Metadata recommendations for encoding products in netCDF based on the CF convention

Antonio S. Cofiño[1], Manuel Fuentes[2], Kevin Marsh[2], Sebastien Villaume[2], Richard Mladek[2], Eduardo Penabad[3], Cedric Bergeron[2] and Baudoin Raoult[2]

[1] Dep. of Applied Maths and Computational Sciences. Univ. de Cantabria (ES)

[2] Forecast Department, Products Team. ECMWF

[3] C3S, Seasonal Forecast. ECMWF

UC UNIVERSIDAD DE CANTABRIA MG METEOROLOGY GROUP ECMWF UERRA Uncertainties in Ensembles of Regional ReAnalyses is-enes INFRASTRUCTURE FOR THE EUROPEAN NETWORK FOR EARTH SYSTEM MODELLING

Introduction

The objective is to provide some **recommendations** (and examples) for the **encoding of metadata** and data in a form suitable for archiving.

The aim is to be explicit (as possible), to provide: values for file/record specific attributes (and not for overall collections), richer metadata and conventions.

The **intention** is to be minimalist to allow downstream data **re-use beyond the original intent**, product development, scientific quality control and provision of long-term preservation.

This means that this recommendation **is not intended to provide metadata to specific project,** experiment or simulation, like attributes for data discovery, or special characteristics.

This recommendation is **based on CF-1.6 Convention Document** and the **Standard Name Table**.

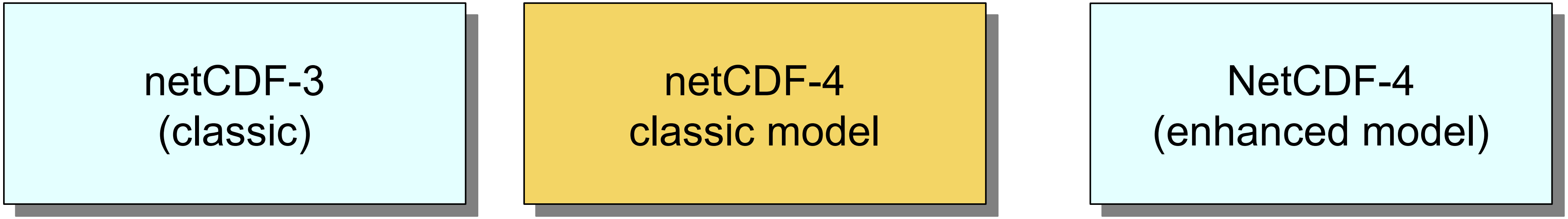
The **encoding reference** used is the **netCDF-classic data model**, but extension to other encoding formats should be possible.

Encoding formt

*The **encoding reference** used is the **netCDF-classic data model**, but extension to other encoding formats should be possible:*

When netCDF is used the recommendation is to use the **netCDF-4 classic model format** which follows the netCDF classic data model, but provides features from the **HDF5 storage layer**:

- Uses classic API for compatibility
- Uses netCDF-4/HDF5 storage for **compression, chunking, performance**
- To use, just recompile, relink



netCDF-3
(classic)

netCDF-4
classic model

NetCDF-4
(enhanced model)

Metadata

- Metadata is *data that provides information about other data*
- Distinct types of metadata exist:
  - **descriptive metadata** describes a resource for purposes such as discovery and identification. It can include elements such as title, abstract, author, and keywords.
  - **structural metadata** is about containers of data and indicates how compound objects are put together, for example, how pages are ordered to form chapters. It describes the types, versions, relationships and other characteristics of digital materials.
- **This recommendation** it's about structural metadata enabling users and tools to **identify comparable data** and facilitating the development of software **to store, extract, process, analyze and display.**

Data Fields

A **data field** (DF) is defined as a combination of several data variables.

In general, a **data field may span data variables in more than one file** object: i.e. different ranges of a time coordinate.

Rules for aggregating data variables from one or several files objects into a single data field are needed but **are not defined by CF**. (but some exist)

The assumption is that **each DF is independent**.

But data stored in netCDF files are **often not independent**, because **coordinate** variables are **shared between them**.

- This sharing is a means of saving disk space, and any software should be able to alter any DF without altering any other DF.
- If a given coordinate of a DF is altered this should not affect other DF's in the same file.

For example, daily near-surface minimum temperature could share longitude and latitude coordinates variables with 6 hourly instantaneous temperature at 500mb pressure level, but different temporal and vertical coordinate variables should be used.

