

University of New Hampshire

University of New Hampshire Scholars' Repository

Earth Sciences Scholarship

Earth Sciences

8-1-2010

GOES-R Algorithms: A Common Science and Engineering Design and Development Approach for Delivering Next Generation Environmental Data Products

John L. Baldwin

Atmospheric and Environmental Research, Inc., Lexington, MA

Rob Braswell

University of New Hampshire - Main Campus, rob.braswell@unh.edu

David B. Hogan

Atmospheric and Environmental Research, Inc., Lexington, MA

Edward Kennelly

Atmospheric and Environmental Research, Inc., Lexington, MA

Xanthe Papadakis

Atmospheric and Environmental Research, Inc., Lexington, MA

See next page for additional authors

Follow this and additional works at: https://scholars.unh.edu/earthsci_facpub

Recommended Citation

John L. Baldwin; Bobby H. Braswell; David B. Hogan; Edward Kennelly; Xanthe Papadakis; Michael Sze; Alexander Werbos and T. Scott Zaccheo, "GOES-R algorithms: a common science and engineering design and development approach for delivering next generation environmental data products", Proc. SPIE 7813, Remote Sensing System Engineering III, 781306 (August 26, 2010) doi:10.1117/12.860855; <http://dx.doi.org/10.1117/12.860855>

This Article is brought to you for free and open access by the Earth Sciences at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Earth Sciences Scholarship by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact Scholarly.Communication@unh.edu.

Authors

John L. Baldwin, Rob Braswell, David B. Hogan, Edward Kennelly, Xanthe Papadakis, Michael Sze, Alexander Werbos, and T Scott Zaccheo

GOES-R Algorithms: A Common Science and Engineering Design and Development Approach for Delivering Next Generation Environmental Data Products

John L. Baldwin^{a*}, Bobby H. Braswell^a, David B. Hogan^a, Edward Kennelly^a, Xanthe Papadakis^a,
Michael Sze^a, Alexander Werbos^a, T. Scott Zaccheo^a

^aAtmospheric and Environmental Research, Inc., Lexington, MA.02481

ABSTRACT

GOES-R, the next generation of the National Oceanic and Atmospheric Administration's (NOAA) Geostationary Operational Environmental Satellite (GOES) System, represents a new technological era in operational geostationary environmental satellite systems. GOES-R will provide advanced products that describe the state of the atmosphere, land, oceans, and solar/ space environments over the western hemisphere. The Harris GOES-R Ground Segment team will provide the software, based on government-supplied algorithms, and engineering infrastructures designed to produce and distribute these next-generation data products. The Harris GOES-R Team has adopted an integrated applied science and engineering approach that combines rigorous system engineering methods, with modern software design elements to facilitate the transition of algorithms for Level 1 and 2+ products to operational software. The Harris Team GOES-R GS algorithm framework, which includes a common data model interface, provides general design principles and standardized methods for developing general algorithm services, interfacing to external data, generating intermediate and L1b and L2 products and implementing common algorithm features such as metadata generation and error handling.

This work presents the suite of GOES-R products, their properties and the process by which the related requirements are maintained during the complete design/development life-cycle. It also describes the algorithm architecture/engineering approach that will be used to deploy these algorithms, and provides a preliminary implementation road map for the development of the GOES-R GS software infrastructure, and a view into the integration of the framework and data model into the final design.

Keywords: Remote Sensing, GOES-R, Scientific Software Design

1. INTRODUCTION

GOES-R is the next generation of the National Oceanic and Atmospheric Administration's (NOAA) ongoing Geostationary Operational Environmental Satellite (GOES) System. GOES-R is scheduled to join the current GOES constellation in 2015. The current GOES system includes 2 operational spacecraft in geostationary orbits over the equator at 75 degrees West and 137 degrees West longitude.¹

GOES-R represents a new technological era in operational geostationary environmental satellite systems, with a significant increase in image/data resolution and refresh-rate. GOES-R will provide the nation with a 45-fold increase in data covering the western hemisphere, and will include such products as:

- Visual and infrared imagery of the entire hemisphere every 5 minutes at up to 0.5km resolution
- Coverage of severe weather events with 30 second refresh intervals with simultaneous routine hemispheric and CONUS imaging
- Hemispheric detection of cloud-to-cloud and cloud-to-ground lightning events

