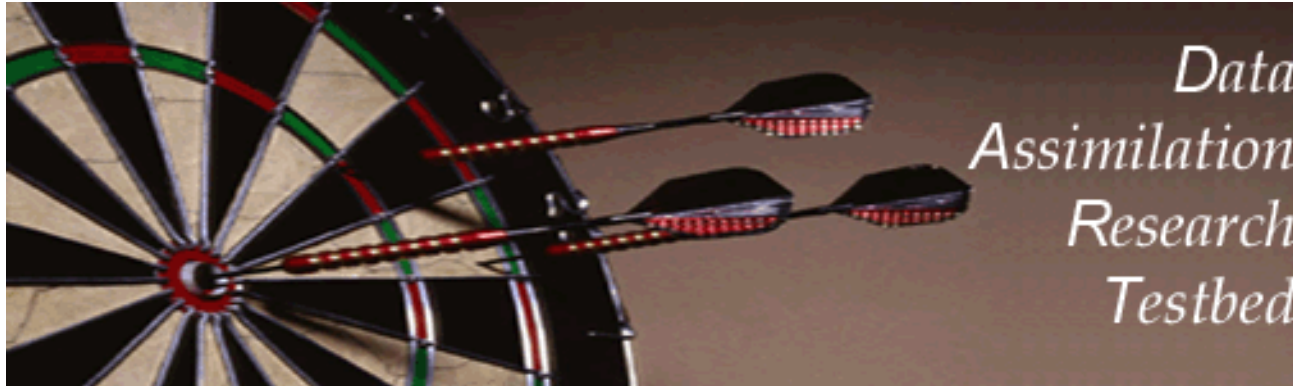


# Data Assimilation Research Testbed Tutorial

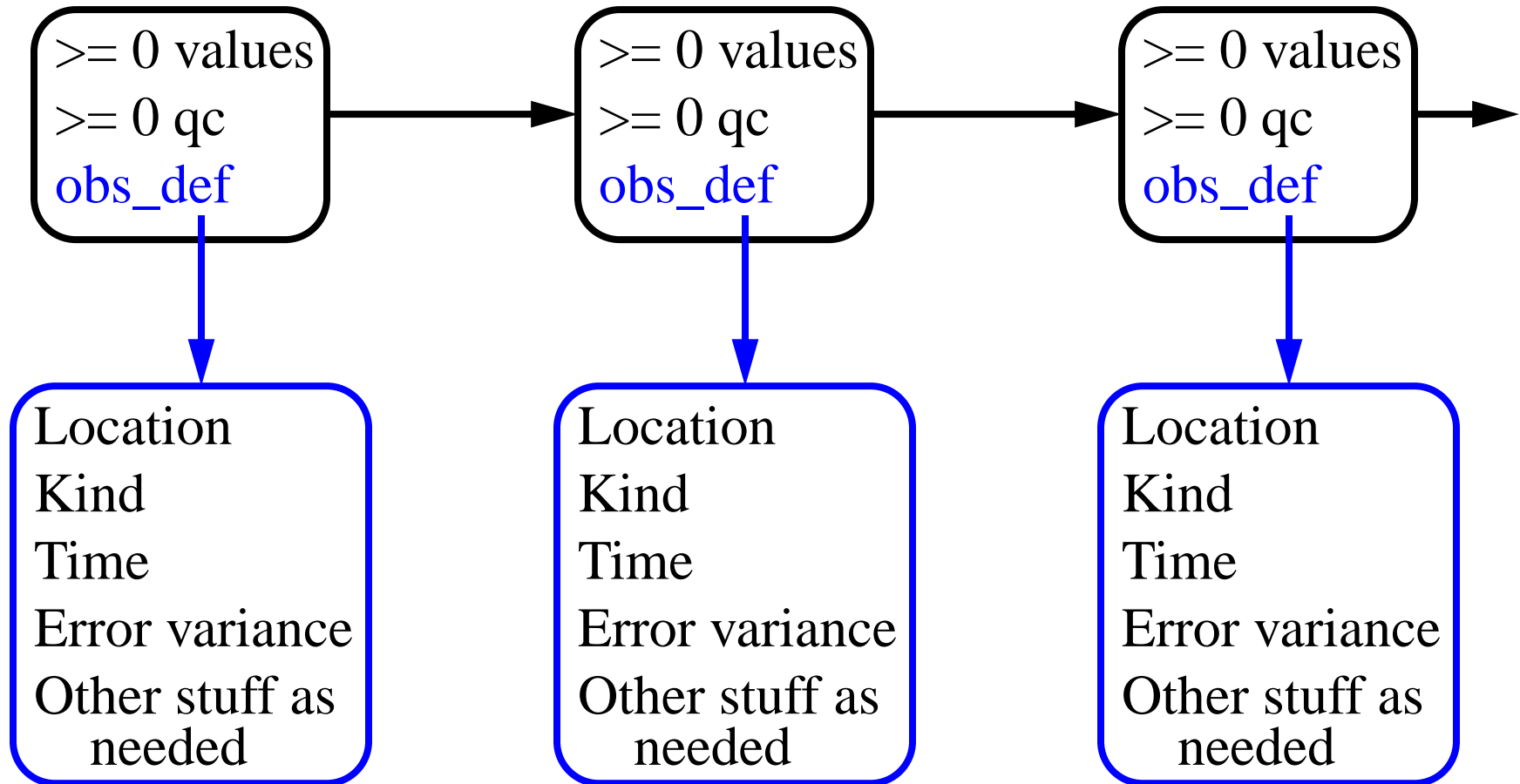


## Section 17: Creating Observation Sequences

Version 1.0: June, 2005

## Structure of an obs\_sequence file:

Sequence with non-decreasing times in definitions.



## Building Real observation sequences:

### 1. Interactive direct construction: program *create\_obs\_sequence*.

Queries for information for each observation in turn.

Enter kind, location, time, error variance, value, qc value(s).

Often convenient to create an input file.

Then redirect this file to standard input for *create\_obs\_sequence*.

### 2. Creating your own program.

The *obs\_sequence* module provides full set of interfaces to create.

Example: translation from NCEP BUFR file format.

Reads BUFR files, writes *obs\_sequence*.

## Creating Synthetic Observation Sequences (OSSEs):

Step 1: Create an observation sequence with no values.

A. Direct use of *create\_obs\_sequence*: no need to specify value for obs.

OR...

B. Synthetic observing network fixed in time:

1. First, use *create\_obs\_sequence* to specify observations in fixed network, all with time 0 days, 0 seconds.
2. Use *create\_fixed\_network\_seq* to specify times at which fixed network is observed.
3. Times can be regularly or irregularly spaced.

