy.gp.Aspect_sa(fill_dem_img, aspect_Raster_img)'
tool6 = GetToolName(strAspect)
toolList.append(tool6)

tool6ST = getStartTime(aspect)
stList.append(tool6ST)

tool6ET = getEndTime(aspect)
etList.append(tool6ET)
```

```
        tool6Input = str(getInputs(aspect))
        inputList.append(tool6Input)

        tool6Output = getOutputs(aspect)
        outputList.append(tool6Output)

        uniID6 = str(uuid.uuid4())
        uID.append(uniID6)
        # Process: Flow Direction
        flDir = arcpy.gp.FlowDirection_sa(fill_dem_img, flowDir_img, "NORMAL", 
Output_drop_raster)
        strFlDir = 'arcpy.gp.FlowDirection_sa(fill_dem_img, flowDir_img, "NORMAL", 
Output_drop_raster)'
        tool7 = GetToolName(strFlDir)
        toolList.append(tool7)

        tool7ST = getStartTime(flDir)
        stList.append(tool7ST)

        tool7ET = getEndTime(flDir)
        etList.append(tool7ET)

        tool7Input = str(getInputs(flDir))
        inputList.append(tool7Input)

        tool7Output = getOutputs(flDir)
        outputList.append(tool7Output)

        uniID7 = str(uuid.uuid4())
        uID.append(uniID7)
        # Process: Flow Accumulation
        flAc = arcpy.gp.FlowAccumulation_sa(flowDir_img, flowAc_img, "", "FLOAT")
        strFlAc = 'arcpy.gp.FlowAccumulation_sa(flowDir_img, flowAc_img, "", "FLOAT")'
        tool8 = GetToolName(strFlAc)
        toolList.append(tool8)

        tool8ST = getStartTime(flAc)
        stList.append(tool8ST)

        tool8ET = getEndTime(flAc)
        etList.append(tool8ET)

        tool8Input = str(getInputs(flAc))
        inputList.append(tool8Input)
```

```
    tool8Output = getOutputs(flAc)
    outputList.append(tool8Output)

    uniID8 = str(uuid.uuid4())
    uID.append(uniID8)
    #userInputs(dataSrc)//Not working correctly for this one. Need to fig outprob.

    generateProvXML(toolList,stList,etList,inputList,outputList,uID)

    print 'done'
```

B4. siteSelectionExp.py

```
  def genSiteProv():

    # Import arcpy module
    import arcpy
    from datetime import datetime
    from xmlGenerate import*

    #Workspace
    from arcpy import env
    env.workspace = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial Analyst\\Stowe.gdb"

    # Check out any necessary licenses
    arcpy.CheckOutExtension("spatial")

    # Set Geoprocessing environments
    arcpy.env.extent = "471060.082572495 208312.353396819 494700.082572495
231352.353396819"
    arcpy.env.cellSize = "30"

    #Allowing rewrite
    arcpy.env.overwriteOutput = True

    #Enabling geoprocessing log in case want to check results against it
    arcpy.SetLogHistory(True)



    # Local variables:
    elevation = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\elevation"
```

```
    rec_sites = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\rec_sites"
    schools = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial Analyst\\Stowe.gdb\\schools"
    landuse__2_ = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\landuse"
    Weighte_Recl1__2_ = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Weighte_Recl1"
    #RasterT_Majorit1__2_ = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\RasterT_Majorit1"
    roads = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial Analyst\\Stowe.gdb\\roads"
    RasterT_Majorit1__4_ = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\RasterT_Majorit1"
    HillSha_elev2 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\HillSha_elev2"
    Slope_Out = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Slope_Out"
    EucDist_rec_1 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\EucDist_rec_1"
    Output_direction_raster = ""
    EucDist_scho1 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\EucDist_scho1"
    Output_direction_raster__2_ = ""
    Reclass_Slop1 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Reclass_Slop1"
    Reclass_EucD1 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Reclass_EucD1"
    Reclass_EucD2 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Reclass_EucD2"
    Weighte_Recl1 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Weighte_Recl1"
    Con_Weighte_1 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Con_Weighte_1"
    Majorit_Con_1 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Majorit_Con_1"
    RasterT_Majorit1 = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\RasterT_Majorit1"
    RasterT_Majorit1__5_ = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\RasterT_Majorit1"
    final_site_shp = "C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\final_site.shp"

    #Lists/Arrays for holding data on layers
    toolList = []
    stList = []
    etList = []
```

```
        inputList = []
        outputList = []
        dataSrc = []
        uID = []

      #First Process
        # Process: Hillshade
        hillShade1 = arcpy.gp.HillShade_sa(elevation, HillSha_elev2, "315", "45",
"NO_SHADOWS", "0.3048")

        #Converting to string for parsing tool name and appending to list
        sHillshadCon = 'arcpy.gp.HillShade_sa(elevation, HillSha_elev2, "315", "45",
"NO_SHADOWS", "0.3048")'
        tool1 = GetToolName(sHillshadCon)
        toolList.append(tool1)

        #Acessing Describe Object
        hillDescrip = arcpy.Describe(hillShade1)

        tool1ST = getStartTime(hillShade1)
        stList.append(tool1ST)

        tool1ET = getEndTime(hillShade1)
        etList.append(tool1ET)

        tool1Input = str(getInputs(hillShade1))
        inputList.append(tool1Input)

        tool1Output = getOutputs(hillShade1)
        outputList.append(tool1Output)

        uniId1 = str(uuid.uuid4())
        uID.append(uniId1)

      #Second Process
        # Process: Euclidean Distance
        eucDist1 = arcpy.gp.EucDistance_sa(rec_sites, EucDist_rec_1, "", "30",
Output_direction_raster)

        #Converting to string for parsing tool name
        seucDist1 = 'arcpy.gp.EucDistance_sa(rec_sites, EucDist_rec_1, "", "30",
Output_direction_raster)'
        tool2 = GetToolName(seucDist1)
        toolList.append(tool2)
        #Accessing Describe Object
        eucDistDescr = arcpy.Describe(eucDist1)

        #Continuing appending lists
        tool2ST = getStartTime(eucDist1)
        stList.append(tool2ST)

        tool2ET = getEndTime(eucDist1)
        etList.append(tool2ET)

        tool2Input = str(getInputs(eucDist1))
        inputList.append(tool2Input)

        tool2Output = getOutputs(eucDist1)
        outputList.append(tool2Output)

        uniId2 = str(uuid.uuid4())
        uID.append(uniId2)

      #Third Process
        # Process: Euclidean Distance (2)
        eucDist2 = arcpy.gp.EucDistance_sa(schools, EucDist_scho1, "", "30",
Output_direction_raster__2_)

        #Converting to string for parsing tool name
        seucDist2 = 'arcpy.gp.EucDistance_sa(schools, EucDist_scho1, "", "30",
Output_direction_raster__2_)'
        tool3 = GetToolName(seucDist2)
        toolList.append(tool3)

        #Accessing Describe Object
        eucDistDescr2 = arcpy.Describe(eucDist2)

        #Continuing appending lists
        tool3ST = getStartTime(eucDist2)
        stList.append(tool3ST)

        tool3ET = getEndTime(eucDist1)
        etList.append(tool3ET)

        tool3Input = str(getInputs(eucDist2))
        inputList.append(tool3Input)

        tool3Output = getOutputs(eucDist2)
        outputList.append(tool3Output)
        uniId3 = str(uuid.uuid4())
        uID.append(uniId3)

      #Fourth Process
        # Process: Reclassify (3)
        reclas3 = arcpy.gp.Reclassify_sa(EucDist_scho1, "Value", "0 1694.7212890625001
1;1694.7212890625001 3389.4425781250002 2;3389.4425781250002 5084.1638671875007
3;5084.1638671875007 6778.8851562500004 4;6778.8851562500004 8473.6064453125
5;8473.6064453125 10168.327734375 6;10168.327734375 11863.049023437499
7;11863.049023437499 13557.770312499999 8;13557.770312499999 15252.491601562499
9;15252.491601562499 16947.212890625 10", Reclass_EucD2, "DATA")