- Enhanced space¹ weather and solar monitoring and event detection

The Harris GOES-R Core Ground Segment (GS) Team will be responsible for transitioning Government provided algorithms to 24/7/365 operational software designed to provide all calibrated imagery/data (Level 1b products) from the six GOES-R sensors as well as all of the GOES-R operational data (Level 2+ products). These products include:

- Aerosol Products: Aerosol detection suspended matter/optical depth and particle size
- Aviation Products: Volcanic ash detection and height
- Cloud Products: Cloud clear sky mask, phase, optical depth, particle size distribution, cloud liquid water
- Hydrology: Rainfall Rate / QPE
- Land Products: Land temperature and fire/hot spot characterization
- Lightning Products: Lightning events, groups and flashes
- Radiation Products: Downward and reflected solar insolation
- Soundings Products: Legacy vertical moisture and temperature profiles, stability indices and total precipitable water
- Sea Surface Products: Sea surface temperature
- Wind Products: Derived motion winds and hurricane intensity

The GOES-R Product Generation (PG) environment is one of four elements in the overall GS design. The others, described in Hanson² et al, are the Enterprise Management, Mission Management and Product Distribution elements. This work focuses on the PG architecture and its software elements and design concepts which include the application of a common Data Model Interface (DMI) and Algorithm Framework^{2,3}.

2. GOES-R PG ARCHITECTURE

The overall base-line PG algorithm flow diagram is shown in Figure 1. This diagram outlines the Level 1b and Level 2+ processing of the GOES-R Advance Baseline Imager (ABI) and the GOES-R Geostationary Lightning Mapper (GLM) data. Even though this diagram focuses on the ABI and GLM data flow, the data flow for the GOES-R suite of Solar and Space Level 1b products is implemented in an identical fashion using the same architectural constructs and software design elements described below. The two major elements shown in this figure are the GOES-R PG L1b/L2 Processing element at the Wallops Command and Data Acquisition Station (WCDAS), and the L2 Processing element at NOAA Satellite Operations Facility (NSOF). The other sub-system shown in this diagram is the Ancillary Data Relay System (ADRS), a system external to the GOES-R GS responsible for providing the GOES-R GS at NSOF with dynamic ancillary data, e.g. Numerical Weather Prediction (NWP) data.

Raw GOES-R sensor data are down-linked then calibrated, geo-located, and processed at WCDAS to form ABI Level 1b and GLM Level 2+ products. These data, along with the Level 1b space and solar products, are then distributed via the GOES-R ReBroadcast (GRB) sub-system, a transponder based communication link integrated into the GOES-R satellite design, to NSOF and other real-time Level 1b users. At NSOF, the ABI L1b data are combined with ancillary data and static/semi-static algorithm parameter to produce GOES-R Level 2+ products. Many of the WCDAS and NSOF functions are also replicated at the GOES-R Remote Backup Unit (RBU), a remote and redundant “hot” backup facility not illustrated in Figure 1. This facility ensures continuous production of GOES-R key performance parameters even in the event of a catastrophic failure at WCDAS and/or NSOF.

*jbaldwin@aer.com; phone 1 781.761 2288; www.aer.com

The Harris GOES-R GS core team has proposed a common algorithm architecture to implement this high-level design. This architecture serves two key functions. 1) Support the production of L1b and L2+ products at WCDAS and NSOF. 2) Facilitate the implementation of the GOES-R Algorithm Engineering (AE) environment used to transition Government supplied material to pre-operational algorithms. The Harris GOES-R GS core team is currently in process of designing and implementing this common algorithm architecture.

