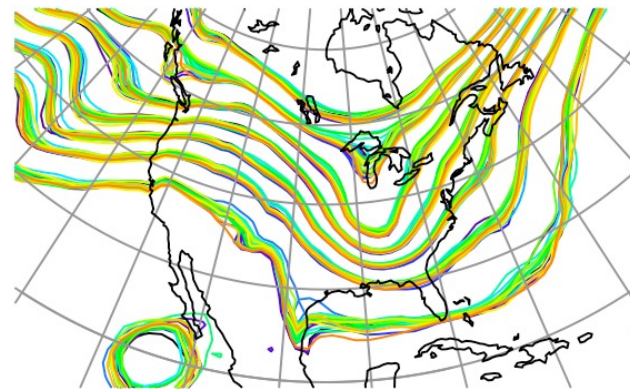


D
A
R
T

ata
ssimilation
esearch
estbed



DART Tutorial Section 2: The DART Directory Tree



©UCAR



The National Center for Atmospheric Research is sponsored by the National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

NCAR | National Center for
UCAR | Atmospheric Research

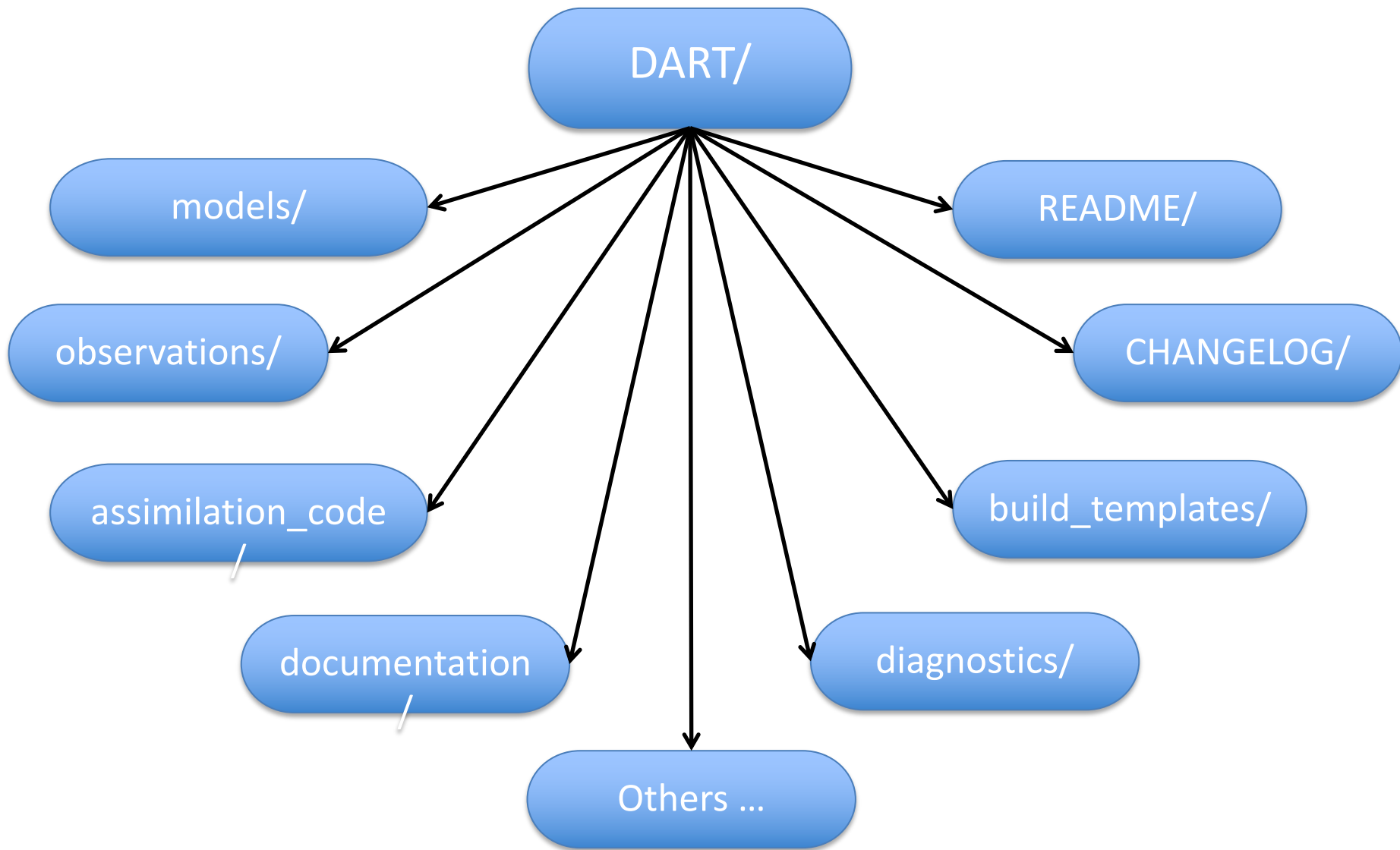
The DART Code Tree

Much of DART is implemented as Fortran-90 modules and programs.

DART also contains:

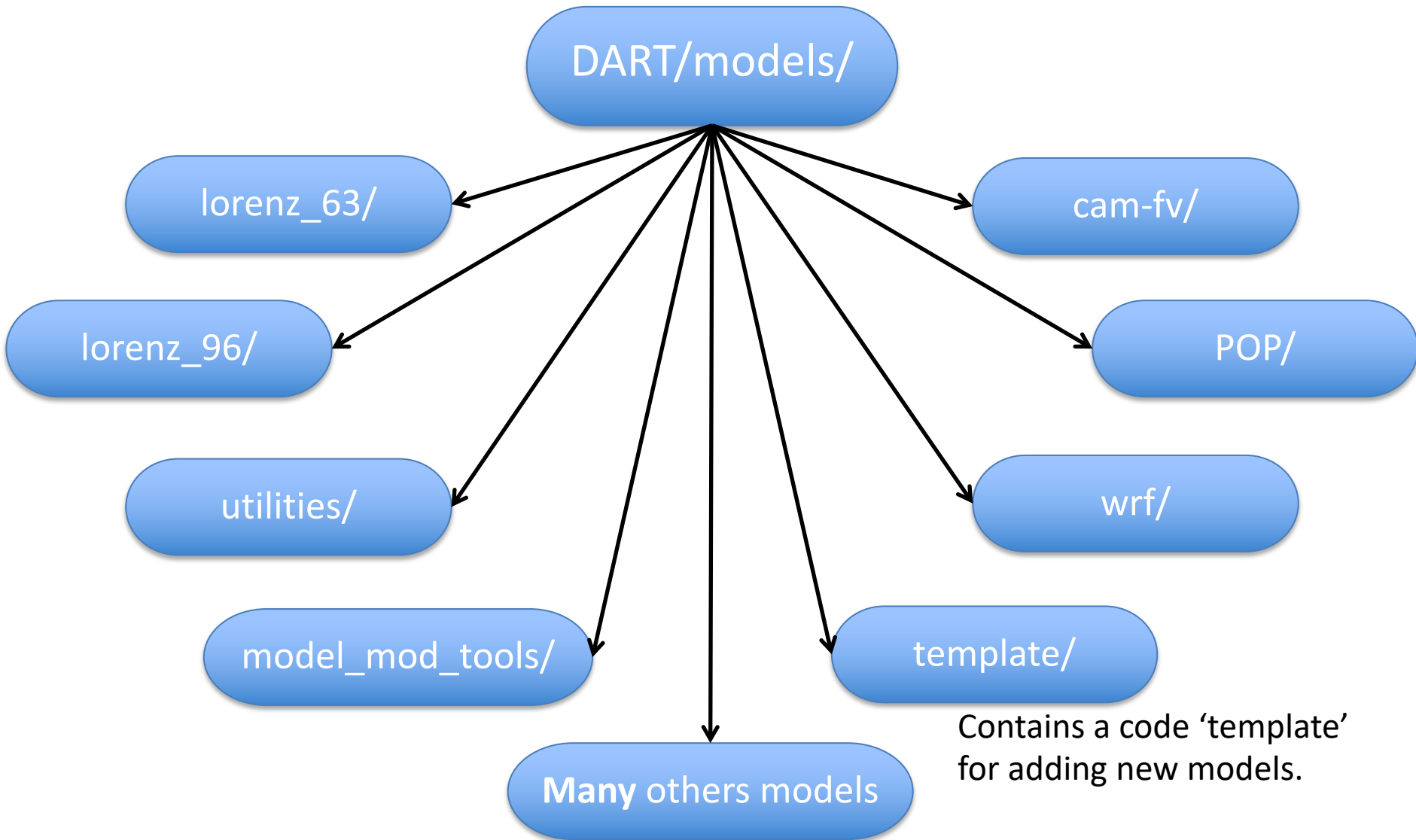
- Documentation (really!),
- Namelist control files,
- Compilation tools,
- Shell scripts for managing large applications, and
- Diagnostic tools.