        #Converting to string for parsing tool name
        sreclas3 = 'arcpy.gp.Reclassify_sa(EucDist_scho1, "Value", "0 1694.7212890625001
1;1694.7212890625001 3389.4425781250002 2;3389.4425781250002 5084.1638671875007
3;5084.1638671875007 6778.8851562500004 4;6778.8851562500004 8473.6064453125
5;8473.6064453125 10168.327734375 6;10168.327734375 11863.049023437499
7;11863.049023437499 13557.770312499999 8;13557.770312499999 15252.491601562499
9;15252.491601562499 16947.212890625 10", Reclass_EucD2, "DATA")'
        tool4 = GetToolName(sreclas3)
        toolList.append(tool4)

        #Accessing Describe Object
        reclas3Des = arcpy.Describe(reclas3)

        #Continuing appending lists
        tool4ST = getStartTime(reclas3)
        stList.append(tool4ST)

        tool4ET = getEndTime(reclas3)
        etList.append(tool4ET)

        tool4Input = str(getInputs(reclas3))
        inputList.append(tool4Input)

        tool4Output = getOutputs(reclas3)
        outputList.append(tool4Output)

        uniId4 = str(uuid.uuid4())
        uID.append(uniId4)

      #Fifth Process
        # Process: Reclassify (2)
        reclas2 = arcpy.gp.Reclassify_sa(EucDist_rec_1, "Value", "0 1352.92021484375
10;1352.92021484375 2705.8404296875001 9;2705.8404296875001 4058.7606445312504
8;4058.7606445312504 5411.6808593750002 7;5411.6808593750002 6764.60107421875
6;6764.60107421875 8117.5212890624998 5;8117.5212890624998 9470.4415039062496
4;9470.4415039062496 10823.36171875 3;10823.36171875 12176.281933593751
2;12176.281933593751 13529.2021484375 1", Reclass_EucD1, "DATA")

        #Converting to string for parsing tool name
        sreclas2 = 'arcpy.gp.Reclassify_sa(EucDist_rec_1, "Value", "0 1352.92021484375
10;1352.92021484375 2705.8404296875001 9;2705.8404296875001 4058.7606445312504
8;4058.7606445312504 5411.6808593750002 7;5411.6808593750002 6764.60107421875
6;6764.60107421875 8117.5212890624998 5;8117.5212890624998 9470.4415039062496
4;9470.4415039062496 10823.36171875 3;10823.36171875 12176.281933593751
2;12176.281933593751 13529.2021484375 1", Reclass_EucD1, "DATA")'


        tool5 = GetToolName(sreclas2)
        toolList.append(tool5)

        #Accessing Describe Object
        reclas2Des = arcpy.Describe(reclas2)

        #Continuing appending lists
        tool5ST = getStartTime(reclas2)
        stList.append(tool5ST)

        tool5ET = getEndTime(reclas2)
        etList.append(tool5ET)

        tool5Input = str(getInputs(reclas2))
        inputList.append(tool5Input)

        tool5Output = getOutputs(reclas2)
        outputList.append(tool5Output)

        uniId5 = str(uuid.uuid4())
        uID.append(uniId5)

      #Sixth Process
        # Process: Slope
        slp1 = arcpy.gp.Slope_sa(elevation, Slope_Out, "DEGREE", "0.3048")

        #Converting to string for parsing tool name
        sSlp1 = 'arcpy.gp.Slope_sa(elevation, Slope_Out, "DEGREE", "0.3048")'
        tool6 = GetToolName(sSlp1)
        toolList.append(tool6)

        #Accessing Describe Object
        slp1Des = arcpy.Describe(slp1)

        #Continuing appending lists
        tool6ST = getStartTime(slp1)
        stList.append(tool6ST)

        tool6ET = getEndTime(slp1)
        etList.append(tool6ET)

        tool6Input = str(getInputs(slp1))
        inputList.append(tool6Input)

        tool6Output = getOutputs(slp1)
        outputList.append(tool6Output)

        uniId6 = str(uuid.uuid4())
        uID.append(uniId6)

      #Seventh Process
        # Process: Reclassify
        reclas1 = arcpy.gp.Reclassify_sa(Slope_Out, "Value", "0 4.7758632659912106
10;4.7758632659912106 9.5517265319824212 9;9.5517265319824212 14.327589797973632
8;14.327589797973632 19.103453063964842 7;19.103453063964842 23.879316329956055
6;23.879316329956055 28.655179595947267 5;28.655179595947267 33.431042861938479
4;33.431042861938479 38.206906127929692 3;38.206906127929692 42.982769393920904
2;42.982769393920904 47.758632659912109 1", Reclass_Slop1, "DATA")

        #Converting to string for parsing tool name
        sreclas1 = 'arcpy.gp.Reclassify_sa(Slope_Out, "Value", "0 4.7758632659912106
10;4.7758632659912106 9.5517265319824212 9;9.5517265319824212 14.327589797973632
8;14.327589797973632 19.103453063964842 7;19.103453063964842 23.879316329956055
6;23.879316329956055 28.655179595947267 5;28.655179595947267 33.431042861938479
4;33.431042861938479 38.206906127929692 3;38.206906127929692 42.982769393920904
2;42.982769393920904 47.758632659912109 1", Reclass_Slop1, "DATA")'

        tool7 = GetToolName(sreclas1)
        toolList.append(tool7)

        #Accessing Describe Object
        reclas1Des = arcpy.Describe(reclas1)
        #Continuing appending lists
        tool7ST = getStartTime(reclas1)
        stList.append(tool7ST)

        tool7ET = getEndTime(reclas1)
        etList.append(tool7ET)

        tool7Input = str(getInputs(reclas1))
        inputList.append(tool7Input)

        tool7Output = getOutputs(reclas1)
        outputList.append(tool7Output)

        uniId7 = str(uuid.uuid4())
        uID.append(uniId7)

       #Eight Process
        # Process: Weighted Overlay
        wOver1 =
arcpy.gp.WeightedOverlay_sa("('C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Reclass_EucD2' 25 'Value' (1 1; 2 2; 3 3; 4 4; 5 5; 6 6; 7 7; 8 8; 9 9; 10
10;NODATA NODATA); 'C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Reclass_EucD1' 50 'Value' (1 1; 2 2; 3 3; 4 4; 5 5; 6 6; 7 7; 8 8; 9 9; 10
10;NODATA NODATA); 'C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Reclass_Slop1' 13 'Value' (1 Restricted; 2 Restricted; 3 Restricted; 4 4; 5 5;
6 6; 7 7; 8 8; 9 9; 10 10;NODATA NODATA); 'landuse' 12 'LANDUSE' ('Brush/transitional' 5;
'Water' Restricted; 'Barren land' 10; 'Built up' 3; 'Agriculture' 9; 'Forest' 4; 'Wetlands'
1;NODATA NODATA));1 10 1", Weighte_Recl1)

        #Converting to string for parsing tool name
        swOver1 =
'arcpy.gp.WeightedOverlay_sa("(C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Spatial
Analyst\\Stowe.gdb\\Reclass_EucD2'
        tool8 = GetToolName(swOver1)
        toolList.append(tool8)

        #Accessing Describe Object
        wOver1Des = arcpy.Describe(wOver1)

        #Continuing appending lists
        tool8ST = getStartTime(wOver1)
        stList.append(tool8ST)

        tool8ET = getEndTime(wOver1)
        etList.append(tool8ET)
        tool8Input = str(getInputs(wOver1))
        inputList.append(tool8Input)

        tool8Output = getOutputs(wOver1)
        outputList.append(tool8Output)

        uniId8 = str(uuid.uuid4())
        uID.append(uniId8)

      #Ninth Process
        # Process: Con
        con1 = arcpy.gp.Con_sa(Weighte_Recl1__2_, Weighte_Recl1, Con_Weighte_1, "", "Value
= 9")

