

Report on the GeoNIS meeting at the LTER Network Office, January 8-11, 2013

The following people were in attendance: Jamie Hollingsworth (BNZ), Aaron Stephenson (NTL), Adam Skibbe(KNZ), Theresa Valentine(AND), Ron Beloin (contractor), and Inigo San Gil. Additionally the team met with Mark Servilla, Dez Wyman, and Marshall White from the Network Office.

The GIS Work Group met January 8-10, 2014 for a 2 ½ day meeting at the LTER Network Office (LNO). The purpose of the meeting was to develop the requirements for workflows for ingesting PASTA data into the GeoNIS, and provide direction and coordination between LNO PASTA developers, system administrators for the GeoNIS server, the GeoNIS programming contractor, and the LTER IM site representatives that have worked on designing the GeoNIS module. Theresa Valentine also worked on testing and validating EML for the LandSat database prepared by Kyle Cavanaugh.

The team completed the following products:

1. Requirement documents for contractor (includes workflows and user application)
  - a. Deliverable document
  - b. Workflow diagram for SpatialVector data sets.
  - c. Preliminary workflow diagram for SpatialRaster data sets.
  - d. Identified quality checks for spatial data. These checks will be provided to PASTA as a service to check spatial data prior to injection into PASTA.
2. Upgraded spatial data server at LNO and designed directory structure.
3. Worked with Marshal White to update LTERMapS application to latest version.
4. Request submitted to the Pasta development team to modify the listing API to allow “gets” by data type class (which includes spatial vector, spatial raster) (data package manager)
5. Communication strategy for internal working group, LTER sites, and LTER IM Committee.
6. List of value added products and priorities.
7. Plan of action with assignments for team members

Participants tested access to the LNO system, worked with test EML documents and datasets, and discussed implementation of LTERMapS Phase 2. The ESRI2EML metadata stylesheet and the Best Practices for Documenting Spatial Data document were updated

Next Steps include:

1. the development of a GeoNIS workflow implementation plan by the contractor,
2. remote access to GEONIS server for team, developing sample data sets (Jamie to lead)
3. Meeting notes, workshop report, schedule next conference call (Theresa)
4. ESRI metadata and EML mapping (Theresa and Ron)
5. Raster Testing: continue working on solution with ESRI (team)

Detailed notes from the meeting are available at the LTER IM Webpage:  
<http://im.lternet.edu/project/GEONIS>

Notes from Meeting with Mark S. on 1/9/2013:

1. GeoNIS Team requests that the Pasta development team modify the listing API to allow “gets” by data type class (which includes spatial vector, spatial raster) (data package manager)
2. The spatial QA check should be called by PASTA during the validation stage. Spatial QA checks we are considering include consistency between EML and the data, internal consistency of the data, things that are currently not being checked on spatial data types in the congruency checker. The GEONIS will write the QA checks in such a way that could be turned into a web service that could be called from PASTA/data congruency folks.
3. The goal is to integrate the GeoNIS into PASTA more closely. Currently we’ll be working with Level 1 EML, but in the future, we most likely will be working with site EML... (before it gets into PASTA.. see QA above)
4. We should be looking at Open Source/GDAL for spatial quality checks, but as a service we can do what we want...so Ron will look into this. May need to add some additional software (Jamie and Ron)
5. Scopes might change... and we don’t need to keep the knb-lter as long as we can recreate the package-id.
6. Authorization: Need to figure out users/access maybe make a geonis user.. we need contact James B. about this. Could help with keeping track of activity
7. Data accessibility: GeoNIS would only access data that is publicly available, however the quality checks should still work on restricted data sets.
- 8.

