Work Package 1: Harmonization

M. Umer Altaf, Nils Van Velzen, Martin Verlaan, A. W. Heemink

SANGOMA Annual Meeting: Leige, 26 – 28 November, 2012

Objectives

- Analyze the existing tools as a series of modules, some of which are common to several assimilation tools.
- Propose new modules for WP2 and WP4
- These modules together with the newly developed modules of WP4 and WP2 will then serve as the toolbox for designing new data assimilation systems
- Adoption of common standards and naming convention

Deliverables

- **Task 1.1** Identification of common tools.
  - List of commonly available tools (**Month 3)**
- **Task 1.2** Identification of new tools to be shared
  - List of tools to be adapted (**Month 6**)
- **Task 1.3** Specification of tool interface data model
  - Specification of data model (**Month 6**)
- **Task 1.4** Documentation of physical interfaces
  - Augmented list of common and new tools (**Month 12**)
  - Documentation of specifications (**Month 12**)

Task 1.1 Identification of common tools

- A shared document was prepared to gather information on:
  - Features of available data assimilation toolboxes.
  - The presently available modules.
  - Essential modules for SANGOMA
- Outcome (Deliverable 1.1)
  - Overview of the current systems with details of programming languages
  - Initial list of common components for SANGOMA

Example (compute_histogram)

- Description of functionality/purpose : Compute rank histogram of an ensemble about some state (e.g. ensemble mean or true state). Computation is done for a single location. It increments the information stored in a histogram array.

- Inputs: Ensemble values for a single grid point. Single state entry. Size of ensemble. Histogram array (size ensemble size+1. It has to be initialized to zero before the first call).

- Outputs: Histogram array

- Required work: The module needs to be adapted to be generally usable. The input/output is currently different.

- Language: F95

- Needs: no libraries

- Host: AWI

Task 1.2 Identification of new tools

- A similar document was circulated to gather information on:
  - New modules for SANGOMA
  - Who will develop new modules.

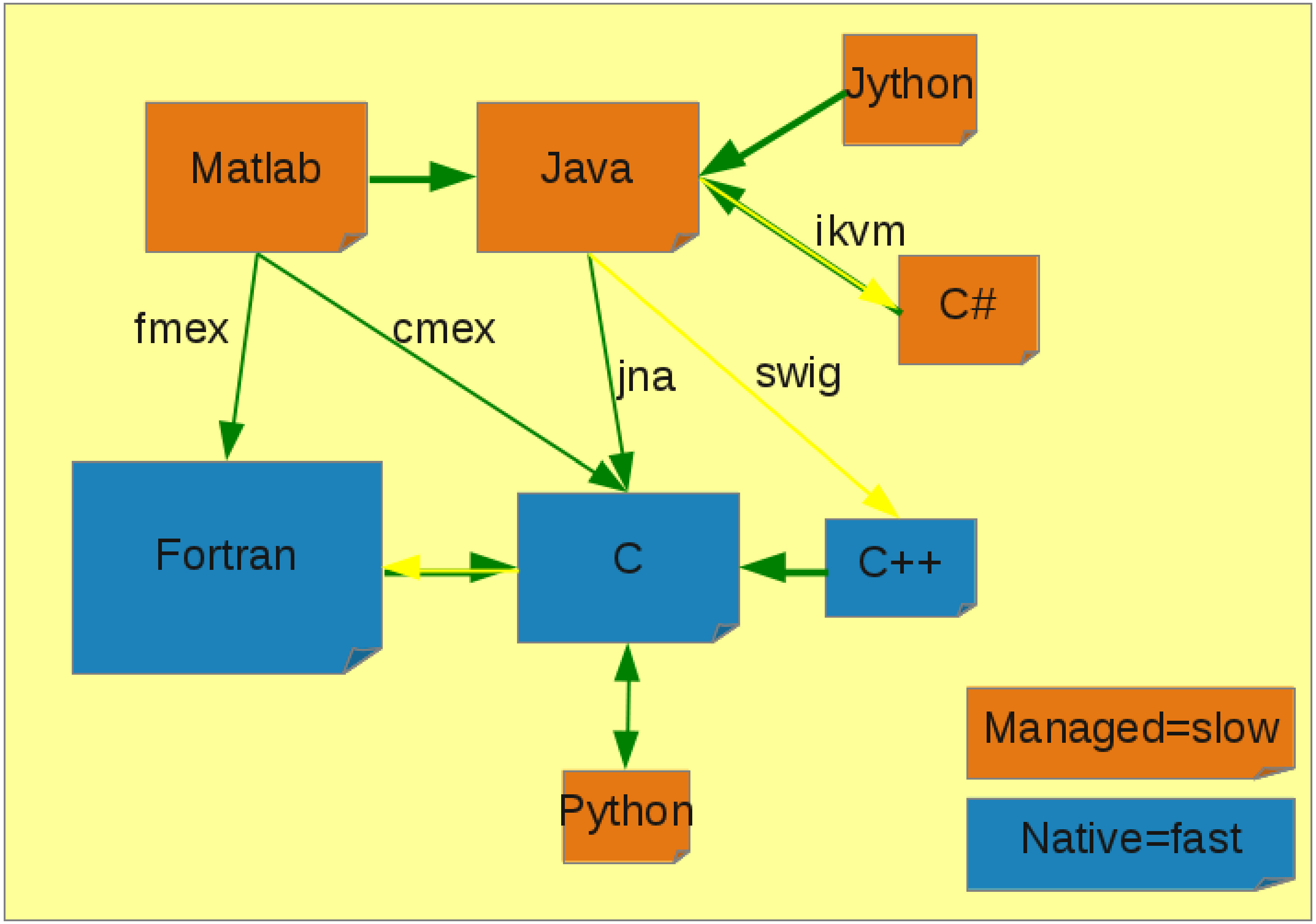
- Outcome (Deliverable 1.2)
  - An updated list of common and new components is prepared.