A test on the equality, and/or equivalence, of coordinates between different DF's should be made when attempting to merge DF from different files or records.

elds

A DF *may* have:

1. 0..N **domain indexes:** each domain coordinate has a size (an integer value greater than zero)**.**
2. 1 **data array** whose shape is been determined by the domain indexes. All elements on the data array must be of the same data type (numeric, char or string).
3. A collection of **domain coordinates**: A domain coordinate construct indicates the physical meaning and locations of the cells for a unique **domain indexes** of the field.
4. A collection of **auxiliary coordinates**: An auxiliary coordinate provides auxiliary information for interpreting the cells of an ordered list of one or more **domain coordinate** of the field.
5. A collection of **cell-measures**: A cell measure provides information about the size, shape or location of the cells (n-dimensional in general) defined by an ordered list of one or more domain coordinates of the DF.
6. An ordered collection of **cell-methods**: A cell methods describes how the data values represent variation of the quantity within cells. The methods are not necessary commutative, therefore it is an ordered list of methods.
7. A **Coordinate reference systems**: A coordinate reference system relates the field's coordinate values to locations in a earth reference frame.
8. Other **properties** which represents metadata about the DF. Not all attributes in a netCDF file are properties in this sense. Some of these can be global attributes in a netCDF file. It is assumed that global attribute is also an attribute of every data variable, although it is superseded if the data variable has its own attribute with an identical name.
9. Ancillary fields which are used to identify fields which provide additional metadata (i.e. quality of the data).

The **domain axes, domain coordinates, auxiliary coordinates, cell measures, and cell method describe the domain** in which the DF resides.

Coordinate systems

Explicit list of **domain coordinates** must be provided by **the** `coordinates` attribute on data variable.

The coordinates define the provides the **coordinate system**

Identification of a **coordinate type** by its units alone is complicated, the `standard_name` attribute provides a direct identification.

Additionally, to identify generic spatial and/or temporal coordinates the use of the attribute `axis` may be added to a coordinate-variable and given one of the values *X, Y, Z or T*.

The **values of a coordinate** are the locations of the **boundaries between cells**. The `bounds` attribute attached to coordinate variable indicates the variable with those values.

The values on **coordinate variable** are **labels for cell locations**

The dimensions on coordinates variables must comply with some rules wrt dimensions on data variable.

Coordinate reference Systems
Regular longitude and latitude

```
netcdf regular_latitude_longitude_grid {
  //global attributes:
    :Conventions = "CF-1.6";
  dimensions:
    latitude = 180 ;
    longitude = 360 ;
  variables:
    double mslp(latitude, longitude) ;
      mslp:standard_name = "air_pressure_at_sea_level" ;
      mslp:units = "Pa" ;
      mslp:grid_mapping = "hcrs" ;
      mslp:coordinates = "latitude longitude" ;
    double latitude(latitude) ;
      latitude:standard_name = "latitude" ;
      latitude:units = "degrees_north" ;
      latitude:axis = "Y" ;
    double longitude(longitude) ;
      longitude:standard_name = "longitude" ;
      longitude:units = "degrees_east" ;
      longitude:axis = "X" ;
    char hcrs ;
      hcrs:grid_mapping_name = "latitude_longitude" ;
}
```