Background:

The LNO has provided funding for a programmer to code the workflows for ingesting PASTA data into the GeoNIS, creating web services for value added products, automating the submission of said products into PASTA, and to develop web mapping applications for cross site and site data. This product oriented workgroup has been integrated closely with the GeoNIS project and with efforts by individual LTER site data preparation efforts (funded through 2012 IM supplements

The purpose of the GeoNIS project is to provide the integration of geospatial data into the NIS through linkages with EML and PASTA. The GeoNIS is intended to provide dynamic access spatial data stored within the Pasta system, support value-added products, and include the ability to generate synthetically derived data products which will be added to the network data portal. By integrating GIS data across sites into one repository, the GeoNIS will provide researchers with the ability to create new products and services, and provide the spatial framework for cross-site science.

The GeoNIS project started as a subgroup of the LTER GIS Working Group. The group gathered momentum for the 2009 All Scientist meeting and conducted a successful workshop. The group applied for a post-ASM workshop funds and held a meeting at the LNO in 2010 where they successfully transferred and deployed the LTERMapS application to the LTER Network website (<http://www.lternet.edu/map/index.php>). Subsequent workshops were held in 2011 and 2012. These two workshops provided the foundation for the development of the GeoNIS to provide workflow tools to extract spatial data from site level EML housed in the PASTA framework, develop web mapping services for individual LTER sites, and cross site efforts ( StreamChemDB and ClimDB/HydroDB). Products from the 2011/12 workshops included Best Practices for Documenting Spatial Data, updated LTERmapS interface, and a conceptual diagram for the GeoNIS, with defined components.

### 3. Products:

- Requirement documents for contractor (includes workflows and user application)
- Map template standards for symbology and cartography for LTER sites and cross site efforts.
- Plan for testing data flow from eml through PASTA, into the GeoNIS, with derived products flowing back through PASTA.

### 4. Tasks:

- Continue development of map templates for a common symbology and cartography for LTER Sites.
- Define workflows for the following:
  - data ingestion from PASTA (including process for updated EML)
  - conversion of base data to standard all-LTER layers
  - creation of web mapping services
  - derived product documentation back into PASTA
- Meet with LNO staff to insure that server is operational, and to further discussions on PASTA integration.
- Develop specifications for user applications.
- Develop plan for integration of LTER Landsat images.

## GEONIS Project Report

Theresa Valentine, Andrews LTER

September 2013

The GeoNIS working group, coordinated with University of New Mexico to contract with Ron Beloin to automate a workflow to bring PASTA data with a spatial component (SpatialRaster and SpatialVector EML tags) into a centralized geodatabase, quality check the EML and data, and create web services for each LTER site. The contractor also created a user interface where sites can view their spatial data, link to map and image web services, and view a report on their data (check for errors). The programs are developed as tools within the esri toolbox structure (programed in Python), and are located on a server in the LNO office. The application is programmed with the ArcGIS Javascript API, and is available at the following link: <http://maps3.lternet.edu/geonis/>.

Specifications for the projects were developed at a meeting with the contractor, GeoNIS working group, and LNO staff in January of 2013. Additional on-line meetings were held throughout the contract period, and a demonstration of the capabilities of the system was presented to the Information Managers Committee and during a working group breakout session, at their 2013 annual meeting in Fairbanks, Alaska.

Following discussions with LNO staff at the LTER Information Managers Committee in Fairbanks, a value added product was discussed to provide a geoprocessing service that PASTA could initiate for a data check, prior to data being placed in the production PASTA portal. The current GeoNIS tools work in production and testing mode. It would be possible to have sites submit data to the testing mode before submitting their data to PASTA. The contractors did not have time to complete this task on this contract. They did outline a process for the task, and estimate that it would take about 80 hours to complete the task at a cost of \$3,200.

The GeoNIS working group has committed to monitoring the project, and will be moving it to production mode (where the workflow runs on a schedule, searching PASTA for new spatial data, and running the workflow and updating the web services).

The GeoNIS working group would like to acknowledge the support it received from the LTER Network Office to fund this development. The project has helped sites document their spatial data with EML, incorporating said data into PASTA, and provides services to the sites for their spatial data. In addition, several LTER sites provided sample datasets for testing, and worked to improve the workflow and best practices for documenting spatial data.