Task 1.3 Specification of tool interface data model

- Overview of the presently available data models:
  - Basic dimesions and Arrays
  - Observation operator.
  - Programming concepts.

- Design a data model that is consistent with
  - Data models of the existing systems
  - MyOcean data models

Programming Languages (Overview)



Overview of the existing systems

- PDAF

- In-memory Interface
- Dimensions (size of state vector, observation vector, ensemble)
- Common Arrays (state vector, observation vector, ensemble matrix)
- The data assimilation process through set of routines

I. Data transfer (get_state, put_state)

II. Model forecast (next_observation)

III. Observation handling (init_dim_obs, init_obs, obs_ob)

IV. Ensemble method (prodRinvA, add_obs_noise etc)

V. Prepoststep to compute ensemble mean variance before and after analysis

Overview of the existing systems

- OpenDA

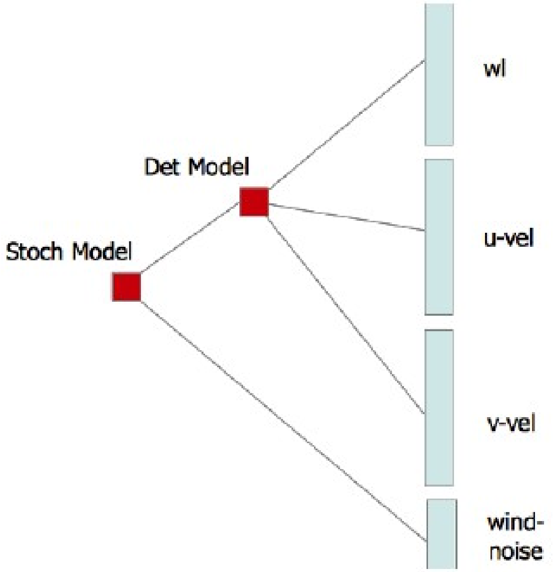
  - Input-output files interface
  - Object oriented features (no direct access to data)
  - Functions are used to do all operations (e.g., getstate, setstate etc)
  - Main Objects

I. Treevector

II. StochasticModel

III. StochasticObserver

IV. Data assimilation / calibration method.



More SANGOMA Data Models

- Ocean Assimilation Kit (OAK)

  The data model bases on Fortran arrays.

  derived types for data structure in the model and the memory layout.

- Beluga / Sequoia

  Data model contains state vector and grid, derived types for observations, sampled libararies

- SESAM

  Provides a black-box coupling to a model through NetCDF files.

- NERSC EnKF

  Derived tyes for observations. Basic types in the analysis step.

MyOcean Data Models

- MyOcean uses different Ocean models
  - TOPAZ system (HYCOM ocean Model)
  - NEMO Model.

- NEMO Model
  - Benchmark for SANGOMA
  - Uses CF-complaint NetCDF files
  - Basic datatypes (internally)
  - Derived datatypes (diagnostic purposes, input forcings)

SANGOMA Data Model

- Initially specified as logical data model
- Programming languages (Fortran, Matlab/Octave)
- Implementation should also be possible in C, Java
- Data model was discussed in virtual meeting on May 8, 2012.
- Two main interfaces are finalized:
  - File Interface
  - In-memory Interface

SANGOMA Data Model

- File Interface

I. The NetCDF files will be used to connect models to DA tools

II. SANGOMA tools should work for CF-complaint input files

III. Model output standards

IV. Tools should be compiled with version 3 and 4

V. Separate NetCDF files for each ensemble member

SANGOMA Data Model

- In-memory Interface

I. Basic data structures (no complex data types)

II. Compatibility of Fortran based codes with C, Matlab, Java

III. No derived datatypes

IV. Basic dimensions as in PDAF (statevector, ensemble, observations)

V. Call-back functions to make use of complex data structures

Task 1.4 Documentation of physical interfaces

- Updated the list of common and new modules
- A template of detailed specification of inputs and outputs of one SANGOMA module is prepared
- This template document was shared within SANGOMA to gather information on each listed modules
- Outcome (Deliverable 1.4 and 1.5)
  - An updated list of modules for SANGOMA
  - A detailed specification of these modules (Living document)

Example of Specification: POD

1. Function Name: **Compute_POD**

- **Operation:** This function reads an initial ensemble (***nens*** ensemble members) from Netcdef files of model states and store them to an array of vectors ***(nstate, nens)*** of size nstate. Then the eigenvalue problem is solved using this array of ensemble. An example of such a Netcdef file is already given in data model description. Before solving the eigenvalue problem the individual vectors may be normalized. The outputs of the function are truncated eigenvalues and eigenvectors.

- **Inputs:** Netcdef files containing model states (see section 2.2)

- **Output:** A data file containing truncated eigenvectors and eigenvalues. The elements of output file are:

P(nstate,nvals)	truncated eigenvectors
Evals(nvals)	vector containing eigenvalues

Example of Specification: POD

```
program compute_POD
        read ensemble(nens)   // Assumes ensemble already available
        do i = 1 to nens
                read ensemble(i)
                set X(ntate,i) = ensemble(i)
        end do
        Normalize X
        Compute X'X
        Compute EVD(X'X)
        Return P(nstate,nvals)
        Return Evals(nvals)
end compute_POD
```

Summary

- Listed the modules to be shared for SANGOMA
- Logical data model design
- A detailed specification of shared modules is formulated
- We can now proceed to the implementation of new methods as well as modification in existing ones (WP2 and WP3).