Coordinate Systems
Vertical

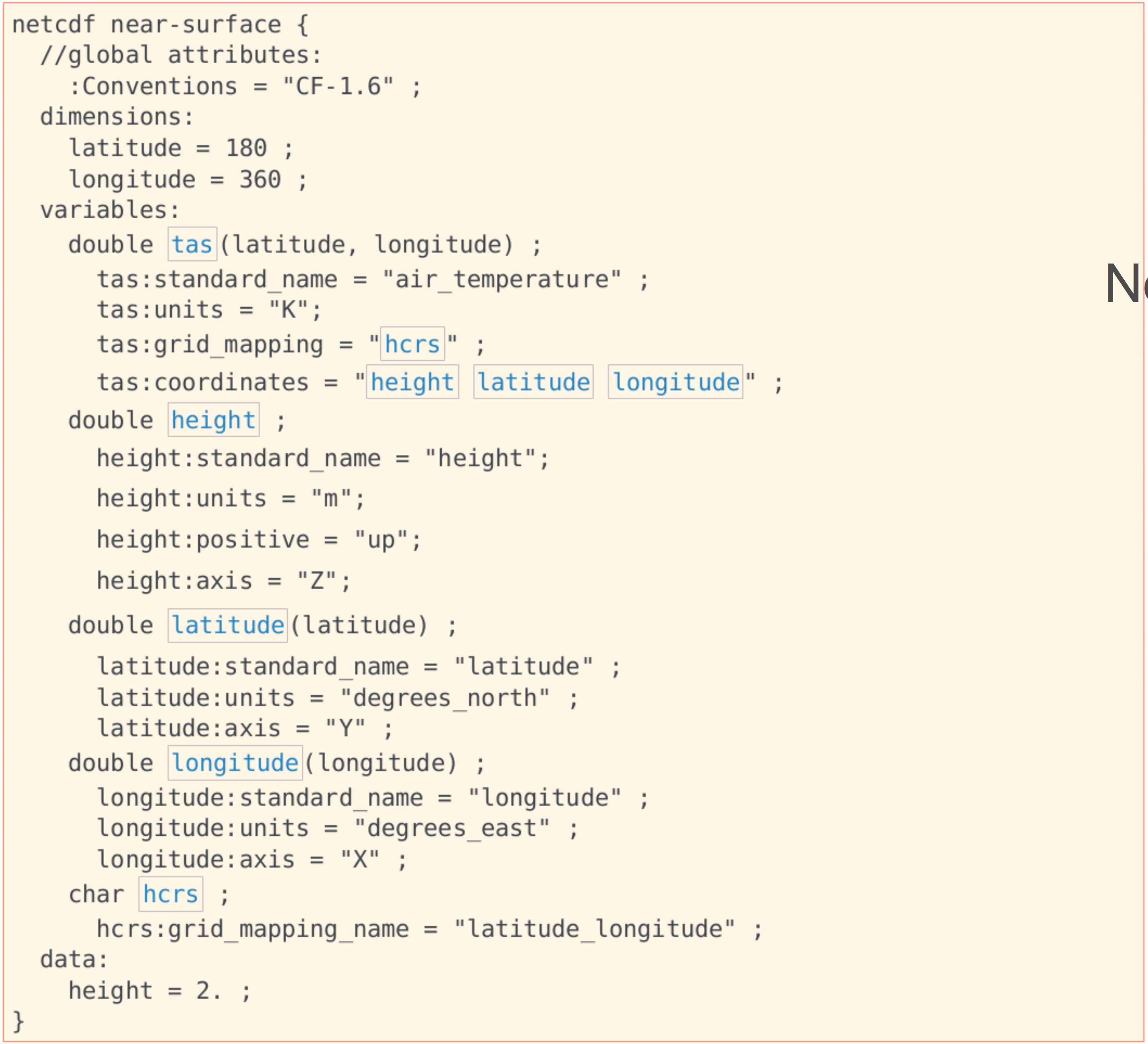
Variables representing dimensional height or depth axes must always explicitly include the `units` attribute; there is no default value for this attribute. If the `units` attribute value is a valid pressure unit the default value of the `positive` attribute is `down`.

A vertical coordinate will be identifiable by:

- units of pressure; and/or
- the presence of the `positive` attribute with a value of `up` or `down` (case insensitive); and/or
- by providing the `standard_name` attribute with an appropriate value; and/or
- the `axis` attribute with the value `Z`.

