Plans to transition the NCEP data assimilation system to JEDI

Andrew Collard[1], Cory Martin[1], Emily Liu[1], Ron McLaren[2], Kevin Dougherty[3],Nicholas Esposito[3], Azadeh Gholoubi[4], Praveen Kumar[2], Xuanli Li[3], Haixia Liu[2], Russ Treadon[1], Daryl Kleist[1]
Ben Ruston, Greg Thompson and All Our Colleagues from JCSDA and Partners

[1]NOAA/NCEP/EMC [2] Lynker @ NOAA/NCEP/EMC [3]SAIC@NOAA/NCEP/EMC
[4]Axiom@ NOAA/NCEP/EMC [5]JCSDA

andrew.collard@noaa.gov

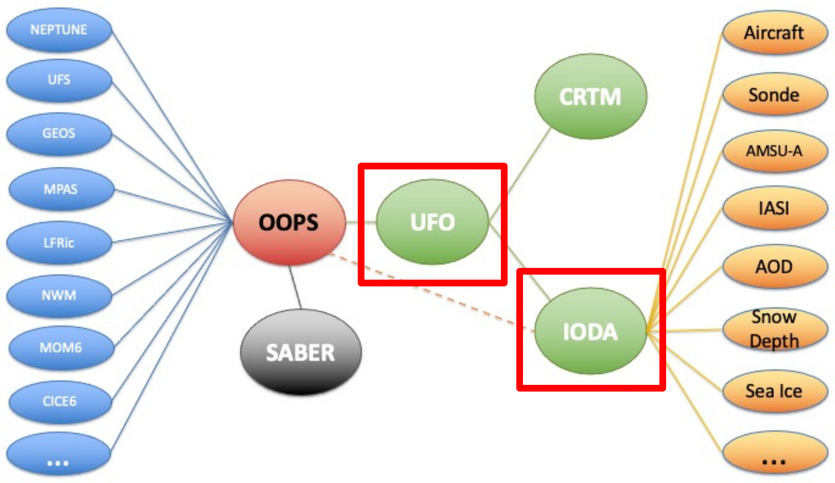
Joint Effort for Data assimilation Integration (JEDI)
Infrastructure for Unified Data Assimilation

JEDI is a project within the Joint Center for Satellite Data Assimilation (JCSDA)

JEDI provides a **software infrastructure for data assimilation** that
- is model agnostic
- is generic and portable, from toy models running on laptops to operational Earth system coupled models running in the cloud.
- enables DA on the model native grid
- does not impose one specific DA methodology or algorithm
- provides a framework for rapid uptake of new observations into operations with generic observation handling and modeling
- encourages implementation of model-independent observation operators
- provides a unified Interface for Observation Data Access (IODA)

JEDI is intended for **scientific exploration** and **operational forecasting**.

The keys to success are **separation of concerns** and **interfaces**.

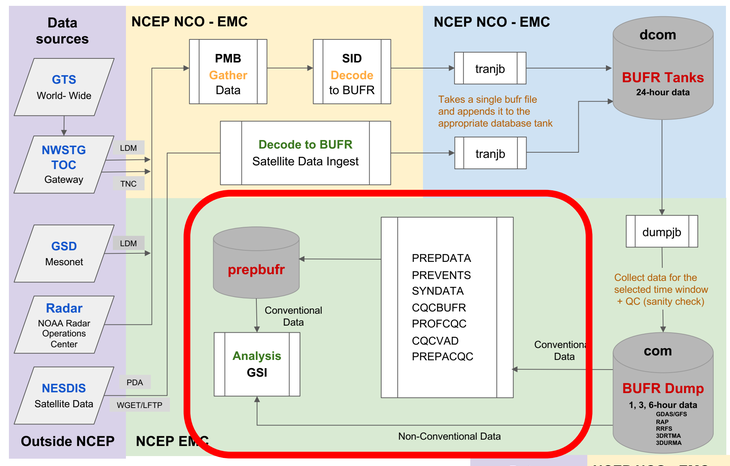


This poster concerns:

