Uncertainty Metadata Conventions Specifications

version draft-v0.1

CoMet Toolkit Team

December 19, 2024

Contents

Uncertainty Metadata Conventions (UNC)

Authors

Lead Authors

- Sam Hunt, NPL
- Pieter De Vis, NPL

Introduction

Goals

Measurement datasets are becoming larger, more complex, and are increasingly used to support critical applications such as manufacturing, health, and environmental monitoring. Reliable interpretation of these measurements requires accompanying uncertainty and error-covariance information - however, this is often overlooked. Where available, such information lacks standardisation and could, in principle, be highly complex and large.

The goal of this specification is to provide a standardised metadata format for storing the accompanying uncertainty/error-covariance information with measurement datasets. This format is intended to support fully capturing the content of the error-covariance matrices associated with measurement data in a compact structure, by parameterising error-covariance with a simple set of metadata.

Philosophy

This specification is intended to contribute to and build upon an existing ecosystem of standards and best practices. In particular the following are adhered to, to the extent possible:

- The understanding of uncertainty concepts defined in the JGCM GUM (Guide to the expression of uncertainty in measurement) suite of documents.
- The definition of uncertainty-related terminology defined in the JGCM VIM (International Vocabulary for Metrology).
- The NetCDF data model for creating self-describing, array-oriented scientific datasets.
- The Climate and Forecast (CF) conventions on metadata for weather and climate data.

The work builds on previous work on the standardisation uncertainty information for climate data developed within the H2020 FIDUCEO project.

Within this context, this specification also attempts to adhere to the following principles:

- **Scalability of Complexity**

  The complexity of the metadata should align with the use case:

  - *Simple use cases* should be achievable with a *simple implementation*.
  - *Complex use cases* should be *possible without unnecessary restrictions*.
- **Minimisation of Redundancy**

  The metadata specification should *avoid duplication* of information to prevent potential inconsistencies.
- **Human and Machine Readability**

  Metadata must be:

  - *Comprehensible* for humans.
  - *Parsable* by machines.

Terminology

For terminology related to measurements and associated uncertainties, definitions within the VIM (International Vocabulary for Metrology) are adopted to the extent possible. This includes the following important terms:

- Error
- Uncertainty
- Coverage factor

The following terms are derived from X:

- Error-correlation
- Error-covariance
- fractional uncertainty

Format for Examples

The ASCII format used to describe the contents of a NetCDF dataset is called CDL (NetCDF Common Data form Language). This follows C-style indexing where indices start at 0, and the last declared dimension varies fastest in storage order. For example, in a 2D array `data(time, lat)`, the `lat` dimension changes faster than `time` during indexing.

Snippets of CDL are used to present examples in this specification. A minimal example of a measurement dataset in CDL is given below. Here a dataset of `temperature` with its metadata (units and description) is defined along `time`, `lat` and `lon` dimensions.

```
netcdf short_example {
  dimensions:
    time = 2 ;
    lat = 2 ;
    lon = 2 ;

  variables:
    float temperature(time, lat, lon) ;
      temperature:units = "K" ;
      temperature:long_name = "Temperature" ;

  data:
    temperature =
      290.1, 291.2,
      292.3, 293.4,
      294.5, 295.6,
      296.7, 297.8 ;
}
```

Measurement Dataset Structures

As mentioned above, the NetCDF data model for creating self-describing, array-oriented scientific datasets is adopted. The components of NetCDF datasets are described in Section 2 of the NUG (NetCDF Users Guide). In this section, we introduce the core components of this data model relevant to this standard.

Variables

Datasets are composed of variables, which are multidimensional data arrays.

This standard defines the following categories of variables:

- **Observation Variables**

  Observation variables represent a multidimensional array of measurements.

- **Uncertainty Variables**

> *Uncertainty variables* represent a component of uncertainty associated with an *observation variable*. An *observation variable* may have multiple *uncertainty variables* associated with them.
>
> *Uncertainty variables* must have the same dimensions as the *observation variable* they are associated with.

A dataset may also contain variables that are neither *observation variables* or *uncertainty variables*.

Dimensions

A variable may have any number of named dimensions, including zero – e.g., `"x"`, `"y"`, `"time"`. Dimensions may be of any size, including unity.

Data Types

Observation variables and *uncertainty variables* must be `floats`.

Note: these variables may be encoded as e.g. integers for efficient storage on disc.

Attributes

Dataset attributes provide metadata about the dataset, its variables, and dimensions. Global attributes describe the entire dataset (e.g., title, institution, history). Variable attributes define specific properties of the variable (e.g., units, valid ranges). These attributes ensure data is interpretable, support automated processing, and facilitate sharing by following standardised conventions.

This standard defines a set of variable attributes to:

- link *observation variables* with their associated *uncertainty variables*
- define the error-correlation properties of a given *uncertainty variables* in a compact way.

A dataset may also contain non-standard attributes.

Uncertainty Attributes

Assigning Uncertainty Components

Uncertainty variables are associated with their *observation variable* through the *observation variable*'s `"unc_comps"` attribute. This attribute contains a list of the names of all of the *uncertainty variables* associated with an *observation variable*.

