End-to-End Design and Development of GOES-R Level 2 Algorithms: Cloud Pathfinder Algorithm Case Study

Andra Ivan, Paul Van Rompay*, Larry Belvin, Xanthe Papadakis and T Scott Zaccheo
AER, Inc, Lexington, MA 02421 (*AER, Greenbelt, MD 20706)

Introduction

GOES-R is the next generation of NOAA's Geostationary Operational Environmental Satellite (GOES) System

- GOES-R will provide advanced products which describe the state of the atmosphere, land, and oceans over the Western Hemisphere.
- Harris GOES-R Core Ground Segment (GS) Team will provide the ground processing software and infrastructure needed to produce and distribute these products.
- GOES-R L2+ algorithms ingest calibrated and geo-located GOES-R L1b data and other ancillary/auxiliary/intermediate data to produce L2+ products such as aerosol optical depth, rainfall rate, derived motion winds, and snow cover.
- This work describes the ABI Cloud Mask (ACM) algorithm's journey from its scientific description to a working engineering solution, and highlights milestones in the software development life cycle.

ATBD (AWG) → **ADD** (AER) → **Requirements Analysis** → **Preliminary Design** → **Detailed Design** → **Code and Unit Test** → **SWIT** → **SIQT**

Algorithm Theoretical Basis Document
- Inputs, Outputs
- Physics, Math
- Test Data Sets
- Practical Considerations
- Quality Assessment
- Diagnostics
- Exception Handling

Algorithm Description Document
- Inputs, Outputs
- Processing Parameters
- Flow Diagram
- Functional Blocks
- Library Services Functions

- Use Cases:
  Run Algorithm
  Retrieve Output Data
- Requirements
- SADT Diagrams
- DMI Activity Diagram
- Data Dictionary
- Acronym List, Glossary

- Algorithm CSUs
- Expanded SADT Model
- External Interfaces Provided by Algorithm
- Class Diagram
- Requirements Updates

Design Decision: Stateless Algorithm
- No data are persisted in the algorithm.
- All intermediate data must be written.
- PGIM may persist data, as needed.

- Complete Design
- All Interfaces: Provided / Used, Internal / External
- Code Headers & Stubs
- Exception Handling
- Requirements Updates

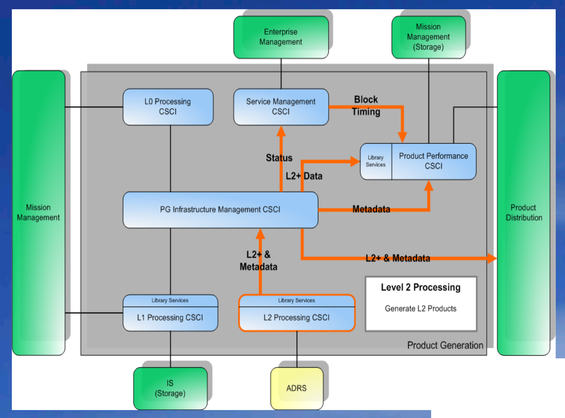
- Code Units
- Test Classes
- Test Data
- White-Box Test Results
- Black-Box Test Results

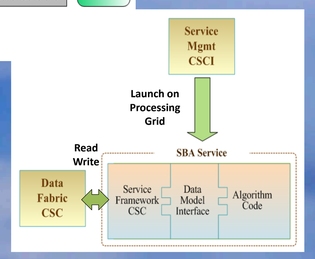
SW Infrastructure
- Library Services: algorithm framework, DMI, CRTM, shared library functions
- Mock objects as needed

SW Integration and Test
Implement requirement verification tests.
Implement integration tests running the algorithm end to end on realistic data.

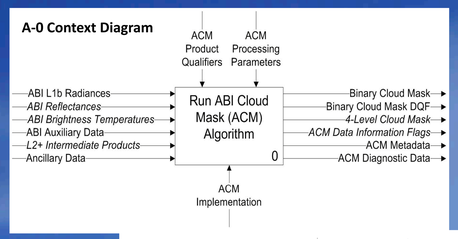
Software Item Qualification Test

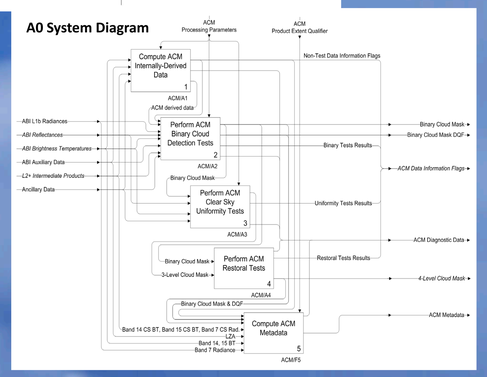
Interfaces between Components





Structured Analysis for Algorithms





SADT Model and System Diagrams
- Breaks down algorithm into activity boxes or functional blocks
- Shows flow of data at various levels of detail
- Augmented with flowcharts, pseudocode, functionality tables detailing operations
- Basis for class diagram, member variables, member functions, and code units

ACM Algorithm:
- 14 binary cloud detection tests
- 2 clear-sky uniformity tests
- 2 restoral tests

Outputs
- Binary Cloud Mask: cloudy, clear
- 4-Level Cloud Mask: cloudy, prob. cloudy, prob. clear, clear
- Test Results
- Data Quality Flags
- Metadata

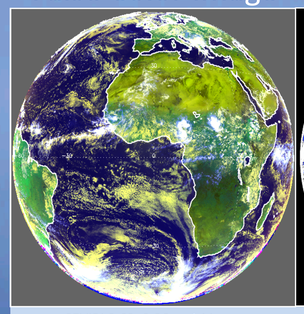
Current Status

- Completed Software Integration and Test phase
- Integrated Pathfinder ACM with Algorithm Test Tools and produced a cloud mask and associated products on AWG-reproducibility data
- Showed example on how to promote unit tests to verification tests.
- From Pathfinder work, developed process and software infrastructure for next algorithms

Future Work

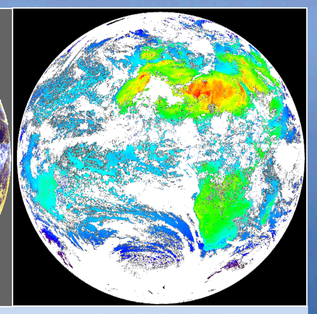
- Next Stage: Cloud Algorithms (including Cloud Mask) will follow Pathfinder example to complete formal SWIT.

False Color Image



SEVIRI 0.63, 0.86 and 11 µm:
12 UTC, August 10, 2006
(from ABI Cloud Mask ATBD, NOAA/NESDIS/STAR)

4-Level Cloud Mask



Resulting Cloud Mask Product
white: cloudy, grays: probably cloudy/clear
(from ABI Cloud Mask ATBD, NOAA/NESDIS/STAR)

Acknowledgements

The authors would like to thank Charles Paxson at AER, and Jeff Copes at Harris Corp.