Members of the GeoNIS working group are: Adam Skibbie (KNZ), Theresa Valentine (AND), Jamie Hollingsworth (BNZ), and Aaron Stephenson (NTL). Contractors were Ron Beloin and Jack Peterson

The following are reports on the progress of the project, by the contractor.

## PASTA spatial data workflow

### Deliverable Report

Date: August 30, 2013

This document describes progress on the workflow, reporting web page and services, and testing as of August 30, 2013.

For the value add product, a service for spatial data checking, we performed some design and analysis for this product but did not have time to implement it. The workflow code could be branched from the main trunk of the code, and this branch would be given its own set of file directories, database tables, feature datasets in the geodatabase, and database connection credentials. Minor modifications would be made to this branched code to allow the workflow to run as a geoprocessing task. It would be a single Python script that performed the entire workflow. It would accept a single URL to an EML file for input, and send its report findings to an output parameter. A geoprocessing task would be created for this script. This task would be set up as an asynchronous task, and limited to a single instance. The PASTA server could use the ESRI geoprocessing REST API to call the task and get the output. The script will simply discard files and delete database entries once it has generated the report findings.

The image service is now being refreshed by re-publishing the mosaic raster datasets when new raster data are added, similar to the map service refreshing.

Discussions were held on naming conventions and how the workflow treats the various EML elements and other sources having to do with naming—Title, Entity Name, Object Name, zip file name, and actual file name of data.

Finally, additional fixes were made to the workflow, and an issue where the workflow would lose its database connection was investigated.

## Testing

Currently, all available data are being processed through the testing database as we continue to analyze the source of errors for those packages that don't complete the workflow.

XSL:

- metadataMerge.xsl
- emlSubset.xsl
- emlSubsetToDataFields.xsl
- emlSubsetToSupp.xsl

Python:

- Geonis PASTA Processing.pyt (Python toolbox)
- Ino\_geonis\_base.py

lno\_geonis\_wf.py  
geonis\_pyconfig.py  
geonis\_log.py  
geonis\_helpers.py  
geonis\_emlparse.py  
geonis\_postgresql.py  
main.py (new, runs entire workflow)

SQL:

config.sql  
functions.sql  
tables.sql  
views.sql

The source files are located on Maps3, and in the source code repository GitHub (<https://github.com/rbeloin/pasta2geonis.git>).

## Deliverable Report

Date: August 21, 2013

This document describes progress on the workflow, reporting web page and services, and testing as of August 19, 2013.

A demo was held for the all scientists meeting in Alaska, and support was provided leading up to, and during, the demo. Most of the effort for this period went into creating and refining the web reporting function, which allows data managers to query and browse the Geonis server for the status of their spatial data.

Numerous bug fixes and exception handling code for unexpected input, such as double sets of quotes, invalid characters, and extra directory levels within zip files.

### **Reporting Web Page**

A dynamic web page was created, along with a map query service, which allows a user to browse spatial datasets by site and name. Map data are displayed on the page as layers are selected. The page also serves as feedback to the data managers on the status of their data and any errors generated by the workflow process. A third use of the web page is to act as a template and demonstration for any custom web pages that one may want to create using the spatial data services running on the Geonis server.

An image service was created for raster type datasets. Many individual raster datasets are combined in an image service for the LTER site. The reporting web page was then enhanced to allow query and display of individual images from the image service.

The error reporting system itself was re-structured, to allow more flexibility and to better support the web reporting page. In addition, the re-design facilitates another, upcoming value add product, which is

a server-to-server service for examining spatial data as it enters the PASTA system, and reporting results back to the PASTA server.

See <http://maps3.lternet.edu/geonis/report.html>

## Testing

Currently, all available data are being processed through the testing database as we continue to analyze the source of errors for those packages that don't complete the workflow.