Interface for Observation Data Access (IODA):
Replaces functionality of observation pre-processing (ObsProc) and GSI data ingest at NCEP

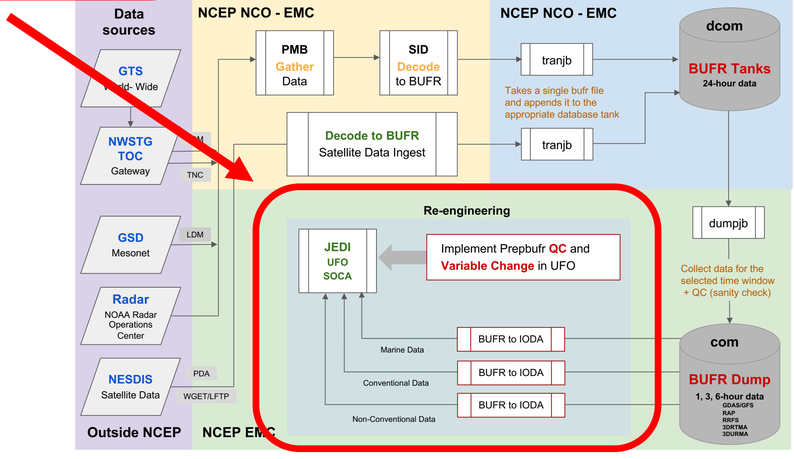
Universal Forward Operator (UFO):
Functionality of the "observer" component of GSI

From https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/inside/jedi-components/ioda/introduction.html

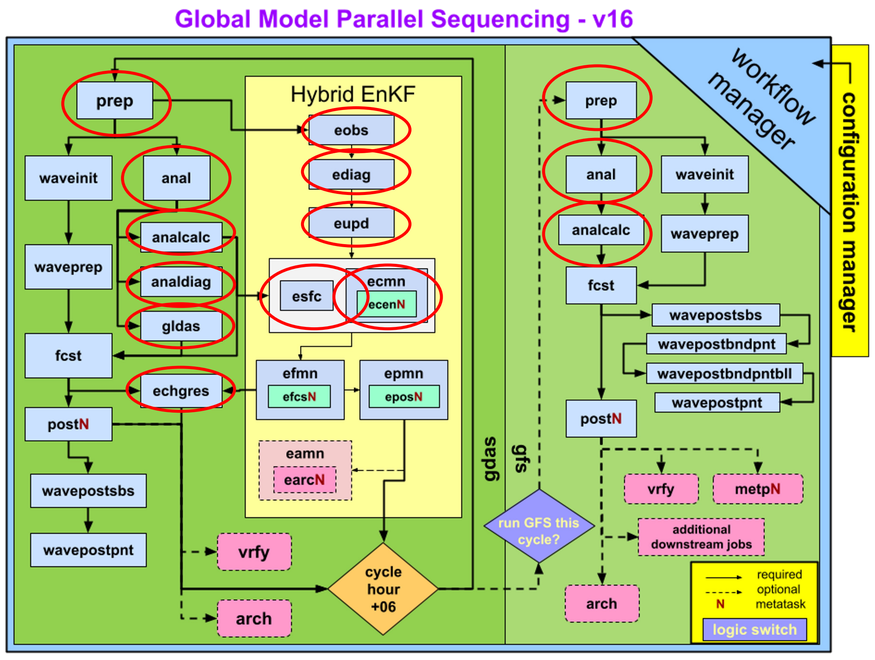


"ObsProc Re-engineering" replaces ingest of BUFR into GSI with IODA converters.

Eventually will replace the individual BUFR dumps tuned to each model's assimilation window with a database solution.



Eventually JEDI will replace at least 11 components of the atmospheric data assimilation system.

