acve

ATMOSPHERIC COMPOSITION VALIDATION AND EVOLUTION

18–20 October 2016 | ESA–ESRIN | Frascati (Rome) Italy
HARP

Data harmonization toolset for scientific earth observation data

Sander Niemeijer S[T]
HARP – Introduction

• One of the main components of the **ESA Atmospheric Toolbox** (BEAT) (other components are CODA and VISAN)

• Successor of BEAT layer 2 (beatl2 on steroids)
• Built on heritage from the ESA GECA and FP7 NORS projects

• Goals:
  • Convert data into easy-to-use data structures
  • Provide common operations to harmonize data
  • Open interfaces
  • Open Source and Cross Platform
HARP – data ingestion

HARP product ingestion similar to beatl2 ingest() but much improved (addition of attributes, named dimensions, etc.)

```python
>>> product = harp.ingest_product(......)
>>> print product
source product = 'MIP_NL__2PODPA20020925_010931_000060262009_00418_02977_0147.N1'

double datetime {time=74} [seconds since 2000-01-01]
double altitude {time=74, vertical=17} [km]
double latitude {time=74} [degree_north]
double longitude {time=74} [degree_east]
double pressure {time=74, vertical=17} [hPa]
double pressure_uncertainty {time=74, vertical=17} [hPa]
double temperature {time=74, vertical=17} [K]
double temperature_uncertainty {time=74, vertical=17} [K]
double O3_number_density {time=74, vertical=17} [molec/cm^3]
double O3_number_density_uncertainty {time=74, vertical=17} [molec/cm^3]
double O3_volume_mixing_ratio {time=74, vertical=17} [ppmv]
double O3_volume_mixing_ratio_uncertainty {time=74, vertical=17} [ppmv]
long index {time=74}
```
HARP – operations

Ability to perform operations during ingestion.
Provide sequence of operations as a single string parameter (e.g. script).

```python
>>> harp.ingest_product("filename", "operations")
```

Supported operations:

- filtering
  - which variables to include/exclude
  - which elements in a dimension to include/exclude
- variable conversions
- regridding
- smoothing
- ...

HARP – unit conversions

Your data contains O3 total column density in molec/cm², but you want DU?

"derive(O3_column_number_density {time} [DU])"

It will take as input:

O3_column_number_density {time} [molec/cm²]

And replace this with:

O3_column_number_density {time} [DU]
HARP – variable conversions

What if you have O3 volume mixing ratio and total air number density, but you want O3 number densities? No problem:

\[
\text{derive(O3\_number\_density } \{\text{time, vertical}\} \ [\text{molec/cm3}\})
\]

It will calculate:

\[
\begin{align*}
\text{O3\_number\_density } \{\text{time, vertical}\} \ [\text{molec/cm3}] & \text{ from } \\
\text{number\_density } \{\text{time, vertical}\} \ [\text{molec/cm3}] & \\
\text{O3\_volume\_mixing\_ratio } \{\text{time, vertical}\} \ [\text{ppmv}] & 
\end{align*}
\]

Also, if the air number density or O3 vmr are in a different unit, HARP will automatically convert them before using them as input to the algorithm.
HARP – variable conversions

What if you don't have the total air number density, but you do have pressure and temperature profiles?

Again, no problem, HARP tries to recursively try to derive the needed inputs:

$$\text{O}_3\text{-number\_density}\ \{\text{time,vertical}\} \ [\text{molec/cm}^3] \ \text{from}$$
$$\text{number\_density}\ \{\text{time,vertical}\} \ [\text{molec/cm}^3] \ \text{from}$$
$$\text{temperature}\ \{\text{time,vertical}\} \ [\text{K}]$$
$$\text{pressure}\ \{\text{time,vertical}\} \ [\text{hPa}]$$
$$\text{O}_3\text{-volume\_mixing\_ratio}\ \{\text{time,vertical}\} \ [\text{ppmv}]$$
HARP – vertical column integration