DART Top-level directory structure

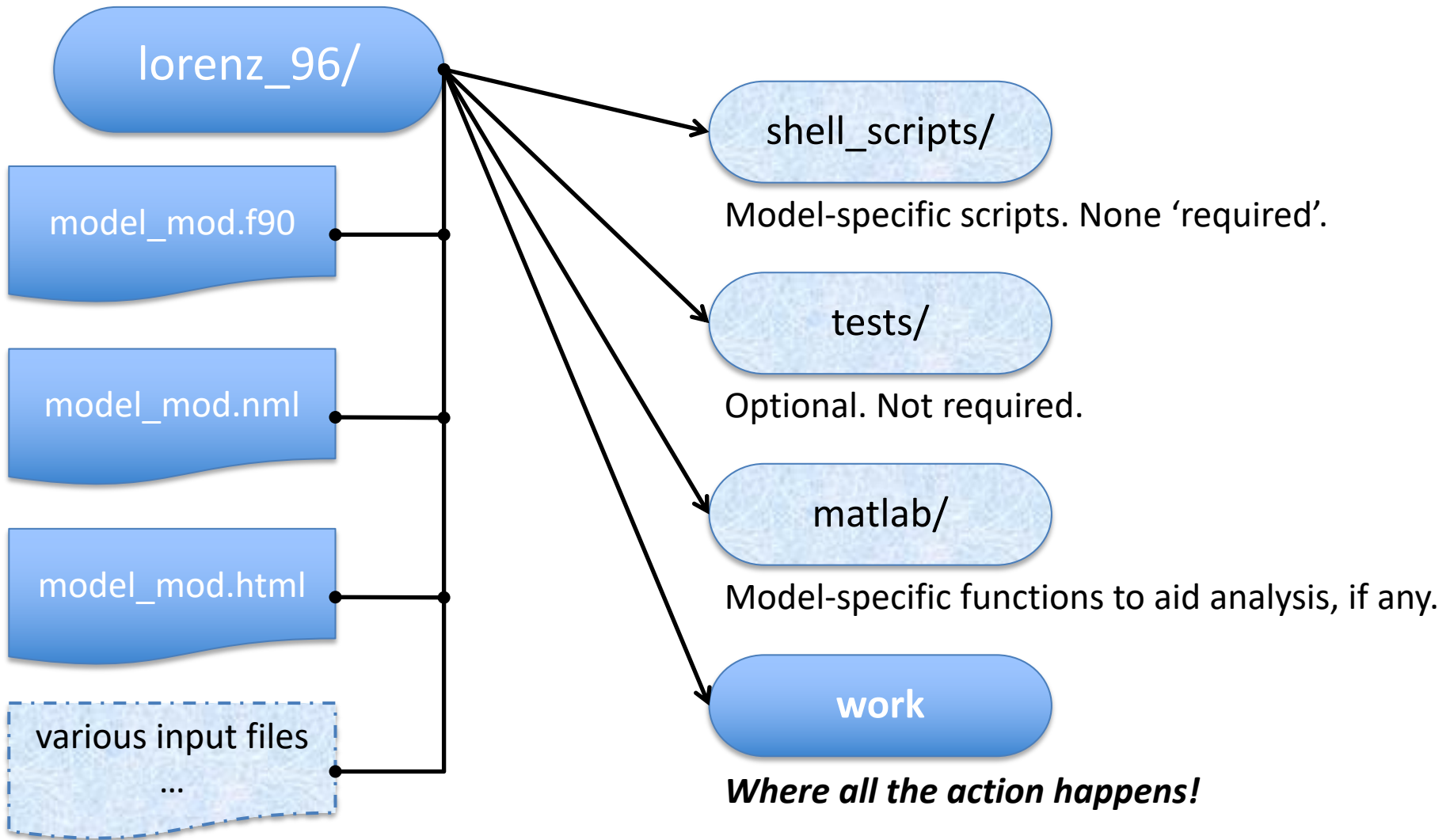


Peruse *your* DART subdirectories!

DART models directory:



Example DART model directory detail:



Anything with this background is optional.

DART model/work directory details:

work/

Executables are built and run in model work directories. Makefiles and compiler output files reside here. Input and output files generally reside here. Lots of other junk files tend to accumulate here. Check out contents of *models/lorenz_63/work*.

input.nml

file used by all DART programs for user control

workshop_setup.sh

script used to run 'set' experiments for some workshop exercises. Not all models run workshop experiments.

quickbuild.sh

script used to compile ALL applicable DART programs for this model. Feel free to take a peek, but no need to understand the details.
./quickbuild.sh help

obs_seq.out.xxxxxx

Sequence of observations to be assimilated for case xxxxxx

DART module files:

DART Fortran-90 code comes as code, documentation, and run-time control files. For instance, the directory *assimilation_code/modules/assimilation/* contains the following three files that implement localization (more on this later).

`cov_cutoff_mod.f90`

Code for module ***cov_cutoff_mod***

`cov_cutoff_mod.html`

Documentation for module

`cov_cutoff_mod.nml`

Run-time control for module

DART observations directory:

DART/observations/

```
graph TD; A[DART/observations/] --> B[forward_operators/]; A --> C[obs_converters/]; A --> D[utilities/];
```

forward_operators/

Code to computer forward operators for many types of instruments and for some idealized models.

