Software Design Document

for

Layer Analyzing

Version 2.0 approved

Prepared by: Ken Luo, Christopher P. Nguyen,
Kimberly Reyes, Edwin Ruiz,
Elizabeth Sanchez, & Alyssa Solon

The Jet Propulsion Laboratory

April 14, 2018
California State University, Los Angeles

Table of Contents

Revision History

Name	**Date**	**Reason For Changes**	**Version**
First Draft	11/15/17	Original Version	1.0
Final Draft	4/14/18	Final Version	2.0

1. Introduction

1.1 Purpose

The purpose of this document is to outline, in detail, the design for The Jet Propulsion Laboratory (JPL) project: Layer Analyzing. The Layer Analyzing software framework will be specified throughout the document along with new revisions implemented.

1.2 Document Conventions

Throughout this document, there will be bolded words in different sizes. The large words represent the name of the section and the medium-sized words will represent the subsections for each section.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document are developers, project managers, astronomers, scientists, geologists, researchers, students, and anyone interested in accessing or utilizing lunar data.

The organization of this document provides a natural flow of requirements to each succeeding section. It is suggested that this document is read in its current order to fully understand the intention of the project.

1.4 System Overview

The Layer Analyzer's goal is to educate the public and researchers, to meet the needs of lunar exploration creating a comprehensive lunar research through lunar data layers. Its analysis allows users to perform a wide range of analyses such as measuring distances, elevation, and slope analysis. The Layer Analyzer is a function that will be designed to extract data from the layers on the Moon Trek. The retrieved data will be used to show information about various parts of the Moon and any correlations between the layers.

2. Design Considerations

2.1 Assumptions and Dependencies

The Layer Analyzing project is executable with the assumption that the user has an operating system that is capable of supporting Python and its additional libraries.

2.2 Goals

- To discover and present any correlations between the various layers that are available on JPL's Moon Trek API.
- To enhance the public or researcher's knowledge of the Moon
- For users to perform a wide range of analyses utilizing lunar data.
- To speed up the time used for computing the data from the layers.

2.3 Development Methods

The 'Waterfall Development' method was used for this software design. We had to learn about a certain python library and complete a few tasks each week before moving onto the next step in the project. We would not have been able to make progress without doing so.

3. Architectural Strategies

3.1 Use of Certain Products

The Layer Analyzing project is being developed using Python and the following Python libraries and modules to handle data:

- NumPy ver. 1.13.1
- GDAL ver. 2.2.1
- Matplotlib ver. 2.0.2
- Pandas ver. 0.20.3
- Georasters ver. 0.5.11
- Inquirer ver. 2.2.0
- Scikit-learn ver. 0.19.1

3.2 Reuse of Existing Software Components

The layers that are currently available on JPL's Moon Trek API are being used in this project to obtain data for making correlations. These layers are stored as GeoTIFF files.

3.3 User Interface Paradigms (or system input and output models)

GeoTIFF and .xyz files are taken in as input for this project. The output shall be statistical data that will be computed from the entered files. Visual representation of the statistical data is also returned as output.

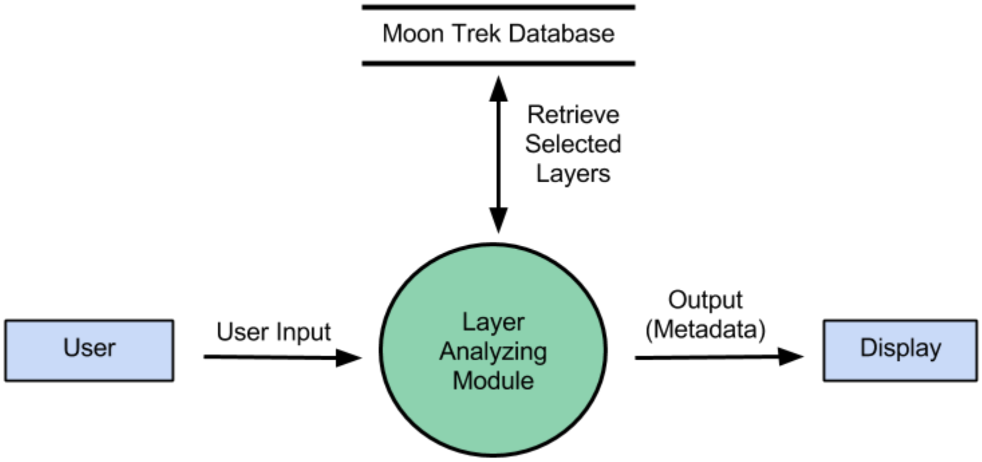
3.4 Future Plans for Extending/Enhancing Software

Future plans for the Layer Analyzing Project are including but not limited to:

- Using Seaborn to visually display the data retrieved from the .xyz files.
- Implementing Spark to minimize the data processing time.
- Reading in multiple band GeoTIFF files.