        #Converting to string for parsing tool name
        scon1 = 'arcpy.gp.Con_sa(Weighte_Recl1__2_, Weighte_Recl1, Con_Weighte_1, "",
"Value = 9")'
        tool9 = GetToolName(scon1)
        toolList.append(tool9)

        #Accessing Describe Object
        con1Desc = arcpy.Describe(con1)

        #Continuing appending lists
        tool9ST = getStartTime(con1)
        stList.append(tool9ST)

        tool9ET = getEndTime(con1)
        etList.append(tool9ET)

        tool9Input = str(getInputs(con1))
        inputList.append(tool9Input)

        tool9Output = getOutputs(con1)
        outputList.append(tool9Output)

        uniId9 = str(uuid.uuid4())
        uID.append(uniId9)

      #Tenth Process
        # Process: Majority Filter
        majFi = arcpy.gp.MajorityFilter_sa(Con_Weighte_1, Majorit_Con_1, "EIGHT",
"MAJORITY")
        #Converting to string for parsing tool name
        smajFi = 'arcpy.gp.MajorityFilter_sa(Con_Weighte_1, Majorit_Con_1, "EIGHT",
"MAJORITY")'
        tool10 = GetToolName(smajFi)
        toolList.append(tool10)

        #Accessing Describe Object
        majFiDesc = arcpy.Describe(majFi)

        #Continuing appending lists
        tool10ST = getStartTime(majFi)
        stList.append(tool10ST)

        tool10ET = getEndTime(majFi)
        etList.append(tool10ET)

        tool10Input = str(getInputs(majFi))
        inputList.append(tool10Input)

        tool10Output = getOutputs(majFi)
        outputList.append(tool10Output)

        uniId10 = str(uuid.uuid4())
        uID.append(uniId10)

      #Eleventh Process
        # Process: Raster to Polygon
        ras2pol = arcpy.RasterToPolygon_conversion(Majorit_Con_1, RasterT_Majorit1,
"SIMPLIFY", "Value")

        #Converting to string for parsing tool name
        sras2pol = 'arcpy.RasterToPolygon_conversion(Majorit_Con_1, RasterT_Majorit1,
"SIMPLIFY", "Value")'
        tool11 = GetToolName(sras2pol)
        toolList.append(tool11)

        #Accessing Describe Object
        ras2polDesc = arcpy.Describe(ras2pol)

        #Continuing appending lists
        tool11ST = getStartTime(ras2pol)
        stList.append(tool11ST)

        tool11ET = getEndTime(ras2pol)
        etList.append(tool11ET)
        tool11Input = str(getInputs(ras2pol))
        inputList.append(tool11Input)

        tool11Output = getOutputs(ras2pol)
        outputList.append(tool11Output)

        uniId11 = str(uuid.uuid4())
        uID.append(uniId11)

      #Twelth Process
        #making a feature layer
        featLayer1 = arcpy.MakeFeatureLayer_management(RasterT_Majorit1,
"RasterT_Majorit1__2_")

        #Converting to string for parsing tool name
        sfeatLayer1 = 'arcpy.MakeFeatureLayer_management(RasterT_Majorit1,
"RasterT_Majorit1__2_")'
        tool12 = GetToolName(sfeatLayer1)
        toolList.append(tool12)

        #Accessing Describe Object
        featLayer1Desc = arcpy.Describe(featLayer1)

        #Continuing appending lists
        tool12ST = getStartTime(featLayer1)
        stList.append(tool12ST)

        tool12ET = getEndTime(featLayer1)
        etList.append(tool12ET)

        tool12Input = str(getInputs(featLayer1))
        inputList.append(tool12Input)

        tool12Output = getOutputs(featLayer1)
        outputList.append(tool12Output)

        uniId12 = str(uuid.uuid4())
        uID.append(uniId12)

      #Thirteenth Process
        # Process: Select Layer By Location
        RasterT_Majorit1__3_ =
arcpy.SelectLayerByLocation_management("RasterT_Majorit1__2_", "INTERSECT", roads, "",
"NEW_SELECTION")
```

```
        #Converting to string for parsing tool name
        sRasterT_Majorit1__3_ =
'arcpy.SelectLayerByLocation_management("RasterT_Majorit1__2_", "INTERSECT", roads,
"", "NEW_SELECTION")'
        tool13 = GetToolName(sRasterT_Majorit1__3_)
        toolList.append(tool13)

        #Accessing Describe Object
        sRasterT_Majorit1__3_Desc = arcpy.Describe(RasterT_Majorit1__3_)

        #Continuing appending lists
        tool13ST = getStartTime(RasterT_Majorit1__3_)
        stList.append(tool13ST)

        tool13ET = getEndTime(RasterT_Majorit1__3_)
        etList.append(tool13ET)

        tool13Input = str(getInputs(RasterT_Majorit1__3_))
        inputList.append(tool13Input)

        tool13Output = getOutputs(RasterT_Majorit1__3_)
        outputList.append(tool13Output)

        uniId13 = str(uuid.uuid4())
        uID.append(uniId13)

      #Fourteenth Process
        #making a feature layer
        featLayer2 = arcpy.MakeFeatureLayer_management(RasterT_Majorit1__3_,
"RasterT_Majorit1__4_")

        #Converting to string for parsing tool name
        sfeatLayer2= 'arcpy.MakeFeatureLayer_management(RasterT_Majorit1__3_,
"RasterT_Majorit1__4_")'
        tool14 = GetToolName(sfeatLayer2)
        toolList.append(tool14)

        #Accessing Describe Object
        featLayer2Desc = arcpy.Describe(featLayer2)

        #Continuing appending lists
        tool14ST = getStartTime(featLayer2)
        stList.append(tool14ST)

        tool14ET = getEndTime(featLayer2)
```

```
    etList.append(tool14ET)

    tool14Input = str(getInputs(featLayer2))
    inputList.append(tool14Input)

    tool14Output = getOutputs(featLayer2)
    outputList.append(tool14Output)

    uniId14 = str(uuid.uuid4())
    uID.append(uniId14)


   #Fifthteenth Process
    # Process: Select Layer By Attribute
    RasterT_Majorit1__5_ =
arcpy.SelectLayerByAttribute_management("RasterT_Majorit1__4_",
"SUBSET_SELECTION", "Shape_Area >= 40469")

    #Converting to string for parsing tool name
    sRasterT_Majorit1__5_=
'arcpy.SelectLayerByAttribute_management("RasterT_Majorit1__4_",
"SUBSET_SELECTION", "Shape_Area >= 40469")'
    tool15 = GetToolName(sRasterT_Majorit1__5_)
    toolList.append(tool15)

    #Accessing Describe Object
    RasterT_Majorit1__5_Desc = arcpy.Describe(RasterT_Majorit1__5_)

    #Continuing appending lists
    tool15ST = getStartTime(RasterT_Majorit1__5_)
    stList.append(tool15ST)

    tool15ET = getEndTime(RasterT_Majorit1__5_)
    etList.append(tool15ET)

    tool15Input = str(getInputs(RasterT_Majorit1__5_))
    inputList.append(tool15Input)

    tool15Output = getOutputs(RasterT_Majorit1__5_)
    outputList.append(tool15Output)

    uniId15 = str(uuid.uuid4())
    uID.append(uniId15)
   #Sixteenth Process
    # Process: Copy Features
    cpyft = arcpy.CopyFeatures_management("RasterT_Majorit1__4_",
"C:\Users\Kcnil14\Documents\ArcGIS\Spatial Analyst\Stowe.gdb\final_site", "", "0", "0", "0")

    #Converting to string for parsing tool name
    scpyft= 'arcpy.CopyFeatures_management("RasterT_Majorit1__4_", "final_site", "", "0",
"0", "0")'
    tool16 = GetToolName(scpyft)
    toolList.append(tool16)

    #Accessing Describe Object
    cpyftDesc = arcpy.Describe(cpyft)