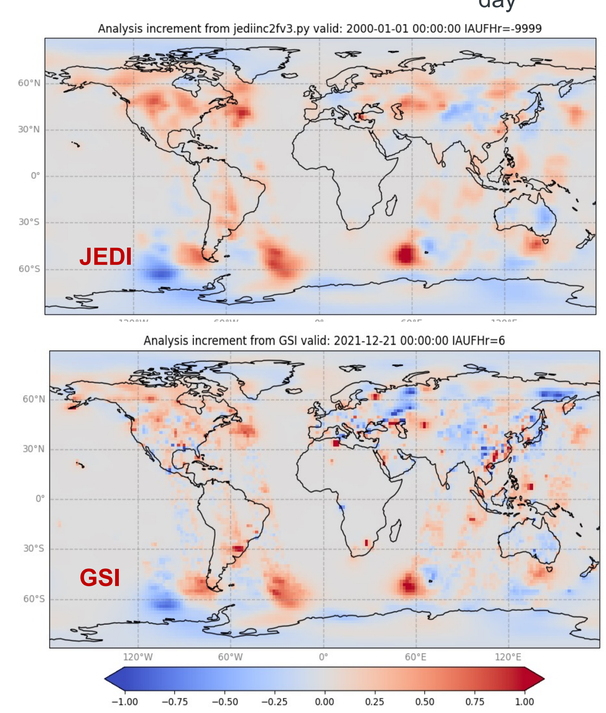


GDASApp

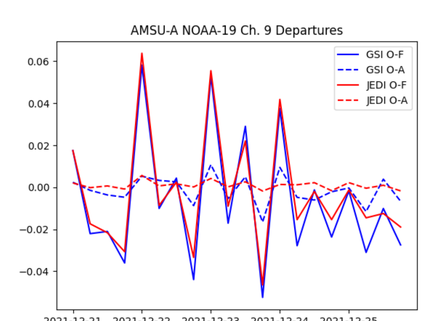
- NOAA-EMC github repository
  - https://github.com/noaa-emc/gdasapp
  - Connects JEDI to the global-workflow
  - Includes scripts and YAMLs for standalone testing
  - Enables easier validation of JEDI components
- Components currently being developed:
  - Atmosphere (v18)
  - Marine
  - Snow
  - Aerosol

A low-resolution, simplified proof of concept

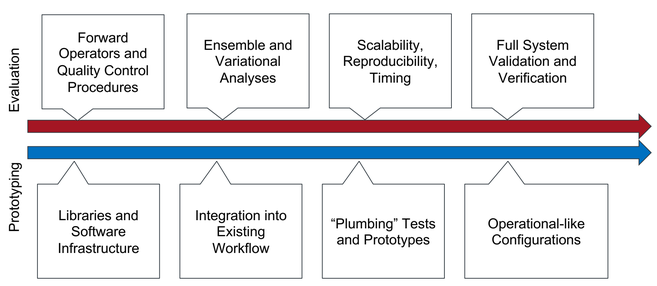
- Atmosphere-only, 3DVar cycling experiments
  - 1 degree forecast, analysis performed at 2 degree resolution
  - Prototype GSI-based static **B** used in JEDI, default static **B** used in GSI
  - All GSI 'bells and whistles' turned off for more direct comparison
    - No TLNMC, no humidity constraints, no dry mass constraint, no VarQC, etc.
  - Assimilated radiosondes and AMSU-A NOAA-19 radiances
  - Performed using existing GFS workflow
- Cycled for 5 days, ran a 120 hour forecast at 00z each day



Temperature analysis increments at approximately 500 hPa for FV3-JEDI and GSI look qualitatively similar as are fits to AMSU radiances.



Parallel Evaluation and Prototyping



Translation of Quality Control Procedures

```
! QC for clear condition
    else  ! <clw4crtm>
       qc4emiss=.false.
       if(factch6 >= one .or. latms_surfaceqc)then
          efactmc=zero
          vfactmc=zero
          errf(1:ich544)=zero
          varinv(1:ich544)=zero
          do i=1,ich544
             if(id_qc(i) == igood_qc)id_qc(i)=ifail_factch6_qc
          end do
          if(id_qc(ich890) == igood_qc)id_qc(ich890)=ifail_factch6_qc
          errf(ich890) = zero
          varinv(ich890) = zero
          if (latms) then
             do i=17,22   !  AMSU-B/MHS like channels
                if(id_qc(i) == igood_qc)id_qc(i)=ifail_factch6_qc
                errf(i) = zero
                varinv(i) = zero
             enddo
          endif
!      QC3 in statsrad
          if(.not. mixed.and. luse)aivals(10,is) = aivals(10,is) + one
```

