DESIGN OF A FREE AND OPEN SOURCE DATA PROCESSING, ARCHIVING, AND DISTRIBUTION SUBSYSTEM FOR THE GROUND RECEIVING STATION OF THE PHILIPPINE SCIENTIFIC EARTH OBSERVATION MICRO-SATELLITE

R.K.D. Aranas[a,*1], B.J.D. Jiao [a], B.J.P. Magallon [a,b] , M.K.F. Ramos [a], J.A. Amado[a], A.M. Tamondong [a], and M.E.A. Tupas
[a] University of the Philippines, Diliman, Quezon City, 1101, Metro Manila, Philippines,
(jaamado,bdjiao, mframos,rdaranas,amtamondong, matupas)@up.edu.ph
[b] Space Mission Center, Creative Research Institution, Hokkaido University
benjamin@ep.sci.hokudai.ac.jp

Commission VII, SpS10 - FOSS4G: FOSS4G Session (coorganized with OSGeo)

KEY WORDS: Micro-satellite, Data Processing, Open Source Software, System Design

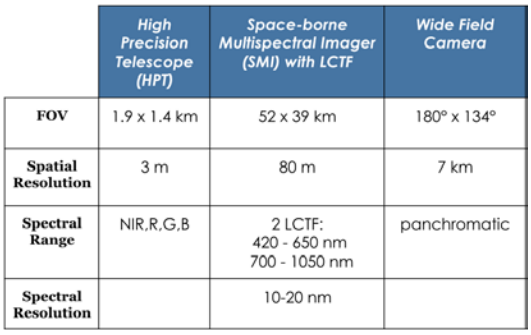
ABSTRACT:

The Philippines's PHL-Microsat program aims to launch its first earth observation satellite, DIWATA, on the first quarter of 2016. DIWATA's payload consists of a high-precision telescope (HPT), spaceborne multispectral imager (SMI) with liquid crystal tunable filter (LCTF), and a wide field camera (WFC). Once launched, it will provide information about the Philippines, both for disaster and environmental applications. Depending on the need, different remote sensing products will be generated from the microsatellite sensors. This necessitates data processing capability on the ground control segment. Rather than rely on commercial turnkey solutions, the PHL-Microsat team, specifically Project 3:DPAD, opted to design its own ground receiving station data subsystems. This paper describes the design of the data subsystems of the ground receiving station (GRS) for DIWATA. The data subsystems include: data processing subsystem for automatic calibration and georeferencing of raw images as well as the generation of higher level processed data products; data archiving subsystem for storage and backups of both raw and processed data products; and data distribution subsystem for providing a web-based interface and product download facility for the user community. The design covers the conceptual design of the abovementioned subsystems, the free and open source software (FOSS) packages used to implement them, and the challenges encountered in adapting the existing FOSS packages to DIWATA GRS requirements.

1. INTRODUCTION

1.1 Background

The Philippines's PHL-Microsat program aims to launch its first earth observation satellite, DIWATA, on the first quarter of 2016. DIWATA's payload consists of a high-precision telescope (HPT), spaceborne multispectral imager (SMI) with liquid crystal tunable filter (LCTF), and a wide field camera (WFC). Figure 1 shows the technical specifications of DIWATA's payloads.



	High Precision Telescope (HPT)	Space-borne Multispectral Imager (SMI) with LCTF	Wide Field Camera
FOV	1.9 x 1.4 km	52 x 39 km	180° x 134°
Spatial Resolution	3 m	80 m	7 km
Spectral Range	NIR,R,G,B	2 LCTF: 420 - 650 nm 700 - 1050 nm	panchromatic
Spectral Resolution		10-20 nm	

Figure 1. DIWATA payloads

Depending on the need, different remote sensing products will be generated from the microsatellite payloads. This necessitates data processing capability on the ground control segment.

The PHL-Microsat program is composed of five components:

- Project 1: Microsatellite BUS Development
- Project 2: Ground Receiving Station
- Project 3: Development of a Data Processing, Archiving, and Distribution Sub-system (DPAD)
- Project 4: Calibration and validation of remote sensing instruments
- Project 5: Data product generation

Of the five projects, four (projects 3-5) are involved in the data processing chain. Project 2 is in charge of receiving the data from the satellite. Project 4 provides the calibration values for the instrument, whereas Project 5 is in charge of researching and developing data product generation algorithms. Project 4 and Project 5 both provide inputs to Project 3 which is in charge of the data processing pipeline.

Rather than rely on commercial turnkey solutions, the PHL-Microsat program, specifically Project 3: DPAD, opted to design its own ground receiving station data subsystems. The decision to design and implement the system was made due to the following reasons:

- **Capacity building.** Turnkey solutions are black boxes. They will work but the Philippines will not have learned anything from it.

