Open Source Principles Utilized by the CERES Edition 5 Level-3 Framework

Thomas.N.Hillyer@nasa.gov, ADNET
Joshua.C.Wilkins@nasa.gov, ADNET

0. Abstract

The Clouds and the Earth's Radiant Energy System (CERES) Science Team integrates and fuses observations from six CERES instruments aboard the Terra, Aqua, S-NPP, and NOAA-20 missions with data from twenty-five geostationary imagers, in creating a nearly 25-year, ongoing record of the Earth's radiation budget. In preparation for the next version of CERES data products, the team has been exploring software development methods leveraging open-source principles and software to streamline data product configuration and algorithm implementation. This effort will provide robust software for production while maintaining greater flexibility for the algorithm developers to explore new science.

Developers on the CERES Level 3 products have created a framework allowing greater modularization and flexibility to design and configure CERES Level 3 products while reducing development time. The framework leverages open data formats to configure data variables and attributes within the product, declare tolerances for input validation, select resolution and binning schemes, and allow algorithm selection at runtime without requiring recompilation of the product generator. The framework incorporates means to perform unit and integration testing allowing more rapid feedback to the science team. The framework has been used for one current CERES data product and has demonstrated promise for upcoming products in development.

1. Requirements

The CERES Level 3 working group lead's goals for Edition 5 framework are the following improvements over Edition 4:

- Modular and Flexible
  - Single Purpose Algorithms and Functions
  - Minimal Disruption When Swapping Algorithms
- Configurable
  - Minimize Need for Constant Source Code Changes
  - Favor Descriptive and Structured Configuration Files Where Possible
  - Enable/Disable Scientific Parameters as Needed
  - Easily Change Processing Modes
    - Binning - how footprints are assembled
    - Gridding
      - 1°, 1/2°, etc.
      - equal angle or equal area gridding
- Rapid Development and Analysis
  - Adjustable Granularity
    - Daily
    - Monthly
  - Quick Tests or Long-Term Analysis

2. Programming Languages

Part of the updated Edition 5 framework involves extensive refactoring of the existing Edition 4 Fortran 90 codebase to take greater advantage of object oriented facilities in modern languages.

- C++-17
  TISA Gridding
- Fortran 2018
  TISA Averaging

3. Improved Unit Testing

Increased modularization allows easier incorporation of unit testing for Level 3 software. CERES Edition 5 framework is using the open source **doctest** framework allowing tests to be written into production code.

https://github.com/doctest/doctest

4. JSON-based Product Configuration

JSON-based configuration files are used to define the structure of data products, valid ranges for input and output products, and various attributes to be included with the final product.

Many libraries exist allowing JSON-based configuration parameters to be shared across both product generators and programming languages.

```
{
    "iprop": 8,
    "proplname": "Surface_Pressure",
    "valmin": 0.0,
    "valmax": 1100.0,
    "valdef": 3.4028235e+38,
    "units": "hPa",
    "avescheme": "Normal",
    "weightproperty": "None",
    "log_type": "None",
    "linked_property": "None",
    "gridding" : "Gridded",
    "coverage_content_type" : "physicalMeasurement"
},
{
    "iprop": 9,
    "proplname": "MATCH_AOT",
    "valmin": 0.0,
    "valmax": 8.0,
    "valdef": 3.4028235e+38,
    "units": "N/A",
    "avescheme": "Normal",
    "weightproperty": "None",
    "log_type": "None",
    "linked_property": "None",
    "gridding" : "Gridded",
    "input_adjustment" : "offset_addition_when_less_than_qualifier",
    "offset_value" : 6.5536,
    "qualifier_value" : 0.0,
    "coverage_content_type" : "physicalMeasurement"
},
```