XSL:

- metadataMerge.xsl
- emlSubset.xsl
- emlSubsetToDataFields.xsl
- emlSubsetToSupp.xsl

Python:

- Geonis PASTA Processing.pyt (Python toolbox)
- lno\_geonis\_base.py
- lno\_geonis\_wf.py
- geonis\_pyconfig.py
- geonis\_log.py
- geonis\_helpers.py
- geonis\_emlparse.py
- geonis\_postgresql.py
- main.py (new, runs entire workflow)

SQL:

- config.sql
- functions.sql
- tables.sql
- views.sql

The source files are located on Maps3, and in the source code repository GitHub (<https://github.com/rbeloin/pasta2geonis.git>).

Date: July 29, 2013

This document describes the fourth iteration of the workflow product as of July 28, 2013, and the development of the reporting value added product.

The conceptual steps involved in collecting PASTA spatial data sets and storing and serving the data (the workflow) can be listed as follows (evolved from the first deliverable):

1. Discovery and retrieval of data not yet processed
2. Unpacking of data and parsing the EML file
3. Data checks
4. Load vector type data
5. Augment metadata for vector data

6. Load raster type data
7. Augment metadata for raster data
8. Update map documents, master mosaic datasets
9. Refresh map and image services and update layer information for queries
10. Update a database with tracking and progress information
11. Compress and archive work directories; perform other cleanup tasks

In the fourth iteration of the workflow, improvements were made to all of the tools, as numerous spatial datasets were tested, uncovering opportunities to improve the workflow. Clean up code was added, and a script was created to run the entire workflow outside of the ArcCatalog environment (although it is still possible to run individual tools or chains of tools inside ArcCatalog). Raster support was added to the workflow, supporting a large number of raster storage types. An image service was set up for serving raster data. The reporting system was further enhanced, including a web page that leads the data owner to the status of their spatial data after the workflow has processed it, notifying them via email. At this point, the workflow is functionally complete, with planned improvements to peripheral products such as reporting, map service clients, and searching underway.

## Details

Improvements and clean up. Many large and small changes were made to the entire workflow, as various datasets highlighted where the workflow needed to be modified to accommodate the diversity of packages and EML. The workflow required new code to ensure that any remnants of packages processed earlier, or in earlier versions, were carefully removed to avoid name collisions. This included not only files and folders, but map layers and geodatabase features.

Raster support. The workflow now processes raster datasets of all kinds, placing them in Mosaic Datasets within File Geodatabases. This is the recommended method for organizing raster data with ArcGIS 10. Image services are set up for delivering raster data to outside clients.

Reporting. More detailed information is now collected during the workflow, and reports are now more informative and useful even when the data were successfully processed. A web page was set up with JavaScript code to request the report information via a query service handled by ArcGIS server. In this way, it was not necessary to add application server code to the GEONIS server.

## Testing

A this stage, over one hundred spatial datasets have been processed from sites NTL, NWT, KNZ, AND, and BNZ, as we continue to analyze the source of errors for those packages that don't complete the workflow.

XSL:

- metadataMerge.xsl
- emlSubset.xsl
- emlSubsetToDataFields.xsl
- emlSubsetToSupp.xsl

Python:

Geonis PASTA Processing.pyt (Python toolbox)

lno\_geonis\_base.py

lno\_geonis\_wf.py

geonis\_pyconfig.py

geonis\_log.py

geonis\_helpers.py

geonis\_emlparse.py

geonis\_postgresql.py

main.py (new, runs entire workflow)

SQL:

config.sql

functions.sql

tables.sql

views.sql

The source files are located on Maps3, and in the source code repository GitHub (<https://github.com/rbeloin/pasta2geonis.git>).

Date: June 23, 2013

This document describes the third iteration of the workflow product as of June 23, 2013.