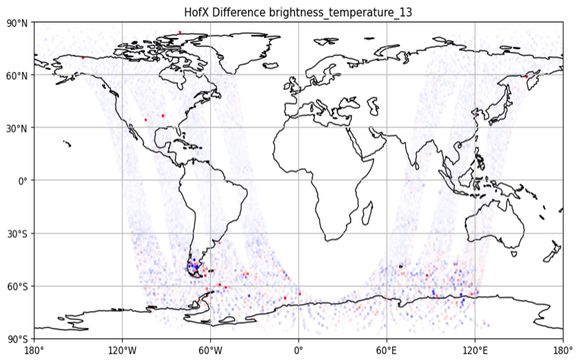
Compiled Fortran code in the GSI

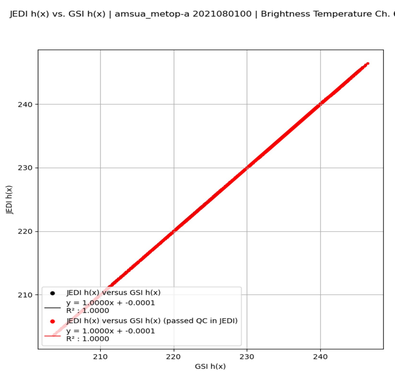
```
# Step 3: CLW Retrieval Check (background_based)
- filter: Bounds Check
  filter variables:
  - name: brightnessTemperature
    channels: 1-6, 15
  test variables:
  - name: DerivedMetaData/CLWRetFromBkg
  maxvalue: 999.0
  actions:
    - name: set
      flag: CLWRetrievalReject
    - name: reject

# Step 4: Window channel sanity check
- filter: Bounds Check
  filter variables:
  - name: brightnessTemperature
    channels: 1-6, 15
  test variables:
  - name: DerivedMetaData/Innovation
    channels: 1, 2 4-6, 15
  maxvalue: 200.0
  minvalue: -200.0
  flag all filter variables if any test variable is out of bounds: true
```

YAML input to JEDI/UFO

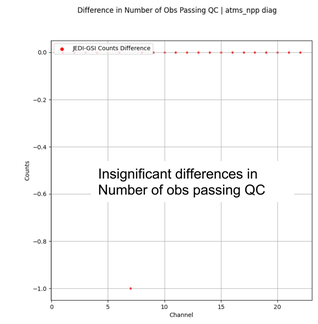
Validation of Forward Operators

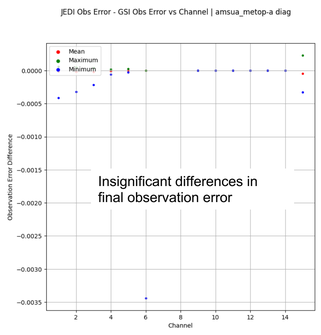
Most validation work to date has focused on the UFO
- Model simulated observations (H(x))
- Quality Control
- Error Assignment





For the same model fields, do we get the same simulated observations? Yes!
<- Interpolation leads to some differences though





JEDI Transition Summary

Notional schedule for UFS-applications is not (and may not perfectly align with) transition of JEDI infrastructure. This necessitates an incremental, multi-tiered approach:

1. Observation Processing Reengineering – Replace legacy "obsproc" software with JEDI-based tools (IODA + UFO)
2. From GSI to JEDI, starting with GFS/GDAS
   a. UFO development and acceptance
   b. Replace GSI-based EnKF with JEDI-LETKF (perturbation update)
   c. Completely replace GSI-based solver with JEDI (starting with hybrid 4DEnVar as is currently operational)
3. From GSI to JEDI for other atmospheric applications (RRFS, HAFS, and RTMA, all of which will leverage GSI in initial operational capabilities)
4. New capabilities for non-atmospheric components directly in JEDI
   a. **marine, land, atmospheric composition**