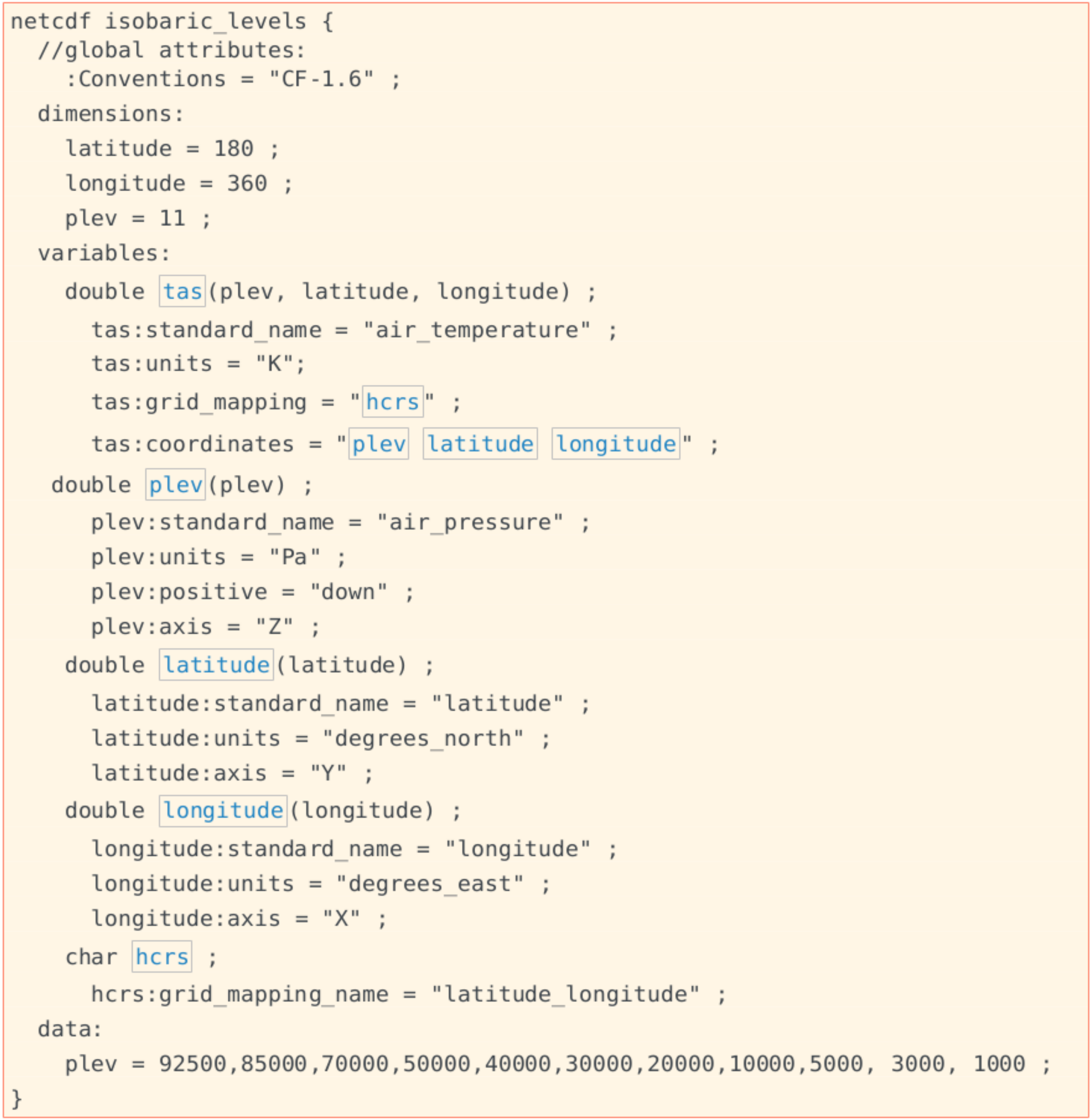
ence Systems

Near-surface fields



```
netcdf near-surface {
  //global attributes:
    :Conventions = "CF-1.6" ;
  dimensions:
    latitude = 180 ;
    longitude = 360 ;
  variables:
    double tas(latitude, longitude) ;
      tas:standard_name = "air_temperature" ;
      tas:units = "K";
      tas:grid_mapping = "hcrs" ;
      tas:coordinates = "height latitude longitude" ;
    double height ;
      height:standard_name = "height";
      height:units = "m";
      height:positive = "up";
      height:axis = "Z";
    double latitude(latitude) ;
      latitude:standard_name = "latitude" ;
      latitude:units = "degrees_north" ;
      latitude:axis = "Y" ;
    double longitude(longitude) ;
      longitude:standard_name = "longitude" ;
      longitude:units = "degrees_east" ;
      longitude:axis = "X" ;
    char hcrs ;
      hcrs:grid_mapping_name = "latitude_longitude" ;
  data:
    height = 2. ;
}
```