The conceptual steps involved in collecting PASTA spatial data sets and storing and serving the data (the workflow) can be listed as follows (evolved from the first deliverable):

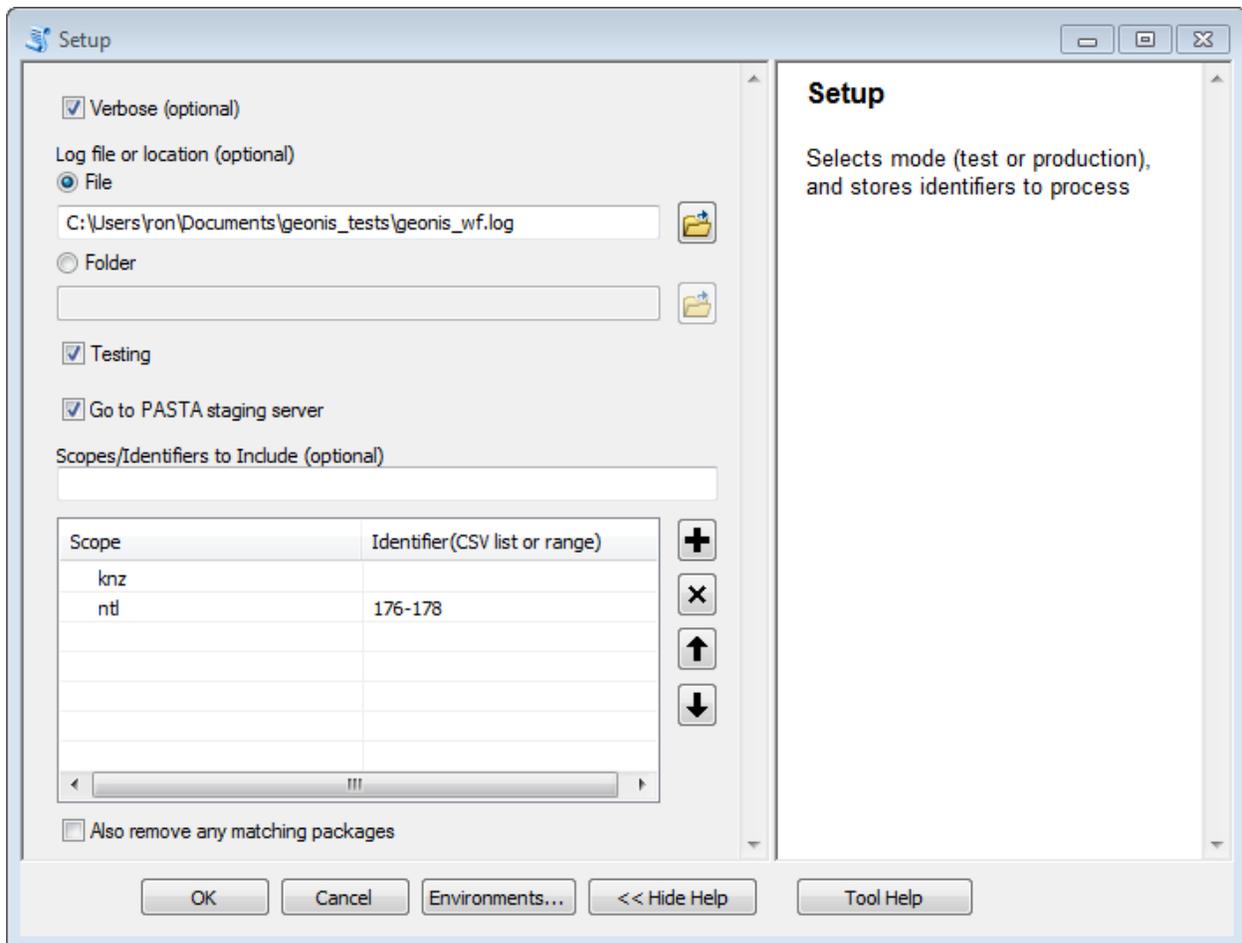
1. Discovery and retrieval of data not yet processed
2. Unpacking of data and parsing the EML file
3. Data checks
4. Load vector type data
5. Augment metadata for vector data
6. Load raster type data
7. Augment metadata for raster data
8. Update map documents, master mosaic datasets
9. Refresh map and image services and update layer information for queries
10. Update a database with tracking and progress information
11. Compress and archive work directories; perform other cleanup tasks

In the third iteration of the workflow, a system was created to allow both testing workflows and production workflows to exist on the same server. A new tool was created, the Setup tool, to select testing or production modes. Database tables and stored procedures were created to temporarily store information on data entities being superseded by newer versions. Configuration data has been moved to the database. Reporting of package errors was enhanced. Restarting of map services was further developed. Finally, new checks were added to the data checking tool.