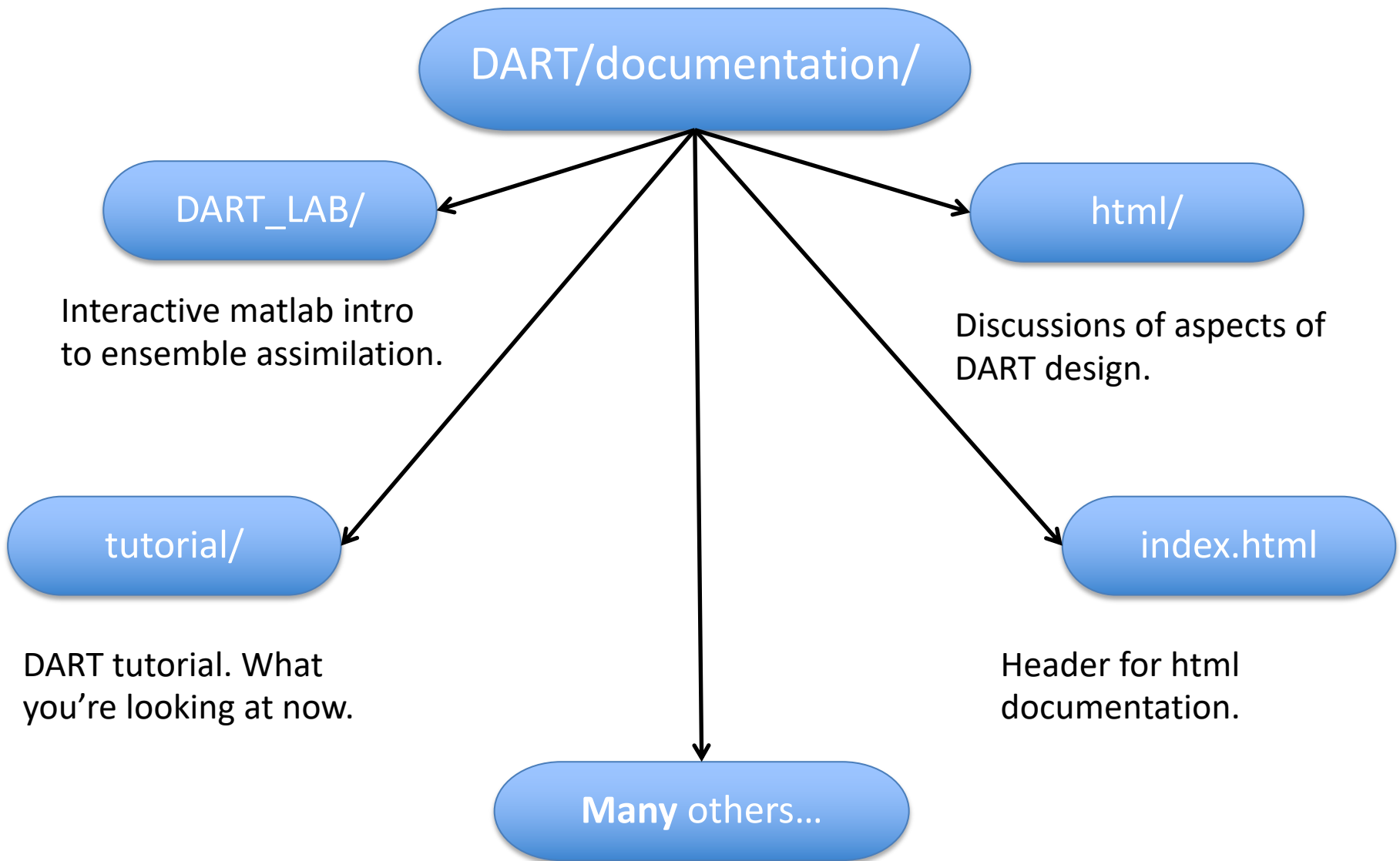
obs_converters/

Directories containing code and build tools for programs that create observation sequence files from many data sources.