## Creating Synthetic Observation Sequences (OSSEs):

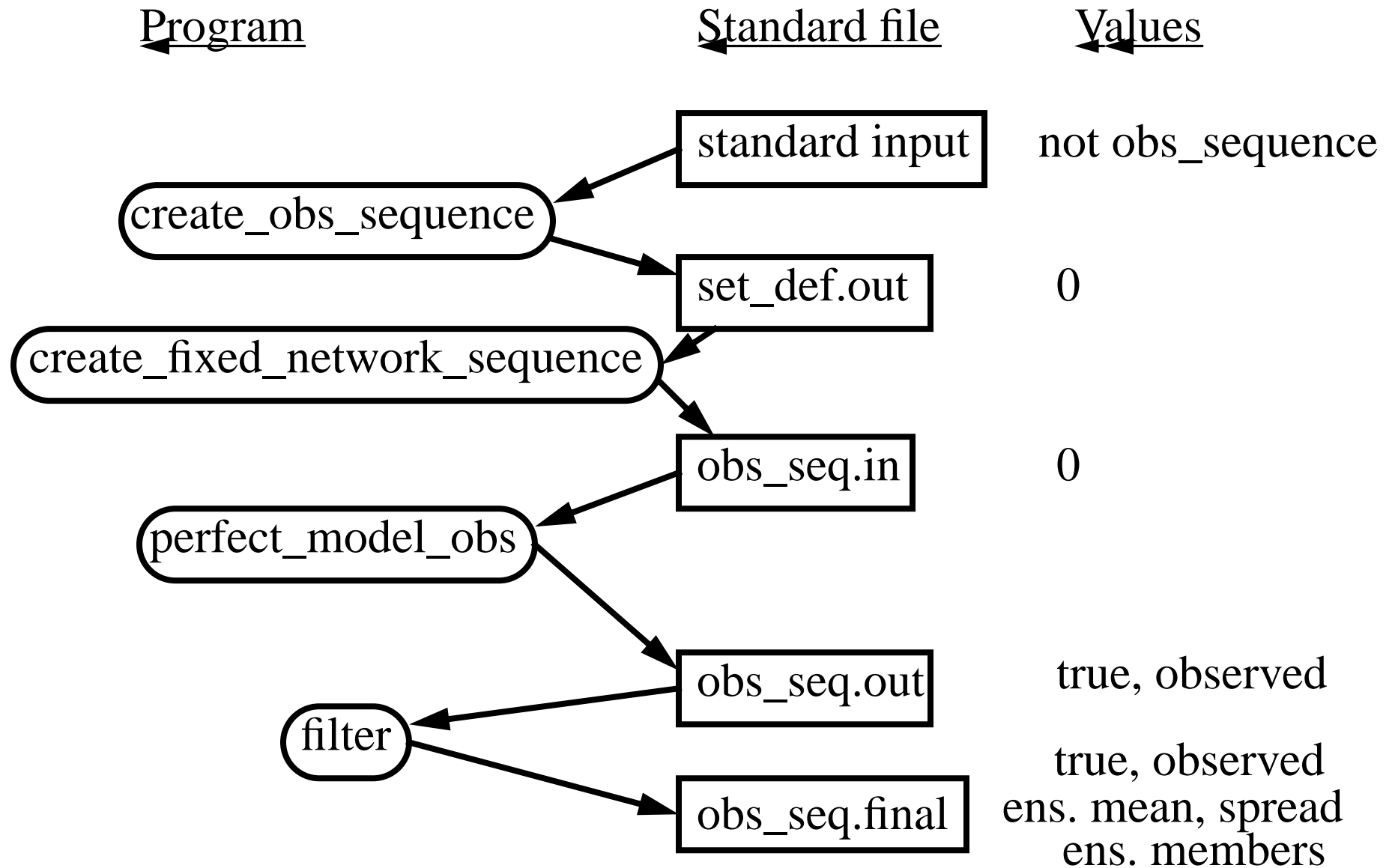
Step 2: Use perfect model\_obs to add observed values:

1. Integrates model.
2. Applies forward operators to get 'true' observed values.
3. Adds sample from observational error to get observed value.  
(Output obs\_sequence has 2 values for each observation).

Step 3: Run the filter:

Ensemble mean, spread, and individual ensemble members are added as values if requested (*filter\_nml*, see section 16).

## Creating Synthetic Observation Sequences (OSSEs):



## Selecting set of observation definitions:

Need to specify via name (character string) in namelist:

- Kind of all observations to be assimilated;

- Kind of all observations to be evaluated but not assimilated.

(Forward operators are computed and stored in `obs_sequence`).

List of available observation types found in *obs\_kind\_mod.f90*.

(see declaration for `obs_kind_info`).

Specify in *obs\_kind\_nml* using names:

- `&obs_kind_nml`

  - `assimilate_these_obs_types = 'RAW_STATE_VARIABLE'`

  - `evaluate_these_obs_types = 'RAW_STATE_1D_INTGRAL'/'`

## Selecting set of observation definitions:

Have to preprocess using program *preprocess*.

DEFAULT\_obs\_def\_mod.f90 and DEFAULT\_obs\_kind\_mod.f90 are merged with additional special observation definition files to create obs\_def\_mod.f90 and obs\_kind\_mod.f90

See section 21 for more details.