The following example of a dataset, in CDL syntax, shows a `temperature` variable defined along 3 dimensions - `time`, `lat`, and `lon`. `temperature` has two uncertainty components associated with it - `u_calibration` and `u_noise`.

```
variables:
  float temperature(time, lat, lon) ;
    temperature:unc_comps=["u_calibration", "u_noise"];
  float u_calibration(time, lat, lon);
  float u_noise(time, lat, lon);
```

Units

The physical units associated with *observation variables* and *uncertainty variables* should be defined by the `"units"` variable attribute as a string.

Observation variables are assumed dimensionless if the variable attribute `"units"` is not defined.

uncertainty variables must have the same `"units"` as the *observation variables* they are associated with. If `"units"` is not defined, the *uncertainty variable* is assumed fractional.

The following example of a dataset again shows a `temperature` variable associated with two uncertainty components - `u_calibration` and `u_noise`. Here, `u_calibration` is defined with units K, matching `temperature`. `u_noise` has no defined units and so is a fractional uncertainty

```
variables:
  float temperature(time, lat, lon);
    temperature:unc_comps=["u_calibration", "u_noise"];
    temperature:units="K"
  float u_calibration(time, lat, lon);
    u_calibration:units="K"
  float u_noise(time, lat, lon);
```

Uncertainty PDF Shape

The probability density function (PDF) shape associated with the uncertainty estimate values in an *uncertainty variable* is defined with the variable attribute `"pdf_shape"`.

`"pdf_shape"` can have one of the following values:

- `"gaussian"` - for uncertainties represented by a Gaussian PDF
- `"rectangular"` - for uncertainties represented by a uniform PDF
- …

If `"pdf_shape"` is not defined for an *uncertainty variable* it is assumed to be `"gaussian"`.

The following example of a dataset again shows a `temperature` variable associated with two uncertainty components - `u_calibration` and `u_noise`. Here, `u_calibration` is defined to be represented by a rectangular PDF. `u_noise` has no defined `"pdf_shape"` and so is assumed Gaussian.

```
variables:
  float temperature(time, lat, lon);
    temperature:unc_comps=["u_calibration", "u_noise"];
    temperature:units="K"
  float u_calibration(time, lat, lon);
    u_calibration:units="K"
    u_calibration:pdf_shape="rectangular"
  float u_noise(time, lat, lon);
```

Error-Correlation Structure

For *observation variables* with elements, the associated error-covariance matrix per *uncertainty variables* has elements. Where the *observation variables* are large, it an quickly become impractical to store this data.

However, in many cases the associated error-correlation matrix can in fact be simply parameterised in a compact form (e.g., identity, full, banded). This standard defines a set of *uncertainty variable* variable attributes to store this parameterisation.

To allow maximum flexibility, different parameterisations can be defined along each *uncertainty variable* dimension, `dim_i` (or sets of dimensions, [`dim_i`, `dim_j`, …]). For example, an error could be systematic in longitude and latitude at each time step, but random between time steps.

Parameterisations have the form…

Error-correlation attributes

Attribute name	Type	Description	Example
err_corr_dimi_name	str	Dimension name	err_corr_dim1_name=”time”
err_corr_dimi_form	str	Parameterisation name	err_corr_dim1_form=”random”
err_corr_dimi_params	list[Any]	Parameterisation parameters	err_corr_dim1_params=[1,2,3]

err_corr_dimi_units	list[str]	Parameterisation parameter units	err_corr_dim1_params=["second", "m", "K"]

Existing parmaterisations:

Error-correlation parameterisations

Parameterisation Form	Parameters	Description
random	[]	No error-correlation between elements in observation variable.
systematic	[]	Full error-correlation between elements in observation variable.

The following example of a dataset again shows a `"temperature"` variable associated with two uncertainty components - `"u_calibration"` and `"u_noise"`.

Here, `"u_calibration"` is defined to have a systematic error-correlation in the `lat` and `lon` dimensions, and random in `time` dimension (perhaps, there is a recalibration between the measurements at each time step!).

`"u_noise"` have a error-correlation defined random in all dimensions.

```
variables:
  float temperature(time, lat, lon);
    temperature:unc_comps=["u_calibration", "u_noise"];
    temperature:units="K"
  float u_calibration(time, lat, lon);
    u_calibration:units="K";
    u_calibration:pdf_shape="rectangular";
    u_calibration:err_corr_dim1_name=["lat", "lon"];
    u_calibration:err_corr_dim1_form="systematic";
    u_calibration:err_corr_dim1_params=[];
    u_calibration:err_corr_dim1_units=[];
    u_calibration:err_corr_dim2_name="time";
    u_calibration:err_corr_dim2_form="random";
    u_calibration:err_corr_dim2_params=[];
    u_calibration:err_corr_dim2_units=[];
  float u_noise(time, lat, lon);
    u_calibration:err_corr_dim1_name=["time", "lat", "lon"];
    u_calibration:err_corr_dim1_form="random";
    u_calibration:err_corr_dim1_params=[];
    u_calibration:err_corr_dim1_units=[];
```