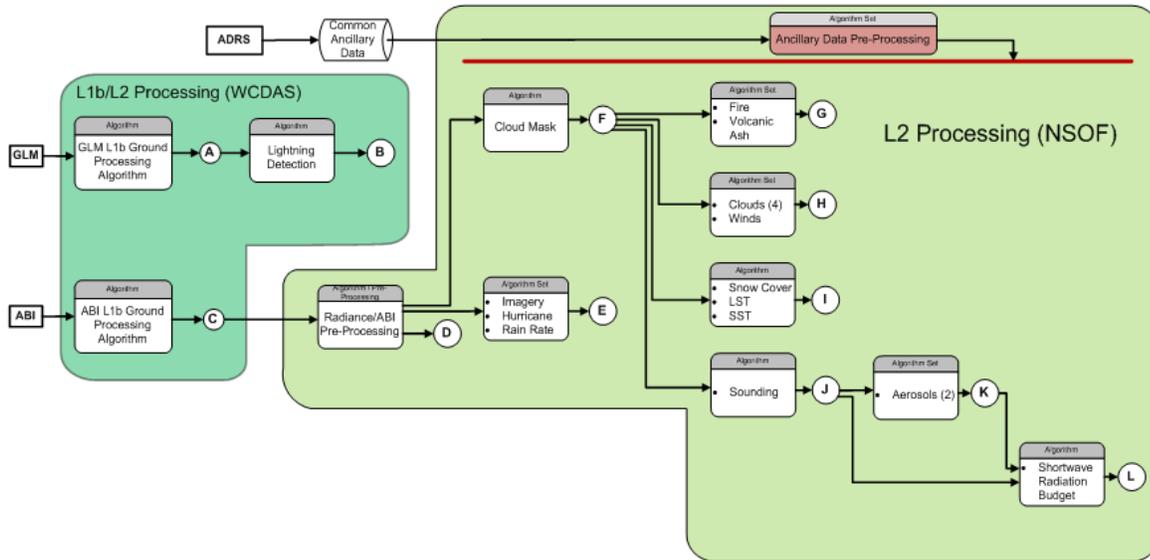


Figure 1. High level base-line GOES-R PG data flow diagram for ABI and GLM product generation from L0 data through L2+ products. The major output data elements denoted in this figure are: A) L1b GLM data, B) L2+ GLM GRB packets, C) ABI L1b GRB packets, D) L1b ABI radiance products, E) Cloud and Moisture Imagery, Hurricane Intensity and Rainfall Rate products, F) ABI Clear-sky Mask product, G) Fire and Volcanic Ash products, H) Cloud Phase/Properties and Derived Motion Wind products, I) Snow Cover and Land/Sea Surface Temperature products, J) Sounding and Stability Indices products, K) Aerosol Detection and Properties products and L) Shortwave Radiation Budget products.

3. GOES-R ALGORITHM FRAMEWORK AND DATA MODEL INTERFACE

The three basic architectural elements used to define both the Algorithm Engineering and GOES-R PG production environment are illustrated in Figure 2. These elements are: 1) An Algorithm Executor/Infrastructure, 2) A Data Model Interface, and 3) The algorithm itself. These generic constructs are design to facilitate the deployment of both Level 0 to Level 1b instrument calibration/geo-location and Level 2 algorithm at the primary production facilities (WCDAS and NSOF) as well as at the Remote Backup (RBU) facility. These three components together form a modular platform that separates scientific algorithm development from operational systems engineering. The separation of science from operations allows for algorithms to be developed with a focus on scientific correctness, while providing broad flexibility to the operational systems that will be controlling them. This flexibility allows for baseline implementations of the product generation system to be operated at all three different production facilities. The system's modular approach provides for a seamless mechanism for expanding the GOES-R GS operational capabilities as new sensors are added to future satellite platforms, e.g. GOES-S or GOES-T, and new algorithms are added to the Level 2 processing streams.⁴

In this design an “Algorithm”, denoted by the lower box in Figure 2, is a passive and stateless software entity that can either be instantiated on demand or requested from an object cache. The Algorithm Executor/Framework is responsible for managing algorithm instances, collecting the inputs needed to process any Level 1b or 2+ product data block, and invoking an algorithm based on a derived data “context” (described below). The final architectural element, which is illustrated by the central box in Figure 2, is the Data Model Interface or DMI. The DMI provides a common object-oriented interface enabling all algorithms, whether they are related to Level 0, Level 1b or Level 2 processing, to ingest data from and to export their resulting product data block to other algorithms or PG infrastructure services irrespective of the execution framework or operational environment. It provides a common, standardized, structure, memory-based interface to input data and output products for all scientific algorithms. In a development/test environment, the DMI provides methods for converting user selectable archived data into algorithm inputs, and standardize method for collecting and assembling algorithm outputs into test/analysis products. In the data-driven production environment, it is responsible for converting serial data streams into input data blocks and converting algorithm output data blocks into data items managed by the PG infrastructure.

In this design, Level 0, Level 1b, as well as other data enters the GOES-R PG environment via external interfaces in a serial/data driven fashion. In the operational environments, these interfaces include the Satellite data downlink interface and the GRB. The Algorithm Executor/Infrastructure is responsible for segmenting these streams into defined processing blocks/elements. These segmented processing elements are defined by “Contexts”. Each context encapsulates the spatial and/or temporal area on which an algorithm should be executed. Each algorithm’s implementation is developed to be fully parameterized by these contexts, allowing them to execute on a variety of differently sized areas depending on the context they are passed. This is the key to decoupling an algorithm’s scientific procedure from the operational process of determining optimal hardware block size and execution assignment.

