ESA Atmospheric Toolbox (BEAT)

Support for Aeolus

Sander Niemeijer (S&T)
BEAT started as Basic Envisat/ERS Atmospheric Toolbox (2002)

Now: ESA Atmospheric Toolbox as one of the Sentinel toolboxes

Goal: to provide the necessary software to support scientific analysis of atmospheric data.

Three layers of functionality:

- **CODA** direct product reading interface (BEAT-I)
- **HARP** data harmonization interface (BEAT-II)
- **VISAN** visualization & analysis application

*All Open Source (BSD/GPL) and Cross Platform!*
CODA

• Common Data Access reading interface for earth observation data products
• Single interface to read any data from a product
• Direct mapping to the data in a product using hierarchy of 'records', 'arrays', and 'basic types'.
• Uses 'codadef' files to describe data format
• Mandatory for raw binary and structured ascii data
• Not needed for self-describing formats (e.g. HDF, netCDF, ...)
• Recommended for XML data (comparable to XML Schema)
ADM-Aeolus CAL/VAL Rehearsal Workshop

CODA C library

- XML backend
- netCDF backend
- CDF backend
- GRIB backend
- HDF4 backend
- HDF5 backend
- RINEX backend
- SP3 backend

- Mission #1 codadef
- Mission #2 codadef
- Mission #3 codadef

- HDF4 library
- HDF5 library

- Fortran interface
- Python interface
- Java interface
- IDL interface
- MATLAB interface

- codadd
- codacheck
- codaeval
- codadump
- codafind
- codacmp
CODA

```python
>>> product = coda.open("AE_OPER_ALD_U_N_1B_20151002T001857059_005787000_046339_0001.DBL")

>>> latitude = coda.fetch(product, 'geolocation', -1,
   'observation_geolocation/observation_rayleigh_geolocation', -1, 'latitude_of_height_bin')
>>> latitude = vstack(latitude)

>>> longitude = coda.fetch(product, 'geolocation', -1,
   'observation_geolocation/observation_rayleigh_geolocation', -1, 'longitude_of_height_bin')
>>> longitude = vstack(longitude)

>>> altitude = coda.fetch(product, 'geolocation', -1,
   'observation_geolocation/observation_rayleigh_geolocation', -1, 'altitude_of_height_bin')
>>> altitude = vstack(altitude)

>>> wind_velocity = coda.fetch(product, 'wind_velocity', -1,
   'observation_wind_profile/rayleigh_altitude_bin_wind_info', -1, 'wind_velocity')
>>> wind_velocity = vstack(wind_velocity)

>>> coda.close(product)
```
CODA

- Need to know format of products
  
  http://www.stcorp.nl/coda/codadef/AEOLUS/index.html
  
  (or use VISAN Product Browser)
HARP

Toolset for Harmonization of Products:

• Format
• Dimensions
• Names
• Units
• Time
• Location
• Resolution

Focuses on (but not limited to) atmospheric data products
HARP

- Import of data products to common data format
- Introducing HARP data format standard
- On top of netCDF, HDF4, HDF5
- Focus is not on being descriptive, but on being able to standardize automated operations on data
- Main aspect is variable names and dimension names/ordering
HARP

Standardized processing components:

- Filtering
- Unit conversion
- Derived variables
- Collocation (find matching pairs for time/location/...)
- Vertical regridding/smoothing/integration
- Horizontal regridding
- Uncertainty/error propagation
HARP C library

Ingestion modules:
ADM-Aeolus, Sentinel 5P, OMI/TES/MLS/HIRDLS,
GOME-2/IASI, GOMOS/MIPAS/SCIAMACHY, GOSAT,
Odin/OSIRIS, Atmospheric CCI, CAMS, NDACC

Unit conversion
Derived variables
Collocation
Filtering
Regridding
Import/Export

CODA C library

HDF4 library
HDF5 library

Matlab interface
Python interface

harpconvert harpdump harpcollocate other...
harpcfilter harpmerge harpcheck other...
other...
HARP