## Details

Test and Production Modes. The SDE database was modified to create two versions of all workflow tables, and the workflow code was modified to support two versions of spatial data, map services, image services, raster storage, and the workflow tables.

Setup Tool. To support the two modes, a new tool was created (see figure below). It allows the user to select the mode (Testing check box), to select specific packages or range of packages, to select the PASTA staging server or production server as a source (third check box), and to clean up selected packages from either the test workflow or the production workflow (bottom check box). In the figure below, all packages from the scope 'knz' will be processed, and the latest versions of packages ntl.176 through ntl.178 will be processed. Not specifying any packages will cause all packages to be processed. The tool performs the clean up task immediately, and stores the other choices for when the workflow is started. Therefore, it can be run simply to remove a specific package or packages from the GEONIS system.



Tables for Superseded Versions. Tables were created that mirror the workflow tables to store the records of data entities that are being superseded by newly processed versions. These tables are then used in the workflow to treat new revisions somewhat differently, and to locate map layers and storage paths for the old versions.

Reporting. As errors are encountered in the processing of spatial data, the code attempts to capture all error messages and activities and stores this information in the entity table or package table. A database view presents these error reports. A query service was created to make this information available as a REST service. Also, the contact information is extracted from the EML file, and an email is composed indicating that a spatial data entity failed to be processed. The email contains a link to a web page (under development) served by the GEONIS server that will use the REST interface to retrieve and display the error report. If an error is encountered during processing, the EML data contact receives an email with a link to a page with the error messages.

Data Checks. Attribute name agreement checks between the EML and the spatial entity were added. Also, checks for suitable precision and scale values set for floating point attributes were added to the data checking tool.

## Testing

The workflow runs completely on the GEONIS server for vector data, although occasionally errors in the EML or data entity highlight where more error checking can be added. As we test more actual packages, more opportunities to detect errors in data, and fix errors in the workflow, will present themselves.

The source files for this deliverable are as follows. SQL code for creating tables, views, and stored procedures has been added, making it possible to completely rebuild all workflow database items from the files:

XSL:

- metadataMerge.xsl
- emlSubset.xsl
- emlSubsetToDataFields.xsl
- emlSubsetToSupp.xsl

Python:

- Geonis PASTA Processing.pyt (Python toolbox)
- lno\_geonis\_base.py
- lno\_geonis\_wf.py
- geonis\_pyconfig.py
- geonis\_log.py
- geonis\_helpers.py
- geonis\_emlparse.py
- geonis\_postgresql.py

SQL:

- config.sql
- functions.sql
- tables.sql
- views.sql

The source files are located on Maps3, and in the source code repository GitHub (<https://github.com/rbeloin/pasta2geonis.git>).

Date: March 29, 2013

This document describes the second iteration of the workflow product as of March 29, 2013.

The conceptual steps involved in collecting PASTA spatial data sets and storing and serving the data (the workflow) can be listed as follows (evolved from the first deliverable):

12. Discovery and retrieval of data not yet processed
13. Unpacking of data and parsing the EML file
14. Data checks
15. Load vector type data
16. Augment metadata for vector data
17. Load raster type data
18. Augment metadata for raster data
19. Update map documents, master mosaic datasets
20. Refresh map and image services and update layer information for queries
21. Update a database with tracking and progress information
22. Compress and archive work directories; perform other cleanup tasks

In this second iteration of the workflow process implementation, steps 1 through 10 are either partially or functionally complete, and are undergoing testing. Also in this iteration, interaction with Postgresql database tables and views has been added, to track and manage progress of the data processing.

