The Joint Effort for Data assimilation Integration (JEDI)

Introduction to JEDI

Joint Center for Satellite Data Assimilation (JCSDA)

AMS Short Course - JEDI Introduction - 10 March 2021

JEDI: Motivations and Objectives

Reduce duplication of effort between JCSDA partners

- Adding new observations (UFO and IODA)
- Implementation of new DA algorithms (OOPS)

Bring all components of Earth-system in one DA system

- Develop DA algorithms once for all components (OOPS)
- Enable future coupled DA developments (OOPS)

For research and operations (and O2R2O)

JEDI: Approach

Modern DA systems are too complex for any one person to grasp entirely

- Collaborative developments
- Separation of concerns

Modernize software

- Speed-up future developments
- Ease maintenance
- Increase portability and efficiency

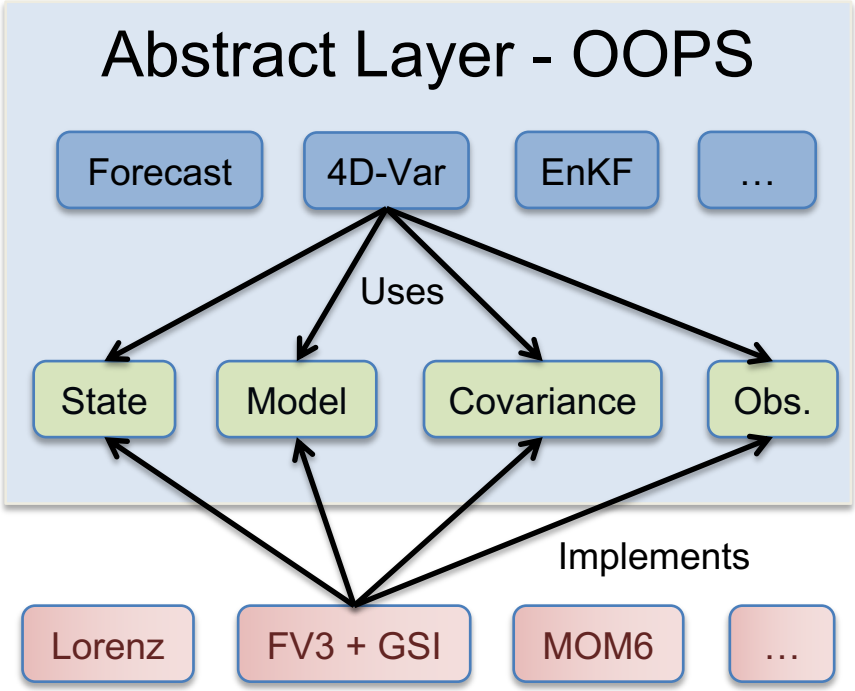
Object Oriented Prediction System (OOPS)

