

Data model of OpenDA

Web meeting May 8, 2012

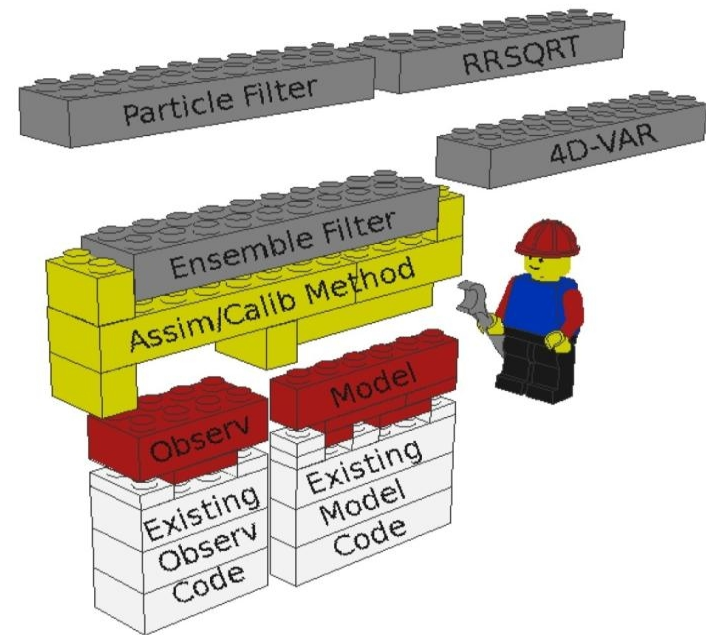
Nils van Velzen, Martin Verlaan & Umer Altaf

“Context” of OpenDA

- A generic toolbox for data-assimilation
 - set of interfaces that define interactions between components
 - library of data-assimilation algorithms
- OpenDA design goals:
 - Shared tools to reduce implementation costs
 - Shared knowledge between applications
 - Development of algorithms with eg universities
 - Easier to test, which should result in fewer bugs
 - Applications are configurable without recompiling
 - Portable to common platforms (windows, apple, linux)
 - “good” performance

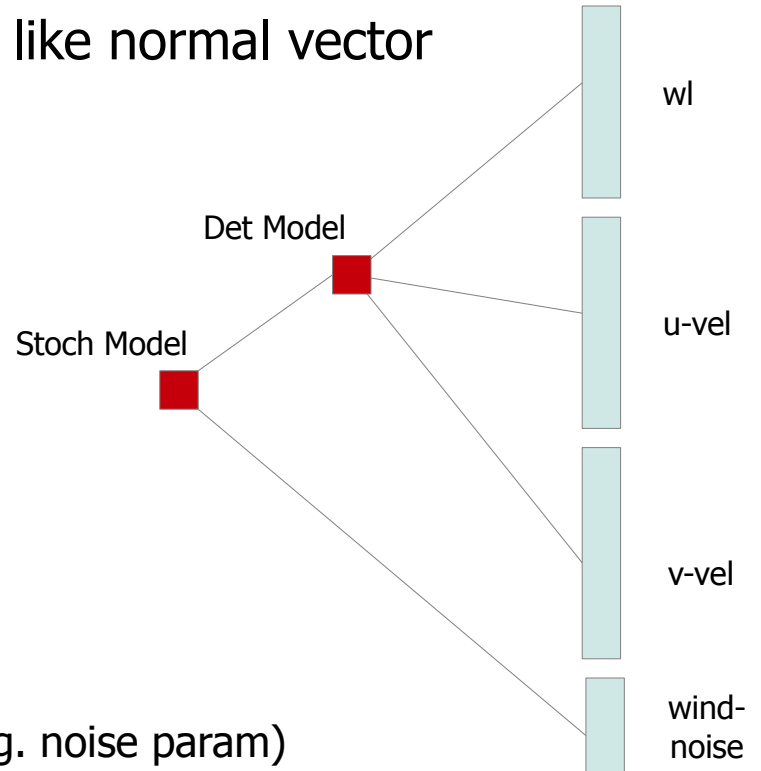
Background, component based system

- Object oriented concepts
 - No direct access to data like Fortran arrays
 - Functions for all kinds of operations
- Main objects
 - Model instance (X,U,P)
 - (Tree)Vector
 - Stochastic Observer
 - Data Assimilation or Calibration method