Once contexts have been generated, the Algorithm Executor/Infrastructure is responsible for pre-fetching all required algorithm inputs, and invoking the algorithms once those data become available. Contexts in combination with an algorithm “Strategy” (not shown in Figure 2) fully define the spatial and/or temporal extent of an algorithm’s input data needs and its output products. This includes the nominal input data dimensions and the provided “padding”, i.e. the number of spatially or temporally adjacent samples/pixels need to compute a single output value. The Strategy defines required algorithm specific static/semi-static configuration information, and all of an algorithm’s data requirement. This includes required and optional Level 0/1b data, Auxiliary data (sensor and satellite geometry), Ancillary (non-instrument data provide by the ADRS) and predecessor data computed by prior algorithm instances. In this paradigm, the resulting Context defines what information an algorithm may request via the DMI, and what it is expected to product in terms of product and intermediate data as well as any state information (scratch data) that maybe needed by subsequent instances of the same or other algorithms. After the Executor pre-fetches all data for a given Context and establishes connectivity between the infrastructure, DMI and algorithm, it invokes the prescribed algorithm.

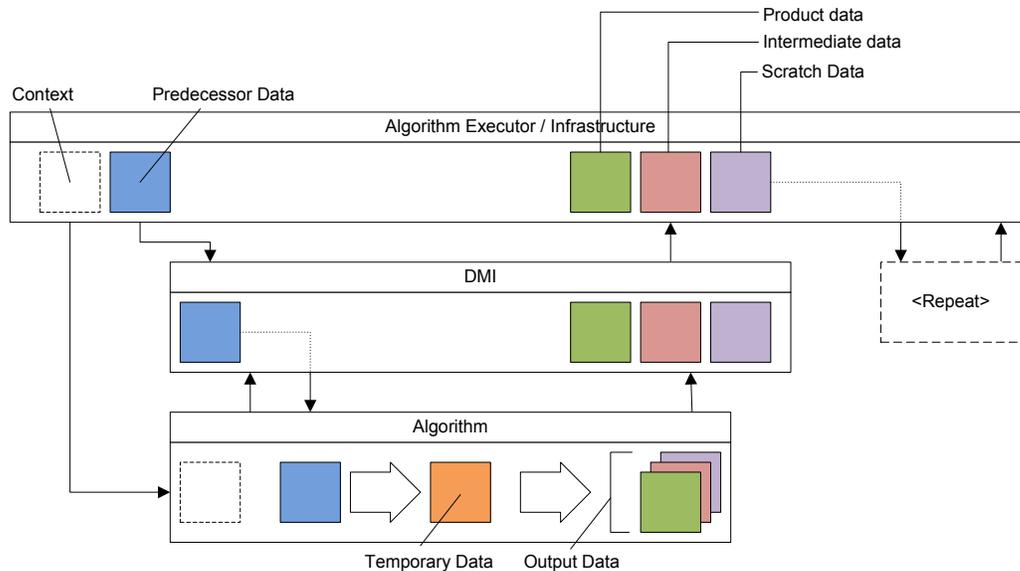


Figure 2. Top-level data flow diagram for GOES-R Level 1b and Level 2 algorithm implementation. Data driven algorithm Executor segments input data streams into logical processing blocks defined by algorithm “Contexts”. Data flows from infrastructure to stateless algorithm instance via the DMI which provides standardized I/O access to input data and configuration information, and common methods for storing resulting scientific data products. Data products

Once the algorithm is invoked, it reads the data required to generate output for its Context via its associated DMI instance. It will use those data to compute an output for all output samples/pixels defined by the given Context. In this design, the algorithm designer is free to compute the required output values one either a pixel by pixel basis or block like fashion. Once all the output product data for a given Context have been computed, the algorithm stores these values and any intermediate product data or dynamic information required by subsequent instances of the same algorithms to the infrastructure via the DMI. Since all algorithms are passive and state-less by nature, all information needed by subsequent down-stream process must be passed via the infrastructure through intermediate products, and any dynamic data needed by subsequent instances of the algorithm must be stored in the infrastructure as scratch data elements. Once an algorithm has provided it data to the infrastructure, it terminates and any temporary data generated during product computation is lost. This is true even if an algorithm resides in an object pool, since the infrastructure does not provide any guarantees that subsequent input Contexts will be executed on by the same instance of an algorithm.

