EDR Transition to NDE and Associated Algorithm Testing

Tom King[2], Shanna Sampson[2], Walter Wolf[1], Meizhu Fan[2], Claire McCaskill[2], Letitia Soulliard[2], Veena Jose[2], and Aiwu Li[2]

[1]NOAA/NESDIS/STAR, College Park, MD 20740 USA

[2]IMSG, College Park, MD 20740, USA

The ASSISTT Role in Algorithm Transition to NDE – Process

- Implementation of QA and Standard Processes
  - Work with teams to conduct a standard set of project reviews
  - Standard set of operational documentation (ATBD, SMM, EUM)
  - Facilitating stakeholder interaction (e.g. NDE, OSPO, JPSS, NCEI, NWS, JCSDA, NASA, DOD, CSPP, etc.)
    - Supporting pIPT, ESPDS PGIPT, TIMs
    - Work with ESPC on availability of operationally supported ancillary data
    - Working with end users to identify specific contents required in product files
  - Requirements development/refinement
  - Risk tracking and mitigation

The ASSISTT Role in Algorithm Transition to NDE – Technical

- Technical Work
  - Code cleanup, refactoring, rewriting
  - Coding standards
  - Use of standard languages, tools, and libraries
  - Implementation of common data formats and metadata (CF & NDE)
  - Configuration management
  - Delivered Algorithm Package (DAP): packaging, verification, and delivery
  - Algorithm package testing

The ASSISTT Algorithm Package Testing – What We Do and Why

- Unit testing
  - Verify functionality new code
  - Regression testing for validating old capabilities.

- End-to-end testing in near real time over long periods using automated offline demonstration systems
  - Ensure code is robust
  - Acquire information on algorithm system requirements (run times, memory, disk usage)
  - Produce data to support science development and validation efforts
    - Provide data for LUTs
    - Provide data to validate the science
  - Provide product distribution to support to end users
    - User pre-operational readiness
    - User-supplied data/product validation
  - Provides a way to check or troubleshoot issues in the NDE I&T or OPS string
    - STAR doesn't have access to NDE and OSPO hardware, compilers, file systems
    - If we use the same input data stream, code, hardware, compilers we may be able to actually diagnose and fix problems
    - May reveal subtle issues with environmental differences, problems with production rule implementation differences or DAP update implementation

Algorithm Testing Examples – NUCAPS Offline Demonstration Processing

- NUCAPS Offline Demonstration Processing System
  - End-to-end processing system generating all products globally
    - Schedules and manages multiple runs, simultaneous jobs, ingests data, executes algorithms, handles processing dependencies, and distributes products.
    - Executes pre and post-processing of data. Runs the Reformatter Toolkit for BUFR generation.
    - Modular such that individual processing units can be extracted from the NUCAPS offline system, put into a DAP, and delivered to NDE. This works because the delivered modules have the interfaces as the NDE system.
    - Automated and running continuously 24/7 for the last 8 years.
    - Environment and compilers match that of NDE (OS and compilers).
  - Data
    - Initially generated simulated CrIS and ATMS from GFS using the SDP Toolkit and then produced output mimicking the CrIS and ATMS SDR and TDR HDF5 formats.
    - Currently pulls the CrIS SDR, ATMS TDR, and VIIRS JPSS Enterprise Clouds from SCDR.
    - Distributes through the STAR ftp server to (e.g. NCO, AWIPS, NCEI, CPC, NPROVS, GMAO, DOD).

Algorithm Testing Examples – NUCAPS Offline Demonstration Processing

- NUCAPS Offline Demonstration Processing System tasks:
  - Supported data flow of CrIS FSR BUFR to NCO/EMC for 1.5 years prior to operational implementation (NCO ingest and GFS implementation)
  - Generate Focus days for the science team in support of regression generation and validation
  - Supported troubleshooting differences between NDE I&T and OPS
  - Running continuously to provide EDRs to NPROVS for T & q validation
  - Running in different modes
    - Comparing different versions of the retrieval
    - Full spectral CrIS vs Nominal spectral CrIS
    - IR-only vs IR+MW
    - S-NPP vs J1
    - With or without VIIRS clouds
      - VIIRS clouds from JPSS Enterprise vs IDPS