ference Systems



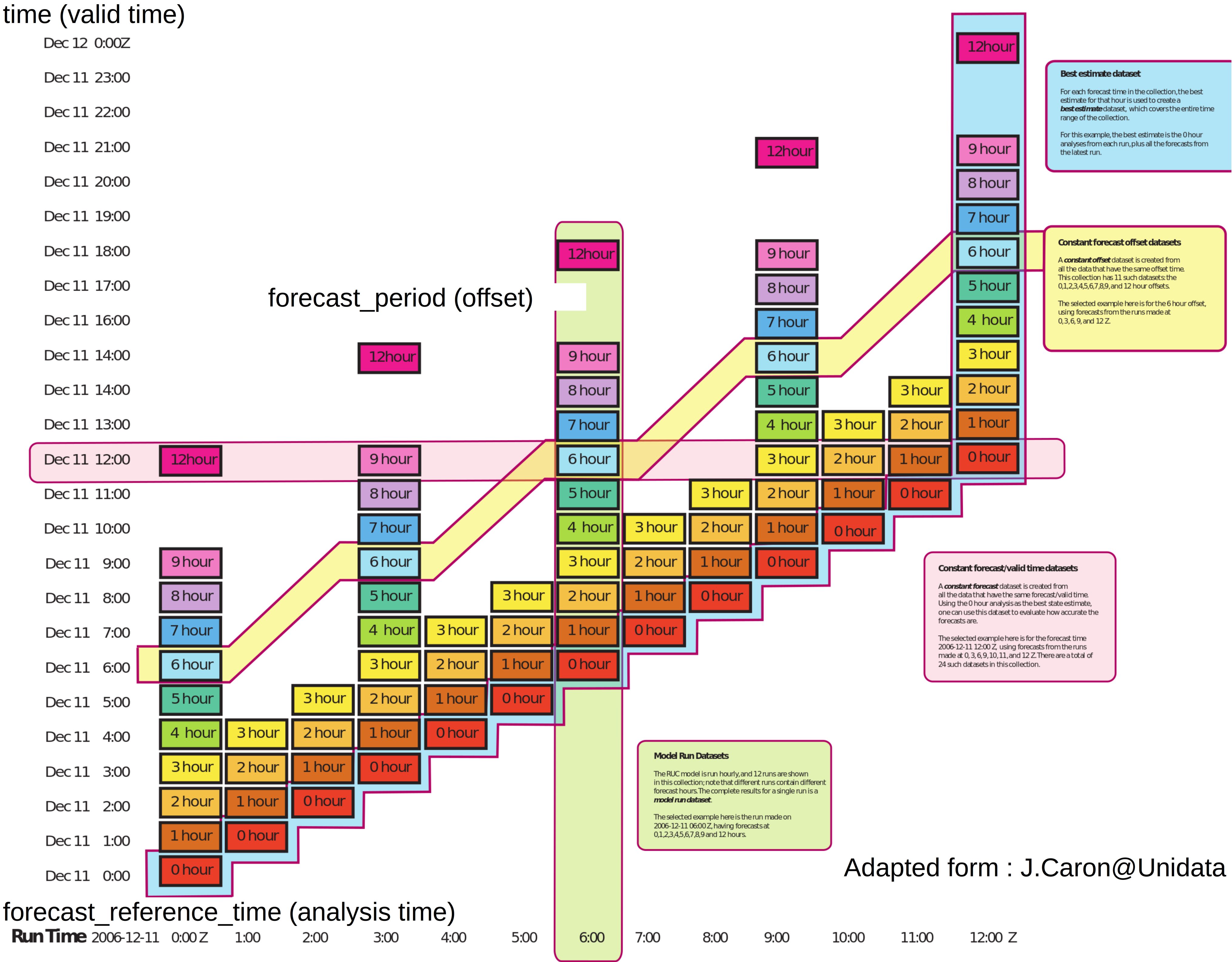
```
netcdf isobaric_levels {
  //global attributes:
    :Conventions = "CF-1.6" ;
  dimensions:
    latitude = 180 ;
    longitude = 360 ;
    plev = 11 ;
  variables:
    double tas(plev, latitude, longitude) ;
      tas:standard_name = "air_temperature" ;
      tas:units = "K";
      tas:grid_mapping = "hcrs" ;
      tas:coordinates = "plev latitude longitude" ;
   double plev(plev) ;
      plev:standard_name = "air_pressure" ;
      plev:units = "Pa" ;
      plev:positive = "down" ;
      plev:axis = "Z" ;
    double latitude(latitude) ;
      latitude:standard_name = "latitude" ;
      latitude:units = "degrees_north" ;
      latitude:axis = "Y" ;
    double longitude(longitude) ;
      longitude:standard_name = "longitude" ;
      longitude:units = "degrees_east" ;
      longitude:axis = "X" ;
    char hcrs ;
      hcrs:grid_mapping_name = "latitude_longitude" ;
  data:
    plev = 92500,85000,70000,50000,40000,30000,20000,10000,5000, 3000, 1000 ;
}
```

Isobaric levels

Time coordinates

- `forecast_reference_time`: The forecast reference time in NWP is the "data time", i.e. the time of the analysis from which the forecast was made.It is not the time for which the forecast is valid; the standard name of `time` should be used for that time
- `forecast_period`: Forecast period is the time interval between the forecast reference time and the validity time. A period is an interval of time, or the time-period of an oscillation
- `time`: The valid time is the time for which the forecast or observation is valid

It is recommended that the calendar be specified by the attribute `calendar` which is assigned to absolute time coordinates (i.e. `time` and `forecast_reference_time`).