All of the tools and supporting Python code has been re-factored to use the database tables, and to use XSL transformations for more transparent conversion of EML file data to parameters for the workflow, and/or supplemental metadata for the spatial data hosted by the GEONIS server.

## **Details**

Step 1. A Python tool has been created to use the PASTA REST API to discover packages that exist and contain spatial data. This tool makes new records in a “package” table for any package found to exist on the PASTA server. The table is updated with counts of data entities found in new packages (if any), and the EML files are downloaded for those packages that do contain spatial data. Views or queries on this table inform the workflow or a user when packages have been discovered, which have been examined for spatial data, which have spatial data, and which EML files have been downloaded.

Steps 2 – 7. The tools implementing these steps have been re-factored to make updates to the database as they process the data, and to use XSL transformations for work involving the parsing or merging of EML and other XML data. They have been updated to handle cases of multiple data entities per package as well.

Step 8. A Python tool was created to add vector data to an existing map (MXD) document, using the geodatabase data as the source of the map layer.

Step 9. A Python tool was created to stop and rebuild a running map service, necessary when changes are made to the corresponding MXD document. At this iteration, each scope (site) has an MXD document and a map service (only KNZ and NTL at this time, while we are testing).

Step 10. During the workflow, records are added to the “entity” table, one for each data entity found. This table tracks the success or failure of the data entity, and its ultimate storage location. Records are added to another table to store service information and fields to support client queries in order to locate spatial data within the services.

Other products of this deliverable include the design and implementation of database tables and views, and a map service specifically for query of map layer information.

## Testing

The following tests have been run successfully on Maps3.

1. Limit Step 1 to two scopes: KNZ and NTL.  
Result: Over 400 packages discovered, 39 with spatial data, many with more than one data entity per package. Package table updated correctly.
2. Limit rest of workflow to six of the 39 packages and run steps 2 through 10.  
Result: 9 data entities downloaded and processed. Two generated errors while loading into the geodatabase (errors not related to the workflow), remaining correctly stored, added to MXD documents, and appeared in refreshed map services. All tables updated correctly, including entities that failed part of the process. Query service correctly updated and indicates correct map layers for the vector data.
3. In other testing, it was demonstrated that raster data is correctly processed in steps 2 through 7. (Image services are still in design.)

The source files for this deliverable are as follows:

XSL:

- metadataMerge.xsl
- emlSubset.xsl
- emlSubsetToDataFields.xsl
- emlSubsetToSupp.xsl

Python:

- Geonis PASTA Processing.pyt (Python toolbox)
- lno\_geonis\_base.py
- lno\_geonis\_wf.py
- geonis\_pyconfig.py
- geonis\_log.py
- geonis\_helpers.py
- geonis\_emlparse.py
- geonis\_postgresql.py

The source files are located on Maps3, and in the source code repository GitHub (<https://github.com/rbeloin/pasta2geonis.git>).

Date: February 15, 2013

This document describes the workflow product iteration as of the above date. As with iterative development methods, the goal was to complete as much as the workflow as possible as a “first pass”, without concern for the completeness of each module.

The steps or modules involved in collecting PASTA spatial data sets and storing and serving can be listed as follows:

23. Retrieval of data not yet processed
24. Unpacking of data and parsing the EML file
25. Data checks
26. Load vector type data
27. Augment metadata for vector data
28. Load raster type data
29. Augment metadata for raster data
30. Update maps, master mosaic datasets, other aspects of services using the data

Overall, the implementation consists of a number of Arcgis Python tools, roughly corresponding to the list above. In general, a tool accepts a list of working directories, does its tasks, and passes on the list of directories, omitting those directories where errors occurred. For example, if a data set failed step 3, data checks, it would not be passed to step 4. This scheme allows for the tools to be chained in a model or Python script, where the output of one becomes the input of the next. The first tool in the chain can set a log file location. Tools can also be run individually, for testing or manually working with a data set. Also, tools can be repurposed, for example, steps 2 and 3 above can be used in a geoprocessing service that checks data and returns a report.