    #Continuing appending lists
    tool16ST = getStartTime(cpyft)
    stList.append(tool16ST)

    tool16ET = getEndTime(cpyft)
    etList.append(tool16ET)

    tool16Input = str(getInputs(cpyft))
    inputList.append(tool16Input)

    tool16Output = getOutputs(cpyft)
    outputList.append(tool16Output)

    uniId16 = str(uuid.uuid4())
    uID.append(uniId16)

    #generating provenance using generateXML
    generateProvXML(toolList, stList, etList, inputList, outputList, uID)

    '''for layers in toolList:
      print(layers)
    print "done" '''
```

B5. xmlGenerate.py

```
    import arcpy
    from lxml import etree
    from datetime import datetime
    import uuid
    from lxml.builder import *


    def generateProvXML(tool,start,end,inputs,outputs,uniID):
     #This function can be used to get user input for sources.
     #For this study set list of input sources will be used
     #myInSrc = userInputs(inputs) #TO GET USER INPUT
     #myInSrc = ['USGS','USGS','Kcnil14','Kcnil14/GeoStor'] FOR NDVI BATCH
     #myInSrc = ['GeoStor', 'USGS', 'Kcnil14/Geostor', 'Kcnil14', 'Kcnil14', 'Kcnil14', 'Kcnil14',
'Kcnil14']
     myInSrc = ['ArcGIS', 'ArcGIS','ArcGIS', 'Kcnil14', 'Kcnil14', 'Kcnil14', 'Kcnil14', 'Kcnil14',
'Kcnil14', 'Kcnil14', 'Kcnil14', 'Kcnil14', 'Kcnil14', 'Kcnil14', 'Kcnil14','Kcnil14']
    #first create variables to hold uris// Maybe add date source back in function later
     nsProv = 'https://www.w3.org/ns/prov#'
     nsXSI = 'https://www.w3.org/2001/XMLSchema-instance'
     nsXSD = 'https://www.w3.org/2001/XMLSchema'
     nsRL = 'http://local_host_/descript#'
     nsDC = 'http://purl.org/dc/terms/'

    #Namespace map is python dictionary the relates ns prefixes to ns ~ nmt.edu
     myNamespaces = {'prov': nsProv,
          'xsi': nsXSI,
          'xsd': nsXSD,
          'rl': nsRL,
          'dct': nsDC}
    #Counter for iteration of nodes
     counter = 0

    #Adding to QName = Wrapper for xml names
     rootName = etree.QName(nsProv, 'document')
     name = etree.QName(nsProv,'activity')
     starts = etree.QName(nsProv, 'startTime')
     ends = etree.QName(nsProv,'endTime')
     nput = etree.QName(nsRL, 'inputFile')
     oput = etree.QName(nsRL, 'outputFile')
     src = etree.QName(nsDC, 'creator') #Creator is == to wasAttributedTo, I can change this if I
want
     uid = etree.QName(nsRL, 'ID')
```

```
  rootElement = etree.Element(rootName,attrib = myNamespaces, nsmap = myNamespaces)

  xml2 = etree.ElementTree(rootElement)

  while counter < len(tool):
    result = etree.SubElement(rootElement, name, nsmap = myNamespaces)
    result.text = tool[counter]
    startTime = etree.SubElement(result, starts, nsmap = myNamespaces)
    startTime.text = start[counter]
    endTime = etree.SubElement(result, ends, nsmap = myNamespaces)
    endTime.text = end[counter]
    inputFile = etree.SubElement(result, nput, nsmap = myNamespaces)
    inputFile.text = inputs[counter]
    outputFile = etree.SubElement(result, oput, nsmap = myNamespaces)
    outputFile.text = outputs[counter]
    addID = etree.SubElement(result, uid, nsmap = myNamespaces)
    addID.text = uniID[counter]
    addSrc = etree.SubElement(result, src, nsmap = myNamespaces)
    addSrc.text = myInSrc[counter]
    counter = counter + 1

  stringXML = etree.tostring(rootElement,pretty_print = True)
  currentDateTime = datetime.now().strftime("%Y%m%d%H%M%S")

  wrkID = currentDateTime
  newXML = open(wrkID + '.xml','w')
  newXML.write(stringXML)

  csvBatch(tool,start,end,inputs,outputs,myInSrc,uniID)

def generateXML(tool,start,end, inputs, outputs):

  #Time Variables
  maToolStartTime = None
  maToolEndTime = None
  maProcessTime = None
  counter = 0

  #root node
  root = etree.Element('Results')

  #creating new document using ElementTree
  newXML = etree.ElementTree(root)
  #Adding elements
  while counter < len(tool):
    result = etree.SubElement(root, 'Result')
    result.text = 'Name='+'"'+tool[counter]+'"'
    startTime = etree.SubElement(result,'StartTime')
    startTime.text = start[counter]
    endTime = etree.SubElement(result,'EndTime')
    endTime.text = end[counter]
    inputFile = etree.SubElement(result,'InputFile')
    inputFile.text = inputs[counter]
    outputFile = etree.SubElement(result,'OutputFile')
    outputFile.text = outputs[counter]
    counter = counter + 1


  #Getting date and time for file name
  currentDateTime = datetime.now().strftime("%Y%m%d%H%M%S")
  ### Save to XML file
  outputXML = open(currentDateTime+".xml", 'w')
  newXML.write(outputXML, xml_declaration=True, encoding='utf-8')

def GetToolName(tool):
  #Determining position of tool start and end
  position1 = tool.find(".")
  position2 = tool.find("(",position1)
  toolName = tool[position1+1:position2]
  return toolName

def SetMapAlgebra():
  mapalgebra = "Map Algebra"
  return mapalgebra

def GetMapAlgebraInfo(tool):
  #Can be used for iteration if needed
  wordCount=tool.count("sa")

  #Determining start and end of Map Algebra expression
  position1 = tool.find("(")
  position2 = tool.find('"',position1)
  mapAlgebraExpression = tool[position1:position2-1]

  return mapAlgebraExpression


def getMapAlgebraInput(tool):
    #Parsing string for input, must pass it a string, not the tool
      info = GetMapAlgebraInfo(tool)

      position1 = info[info.find("("):info.find(")")]

      maInput = info.replace("(","")

      return maInput

    def getMapAlgebraOutput(tool):

      #Accessing describe object properties and using catalogPath to get output save of map
algebra expression
      #Must be the output of the MA process
      toolDescript = arcpy.Describe(tool)
      maOutput = toolDescript.catalogPath

      return maOutput

    def getStartTime(tool):
      #Acessing runtime messages.
      toolMessage = tool.getMessages()

      #Using .find() to locate positions

      position1 = toolMessage.find("Start Time")
      position2 = toolMessage.find("Succeeded")
      startTime = toolMessage[position1+12:position2-5]

      return startTime


    def getEndTime(tool):

      #Accessing the runtime messages
      toolMessage = tool.getMessages()

      #Use rfind because end time is towards end of message output

      position1 = toolMessage.rfind("2015")
      position2 = toolMessage.rfind("Succeeded")
      endTime = toolMessage[position2 + 13:position1]

      return endTime
```

```
def getInputs(tool):