The algorithm framework described about enables processing of subsequent input “Contexts” in either a sequential or parallel fashion as illustrated in Figure 3. This figure illustrates how the Algorithm Executor sub-segments an input data stream based on the data delivery time line and processing requirements to optimize the delivery of each Level 1b and Level 2 end product. The task of assembling products is either the responsibility of downstream PG algorithms whose Context and Strategy describe the full extent of a given end product, or the PD element which acquires data from common GOES-R data storage facility. Level 1b and Level 2 product developed at WCDAS are assembled in the PG element to facilitate the serial data stream needs of the GRB. Level 2 products computed at NSOF are assembled by the PD element for distribution to external users and interfaces.

In addition to the Algorithm Executor/Infrastructure, the PG element has a number of other companion software components. These components provide common data management and delivery services and math/function libraries. Some of the features provided by these services include common geo-spatial and temporal interpolation tools, common Radiative Transfer (RT) modeling approaches and standardized mechanisms for archiving/accessing calibration and algorithm specific coefficients.

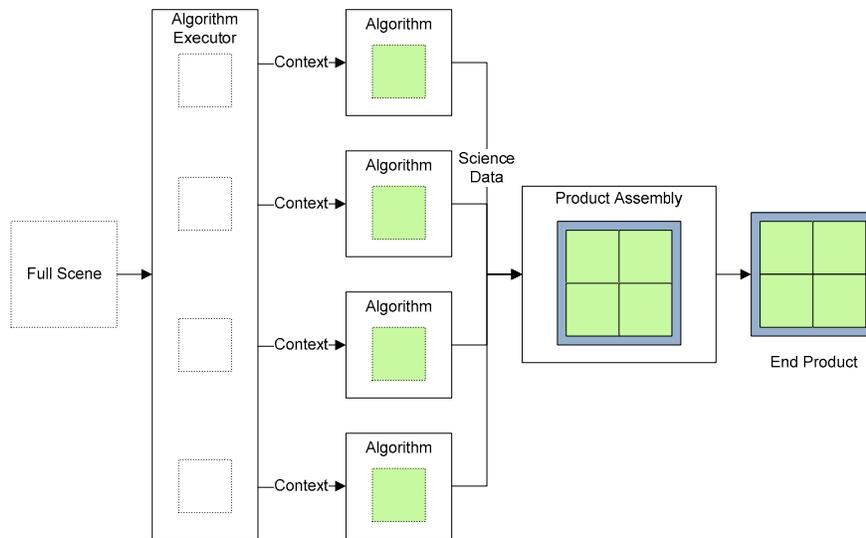


Figure 3. Diagrammatic view of GOES-R Level 1b or Level 2 product generation and assembly. PG Algorithm Executor partitions input data into processing block, which are later assembled by subsequent downstream process into end products

4. DISCUSSION

The GOES-R Product Generation architecture described above provides a highly modular and operationally flexible approach to scientific algorithm implementation. This design approach enables the concurrent development of both an Algorithm Engineering development environment, designed to facilitate transition of Government supplied documentation, and the PG production environment, which will be responsible for executing all GOES-R Level 1b and Level 2 algorithms in real-time high-reliability environment 24/7/365. It also provides for a natural and rapid transition path for algorithms developed in the AE environment to the PG production facility through the use of standardized interface designs, and provides inherent ability to accommodate future GOES-R instruments and algorithms without disrupting or modifying the baseline implementation.

REFERENCES

- [1] www.goes-r.gov
- [2] Hansen D., Bristow, J. Kalluri, S. Weiner, A. and Dittberner, G. J., "An Overview of the GOES-R Ground Segment Architecture", *Proc. SPIE*, **7813**, (2010).
- [3] Dittberner, G. J. Kalluri, S. Weiner, A. and Blanton, M., "The Product Generation Architecture for the GOES-R Ground System", *Proc. SPIE*, **7813**, (2010).
- [4] Zaccheo, T. S., Galantowicz, J., Hogan D. B., Kennelly, E. J. and Snell H. E., "Testbed Architecture for Rapid Prototyping and Assessment of Environmental Remote Sensing Algorithms", *Proc. SPIE*, **5548**, 37-47 (2004).