The Python code logs its progress and any warnings or error conditions, and traps errors that may occur. The goal is to have the tools continue to work, even if one dataset failed a process and generated error conditions.

For this deliverable, progress has been made on steps 2 through 7. Specifically:

## 2. Unpacking of data and parsing EML

Given a “zip” archive of a valid EML file, this tool currently will uncompress the archive, and move information from the EML into a Python object. It uses a pre-made Python object that contains the Xpaths to the desired info, therefore it is straightforward to add new information. One piece of information is the URI of the spatial data, and the tool will retrieve and uncompress the spatial data as well.

### 3. Data checks

This tool currently uses information from the EML and the files/folders that were retrieved to determine the type of spatial data that is present, and if the type found is reasonable given the EML entityName field and spatialVector or spatialRaster node. If a file of type Arcinfo exchange (E00) is encountered, it is imported first, and the result is examined for its type.

### 4., 5. Load vector data and update metadata

This tool adds vector data to the geodatabase, creating a geodatabase feature dataset for the site if none exists. It creates a merge XML document containing information from the EML file. It then uses the metadata associated with the newly add feature class, and runs an XSLT transformation on the metadata. The transformation accepts the merge document and the result is a new augmented metadata file. Only valid metadata nodes are added. The metadata is then re-imported to the dataset using the "Metadata\_importer" tool, which preserves the unique IDs already created for the data. For this iteration, the Title, Abstract, Purpose, package ID, and keywords are added. Also, a timestamp and spatial data URI are added. This information is all under the ISO tag dataIdInfo, and some is in a new Aggregated Citation tag. If the metadata has no dataIdInfo tag, the stylesheet will make it. The loaded vector data is in the WGS 1984 Web Mercator (Auxiliary Sphere) projection.

### 6. 7. Load raster data and update metadata

This tool creates an XML supplemental metadata file as the previous tool, creating the XML merge document. It then copies the raster data to its permanent location, creating a site directory if needed. Then, it updates the metadata on the newly copied dataset, using the same transformation as for vector data. Finally, it either adds the data to an existing mosaic dataset (for the site), or if none exists, it copies a mosaic dataset "template" to a site mosaic dataset, then loads the raster to the new mosaic dataset. The template contains attribute fields that automatically pick up certain information from the metadata. For this iteration, the Title, Purpose, keywords, and package ID are added as attributes.

### Testing

The following test suite has been run successfully on my development computer (where the geodatabase is a file geodatabase) and Maps3.

1. A HJ Andrews site shapefile (provided by TJV)
2. A HJ Andrews site E00 file (which contained a Grid of ground lidar) (provided by TJV)
3. A KMZ file derived from a GPS walking track (provided by the developer. The KML contained both a point and a polyline feature) The next three rasters overlap this location)
4. A Tif aerial photo downloaded from the National Map website
5. A Jpg of the same photo as above
6. A land use image (Tif) downloaded from the National Map site

7. An ASCII grid derived from a NED Grid downloaded from the National Map site, which overlaps HJ Andrews.

All seven cases were run through a model which consisted of the tools described above chained together (Figure 1). (The unpacking/parse EML step was only performed on the two test cases for which EML existed. Other test cases had their “EML” data manually created.) The result was that all files were correctly loaded, all supplemental metadata was in place (Figure 2), and these could be viewed in ArcMap. In particular, select by package ID is possible on the mosaic dataset to isolate one of the rasters in the mosaic.

The source files for this deliverable are as follows:

XSL:

metadataMerge.xsl

Python:

Geonis PASTA Processing.pyt (Python toolbox)

Lno\_geonis\_base.py

Lno\_geonis\_wf.py

Geonis\_pyconfig.py

Geonis\_log.py

Geonis\_helpers.py

Geonis\_emlparse.py

The source files are located on Maps3, and in the source code repository GitHub (<https://github.com/rbeloin/pasta2geonis.git>).