  #Converting results object to string for parsing
  inputCount = tool.inputCount
  i=0
  myString = " "
  list1 = []
  #For process with only one input
  try:
    if inputCount == 1:
      inputs = tool.getInput(0)
      return str(inputs)
  #Returns the input layer and the bounding layer for clip
    elif inputCount==2:
      input1 = tool.getInput(0)
      input2 = tool.getInput(1)
      return str(input1),str(input2)
    else:
      inputCount >=3
      input1 = tool.getInput(0)
      input2 = tool.getInput(1)
      input3 = tool.getInput(2)
      myString = str(input1) + ";" + str(input2) + ";" + str(input3)
      return myString
  except:
    pass
    '''while i < inputCount:
      list1.append(tool.getInput(i))
    i = i + 1

  for item in list1:
    print item
    if item.startswith("C:"):
      myString = myString + item
    print myString
    return myString



    myString = myString + item
        return myString
      print myString
      i=i+1
      #Starts with depends on drive letter
      if str(input2.startswith("C")):
        inputs = str(input1) + " , " + str(input2)
        return inputs
      else:
        return str(input1)
      i=i+1



    x = 0
    while x < inputCount:
      input1 = tool.getInput(x)
      x=x+1
      input2 = tool.getInput(x)
      print str(input1)
      print str(input2)
      return str(input1), str(input2)
    while i < inputCount:
      inputs = tool.getInput(i)
      if i==0:
        input1 = tool.getInput(i)
      elif i ==2:
        input2 = tool.getInput(i)
      i=i+1
  return str(input1),str(input2)'''



def getOutputs(tool):

  outputCount = tool.outputCount
  i = 0
  while i < outputCount:
    toolOutputs = tool.getOutput(i)
    return str(toolOutputs)
    i = i+1



def uniID(tool):
  '''Generating a unique ID. I would like this to be sequential, but haven't
    figured out how to achieve this.'''
  unID = uuid.uuid4(tool)
  return unID

def userInputs(inputs):
  dSrc = []
```

```
    i = 0
    while i < len(inputs):
      source = raw_input('Enter data source for :' + inputs[i])
      dSrc.append(source)
      i = i+1
    return dSrc

  def csvBatch(tool,start,end,inputs,outputs,sources,uniID):

    myList = []
    i = 0
    newTxt = open('batchData.txt','a')

    while i < len(tool):
      longString = ""
      longString= longString + tool[i] + ',' + start[i] + ',' + end[i] + ',' + inputs[i] + ',' + outputs[i]
+ ',' + sources[i] + ',' + uniID[i]
      newTxt.write(longString)
      newTxt.write('\n')
      i = i+1


    newTxt.close()
```

Appendix C: Content Trust for additional workflows

C.1: Content trust for intermediate workflow

Resource	Metadata	Spatial Accuracy	Completeness
WATER_BASE_LAYER_ADEQ.shp	High	Low	High
Boundaries_COUNTIES_AHTD.shp	High	Low	High
ClipStreams.shp	Low	Low	High
Imgn3bw093_13.img	High	High	High
prj_raster	Low	Low	High
clipped_dem.img	Low	Low	High
fill_dem.img	Low	Low	High
slope_Raster.img	Low	Low	High
aspect_Raster.img	Low	Low	High
flowDir.img	Low	Low	High
flowAC.img	Low	Low	High

Table 1: Quality for intermediate workflow

$q = (2/3)+(2/3)+1+[(1/3)*8] = 5/11 = .45$

Resource	Bias
WATER_BASE_LAYER_ADEQ.shp	No perception of bias
Boundaries_COUNTIES_AHTD.shp	No perception of bias
ClipStreams.shp	No perception of bias
Imgn3bw093_13.img	No perception of bias
prj_raster	No perception of bias
clipped_dem.img	No perception of bias
fill_dem.img	No perception of bias
slope_Raster.img	No perception of bias
aspect_Raster.img	No perception of bias
flowDir.img	No perception of bias
flowAC.img	No perception of bias
W2	No perception of bias

Table 2: Bias for intermediate workflow

$b = (1*13)/13 = 13/13 = 1$

Resource	Bias
USGS	High
GeoStor	High
Workflow Creator	Low

Table 3: Authority for intermediate workflow

a = 2/3 = .67

T = (.67+1+.45)/3 = .71 = Trustworthy

C.2: Content trust for advanced workflow

Resource	Metadata	Spatial Accuracy	Completeness
elevation	High	Low	High
HillSha_elev2	Low	Low	High
rec_sites	High	Low	High
EucDist_rec_1	Low	Low	High
schools	High	Low	High
EucDist_scho1	Low	Low	High
Reclass_EucD2	Low	Low	High
Reclass_EucD1	Low	Low	High
Slope_Out	Low	Low	High
Reclass_Slop1	Low	Low	High
Weighte_Recl1	Low	Low	High
Con_Weighte_1	Low	Low	High
Majorit_Con_1	Low	Low	High
RasterT_Majorit1	Low	Low	High
RasterT_Majorit1__2_	Low	Low	High
roads	High	Low	High
RasterT_Majorit1__4	Low	Low	High
Stowe.shp	Low	Low	Low

Table 4: Quality for advanced workflow

q=(4*(2/3)+(14*(1/3))) = .41

Resource	**Bias**
elevation	No perception of bias
HillSha_elev2	No perception of bias
rec_sites	No perception of bias
EucDist_rec_1	No perception of bias
schools	No perception of bias
EucDist_scho1	No perception of bias
Reclass_EucD2	No perception of bias
Reclass_EucD1	No perception of bias
Slope_Out	No perception of bias
Reclass_Slop1	No perception of bias
Weighte_Recl1	No perception of bias
Con_Weighte_1	No perception of bias
Majorit_Con_1	No perception of bias
RasterT_Majorit1	No perception of bias
RasterT_Majorit1__2_	No perception of bias
roads	No perception of bias
RasterT_Majorit1__4	No perception of bias
Stowe.shp	No perception of bias
W3	No perception of bias

Table 5: Bias for advanced workflow

b=19/19 = 1

Resource	**Bias**
ESRI	High
Workflow Creator	Low

Table 6: Authority for advanced workflow.

a = 1/2 = .50

T = (.41+1+.50)/3 = .64 = Trustworthy

Appendix D: Mean elapsed time for each tool used in determining reliability for workflow trust

Tool	Workflow	Mean Elapsed Time in Seconds
Clip Management	1	.98
Map Algebra	1	14.36
Raster to Float	1	3.84
Project Raster	2	1
Extract by Mask	2	9.22
Clip	2	1.26
Fill	2	34.9
Aspect	2	10.2
Flow Direction	2	17.44
Slope	2	8.94
Flow Accumulation	2	166.7
Slope	3	1.86
Hillshade	3	1.78
Reclassify	3	1.9
Euclidian Distance	3	1.57
Weighted Overlay	3	2.76
Con	3	2.12
Majority Filter	3	1.78
Raster to Polygon	3	1.16
Select Layer by Location	3	.08
Select Layer by Attribute	3	.02
Make Feature Layer	3	.03
Copy Features	3	.12

Appendix E: State transition matrices and Observation matrices

Tool Name	Observation Number
Project raster	1
Extract by mask	2
Clip	3
Fill	4
Aspect	5
Slope	6
Flow direction	7
Flow accumulation	8

Table E.1: Assignment of workflow tools to observation number.

	(S_1,S_2)	**(S_2,S_3)**	**(S_3,S_4)**	**(S_4,S_5)**	**(S_5,S_6)**	**(S_6,S_7)**	**(S_7,S_8)**	**Total**
T to U	13	5	2	2	4	1	3	30
U to T	0	5	6	3	2	3	0	19
T to T	37	32	35	39	38	39	39	259
U to U	0	8	7	6	6	7	8	42
Total	50	50	50	50	50	50	50	350

Table E.2: Transition counts for intermediate workflow.

	O_1	**O_2**	**O_3**	**O_4**	**O_5**	**O_6**	**O_7**	**O_8**	**Total**
T	50	37	37	41	42	40	42	39	328
U	0	13	13	9	8	10	8	11	72
Total	50	50	50	50	50	50	50	50	400

Table E.3: Table displaying hidden state counts at a particular observation for intermediate workflow.

$$A = \begin{array}{c|cc} & T & U \\ \hline T & .90 & .10 \\ U & .31 & .69 \end{array} \qquad B = \begin{array}{c|cccccccc} & O_1 & O_2 & O_3 & O_4 & O_5 & O_6 & O_7 & O_8 \\ \hline T & .15 & .11 & .11 & .13 & .13 & .12 & .13 & .12 \\ U & 0 & .18 & .18 & .13 & .11 & .14 & .11 & .15 \end{array}$$

Figure E.1 : State transition and observation matrices estimated from provenance data for intermediate workflow.

	t_1	t_2	t_3	t_4	t_5
T	.075	.007425	.00078111	1.035331e-04	1.361184e-05
U	0	.001350	.00030132	3.718283e-05	3.961041e-06

	t_6	t_7	t_8
T	1.617429e-06	2.123393e-07	2.521292e-08
U	5.732023e-07	6.129777e-08	9.529408e-09

Table E.4: Forward probabilities for intermediate workflow given λ and an observation set.

	o_1t_1	o_2t_2	o_3t_3	o_4t_4	o_5t_5	o_6t_6	o_7t_7	o_8t_8
T	1	.85	.72	.74	.77	.74	.78	.73
U	0	.15	.28	.26	.23	.26	.22	.27

Table E.5: Decoded forward probabilities for intermediate workflow.

Tool Name	**Observation Number**
Euclidian Distance	1
Slope	2
Hillshade	3
Reclassify	4
Weighted Overlay	5
Con	6
Majority Filter	7
Raster to Polygon	8
Select by Location	9
Select by Attribute	10
Make Feature Layer	11
Copy Feature Layer	12

Table E.6: Assignment of workflow tools to observation number advanced workflow.

	(S_1,S_2)	**(S_2,S_3)**	**(S_3,S_4)**	**(S_4,S_5)**	**(S_5,S_6)**	**(S_6,S_7)**	**(S_7,S_8)**	**(S_8,S_9)**
T to U	0	2	6	2	2	2	6	1
U to T	11	0	2	7	2	2	2	5
T to T	38	47	41	41	46	46	42	43
U to U	1	1	1	0	0	0	0	1
Total	50	50	50	50	50	50	50	350

	(S_9,S_{10})	**(S_{10},S_{11})**	**(S_{11},S_{12})**	**(S_{12},S_{13})**	**(S_{13},S_{14})**	**(S_{14},S_{15})**	**(S_{15},S_{16})**	**Total**
T to U	6	3	2	1	3	4	10	50
U to T	1	6	4	2	1	3	3	51
T to T	42	40	44	47	46	43	37	643
U to U	1	1	0	0	0	0	0	6
Total	50	50	50	50	50	50	50	750

Table E.7: Transition counts for advanced workflow

	O_1	**O_2**	**O_3**	**O_4**	**O_5**	**O_6**	**O_7**	**O_8**	**O_9**	**O_{10}**	**O_{11}**	**O_{12}**	**Total**
T	87	47	43	144	44	48	43	46	48	49	93	40	732
U	13	3	7	6	6	2	7	4	2	1	7	10	68
Total	100	50	50	150	50	50	50	50	50	50	100	50	800

Table E.8: Observation counts for advanced workflow

$$\text{A} = \begin{array}{c|cc} & T & U \\ \hline T & .93 & .07 \\ U & .85 & .15 \end{array}$$

$$\begin{array}{c|cccccc} & O_1 & O_2 & O_3 & O_4 & O_5 & O_6 \\ \hline T & .1188525 & .06420765 & .05874317 & .1967213 & .06010929 & .06557377 \\ U & .1911765 & .04411765 & .1029412 & .08823529 & .08823529 & .02941176 \end{array}$$

$$\begin{array}{c|cccccc} & O_7 & O_8 & O_9 & O_{10} & O_{11} & O_{12} \\ \hline T & .05874317 & .06284153 & .06557377 & .06693989 & .1270492 & 0.05464481 \\ U & .1029412 & .05882353 & .02941176 & .01470588 & .1029412 & 0.1470588 \end{array}$$

Figure E.2: State transition and observation matrices estimated from provenance data for advanced workflow.

	t_1	t_2	t_3	t_4	t_5
T	.05942623	.016225309	1.161868e-03	6.714466e-05	1.387393e-05
U	.09558824	.003536396	7.351019e-05	9.507367e-06	5.405498e-07

	t_6	t_7	t_8	t_9	t_{10}
T	2.628633e-06	4.964358e-07	2.864389e-08	1.930595e-09	1.091416e-10
U	9.284624e-08	1.746452e-08	3.297369e-09	7.351992e-11	1.504688e-11

	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}
T	7.182255 e-12	4.704488 e-13	3.027487 e-14	3.633583 e-15	4.537593 e-16	2.443831 e-17
U	5.821732 e-13	1.735541 e-14	5.225695 e-16	2.262262 e-16	2.967637 e-17	5.325677 e-18

Table E.4: Forward probabilities for advanced workflow given λ and an observation set.

	o_1t_1	o_1t_2	o_2t_3	o_3t_4	o_4t_5	o_4t_6	o_4t_7	o_5t_8	o_6t_9	o_7t_{10}	o_8t_{11}
T	.38	.82	.94	.88	.96	.97	.97	.90	.96	.88	.93
U	.62	.18	.06	.12	.04	.03	.03	.10	.04	.12	.07

	o_9t_{12}	$o_{10}t_{13}$	$o_{11}t_{14}$	$o_{11}t_{15}$	$o_{12}t_{16}$
T	.96	.98	.94	.94	.82
U	.04	.02	.06	.06	.18

Table E.5: Decoded forward probabilities for intermediate workflow.

Appendix F: R Code

F.1. Calculations for simple model