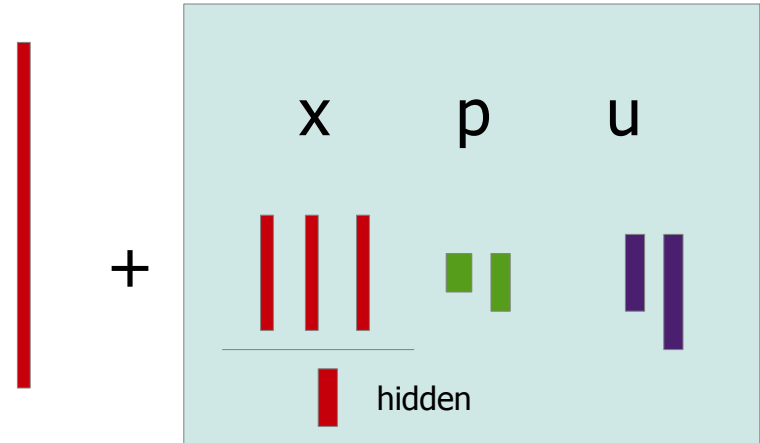
TreeVector

- TreeVector mathematically behaves like normal vector
 - Ordering/distribution of data:
 - nodes (sub-TreeVector)
 - Leafs contain the values
 - BLAS methods implemented e.g.
 - $y=y+\alpha x$
 - $\text{dot}=\langle y,y\rangle$
 - etc
 - Values:
 - In memory
 - On (NetCDF) file
 - Distributed (parallel computing)
 - Meta data (optional)
 - (simple) automatic interpolation (e.g. noise param)
 - Post processing

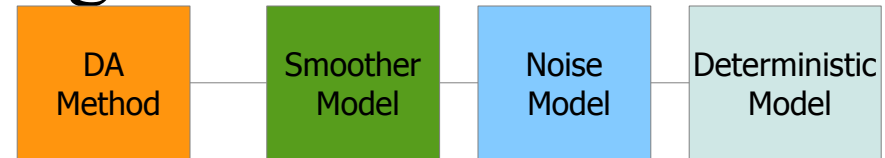


Model

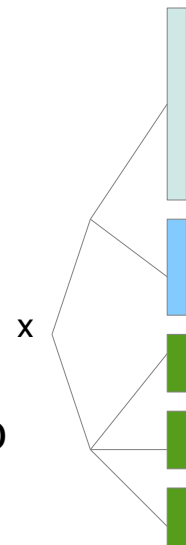
- Internal model state
 - State (x)
 - Forcings (u)
 - Parameters (p)
- Direct operations on internal state
 - `model.axpyState(alpha,dx)`
- Interpolation Hx is method of the model:
`Hx=Model.getObsvalues(observer)`
- Internal representation of x,u and p is up to the model programmer (java, Fortran, C, file).
- Note: formal model state (for restart etc) can contain (much) more data than the state used for filtering!



Why hiding and stacking the data?



- Modularity of model
 - Extend model with a noise model
 - Extending (stochastic) model with a smoother
 - Automatic parallelization of model steps
 - No difference between parallel and sequential model
- Hide parallel computing
- Combine components from different languages
- Goal:
 - Flexible model configuration and extension (without need to change the deterministic models)
 - Single implementation of DA method for all situations



Observations

- Stochastic Observer
 - Observations
 - Meta information (about, time, location quantity, interpol kernel)
 - Uncertainty
- A bit similar to a relational database
- Selections based on time interval, location etc are possible
- Values are available as (tree) vectors