You want to generate the total column from O3 volume mixing ratio profiles? Again, no problem:

\[
\text{O3}\_\text{column\_number\_density} \{\text{time}\} \ [\text{molec/cm}^2] \ \text{from} \\
\text{O3}\_\text{column\_number\_density} \{\text{time, vertical}\} \ [\text{molec/cm}^2] \ \text{from} \\
\text{O3}\_\text{number\_density} \{\text{time,vertical}\} \ [\text{molec/cm}^3] \ \text{from} \\
\text{number\_density} \{\text{time,vertical}\} \ [\text{molec/cm}^3] \ \text{from} \\
\text{temperature} \{\text{time,vertical}\} \ [\text{K}] \\
\text{pressure} \{\text{time,vertical}\} \ [\text{hPa}] \\
\text{O3}\_\text{volume\_mixing\_ratio} \{\text{time,vertical}\} \ [\text{ppmv}] \\
\text{altitude\_bounds} \{\text{time, vertical, 2}\} \ [\text{km}] \ \text{from} \\
\text{altitude} \{\text{time, vertical}\} \ [\text{km}] \ \text{from} \\
\text{altitude} \{\text{vertical}\} \ [\text{km}] 
\]
HARP – Operations

Filters:

- latitude <= 10; latitude >= -10; longitude-range(179,-179)
- exclude(O3_number_density)
- keep(latitude, longitude)

Derived variables:

- derive(<variable> <dimensions> <unit>)

And many more...

You can combine these into small `scripts`:

- derive(solar_zenith_angle {time}); solar_zenith_angle < 80; exclude(solar_zenith_angle)
HARP – Conventions

So how can HARP do these variable conversions?
It requires strict conventions on input variables.

- **Convention for variable names**
  Not targeted to be fully descriptive.
  Only use a different name if variables need to be processed differently.
  (minimize number of different names)

- **Convention for dimensions**
  Fixed ordering; fixed naming (time, vertical, latitude, longitude, spectral)

- **Convention for units**
  Just use udunits2
HARP – HARP Data Format

So what if we apply these conventions to a netcdf, HDF4, or HDF5 file?

It gives us a data format standard!

Data formats

HARP data format

The HARP data format is a set of conventions for files stored in netCDF-3, HDF4, or HDF5 format. For performance and sizing reasons it is recommended to use netCDF-3. The netCDF-4 format actually uses HDF5 underneath and is therefore supported as an HDF5 file by HARP.

The HARP conventions can generally be combined with other conventions that do not contradict it. For instance, a file can be compliant with both netCDF-CF and the HARP data format conventions at the same time.

In order for a product to be compliant with the HARP data format conventions it will need to have the HARP-1.0 value in the Conventions global attribute and follow the restrictions described in this section.
HARP – HARP Data Format

If you create your own files conforming to the HARP format you get all features of HARP for free! You can also create your own 'building blocks'.

The result from a HARP ingest can be exported/imported in HARP format using either netcdf, HDF4, or HDF5.
Currently already ingestion support for many satellite products:

   GOME, SCIAMACHY, MIPAS, GOMOS, GOME-2, IASI,
   OMI, TES, MLS, HIRDLS, GOSAT, ...

But also support for:

- Sentinel-5P
- Atmospheric CCI
- ECMWF GRIB (CAMS model data)
- GEOMS (NDACC ground based data)

And more to come...
HARP - Components

Developed around a single C library.

Command line tools:
- harpconvert
- harpfilter
- harpdump
- harpccheck
- harpcollocate
- harpprofile

Interfaces:
- Python
- MATLAB/IDL (in development)
HARP - Support in VISAN

Support for HARP in VISAN is similar to beatl2.
- Ingest files from their native format.
- Import/export to HARP format.
- Single command to create 2D or worldmap plots.

Now also a dialog for easy HARP ingestion/import:
HARP – Current Status

HARP is currently still in beta (version 0.4), but already usable. Already available:

- command line tools, C library and python interface
- ingestion support for many data products
- support for many variable conversions
- vertical regridding and smoothing

Main things still to do (early 2017):

- ingestion support for additional data products
- MATLAB/IDL interface
- horizontal (lat/lon) regridding

Future plan: uncertainty propagation
HARP – More information

Available on github:

https://github.com/stcorp/harp

There is also extensive documentation in HTML (including description of mapping of all product ingestions and description of algorithms (ATBD)).

HARP is used as the core toolset for the automated validation servers in S5P-MPC, CAMS-84, and QA4ECV.

Please provide us your feedback or get involved:

https://github.com/stcorp/harp/issues

harp@stcorp.nl
HARP – Demonstration

Come and see me during the poster session for a demonstration!