```
> library(HMM)
> states = c("T","U")
> obs = c("ras2flt1","ras2flt2","map algebra","clip")
> initProb = c(.5,.5)
> transProb = matrix(c(.65,.53,.35,.47),2)
> obsProb = matrix(c(.14,.38,.30,.20,.45,.01,.11,.41),2)
> myModel = initHMM(states,obs,initProb,transProb,obsProb)
> forwardProb = forward(myModel,c("ras2flt1","ras2flt2","map algebra","clip"))
> fProb2 = exp(forwardProb)
```

F.2: Calculations for intermediate workflow model .

```
> states = c("T","U")
> obs = c("prj ras","extract","clip","fill","aspect","slope","fd","fa")
> initProb = c(.5,.5)
> transProb = matrix(c(.90,.31,.10,.69),2)
> obsProb = matrix(c(.15,0,.11,.18,.11,.18,.13,.13,.13,.11,.12,.14,.13,.11,.12,.15),2)
> myModel = initHMM(states,obs,initProb,transProb,obsProb)
> forwardProb = forward(myModel,obs)
> fProb2 = exp(forwardProb)
```

F.3: Calculations for advanced workflow model

```
> states = c("T","U")
>obs=c("O1","O2","O3","O4","O5","O6","O7","O8","O9","O10","O11","O12")
> initProb = c(.5,.5)
>transMatrix=matrix(c(.93,.85,.07,.15),2)
>obsMatrix=matrix(c(x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6,x7,y7,x8,y8,x9,y9,x10,y10,x11,y
11,x12,y12),2)
> myModel = initHMM(states,obs,initProb,transMatrix,obsMatrix)
>obs1=c("O1","O1","O2","O3","O4","O4","O4","O5","O6","O7","O8","O9","O10","O11","
O11","O12")
> forwardProb = forward(myModel,obs1)
> fProb2 = exp(forwardProb)
```

Appendix G: Generated Provenance Examples

G.1. Simple Workflow XML Example

```
<?xml version="1.0"?>
<prov:document xsd="https://www.w3.org/2001/XMLSchema"
xmlns:xsd="https://www.w3.org/2001/XMLSchema"
xsi="https://www.w3.org/2001/XMLSchema-instance"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" dct="http://purl.org/dc/terms/"
xmlns:dct="http://purl.org/dc/terms/" prov="https://www.w3.org/ns/prov#"
xmlns:prov="https://www.w3.org/ns/prov#" rl="http://local_host_/descript#"
xmlns:rl="http://local_host_/descript#">

    <prov:activity>
        RasterToFloat_conversion
        <prov:startTime>Sat Apr 25 12:38:10 </prov:startTime>
        <prov:endTime>Sat Apr 25 12:38:13 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\test.gdb\Input\LC80240362014113L
        GN00.tar\LC80240362014113LGN00\LC80240362014113LGN00_B5.TIF</rl:inputFile
        >
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Temp\b5test.flt</rl:outputFile>
        <rl:ID>b961e2ae-bbe1-4b31-aeaa-8cb524b8bce4</rl:ID>
        <dct:creator>USGS</dct:creator>
    </prov:activity>