Best estimate dataset

For each forecast time in the collection, the best estimate for that hour is used to create a ***best estimate*** dataset, which covers the entire time range of the collection.

For this example, the best estimate is the 0 hour analyses from each run, plus all the forecasts from the latest run.

Constant forecast offset datasets

A ***constant offset*** dataset is created from all the data that have the same offset time. This collection has 11 such datasets: the 0,1,2,3,4,5,6,7,8,9, and 12 hour offsets.

The selected example here is for the 6 hour offset, using forecasts from the runs made at 0, 3, 6, 9, and 12 Z.

Constant forecast/valid time datasets

A ***constant forecast*** dataset is created from all the data that have the same forecast/valid time. Using the 0 hour analysis as the best state estimate, one can use this dataset to evaluate how accurate the forecasts are.

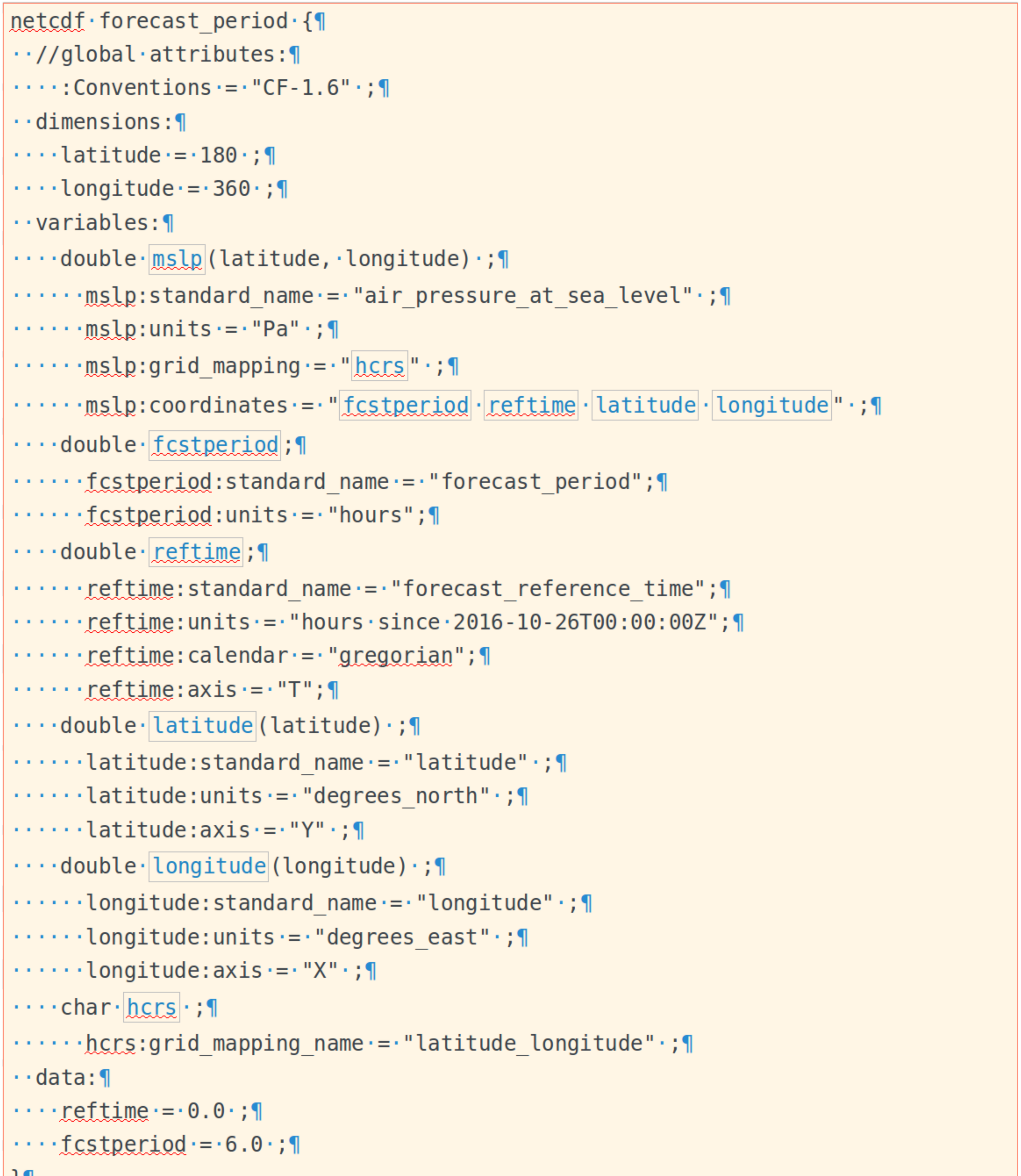
The selected example here is for the forecast time 2006-12-11 12:00 Z, using forecasts from the runs made at 0, 3, 6, 9, 10, 11, and 12 Z. There are a total of 24 such datasets in this collection.