5. JSON Properties Used in Code

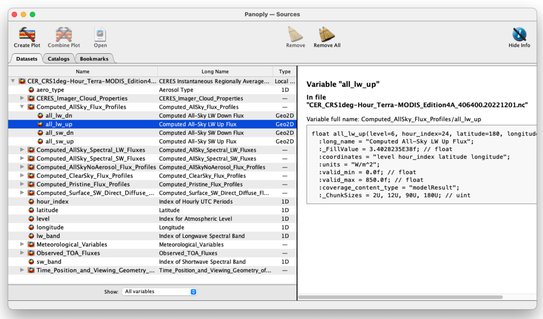
```
if(property.get_averaging() == Averaging::Weighted) {

    if(property.is_array_property()) {
        throw tisa::exceptions::Tisa_Exception("time_box::grid_pixels", "code not implemented for weigh
    } else {
        //create some temporary arrays to populate to use in the averaging routine
        std::vector<float> tmp_values;
        std::vector<float> tmp_weights;
        //get the weighting group index
        int weighting_group_index = property.get_weighting_group_index();
        //get the weighting property index value
        unsigned short weighting_property_index = property.get_weighting_property_index();
        //regardless if weighting_group_index is the same or different from the current group, it will
        //fill up an array of the weights and values
        for(Pixel_Data_Container &p : ungridded_pixels) {
            if(p.has_param(group_index, property_index)) {
                tmp_values.push_back(p.get_param(group_index, property_index));
                tmp_weights.push_back(p.get_param(weighting_group_index, weighting_property_index));
            }
        }

        //average like normal, and save the result
        float average_value = tisa::get_weighted_average<float>(tmp_values, tmp_weights);
        parameters.set_param(group_index, property, average_value);
    }
```

The framework makes use of the following open source C++ JSON library hosted on Github:

https://github.com/nlohmann/json

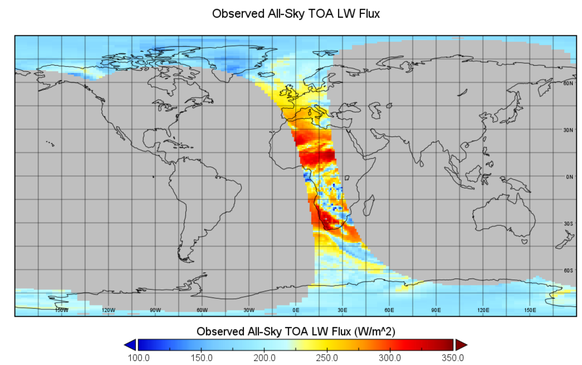
6. Proof of Concept (CRS1deg-Hour)

The CERES Edition 4 CRS1deg-Hour data product was selected to be a proof of concept for exercising the Edition 5 Level 3 framework. The CRS1deg-Hour product provides hourly CERES Clouds and Radiative Swath (CRS) longwave and shortwave computed vertical flux profiles and cloud properties spatially gridded into 1°-regions.



The CRS1deg-Hour PGE was recently delivered to production and uses the Edition 5 framework (using Edition 4 CRS data) and creates a user-friendly, gridded NetCDF-4 data product. Using the framework, it was easy for the team to use the JSON configuration files to define key features and settings to use at runtime, declare tolerances for input data, and define the structure and attributes to be included in the final output product.

7. Proof of Concept (CRS1deg-Hour)



Variable parameters and settings are easier to assign and audit using the framework's JSON configuration approach. This helps to guarantee off-the-shelf visualization tools, like Panoply, can correctly plot and geolocate CERES data.

8. Edition 5: Next Steps

With CRS1deg-Hour product generator in production, the team has qualitative reults that this refactored Edition 5 framework facilitates maintainable and reusable coding. With the framework in place, the team is now developing new Edition 5 product generators, with the CERES TISA SSF1deg-Hour product generator now in its alpha stage. Many framework components used in CRS1deg-Hour, are adaptable for the SSF1deg-Hour product simply by adjusting the JSON configuration files for the product generator.

9. Additional Thanks

The authors would like to thank the following individuals for their guidance and encouragement:

- NASA - Kathleen Dejwakh, David Doelling
- ADNET (RSES) - Walter Miller, Beau Branch, Arun Gopalan, Cathy Nguyen
- AMA (RSES) - Pamela Mlynczak

For more information...

https://ceres.larc.nasa.gov