Generic, portable, model-agnostic DA system

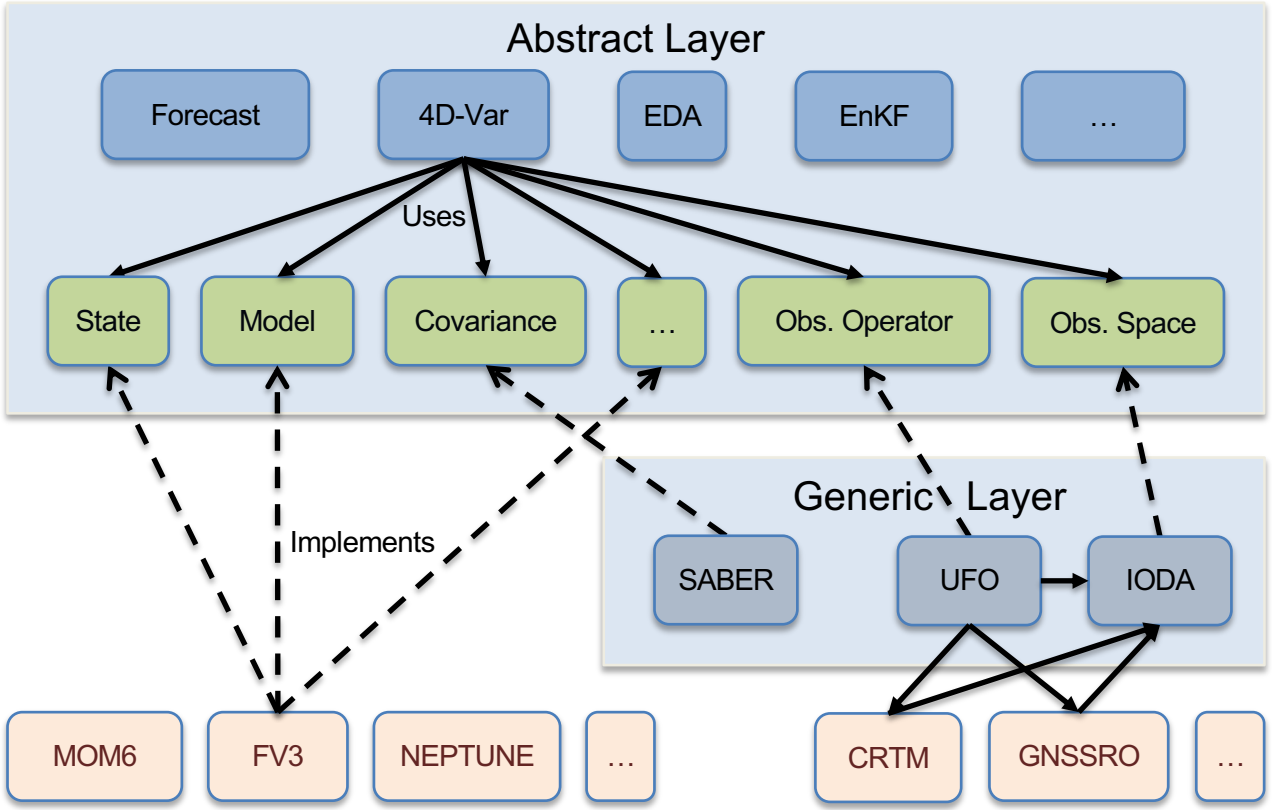
Use **object-oriented** and **generic** programing

Each model implements pre-defined abstract interfaces

Separation of concerns

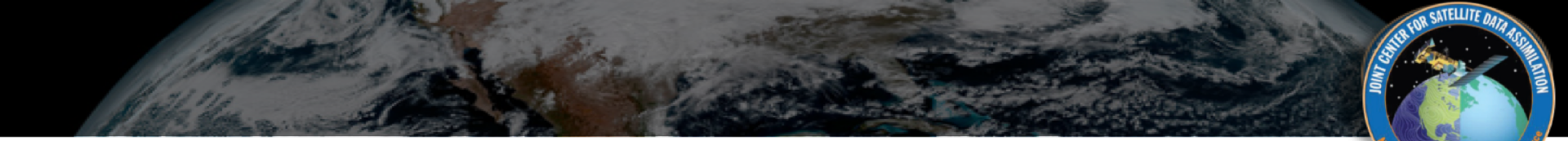


JEDI: Abstraction and Genericity



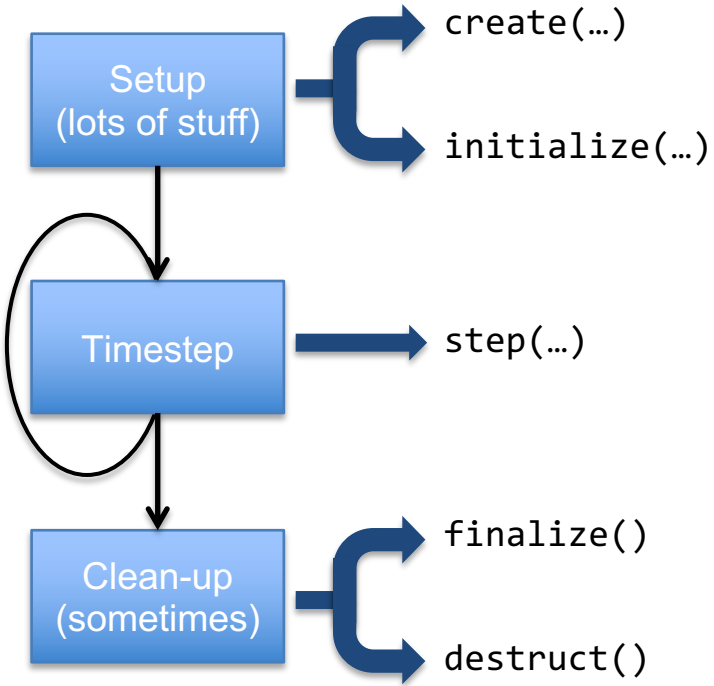
Abstract, model-agnostic DA system

OOPS is complemented by generic (shared) components.