4. System Architecture

4.1 Detailed Context Diagram (DFD Level 0)

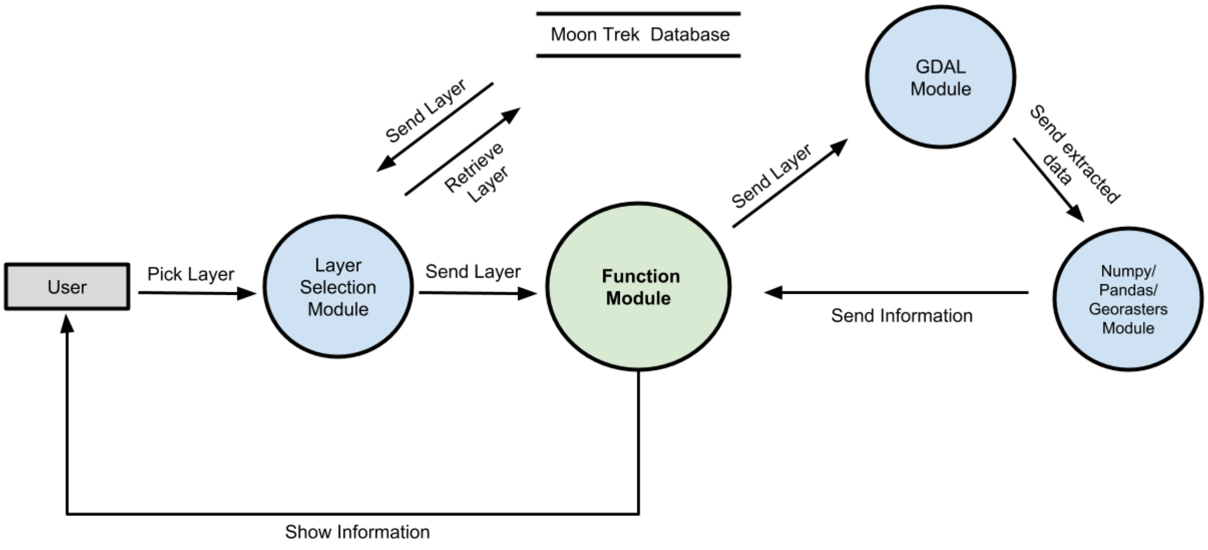


Context Diagram (LEVEL 0 DFD)

4.1.1 Description and Major Functions of the Function Module

The Moon Trek API is a user-friendly interface where the user selects the lunar layer from the Moon Trek database that they want to analyze. The function module will give the user the option to choose which calculations they want to analyze.

4.2 Detailed Functional Description of the Function Module



Context Diagram (Level 1 DFD)

4.2.1 Detailed Description of Function Module Major Subunits

Function Module

This module contains multiple functions which compute calculations on the selected layers depending on what function the user selects. The available functions for implementation are Stats (Average, Standard Deviation, Minimum and Maximum Value, and Median Value), Covariance, Clustering, Logarithm and Correlation Graphs, Scatter Plot, Plotting x values versus y values, Histogram, and 3D Plot.

Layer Selection Module

Selects the layer of interest through the interface (provided)

GDAL Module

GDAL is used to obtain the information from the GeoTIFF file such as the bands being contained, coordinates, pixel size, origin, and the no data values.

Numpy Module/Pandas Module/Georasters Module

These modules read in the .xyz files and the data extracted from the GeoTIFF files. Georasters is used to convert the data extracted from the GeoTIFF files into a dataframe. Pandas is used to organize our different value types. NumPy can only hold one type of value per array while Pandas being able to handle multiple. NumPy also has various functions inside it's own library such as min, max, std, and mean. It is a numerical computation for Python

Moon Trek Database

Contains more than 2500 layers of selection

5. Policies and Tactics

5.1 Specific Products Used

5.1.1 IDE

The IDE being used is Jupyter Notebook and the system is implemented using CLI. Jupyter Notebook is used as an alternative to display various results, debug, and to observe what needs more work as progression was made. Text editors such as Atom is used to create the code to execute in CLI.