Model Run Datasets

The RUC model is run hourly, and 12 runs are shown in this collection; note that different runs contain different forecast hours. The complete results for a single run is a ***model run dataset***.

The selected example here is the run made on 2006-12-11 06:00 Z, having forecasts at 0,1,2,3,4,5,6,7,8,9 and 12 hours.

Adapted form : J.Caron@Unidata



```
netcdf forecast_period {
  //global attributes:
    :Conventions = "CF-1.6" ;
  dimensions:
    latitude = 180 ;
    longitude = 360 ;
  variables:
    double mslp(latitude, longitude) ;
      mslp:standard_name = "air_pressure_at_sea_level" ;
      mslp:units = "Pa" ;
      mslp:grid_mapping = "hcrs" ;
      mslp:coordinates = "fcstperiod reftime latitude longitude" ;
    double fcstperiod;
      fcstperiod:standard_name = "forecast_period";
      fcstperiod:units = "hours";
    double reftime;
      reftime:standard_name = "forecast_reference_time";
      reftime:units = "hours since 2016-10-26T00:00:00Z";
      reftime:calendar = "gregorian";
      reftime:axis = "T";
    double latitude(latitude) ;
      latitude:standard_name = "latitude" ;
      latitude:units = "degrees_north" ;
      latitude:axis = "Y" ;
    double longitude(longitude) ;
      longitude:standard_name = "longitude" ;
      longitude:units = "degrees_east" ;
      longitude:axis = "X" ;
    char hcrs ;
      hcrs:grid_mapping_name = "latitude_longitude" ;
  data:
    reftime = 0.0 ;
    fcstperiod = 6.0 ;
}
```