Algorithm Testing Examples – GCOM Offline Demonstration Processing

- GCOM Offline Demonstration Processing System:
  - End-to-end processing system generating all products globally
    - Schedules and manages multiple runs, simultaneous jobs, ingests data, executes algorithms, handles processing dependencies, and distributes products.
    - Executes pre and post-processing of data. Runs the Reformatter Toolkit for BUFR generation.
    - Modular such that individual processing units can be extracted from the GCOM offline system, put into a DAP, and delivered to NDE. This works because the delivered modules have the interfaces as required by the NDE system.
    - Automated and running continuously 24/7 for the last 4 years.
    - Environment and compilers match that of NDE (OS and compilers).
  - Data
    - Initially used AMSR2 L1B and L1R from JAXA
    - Currently pulls the AMSR2 L1B and L1R from SCDR (from NDE via PDA).
    - Distributes through a dedicated ftp server to end users (NHC, NAVO, NWS, OSPO)

Algorithm Testing Examples – GCOM Offline Demonstration Processing

- GCOM Offline Demonstration Processing System Tasks:
  - Support early data availability for end users
  - Support reprocessing for science team validation
  - Supported troubleshooting differences between NDE I&T and OPS
  - Supported testing of JAXA code in the processing chain to evaluate
    - JAXA code version differences
    - Allocated latency
    - Timestamp bugs
  - Supported testing impacts of granule vs orbital dump processing
  - Supported reprocessing for GHRSST generation and validation

Algorithm Testing Examples – STAR ASSISTT Cluster Offline Demonstration Processing

- STAR ASSISTT Offline Demonstration Processing System:
  - End-to-end processing system generating products globally
    - Schedules and manages multiple runs, simultaneous jobs, ingests data, executes algorithms, handles processing dependencies, and distributes products.
    - Executes pre and post-processing of data. Runs the Reformatter Toolkit for BUFR generation.
    - Modular such that individual processing units can be extracted from the Cluster, put into a DAP, and delivered to NDE. This works because the delivered modules have the interfaces as the NDE system.
    - Automated and running continuously 24/7.
    - Environment and compilers match that of NDE (OS and compilers).
  - Data
    - Pulls JPSS data from SCDR (from GRAVITE via PDA).
    - Distributes through
      - STAR ftp server to end users (NWS, OSPO)
      - STAR ftp THREDDS server to access controlled data

Algorithm Testing Examples – STAR ASSISTT Cluster Offline Demonstration Processing

- STAR ASSISTT Cluster
  - Uses HTCondor for high-throughput distributed processing of jobs
  - 13 machines (Production)
    - 24 cores, 256 GB memory each
  - 4 machines (QA)
    - For testing of upgrades and other software packages before deploying to the rest of the cluster
  - 8 additional servers waiting to be setup
  - Currently Implementing
    - Docker
      - Combined with other tools for continuous integration
  - Currently Evaluating
    - Job schedulers (currently using CRON, looking at JobScheduler)
    - File systems (GlusterFS)

Algorithm Testing Examples – STAR ASSISTT Cluster Offline Demonstration Processing

- STAR ASSISTT Cluster Offline Demonstration Processing System Tasks:
  - Supports end-to-end NRT testing of JPSS Enterprise algorithm packages for limited coverages (see next slide)
  - Future support of early data availability for end users
    - VIIRS Enterprise LST to EMC
  - Run short-term (2.5 month) global processing to validate latest science team updates with CRTM 2.1.3 for next Enterprise DAP delivery (Sep 2017)

JPSS Enterprise Algorithms Running in NRT on the STAR Cluster

Algorithm	Current Coverage
Cloud Mask	CONUS
Cloud Phase	CONUS
Cloud Height	CONUS
Cloud DCOMP	CONUS
Cloud NCOMP	CONUS
Cloud Base	CONUS
Aerosol ADP	CONUS
Aerosol AOD	CONUS
Volcanic Ash	CONUS
Ice Concentration	CONUS and Polar
Ice Age	CONUS and Polar
Snow Cover	CONUS and Polar
Polar Winds	Polar
Land Surface Albedo	CONUS
Land Surface Temperature	CONUS
Coverage of JPSS NRT algorithms: CONUS (126 W, 66 W; 22N, 50 N) Polar regions (above of 50)	

Goal: to have global processing of the Enterprise Algorithms within the HTCondor cluster by the end of 2017

Summary

- ASSISTT facilitates the NDE EDR transition to ops process with near real time algorithm testing in offline demonstration systems as follows:
  - Supports software functional testing
  - Supports science development and validation
  - Supports troubleshooting in I&T and OPS
  - Supports user readiness