The EUMETSAT
Network of
Satellite Application
Facilities

NWC SAF
Support to Nowcasting and
Very Short Range Forecasting

SMHI

User Workshop, April 2010, Madrid

PPS Software Development Status and Plans

Jakob Malm

Outline

Status

- Statistics
- Satellite data
- Auxiliary data
- Libraries
- News

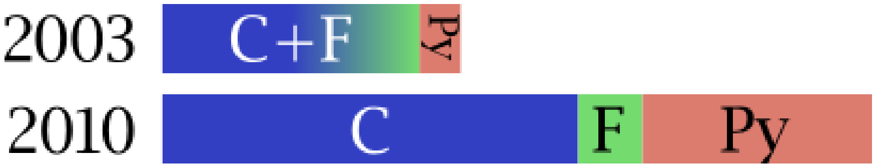
Plans

- Information model common for LEO and GEO
- Satellites and algorithms
- Processing on swath only
- More automated validation
- Installation and configuration
- Portability
- Input and feedback from users needed!

Goal: Raise your interest in the future development of PPS

Status: Statistics

- Code
  - Beta release: C+Fortran: ~75 000 lines, Python: ~12 000 lines
  - C: 121 344 lines
  - Fortran: 18 773 lines
  - Python: 66 993 lines



- Users
  - 6 known to run PPS operationally, 6 more running / setting up
  - How many unknown?

Status: Satellite data

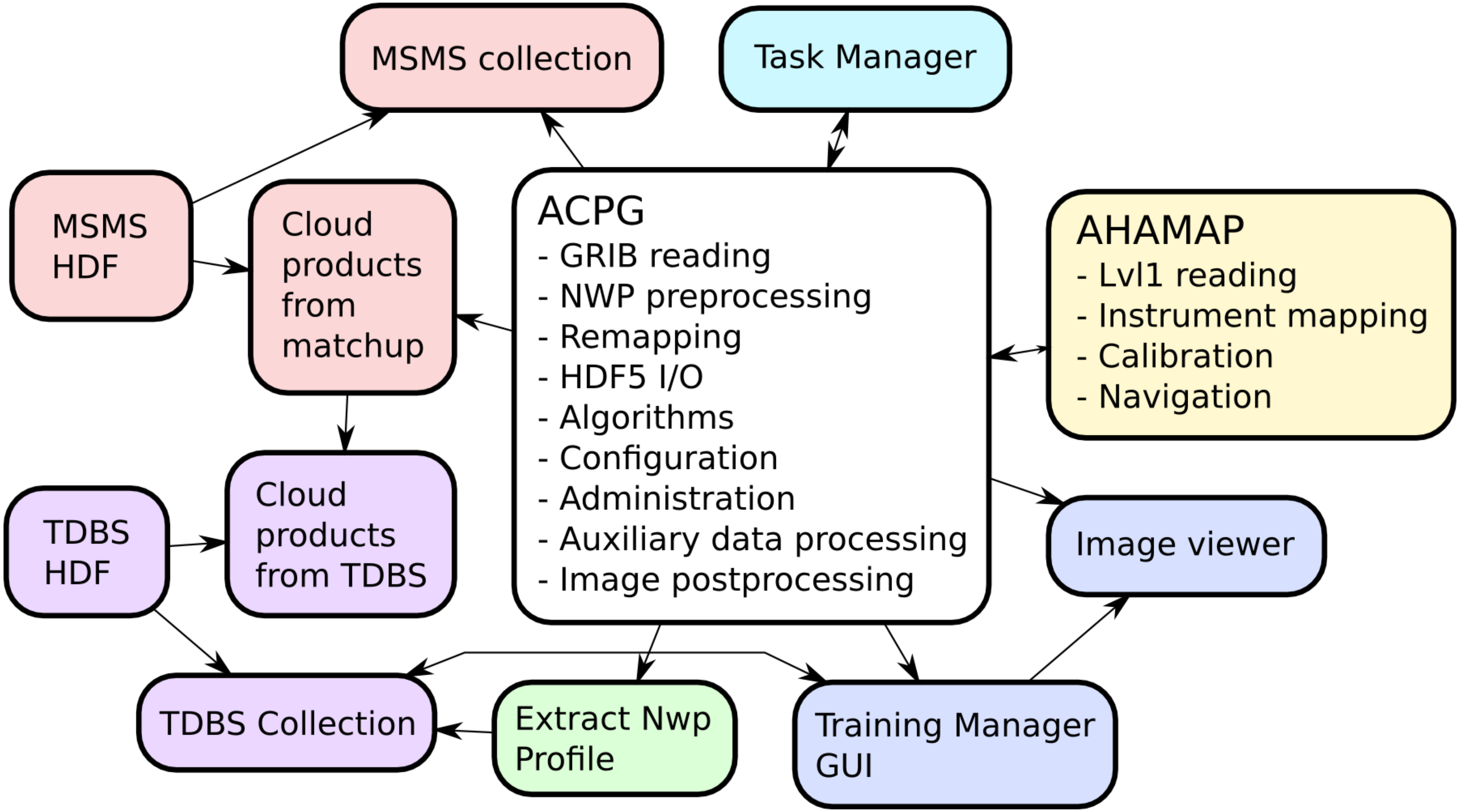
- NOAA (AVHRR, AMSU-A, AMSU-B/MHS)
- Metop (AVHRR, AMSU-A, MHS, IASI)
- Terra, Aqua (MODIS) for prototyping NPP