>>> product =
harp.ingest_product("AE_OPER_ALD_U_N_1B_20151001T001124059_005583000_046323_0001.DBL")
>>> print product

source product = 'AE_OPER_ALD_U_N_1B_20151001T001124059_005583000_046323_0001.DBL'

double datetime_start {time=466} [seconds since 2000-01-01]
double datetime_length {time=466} [s]
double latitude {time=466, vertical=24} [degree_north]
double longitude {time=466, vertical=24} [degree_east]
double altitude_bounds {time=466, vertical=24, 2} [m]
double hlos_wind_velocity {time=466, vertical=24} [m/s]
long hlos_wind_velocity_validity {time=466, vertical=24}
long index {time=466}
HARP

>>> print product.hlos_wind_velocity

  type = double
  dimension = {time=466, vertical=24}
  unit = 'm/s'
  valid_min = -inf
  valid_max = inf
  description = 'HLOS wind velocity at the altitude bin'

  data =

  [[ 9.72355716e+00 -1.61382676e+00 5.37019472e+00 ..., 2.52087225e+00
    1.43891364e+01 6.99138039e+00]
   [ 1.09220012e+01 2.11933840e+00 9.71404005e+00 ..., 9.59876202e+00
    1.53916451e+01 6.12803425e+00]
   [ 3.32102420e+03 -3.10725919e+02 -6.66207524e+01 ..., 1.29843710e+01
    4.66490577e+00 1.17011710e+01]
   ...,
HARP

>>> product =
harp.ingest_product("AE_OPER_ALD_U_N_1B_20151001T001124059_005583000_046323_0001.DBL",
'derive(altitude {time,vertical} [km]);derive(latitude {time});derive(longitude {time})',
'data=rayleigh_measurement')

>>> print product
source product = 'AE_OPER_ALD_U_N_1B_20151001T001124059_005583000_046323_0001.DBL'

double datetime_start {time=13980} [seconds since 2000-01-01]
double datetime_length {time=13980} [s]
double altitude_bounds {time=13980, vertical=24, 2} [m]
double hlos_wind_velocity {time=13980, vertical=24} [m/s]
long hlos_wind_velocity_validity {time=13980, vertical=24}
long index {time=13980}
double altitude {time=13980, vertical=24} [km]
double latitude {time=13980} [degree_north]
double longitude {time=13980} [degree_east]
VISAN

• Ingest data using CODA and HARP
• Python language for command input and performing calculations and manipulations
• With one function call, create interactive 2D and World plot visualizations of your data
• Simple Product Browser
• Open Source and Cross-Platform: Windows, Linux, and Mac OS X.
Welcome to the VISAN/Python Control Shell.

VISAN 3.14 (Python 2.7.11, NumPy 1.11.0, CODA 2.18.1, HARP 0.5, BEAT 6.10.2)

>>> product = harp.ingest_product("/data/AE_OPER_ALD_U_N_1B_20151001T001124059_005583000_046323_0001/AE_OPER_ALD_U_N_1B_20151001T001124059_005583000_046323_0001.DBL", 'derive(altitude {time,vertical} [km]);derive(latitude {time});derive(longitude {time})', 'data=rayleigh_measurement')

>>> print product
source product = 'AE_OPER_ALD_U_N_1B_20151001T001124059_005583000_046323_0001.DBL'

double datetime_start {time=13980} [seconds since 2000-01-01]
double datetime_length {time=13980} [s]
double altitude_bounds {time=13980, vertical=24, 2} [m]
double hlos_wind_velocity {time=13980, vertical=24} [m/s]
long hlos_wind_velocity_validity {time=13980, vertical=24}
long index {time=13980}
double altitude {time=13980, vertical=24} [km]
double latitude {time=13980} [degree_north]
double longitude {time=13980} [degree_east]
ESA Atmospheric Toolbox (BEAT)

Consists of separate software packages:

- **CODA**
- **codadef**
- **HARP** (includes codadefs; does not include CODA)
- **VISAN** (includes CODA python + HARP python + codadefs)

'BEAT package' will become deprecated

Main BEAT website: http://www.stcorp.nl/beat
HARP: http://github.com/stcorp/harp/