Data documentation and dissemination supporting services, some standards and tools that may help

Øystein Godøy, Mathias Bavay and Massimo DiStefano
Data Management

80% of data are unavailable after 20 years from publication.
Gibney and Van Noorden (2013), Nature

http://www.nature.com/news/scientists-losing-data-at-a-rapid-rate-1.14416
Visibility through integration

- Regional and global data management frameworks
  - GEO
  - INSPIRE
  - SAON/IASC Arctic Data Committee
  - WMO Information System
  - WMO Integrated Global Observing System
  - ICSU Word Data System
  - GBIF
  - ...
Challenges during integration

- **Interoperability**
  - Discovery Metadata
    - Exchange Protocols (√)
    - Structures (√)
    - Semantics/terminology (-)
  - Data
    - Exchange Protocols (√)
    - Formats (-)
    - Use metadata (√)
    - Semantics/terminology (-)
    - Common data model (-)

- **Cultural**
  - Sharing data...
The FAIR guiding principles

- **To be Findable:**
  - F1. (meta)data are assigned a *globally unique and persistent identifier*
  - F2. data are described with rich metadata (defined by R1 below)
  - F3. metadata clearly and explicitly include the identifier of the data it describes
  - F4. (meta)data are *registered or indexed* in a searchable resource

- **To be Accessible:**
  - A1. (meta)data are retrievable by their identifier using a *standardized communications protocol*
  - A1.1 the protocol is open, free, and universally implementable
  - A1.2 the protocol allows for an authentication and authorization procedure, where necessary
  - A2. metadata are accessible, even when the data are no longer available

- **To be Interoperable:**
  - I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
  - I2. (meta)data use *vocabularies* that follow FAIR principles
  - I3. (meta)data include *qualified references* to other (meta)data

- **To be Reusable:**
  - R1. (meta)data are richly described with a plurality of accurate and relevant attributes
  - R1.1. (meta)data are released with a *clear and accessible data usage license*
  - R1.2. (meta)data are associated with detailed provenance
  - R1.3. (meta)data meet domain-relevant community *standards*
Standards and tools

- **Standards**
  - Discovery metadata
    - ISO19115
    - GCMD DIF
    - ACDD
    - OGC CSW
    - OAI-PMH
  - Use metadata
    - GBIF
    - CF
  - File formats (standardised)
    - NetCDF/CF
    - Excel/GBIF
    - ...
Must put data in context

- What’s the meaning of a number?
  - Basic metadata are needed for any use of data
- Data can be used in different ways
  - For adequate use of data, adequate information about the data is critical
- The whole is more than the sum of the pieces
  - Smart combination of information has a much larger potential than single observations
- Must
  - Make data talk together
  - Make data traceable
  - Make data count
The promised GCW/SLF software stack

Data flow
- Observation made by a human or a sensor
- Raw data storage
- Quality control and transformation to standardised form
- Documented and compliant data

Data publication
- Discovery metadata OAI-PMH
- Data access OPeNDAP
- Data access Direct HTTP
The concept...

- **Requirements**
  - No-brainer data use (quick usage for any application)
  - Scalability (i.e. minimum effort to add more stations)

- **Constraints**
  - Diversity of formats & protocols
  - Diversity of variable names
  - Diversity of units & other metadata
The magic...

- Data read in native format
- Data converted to standard compliant NetCDF CF1.6 with ACDD metadata
- Standard field names
- Standard metadata
- Standard search metadata
- NetCDF/CF served through OPeNDAP
  - i.e. FAIR data and services
The MeteoIO library is the engine that reads the native data. MeteoIO does name mapping, units conversions, merging, time corrections, filtering...

- MeteoIO writes the data back to NetCDF/CF
- Data are served through pyDAP
MeteoIO’s Workflow

Raw Data → Read Data → Raw data editing → Filter Data → Resample Data → Generate Data → Spatialize → Ready-to-use Data
MeteoIO’s Workflow

Nothing hard-coded, everything from a configuration file

Read Data

Raw data extraction

Prep Data

Generate Data

Spatialize

Ready-to-use Data
DAME: Data standardization

Raw data, vast variety of formats

Read with different plugins
Process to standard levels
Add search metadata
Write with a common plugin

Standardised product
DAME: Data standardization

Raw data, vast variety of formats
Read search metadata
Write with a common plugin

Backup configuration file
Implies traceability

Standardised product
Summary on MeteoIO

- Not only meteorological data
- For each station, a configuration file (a few lines long)
- A small script calls MeteoIO on all configuration files at regular intervals and copies the resulting NetCDF/CF files to the OPeNDAP server
- Further reading
  - Get MeteoIO at http://models.slf.ch
Serving data through web services

- OPeNDAP allows data streaming and integration directly in analysis tools like R, Python and Matlab
- Using the lightweight pyDAP library/application
  - Unicode decoding issues are fixed when dealing with NetCDF files with Unicode characters in the metadata (this involved fix in both pyDAP and one of its dependencies webob)
  - Fixing Key-Value issues when dealing with NetCDF where the dimension are not listed also as variables [ dims not in vars]
  - Adding docker support to serve data through externally mounted docker volume - pyDAP running as Apache WSGI
  - Adding script to generate Debian packages for both webob and pyDAP libraries.
    - The build of packages is automated when building the docker image.
Attributes

- **N_GLOBAL**
  - String station_name "KVI1A";
  - String uom_identifier "00111";
  - String data_created "2010-03-02T07:01:38.404034+00:00";
  - String time_coverage_end "2019-03-02T07:09:00";
  - String title "Observations from station KVI1A 5999388";
  - String retdata_link https://oa3mfn.met.no/oai?verb=GetRecords&metadataPrefix=iso19139&identifier=SN999388;
  - String acknowledgment "frost.met.no";
  - String comment "Observations based on data from frost.met.no";
  - String institution "Norwegian Meteorological Institute";
  - String featureType "TimeSeries";
  - String id "meto_obs_SN999388";
  - String references "";
  - String geospatial_lat_min "58.105606";
  - String Conventions "ADO-1.3; CF-1.6";
  - String creator_name "Norwegian Meteorological Institute";
  - String keywords "observations";
  - String history "2010-03-02T07:01:38.403994+00:00: frost write netcdf";
  - String creator_url https://met.no/;
  - String geospatial_lon_min "31.054309";
  - String geospatial_bounds "31°18.465366 58°10.650606";
  - String geospatial_lon_max "58.105606";
  - String creator_email "observasjon@met.no";
  - String geospatial_bounds_crs "latlon";
  - String source "Meteorological surface observation via frost.met.no";
  - String time_coverage_start "1996-01-01T00:00:00";
  - String wigos_identifier "unknown";

- **latitude**
  - String long_name "latitude";
  - String standard_name "latitude";
  - String units "degree_north";

- **longitude**
  - String long_name "longitude";
  - String standard_name "longitude";
  - String units "degree_east";

- **air_pressure_at_sea_level**
  - String long_name "Air pressure at sea level";
  - String standard_name "air_pressure_at_sea_level";
  - String units "Pa";

- **surface_air_pressure**
  - String long_name "Air pressure at station level";
  - String standard_name "surface_air_pressure";
  - String units "Pa";

- **air_temperature**
  - String long_name "Air temperature";
  - String standard_name "air_temperature";
  - String units "K";
Summary

- MeteoIO for transformation to FAIR and quality control of sensor data
- PyDAP for public access to data through interoperability interfaces
- Need for online transformation services?