- Local
- Global Metop
- GAC (Global Area Coverage, NOAA)
    - State-of-the-art intercalibration (v2010)
    - NOAA7 – NOAA19

Status: Auxiliary Data

- NWP
  - GRIB
  - ECMWF, HIRLAM, any NWP model in regular lon/lat grid (pressure levels)
- Variable land emissivity (v2010)
- Land use (USGS)
  - Fraction of land (coasts)
- Topography (USGS)
- Sea ice (OSISAF)

Status: Different parts of PPS



MSMS collection

Task Manager

MSMS HDF

Cloud products from matchup

ACPG
- GRIB reading
- NWP preprocessing
- Remapping
- HDF5 I/O
- Algorithms
- Configuration
- Administration
- Auxiliary data processing
- Image postprocessing

AHAMAP
- Lvl1 reading
- Instrument mapping
- Calibration
- Navigation

TDBS HDF

Cloud products from TDBS

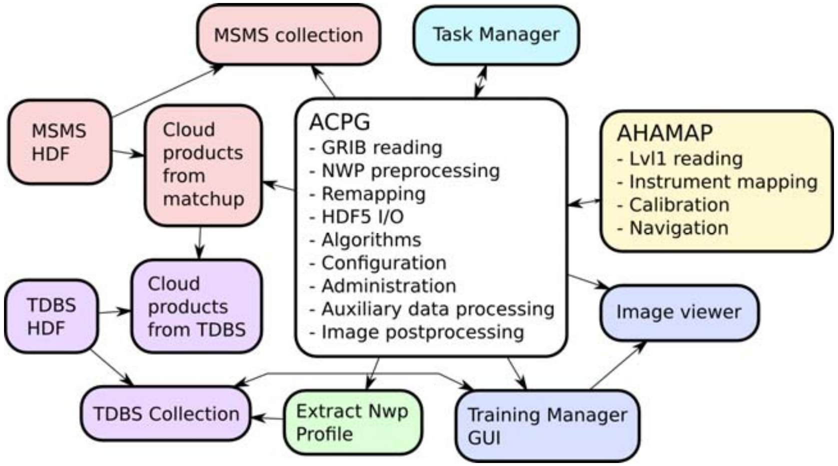
Image viewer

TDBS Collection

Extract Nwp Profile

Training Manager GUI

Status: Internal Libraries



- Common functions
    - HDF5 files
        - Read/write PPS information model
        - HDF5 1.8 (v2010)
    - Read auxilliary data
- GRIB files
    - Read any field
    - PPS uses: Temperature, Column integrated water vapour, OSISAF sea ice
    - NWP postprocessing: model levels to pressure levels, integrate water vapour, tropopause, …
    - GRIB API (v2010) replaces EMOS => seamless use of GRIB edition 2 files
        - Need test data (TIGGE data works)
- Remapping / projection

Use externally? Some insight needed…

We want to make this easier!

OSISAF
MESAN
CM-SAF

LandSAF?
Others?

News 2010

- Variable land emissivity
- Numeric replaced by NumPy, ScientificPython removed
- HDF5 1.8, HLHDF 0.79
- RTTOV9 (CTTH, dynamic feature thresholds)
- GAC processing better
- GRIB API (read both GRIB 1 and GRIB 2)
- CTTH threading configurable
- Bugs smashed
  - Wrong threshold table for t37t12 was used (SPR:376/SMR:354)
  - Memory leaks
  - …
- source_me, .profile_pps now automagically set up during configure

Plans

- Information model common for LEO and GEO
- Satellites and algorithms
- Processing on swath only
- More automated validation
- Installation and configuration
- Portability
- Input and feedback from users needed!