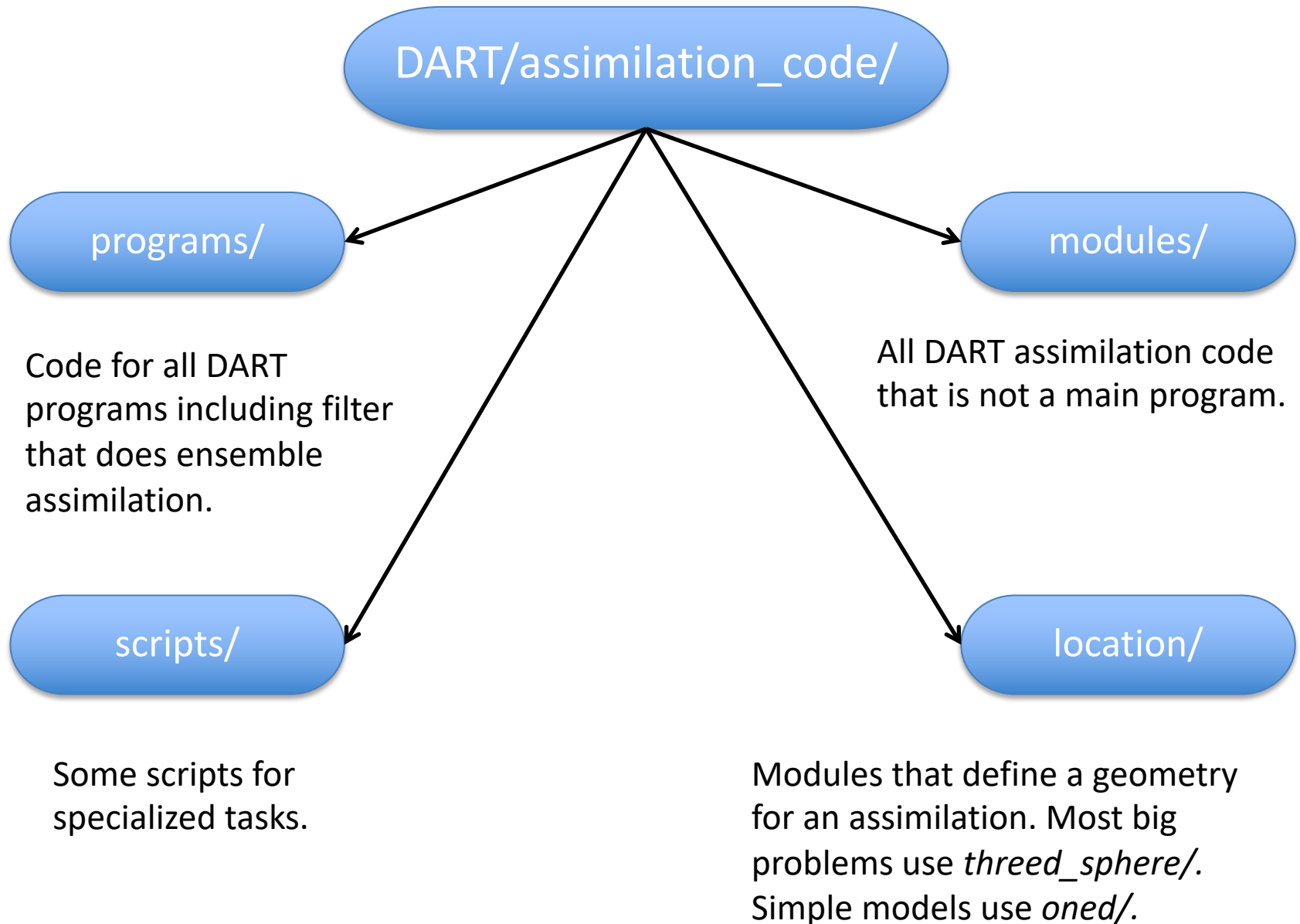
utilities/

Code and build tools for utilities that manipulate observation sequence files. Available for low-order models (oned/) and large models (threed_sphere/).