    <prov:activity>
        RasterToFloat_conversion
        <prov:startTime>Sat Apr 25 12:38:14 </prov:startTime>
        <prov:endTime>Sat Apr 25 12:38:20 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\test.gdb\Input\LC80240362014113L
        GN00.tar\LC80240362014113LGN00\LC80240362014113LGN00_B4.TIF</rl:inputFile
        >
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Temp\b4test.flt</rl:outputFile>
        <rl:ID>e38d7587-42bc-4b36-895d-556cef5e9db3</rl:ID>
        <dct:creator>USGS</dct:creator>
    </prov:activity>

    <prov:activity>
        Map Algebra
        <prov:startTime>Sat Apr 25 12:38:21</prov:startTime>
        <prov:endTime>Sat Apr 25 12:38:33</prov:endTime>
        <rl:inputFile>arcpy.sa.MinusB5flt,B4flt))/arcpy.sa.PlusB5flt,B4flt)</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\test.gdb\Output\ndvi_Output</rl:ou
        tputFile>
        <rl:ID>094ee8b5-f575-4d85-88be-054bd4631209</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
    </prov:activity>

    <prov:activity>
        Clip_management
        <prov:startTime>Sat Apr 25 12:38:33 </prov:startTime>
        <prov:endTime>Sat Apr 25 12:38:34 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\test.gdb\Output\ndvi_Output;551945
        .600299715 3911928.85510621 555608.114799695
        3914858.14920625;C:\Users\Kcnil14\Documents\ArcGIS\test.gdb\Input\Damascus_CL\
        GeoStor\ADMIN_DBO_CITY_LIMITS_AHTD_polygon.shp</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\test.gdb\Output\clipped_NDVI</rl:
        outputFile>
        <rl:ID>7cfefcf0-be3f-4e83-9a75-d22fab33eb55</rl:ID>
        <dct:creator>Kcnil14/GeoStor</dct:creator>
    </prov:activity>
</prov:document>
```

G.2. Intermediate Workflow XML Example

```
<?xml version="1.0"?>
<prov:document xsd="https://www.w3.org/2001/XMLSchema"
xmlns:xsd="https://www.w3.org/2001/XMLSchema"
xsi="https://www.w3.org/2001/XMLSchema-instance"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" dct="http://purl.org/dc/terms/"
xmlns:dct="http://purl.org/dc/terms/" prov="https://www.w3.org/ns/prov#"
xmlns:prov="https://www.w3.org/ns/prov#" rl="http://local_host_/descript#"
xmlns:rl="http://local_host_/descript#">

<prov:activity>
        Clip_analysis
        <prov:startTime>Sun Apr 26 19:58:56 2015 Assembling Features... Reading Features...
         Cracking Feature</prov:startTime>
        <prov:endTime>Sun Apr 26 19:58:57 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Input\WATER_BASE_L
        AYER_ADEQ\WATER_BASE_LAYER_ADEQ.shp;C:\Users\Kcnil14\Documents\Arc
        GIS\Hydro.gdb\Input\FME_011F5D59_1422298114857_26956\GeoStor\Boundaries_C
        OUNTIES_AHTD.shp;</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\ClipStreams.shp</rl:outputFile>
        <rl:ID>2351e153-c762-4da9-8d87-a9889d69e2d5</rl:ID>
        <dct:creator>GeoStor</dct:creator>
</prov:activity>
```

<prov:activity>
	ProjectRaster_management
	<prov:startTime>Sun Apr 26 19:59:00 </prov:startTime>
	<prov:endTime>Sun Apr 26 19:59:00 </prov:endTime>
	<rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Input\n36w093\imgn36w093_13.img;PROJCS['NAD_1983_UTM_Zone_15N',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Central_Meridian',-93.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]];NEAREST</rl:inputFile>
	<rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\prj_raster </rl:outputFile>
	<rl:ID>ca32856a-e8b9-4361-bdaa-bcaa3675c6ec</rl:ID>
	<dct:creator>USGS</dct:creator>
</prov:activity>

<prov:activity>
	gp.ExtractByMask_sa
	<prov:startTime>Sun Apr 26 19:59:48 </prov:startTime>
	<prov:endTime>Sun Apr 26 19:59:56 </prov:endTime>
	<rl:inputFile>('C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\prj_raster', 'C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Input\\FME_011F5D59_1422298114857_26956\\GeoStor\\Boundaries_COUNTIES_AHTD.shp')</rl:inputFile>
	<rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\clipped_dem.img</rl:outputFile>
	<rl:ID>73cc2dbd-bbde-465c-a3bd-20c6076367c7</rl:ID>
	<dct:creator>Kcnil14/Geostor</dct:creator>
</prov:activity>

<prov:activity>
	gp.Fill_sa
	<prov:startTime>Sun Apr 26 19:59:57 </prov:startTime>
	<prov:endTime>Sun Apr 26 20:00:32 </prov:endTime>
	<rl:inputFile>('C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\clipped_dem.img', '')</rl:inputFile>
	<rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\fill_dem.img</rl:outputFile>
	<rl:ID>fdc26f8c-4a56-47af-932a-0f0de2486234</rl:ID>
	<dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>

```
        gp.Slope_sa
        <prov:startTime>Sun Apr 26 20:00:34 </prov:startTime>
        <prov:endTime>Sun Apr 26 20:00:41 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\fill_dem.img;DE
        GREE;1</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\slope_Raster.im
        g</rl:outputFile>
        <rl:ID>976bbd9b-674e-42ff-8f2e-2898314b39ba</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        gp.Aspect_sa
        <prov:startTime>Sun Apr 26 20:00:42 </prov:startTime>
        <prov:endTime>Sun Apr 26 20:00:52 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\fill_dem.img</rl:
        InputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\aspect_Raster.i
        mg</rl:outputFile>
        <rl:ID>f6b3aaf5-a94f-4083-8138-b7d3640c9dc6</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        gp.FlowDirection_sa
        <prov:startTime>Sun Apr 26 20:00:53 </prov:startTime>
        <prov:endTime>Sun Apr 26 20:01:10 </prov:endTime>
        <rl:inputFile>('C:\\Users\\Kcnil14\\Documents\\ArcGIS\\Hydro.gdb\\Output\\fill_dem.im
        g', 'false')</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\flowDir.img</rl:
        outputFile>
        <rl:ID>264d3a8b-f6bb-443c-8b51-0fe7df53a8e1</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        gp.FlowAccumulation_sa
        <prov:startTime>Sun Apr 26 20:01:12 </prov:startTime>
        <prov:endTime>Sun Apr 26 20:03:56 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\flowDir.img;;FL
        OAT</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Hydro.gdb\Output\flowAc.img</rl:
        outputFile>
        <rl:ID>c26cbbb8-6a19-4534-a09a-4d0371383a9c</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
```

```
</prov:activity>
</prov:document>
```

G.3. Advanced Workflow XML Example

```
<?xml version="1.0"?>
<prov:document xsd="https://www.w3.org/2001/XMLSchema"
xmlns:xsd="https://www.w3.org/2001/XMLSchema"
xsi="https://www.w3.org/2001/XMLSchema-instance"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" dct="http://purl.org/dc/terms/"
xmlns:dct="http://purl.org/dc/terms/" prov="https://www.w3.org/ns/prov#"
xmlns:prov="https://www.w3.org/ns/prov#" rl="http://local_host_/descript#"
xmlns:rl="http://local_host_/descript#">

<prov:activity>
        gp.HillShade_sa
        <prov:startTime>Wed Jul 01 14:11:27 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:30 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\elevation;315;45</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\HillSha_elev2</rl:outputFile>
        <rl:ID>6a4a0078-2f71-46df-90b3-4ef802e04aea</rl:ID>
        <dct:creator>ArcGIS</dct:creator>
</prov:activity>