JEDI Model Interfaces

Model design



Between model "steps" OOPS calls post-processors

- OOPS manages when post-processors are called
- Post-processing removed from model code (**separation of concerns**)

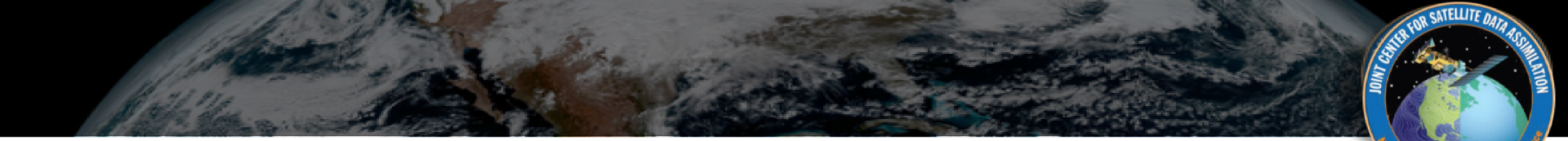
Post-processors isolate data assimilation from the model (**separation of concerns**)

- Computing simulated observations H(x)
- Jc-DFI, …

Post-processors do not modify the State

Models Interfacing

MODEL	TYPE	CENTER
FV3GFS (UFS)	Atmosphere	NOAA-EMC
GEOS	Atmosphere	NASA-GMAO
FV3GFS GSDChem	Atmospheric chemistry	NOAA-ESRL
GEOS-AERO	Atmospheric aerosols	NASA-GMAO
MPAS	Atmosphere	NCAR
LFRic	Atmosphere	Met Office (UK)
UM	Atmosphere	Met Office (UK)
MOM6	Ocean	NOAA-EMC
SIS2	Sea ice	NOAA-EMC
CICE6	Sea ice	NOAA-EMC
NEPTUNE	Atmosphere	NRL
QG	Idealized model	ECMWF
Lorenz 95	Idealized model	ECMWF
Shallow Water	Idealized model	NOAA-ESRL



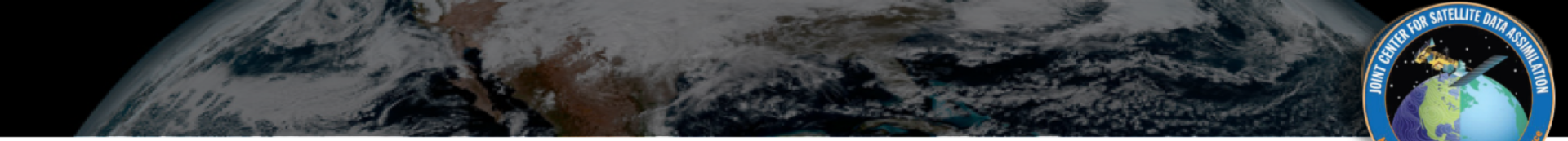
JEDI Observations Interfaces

Unified Forward Operator (UFO)

- Share observation operators between JCSDA partners and reduce duplication of work
  - Taking the model agnostic approach one level down into the observation operators
- Faster use of new observing platforms
  - Regular satellite missions are expensive
  - Cube-sat have short expected life time
  - Include users and instruments science teams
- Unified Forward Operator (UFO)
  - Build a community *app-store* for observation operators

UFO Observation “filters”

- Abstract “observation filters” are called before and after the obs. operator
- Observation filters are generic and have access to
  - Observation values and metadata
  - Model values at observations locations (GeoVaLs)
  - Simulated observation value and diagnostics (for post-filter)
  - Their own private data
- Filters are written once and used with many observation types
- Generic filters already exist for:
  - Gross error check, background check, blacklisting, thinning…
  - Entirely controlled from yaml configuration file(s)
- More filters will be developed as needed
  - Generic filters should cover most needs



JEDI Working Practices

Many people, many organizations, many models

How is fast progress possible?