[*] Corresponding author

- **Minimize costs.** Turnkey solutions do not scale because of the licensing costs. Adding more servers would increase the e cost prohibitive in the future as the project acquires more servers due to the server licensing costs of turnkey solutions. (Kim et al. 2011; Rau et al. 2003)

This paper will focus on Project 3 and the Data Processing, Archiving, and Distribution Subsystem for the ground receiving station of Diwata. The system architecture, data processing chain, and the Free and Open Source Software packages used will be discussed.

2. SYSTEM ARCHITECTURE

As shown in Figure 1, there are eight processes that show the interaction of the different projects and the flow data. Of these eight however, only four (ingestion, archiving, processing, and distribution) fall within DPAD's responsibilities. The architecture is inspired by STORM(Oštir et al. 2015) and CATENA(Krauß et al. 2013), two automated data processing architectures.

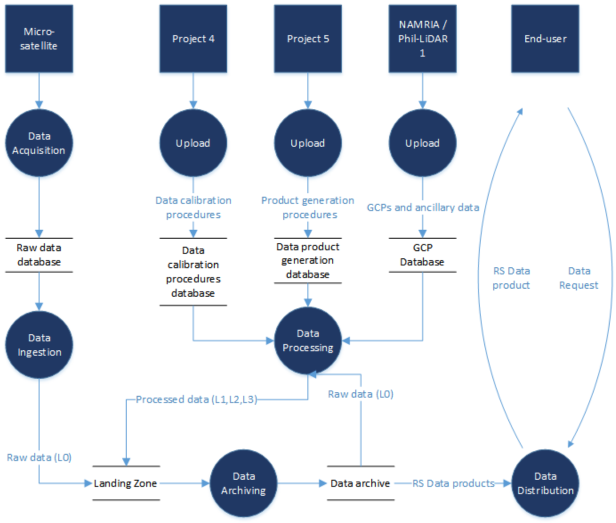


Figure 2 Overview of the system architecture and data flow

2.1 Data Ingestion

DIWATA captures images of the Philippines which are then downloaded by the GRS. Once in the GRS, the data is pulled by the data processing, archiving, and distribution subsystem into the landing zone, a temporary holding place for raw data file (See Fig. 1).

2.2 Data Archiving

Once in the landing zone, the data archiving subsystem copies the raw and unprocessed data and stores them for future use. The data archiving subsystem also stores copies of the processed data.

2.3 Data Processing

The data processing subsystem reads the metadata of data files in the system and performs operations on the data which includes conversion from raw binary file to images, radiometric corrections, georeferencing, and generation of data products. Once the processing is done, the results are pushed to the landing zone. From there, the processed data is copied to archived for future use.

2.4 Data Distribution

The data distribution subsystem is the public facing component of the DPAD subsystems. This subsystem pulls the data from the data archives and serves it to the users which may include GIS practitioners, government agencies, and the general public.

3. DATA PROCESSING CHAIN

3.1 Binary to image

The raw data from DIWATA is in the form of binary files which are not directly usable in the data processing pipeline. It must first be converted into image formats like geotiff and img. The metadata also needs to be extracted from the binary file.

3.2 Radiometric correction

The image extracted from the binary file then undergoes radiometric correction based on Project 4's inputs.

3.3 Cloud cover computation and thumbnail generation

Once corrected, a cloud cover percentage computation algorithm is run on the image. The value from this computation is then written to the metadata.

3.4 Georeferencing

Once the cloud cover computation is done, the image is then passed on to the georeferenced step. An automated georeferencing pipeline is based on Space SI's automatic georeferencing pipeline (Oštir et al. 2015) is currently being tested for the DIWATA images.

3.5 Data product generation

The georeferenced image is then passed to the last step which is the data product generation stage. Based, on the metadata, different data products are produced from the georeferenced image.

4. FREE AND OPEN SOURCE SOFTWARE PACKAGES FOR SATELLITE DATA PIPELINES

4.1 Apache Object Oriented Data Technology

Apache Object Oriented Data Technology (OODT) (Richard Ullman, Peyush Jain 2013)(Leptoukh 2005) is an open source package for data and workflow management. Apache OODT can also execute scripts written in Java or other languages such as Python

Apache OODT is used for the data archiving and processing components of the DPAD subsystem. OODT manages the ingestion, archiving, and data processing. Currently, the implementation of the DIWATA image processing chain in Apache OODT is still in progress.

4.2 Python scientific computing stack

Python (Rossum 1991) is a widely used high-level, general-purpose, interpreted, dynamic programming language. It was chosen as the main language of the data processing chain as it has a large number of data processing libraries(Manoochehri 2013). The presence of Numpy and the Python scientific computing stack also contributed to the decision to use Python.

The team uses Rasterio (Mapbox 2013), a Python binding to the Geospatial Data Abstraction Library (GDAL)(Warmerdam 2008) to load the image data. GDAL supports the reading and writing of most, if not all geospatial data formats, raster and vector.