Plans: Satellites and algorithms

- Additional satellites
  - NPP/JPSS
  - FY-3
  - PCW
  - Post-EPS

- Additions to / improvements of algorithms
  - E.g. probabilistic cloud masking, improved aerosol detection

- We want to make
  - adding new instruments, and
  - testing/adding/replacing algorithms or parts of algorithms

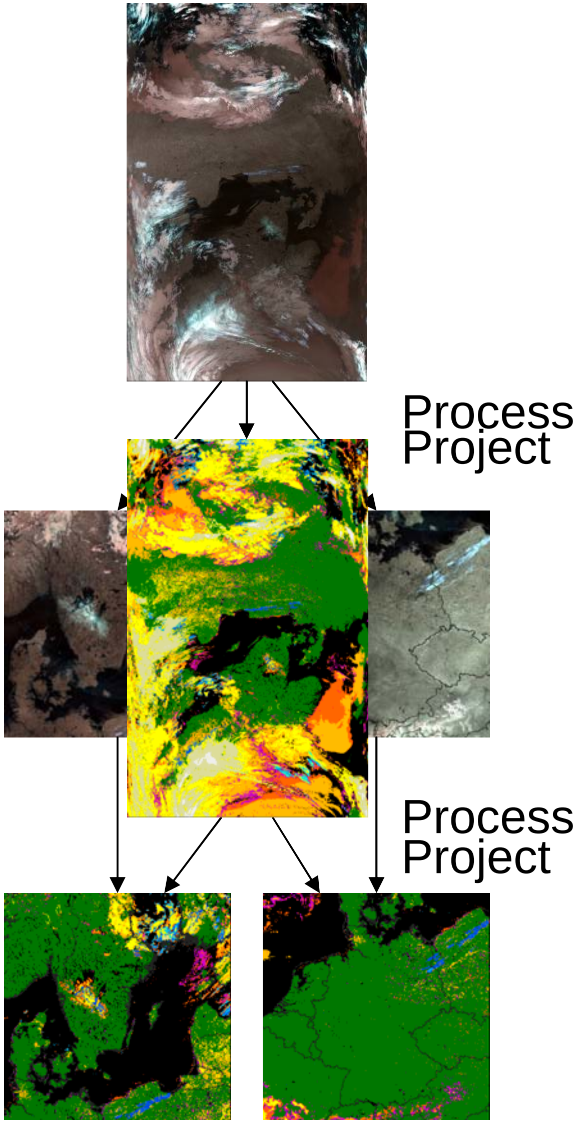
  easy for users and developers alike

Plans: Process on swath only

- Currently two pathways
  - Region: 1) project satellite data, 2) process
  - Swath: 1) process, 2) project result

  => Duplication of code, bugs
- Performance:
  - Region faster for < ~6 1024×1024 areas
  - Swath faster for more processing

How would this affect your processing?



Plans: More automated validation

- PPS currently includes some tools for validation
- Push-a-button validation
- Automatic, periodic, regular validation

Plans: Metadata standards, file formats

- Common information model and metadata standard for LEO and GEO
- Conform to community standards / conventions, such as
  - CF conventions
  - OPeNDAP
- File formats, e.g.
  - HDF5
  - netCDF 4
  - GeoTIFF
- Easier for users to use either LEO or GEO data
- Web services integration

Plans: Simplify Installation, Configuration

- Distribution
  - One complete distribution, including 3rd party dependencies
  - Possibly through script to download needed packages from Aemet servers
- Installation, one of the following candidates
  - Pure GNU build tools, more standardised
    - Umbrella package / installation script
  - Python Distutils
- Configuration
  - Uniform configuration files
  - Even more automated setup
  - Remove multiple definitions

Plans: Portability

- Currently testing only on Intel GNU/Linux
- Solaris user, portability reference

Plans: Technical solutions

- Modularise
- Clean, well defined, persistent interfaces
  - Easier to add functionality without affecting rest of system
  - Users can use whole system or parts
    - I/O, create own products, test/add/replace algorithms
    - **We want to know what you need**
  - Easier to locate bugs
  - Easier to use/replace 3rd party modules, e.g. for projection

- Python as far as possible, C for number crunching
  - Object oriented, cost effective development, easier to build

or

- C core (similar to current system)
  - Users can interface from C programs

Which parts would you like to see in C / Python?

Summary

- More flexible system – for users and developers
  - Easily add satellites/instruments, algorithms
  - Use parts of the PPS software package
  - Modularisation is key, good interfaces

Questions to users

- *Which parts of PPS do you currently use/interface?*
- *Which parts would you like to be able to interface? How would that interface ideally look for you?*
- *What additions/changes to PPS would you like to see in the future?*
- *Do you need support/validation for platforms other than Intel GNU/Linux?*
- *Do you use the Task Manager? Would you like to? On Global Metop?*

- Write down ideas at our poster
- Come talk with us!