<prov:activity>
        gp.EucDistance_sa
        <prov:startTime>Wed Jul 01 14:11:30 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:32 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\rec_sites;;30</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\EucDist_rec_1</rl:outputFile>
        <rl:ID>fbb8c189-b049-4aa5-92cd-411a4f259ea2</rl:ID>
        <dct:creator>ArcGIS</dct:creator>
</prov:activity>

<prov:activity>
        gp.EucDistance_sa
        <prov:startTime>Wed Jul 01 14:11:33 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:32 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\schools;;30</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\EucDist_scho1</rl:outputFile>
        <rl:ID>cdefc40d-f07c-4120-a904-b9682c1fcbf0</rl:ID>
        <dct:creator>ArcGIS</dct:creator>
</prov:activity>

<prov:activity>
        gp.Reclassify_sa
        <prov:startTime>Wed Jul 01 14:11:36 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:37 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\EucDist_scho1;Value;0 1694.7212890625001
        1;1694.7212890625001 3389.4425781250002 2;3389.4425781250002
        5084.1638671875007 3;5084.1638671875007 6778.8851562500004
        4;6778.8851562500004 8473.6064453125 5;8473.6064453125 10168.327734375
        6;10168.327734375 11863.049023437499 7;11863.049023437499 13557.770312499999
        8;13557.770312499999 15252.491601562499 9;15252.491601562499 16947.212890625
        10</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Reclass_EucD2</rl:outputFile>
        <rl:ID>11af5f6e-a6ba-48ef-bda1-fcf38241fad4</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        gp.Reclassify_sa
        <prov:startTime>Wed Jul 01 14:11:38 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:40 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\EucDist_rec_1;Value;0 1352.92021484375 10;1352.92021484375
        2705.8404296875001 9;2705.8404296875001 4058.7606445312504
        8;4058.7606445312504 5411.6808593750002 7;5411.6808593750002
        6764.60107421875 6;6764.60107421875 8117.5212890624998 5;8117.5212890624998
        9470.4415039062496 4;9470.4415039062496 10823.36171875 3;10823.36171875
        12176.281933593751 2;12176.281933593751 13529.2021484375 1</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Reclass_EucD1</rl:outputFile>
        <rl:ID>bec9bce2-7ba7-41ef-af14-fb2e70ba34ee</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        gp.Slope_sa
        <prov:startTime>Wed Jul 01 14:11:40 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:42 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\elevation;DEGREE;0.3048</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Slope_Out</rl:outputFile>
        <rl:ID>ceed1eab-6f48-4dd1-8ec5-77fcb4fb8b6e</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        gp.Reclassify_sa
        <prov:startTime>Wed Jul 01 14:11:43 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:44 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Slope_Out;Value;0 4.7758632659912106 10;4.7758632659912106
        9.5517265319824212 9;9.5517265319824212 14.327589797973632
        8;14.327589797973632 19.103453063964842 7;19.103453063964842
        23.879316329956055 6;23.879316329956055 28.655179595947267
        5;28.655179595947267 33.431042861938479 4;33.431042861938479
        38.206906127929692 3;38.206906127929692 42.982769393920904
        2;42.982769393920904 47.758632659912109 1</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Reclass_Slop1</rl:outputFile>
        <rl:ID>272e52d8-f687-43fc-8299-2058dc6d71ce</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        gp.WeightedOverlay_sa
        <prov:startTime>Wed Jul 01 14:11:45 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:48 </prov:endTime>
        <rl:inputFile>('C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Reclass_EucD2' 25 'Value' (1 1; 2 2; 3 3; 4 4; 5 5; 6 6; 7 7; 8 8; 9 9;
        10 10;NODATA NODATA); 'C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Reclass_EucD1' 50 'Value' (1 1; 2 2; 3 3; 4 4; 5 5; 6 6; 7 7; 8 8; 9 9;
        10 10;NODATA NODATA); 'C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Reclass_Slop1' 13 'Value' (1 Restricted; 2 Restricted; 3 Restricted; 4
        4; 5 5; 6 6; 7 7; 8 8; 9 9; 10 10;NODATA NODATA);
        'C:\Users\Kcnil14\Documents\ArcGIS\Spatial Analyst\Stowe.gdb\landuse' 12
        'LANDUSE' ('Brush/transitional' 5; 'Water' Restricted; 'Barren land' 10; 'Built up' 3;
        'Agriculture' 9; 'Forest' 4; 'Wetlands' 1;NODATA NODATA));1 10 1</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Weighte_Recl1</rl:outputFile>
        <rl:ID>b75686c9-b666-416b-827f-d3ce5b35d613</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        gp.Con_sa
        <prov:startTime>Wed Jul 01 14:11:48 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:50 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Weighte_Recl1;C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Weighte_Recl1;</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Con_Weighte_1</rl:outputFile>
        <rl:ID>2e08557e-50fe-4f61-acb7-     20887e586da8</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity >
        gp.MajorityFilter_sa
        <prov:startTime>Wed Jul 01 14:11:51 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:53 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Con_Weighte_1;EIGHT;MAJORITY</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Majorit_Con_1</rl:outputFile>
        <rl:ID>6b37d3b9-f291-4eda-950f-d562c8a312de</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        RasterToPolygon_conversion
        <prov:startTime>Wed Jul 01 14:11:53 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:54 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\Majorit_Con_1;true;Value</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\RasterT_Majorit1</rl:outputFile>
        <rl:ID>4a87959d-588f-43e6-a1b8-28bb3453b4be</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        MakeFeatureLayer_management
        <prov:startTime>Wed Jul 01 14:11:55 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:55 </prov:endTime>
        <rl:inputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.gdb\RasterT_Majorit1;;</rl:inputFile>
        <rl:outputFile>RasterT_Majorit1__2_</rl:outputFile>
        <rl:ID>14ae135d-f66a-44fb-9f3e-6ed46b66e919</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        SelectLayerByLocation_management
        <prov:startTime>Wed Jul 01 14:11:55 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:55 </prov:endTime>
        <rl:inputFile>RasterT_Majorit1__2_;INTERSECT;C:\Users\Kcnil14\Documents\ArcGIS
        \Spatial Analyst\Stowe.gdb\roads</rl:inputFile>
        <rl:outputFile>RasterT_Majorit1__2_</rl:outputFile>
        <rl:ID>4c4e0aa0-5b05-4c62-8f7f-09c0f27f7bab</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        MakeFeatureLayer_management
        <prov:startTime>Wed Jul 01 14:11:55 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:55 </prov:endTime>
        <rl:inputFile>RasterT_Majorit1__2_;;</rl:inputFile>
        <rl:outputFile>RasterT_Majorit1__4_</rl:outputFile>
        <rl:ID>403a4cc5-735d-4177-93a3-35a14dc2ed6c</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        SelectLayerByAttribute_management
        <prov:startTime>Wed Jul 01 14:11:55 </prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:55 </prov:endTime>
        <rl:inputFile>RasterT_Majorit1__4_;SUBSET_SELECTION;Shape_Area >=
        40469</rl:inputFile>
        <rl:outputFile>RasterT_Majorit1__4_</rl:outputFile>
        <rl:ID>90ff5e99-05be-4bc5-a0d8-61299fb8d6e7</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>

<prov:activity>
        CopyFeatures_management
        <prov:startTime>Wed Jul 01 14:11:55 2015 WARNING 000117: Warning empty output
        genera</prov:startTime>
        <prov:endTime>Wed Jul 01 14:11:55 </prov:endTime>
        <rl:inputFile>RasterT_Majorit1__4_;;0</rl:inputFile>
        <rl:outputFile>C:\Users\Kcnil14\Documents\ArcGIS\Spatial
        Analyst\Stowe.shp</rl:outputFile>
        <rl:ID>0e020452-6a8d-4b8b-90e2-5fa9b3c7c55e</rl:ID>
        <dct:creator>Kcnil14</dct:creator>
</prov:activity>
</prov:document>
```