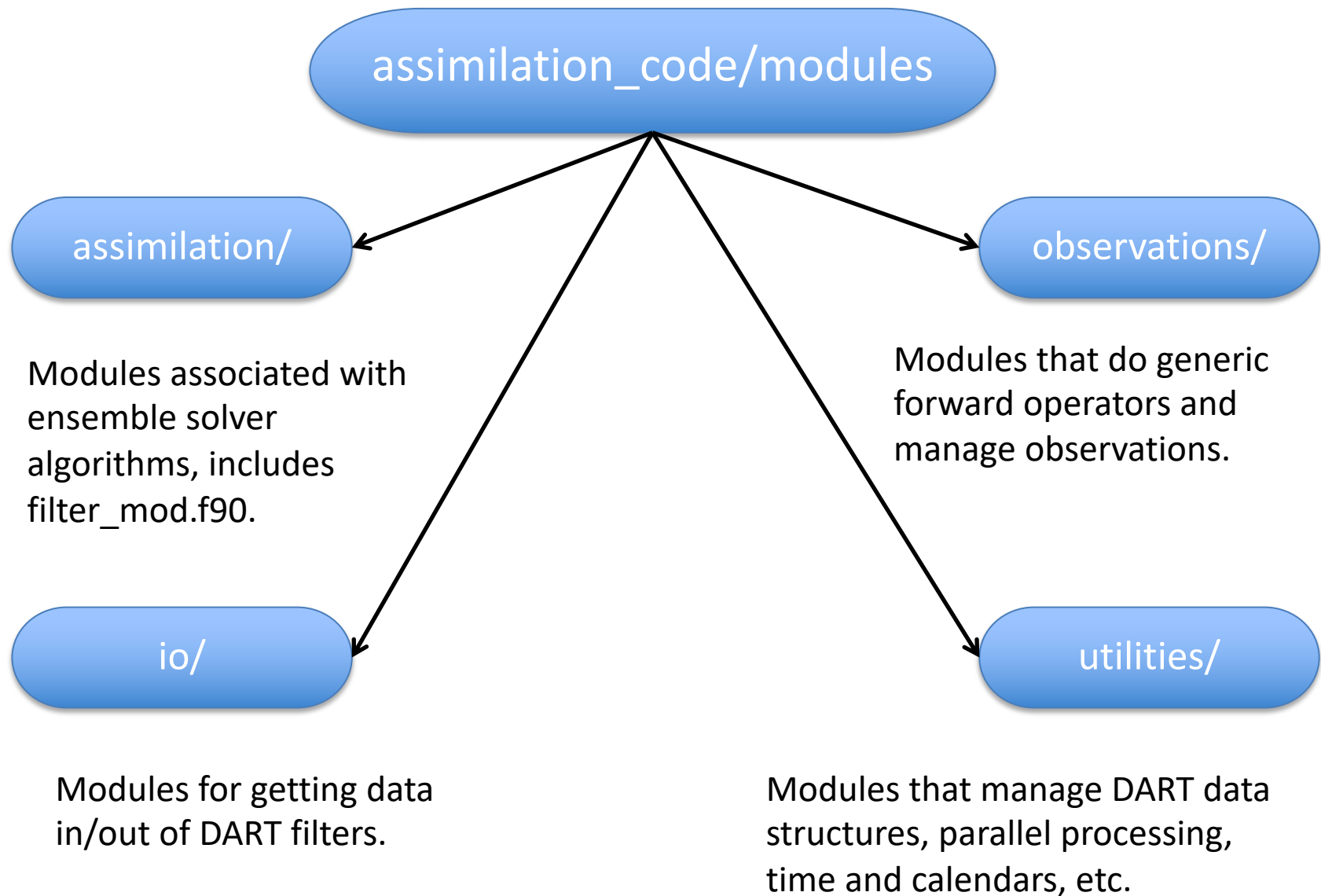
DART documentation directory:



DART assimilation_code/ directory:



DART assimilation_code/modules/ directory:



DART Tutorial Index to Sections

1. Filtering For a One Variable System
2. The DART Directory Tree
3. DART Runtime Control and Documentation
4. How should observations of a state variable impact an unobserved state variable?
Multivariate assimilation.
5. Comprehensive Filtering Theory: Non-Identity Observations and the Joint Phase Space
6. Other Updates for An Observed Variable
7. Some Additional Low-Order Models
8. Dealing with Sampling Error
9. More on Dealing with Error; Inflation
10. Regression and Nonlinear Effects
11. Creating DART Executables
12. Adaptive Inflation
13. Hierarchical Group Filters and Localization
14. Quality Control
15. DART Experiments: Control and Design
16. Diagnostic Output
17. Creating Observation Sequences
18. Lost in Phase Space: The Challenge of Not Knowing the Truth
19. DART-Compliant Models and Making Models Compliant
20. Model Parameter Estimation
21. Observation Types and Observing System Design
22. Parallel Algorithm Implementation
23. Location module design (not available)
24. Fixed lag smoother (not available)
25. A simple 1D advection model: Tracer Data Assimilation