Once the image files are loaded, Numpy (D. Ascher, P.F. Dubois, K. Hinsen, J. Hugunin, Oliphant 2001) is used to transform the images into matrices (e.g. NDVI, cloud masks, etc) for image processing. Numpy adds support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays for Python.

4.3 Libra and Node.js

Libra (Development Seed 2015) is an open source Landsat 8 image browser developed by Development Seed and Astro Digital. Libra makes use of Node.js (Dahl 2009) , an open source project which allows Javascript to be run on the server. The team has already forked Libra and deployed it to serve Landsat 8 datasets of the Philippines. Development is currently underway to extend Libra to accommodate DIWATA image.

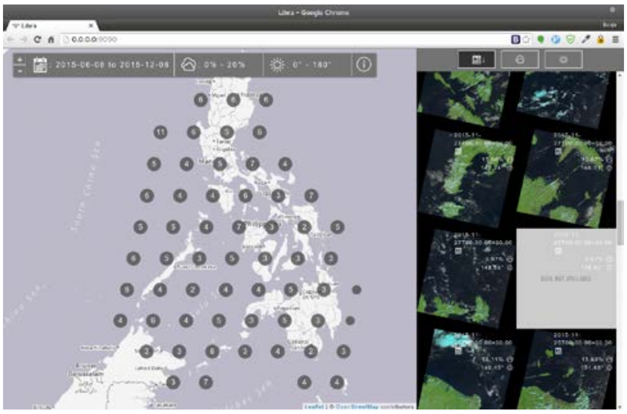


Figure 3 Screenshot of Libra

5. CONCLUSIONS AND RECOMMENDATIONS

The paper has presented a conceptual design and the free and open source software packages used to implement said design. This shows that it is possible to set up a satellite data processing subsystem using only free and open source software. The team is currently in the process of implementing the subsystem. It will be operational by the 3rd quarter of 2016 and the source code and documentation of the whole stack will be shared by then.

ACKNOWLEDGEMENTS

This work is supported by the Department of Science and Technology - Philippine Council of Industry, Energy, and Emerging Technology Research and Development (DOST-PCIEERD) under the Development of the Philippine Scientific Earth Observation Microsatellite (PHL-MICROSAT – Project 3) Program.

REFERENCES

D. Ascher, P.F. Dubois, K. Hinsen, J. Hugunin, Oliphant, Tim T. 2001. “Numerical Python.” http://www.numpy.org/ (February 1, 2016).

Dahl, Ryan. 2009. “Node.js.” https://nodejs.org/en/ (January 15, 2016).

Development Seed. 2015. “Libra.” http://libra.developmentseed.org/ (July 30, 2015).

Kim, KyungHee et al. 2011. “Ground Station Design for STSAT-3.” *International Journal of Aeronautical and Space Sciences* 12(3): 283–87.

Krauß, T., P. d’Angelo, M. Schneider, and V. Gstaiger. 2013. “The Fully Automatic Optical Processing System CATENA at DLR.” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-1/W1(May): 177–83. http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-1-W1/177/2013/isprsarchives-XL-1-W1-177-2013.html.

Leptoukh, Gregory G. 2005. “NASA Remote Sensing Data in Earth Sciences: Processing, Archiving, Distribution, Applications at the GES DISC.”

Manoochehri, Michael. 2013. 30 *Data Just Right: Introduction to Large-Scale Data & Analytics*. Addison-Wesley. https://books.google.com/books?id=kmlCAgAAQBAJ&pgis=1 (February 3, 2016).

Mapbox. 2013. “Rasterio.” https://www.mapbox.com/blog/rasterio-announce/ (March 2, 2016).

Oštir, K. et al. 2015. “Automatic Near-Real-Time Image Processing Chain for Very High Resolution Optical Satellite Data.” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-7/W3(May): 669–76. http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-7-W3/669/2015/.

Rau, Jiann-Yeou et al. 2003. “AN OPERATIONAL MULTI-SENSOR GEOCODED PRODUCTION SYSTEM FOR EARTH RESOURCES SATELLITE IMAGES.” In *Roceedings of the 5th Int. Symposium on the Reducing Cost of Spacecraft and Ground Systems and Operations*, Pasadena, USA.

Richard Ullman, Peyush Jain, Wayne McCullough. 2013. *Case Study: Apache OODT Framework Application for Support to the Joint Polar Satellite System (JPSS)*.

Rossum, Guido van. 1991. “A High-Level Interpreted Language Combining Ideas from ABC, C, Modula-3, Icon, Etc. Intended for Prototyping or as an Extension Language for C Applications. Modules, Classes, User-Defined exceptions. "Linking a Stub Generator (AIL) to a Prototyping Langua.” https://www.python.org/ (January 24, 2016).

Warmerdam, F. 2008. “The Geospatial Data Abstraction Library.” *Open source approaches in spatial data handling*. https://sci-hub.io/http://link.springer.com/chapter/10.1007/978-3-540-74831-1_5 (January 10, 2016).