5.1.2 Libraries

The libraries that will be used are the libraries that were previously mentioned in section 3.1 which are:

- NumPy ver. 1.13.1
- GDAL ver. 2.2.1
- Matplotlib ver. 2.0.2
- Pandas ver. 0.20.3
- Georasters ver. 0.5.11
- Inquirer ver. 2.2.0
- Scikit-learn ver. 0.19.1

We have chosen to use these libraries because they have the most efficient functionality. The GDAL library enables us to retrieve data from the layers. The NumPy and Pandas libraries and Georasters module assist us with storing and manipulating the extracted data. The Matplotlib library allows us to plot the data. Inquirer was used to build the CLI interface.

5.2 Plans for Ensuring Requirements Traceability

We meet once a week our liaisons and advisors to track our progress and the requirements are on the Software Requirements Specification document.

5.3 Plans for Testing the Software

5.3.1 We test our functions first before they are implemented into the backend code.

5.3.2 Our code is being done in Python and will be integrated into JPL's Moon Trek API.

5.3.3 We are working with arrays and data frames to help us work with multiple raster bands.

5.3.4 We are currently working on the functions.

5.3.5 JPL will be providing the interface.

5.3.6 We are using Python shell for now, combining with other python libraries such as Numpy, Pandas, Matplotlib, and Seaborn.

6. Database Design

The database utilized in this project is the database on Moon Trek API which is provided by JPL. The database contains the layers that will be analyzed.

7. User Interface

7.1 Overview of User Interface

The original user interface is already being provided by JPL. Aside from that we have also implemented a user interface that involves command line. A menu depicting the available layers will be presented to the user when the system executes. At this point, the user is required to select either one or multiple layers. Upon making a selection, another menu will appear which contains the available functions to feed the layers and its data into. Depending on which function is chosen, another prompt may come up that requires user input such as asking whether or not the user would like to normalize the layer data before any calculations are done. The statistical calculations are then carried out and visual representation of the data is returned.

8. Requirements Validation and Verification

Functional Requirements		
Requirement No.	**Requirement Description**	**Related Module**
4.1.1	The system shall take in GeoTIFF and .xyz files.	Function Module
4.1.2	The system shall output statistical data from the GeoTIFF or .xyz file.	Function Module
4.1.2.1	The system shall output statistical data and visual representation of the calculated data.	Function Module
4.1.3	The system shall be written in Python.	Function Module
4.1.3.1	The system shall integrate the following Python libraries and modules: GDAL, NumPy, Matplotlib, Pandas, Seaborn, Georasters, and Inquirer.	Function Module
4.1.4	The system shall read raster bands from the GeoTIFF files and translate them into a data format to be used for analysis.	GDAL Module
4.1.4.1	The system shall return coordinates from a pixel coordinate system to a latitude-longitude coordinate system.	NumPy/Pandas/ Georasters Module
4.1.5	The system shall represent the data in graphical formats such as a histogram.	NumPy/Pandas/ Georasters Module

9. Glossary

- **JPL** - Jet Propulsion Laboratory
- **API** - Application Program Interface
- **DFD** - Data Flow Diagram
- **IDE** - Integrated Development Environment
- **std** – Standard Deviation
- **CLI** – Command Line Interface

10. References

Moon Trek (https://moontrek.jpl.nasa.gov/)

GDAL ver. 2.2.1 (http://www.gdal.org/)

Python ver 3.5.2 (https://www.python.org/download/releases/3.0/)

NumPy ver. 1.13.1 (http://www.numpy.org/)

Matplotlib ver. 2.0.2 (https://matplotlib.org/)

Pandas ver. 0.20.3 (http://pandas.pydata.org/)

Georasters ver. 0.5.11 (https://github.com/ozak/georasters)

Inquirer ver. 2.2.0 (https://pypi.python.org/pypi/inquirer)

Scikit-learn ver. 0.19.1 (http://scikit-learn.org)

Visualizing Distributions of Covariance Matrices Document (http://www.stat.columbia.edu/~gelman/research/unpublished/Visualization.pdf)