Infrastructure, working practices

Project methodology inspired by Agile/SCRUM

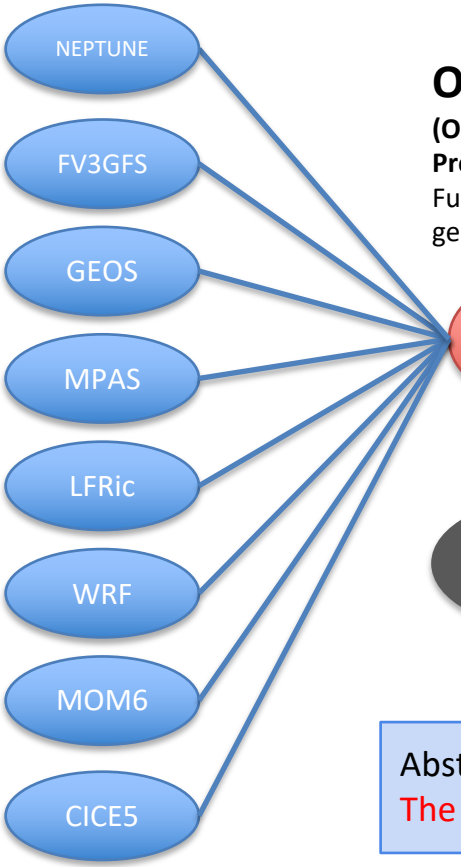
- Adapted to distributed teams and part time members
- Work in small manageable increments with constant feedback

Collaborative environment

- Easy access to up-to-date source code (github)
- Easy exchange of information (zenhub)
- Pull requests and code reviews (all developers actively involved)
- Regular meetings by video
- Code sprints (8-10 developers working together on a specific topic)

Object-oriented programming and independence of code components (separation of concerns)

Code and repositories



OOPS
(Object Oriented Prediction System)
Full data assimilation generic algorithms

UFO
(Unified Forward Operator)
The 'app-store' of model-agnostic observation operators

CRTM
(Community Radiative Transfer Model)
Accurately and efficiently simulate satellite radiances

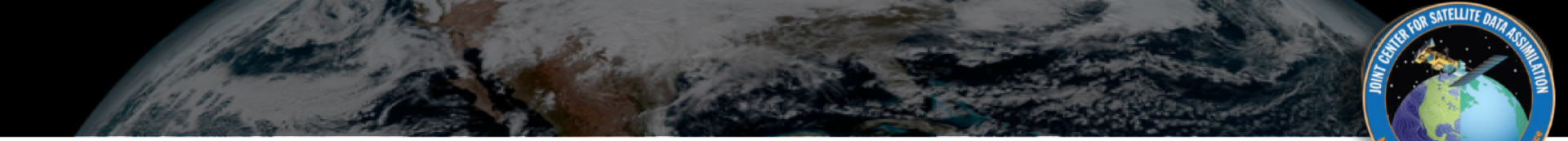
IODA
(Interface for Observation Data Access)
Performs all the I/O of the observations

SABER
(System-Agnostic Background Error Representation)
Generic background error covariance (incl. BUMP)

Abstract interfaces (separation of concerns) are the most important aspect of the design
The end of the monolithic gigantic jumble of code

Infrastructure, working practices

- Enforce software quality
  - Correctness, coding norms, efficiency
- Continuous Integration, Testing framework
  - Toolbox for writing tests
  - Automated running of tests (on pull requests)
- Effort on portability
  - Flexible build system (ecbuild, cmake-based)
  - Automatically run tests with several compilers
  - JEDI available in containers (singularity, charliecloud)
- Documentation, training
  - Doxygen, sphynx, (readthedocs)
  - JEDI Academies, tutorials



Final Comments

Final Comments

JEDI is bringing modern software development technologies and working practices to the data assimilation community

- The technologies in use are all proven in the software industry
- Changing working habits/practices is the most challenging aspect, it takes time…

In the future joint data assimilation environment:

- Technical infrastructure is shared as much as possible
- Common components (**H**, **B**, **R**…) are made available to all the partners when/where it makes sense
- Each partner keeps their own applications and choice of components and data assimilation algorithm they use

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