Discrete coordinates

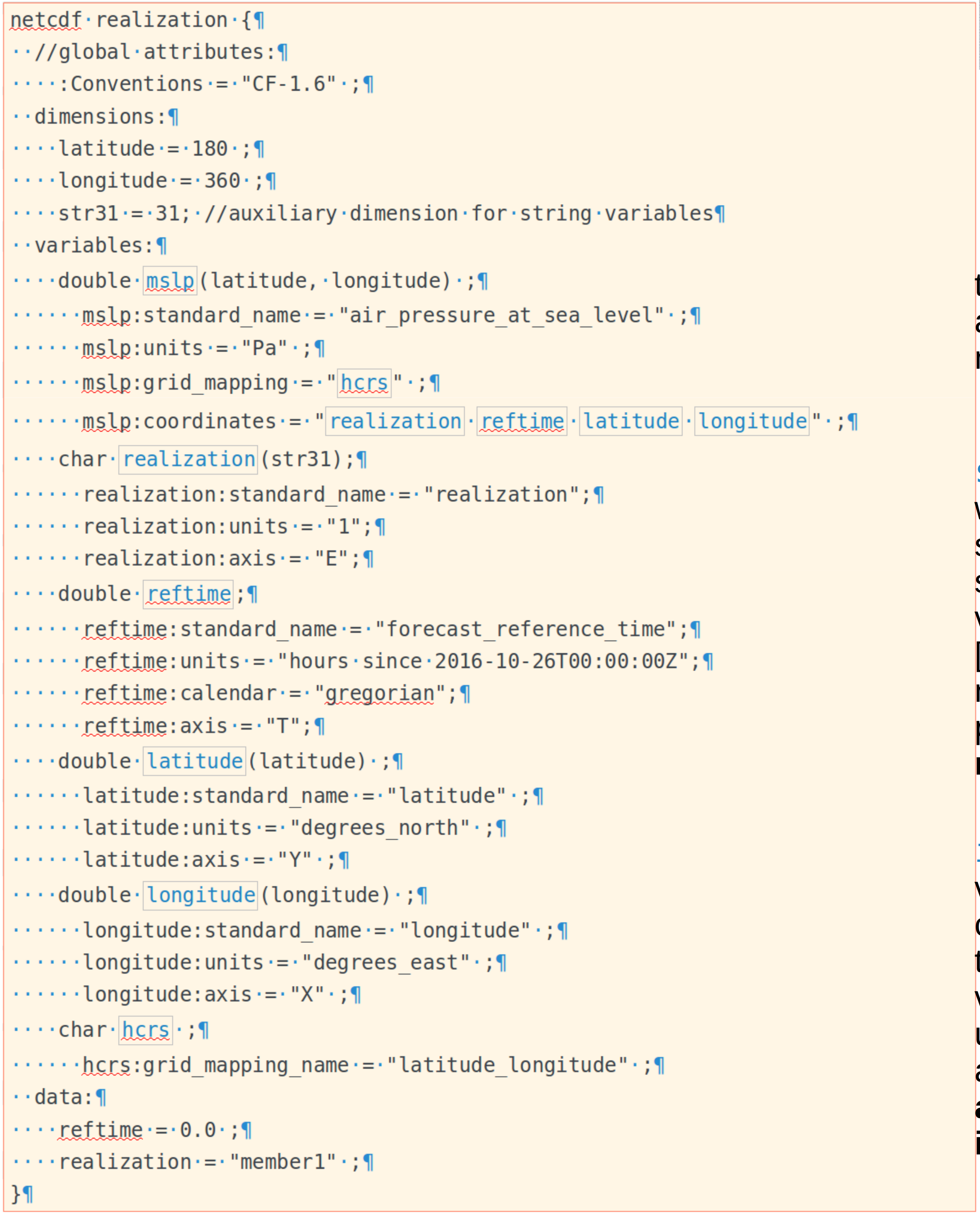
The spatio-temporal coordinates and other geophysical quantities may likewise serve as **continuous coordinates**, for instance density, temperature or radiation wavelength.

But there is a need for coordinates which indicate either an **ordered list** or an **unordered collection**, and does not correspond to any continuous quantity variable.

Such coordinate may be called **discrete coordinate**.

For instance members of a ensemble may be defined as a realization coordinate it is been indicated by providing the `standard_name` attribute with value `realization`,

`realization`: The term “realization” is used to label a dimension that can be thought of as a statistical sample, e.g., labeling members of a model ensemble.



```
netcdf realization {
  //global attributes:
    :Conventions = "CF-1.6" ;
  dimensions:
    latitude = 180 ;
    longitude = 360 ;
    str31 = 31; //auxiliary dimension for string variables
  variables:
    double mslp(latitude, longitude) ;
      mslp:standard_name = "air_pressure_at_sea_level" ;
      mslp:units = "Pa" ;
      mslp:grid_mapping = "hcrs" ;
      mslp:coordinates = "realization reftime latitude longitude" ;
    char realization(str31);
      realization:standard_name = "realization";
      realization:units = "1";
      realization:axis = "E";
    double reftime;
      reftime:standard_name = "forecast_reference_time";
      reftime:units = "hours since 2016-10-26T00:00:00Z";
      reftime:calendar = "gregorian";
      reftime:axis = "T";
    double latitude(latitude) ;
      latitude:standard_name = "latitude" ;
      latitude:units = "degrees_north" ;
      latitude:axis = "Y" ;
    double longitude(longitude) ;
      longitude:standard_name = "longitude" ;
      longitude:units = "degrees_east" ;
      longitude:axis = "X" ;
    char hcrs ;
      hcrs:grid_mapping_name = "latitude_longitude" ;
  data:
    reftime = 0.0 ;
    realization = "member1" ;
}
```

`realization`: The term *realization* is used to label a dimension that can be thought of as a statistical sample, e.g., labeling members of a model ensemble.

`source`: An auxiliary coordinate variable with a standard name of *source* contains string values [...] were model-generated, source should name the model and its version, as specifically as could be useful. [...] The use of *source* as the standard name for an auxiliary coordinate variable permits the **aggregation** of data from **multiple sources** within a single data file.

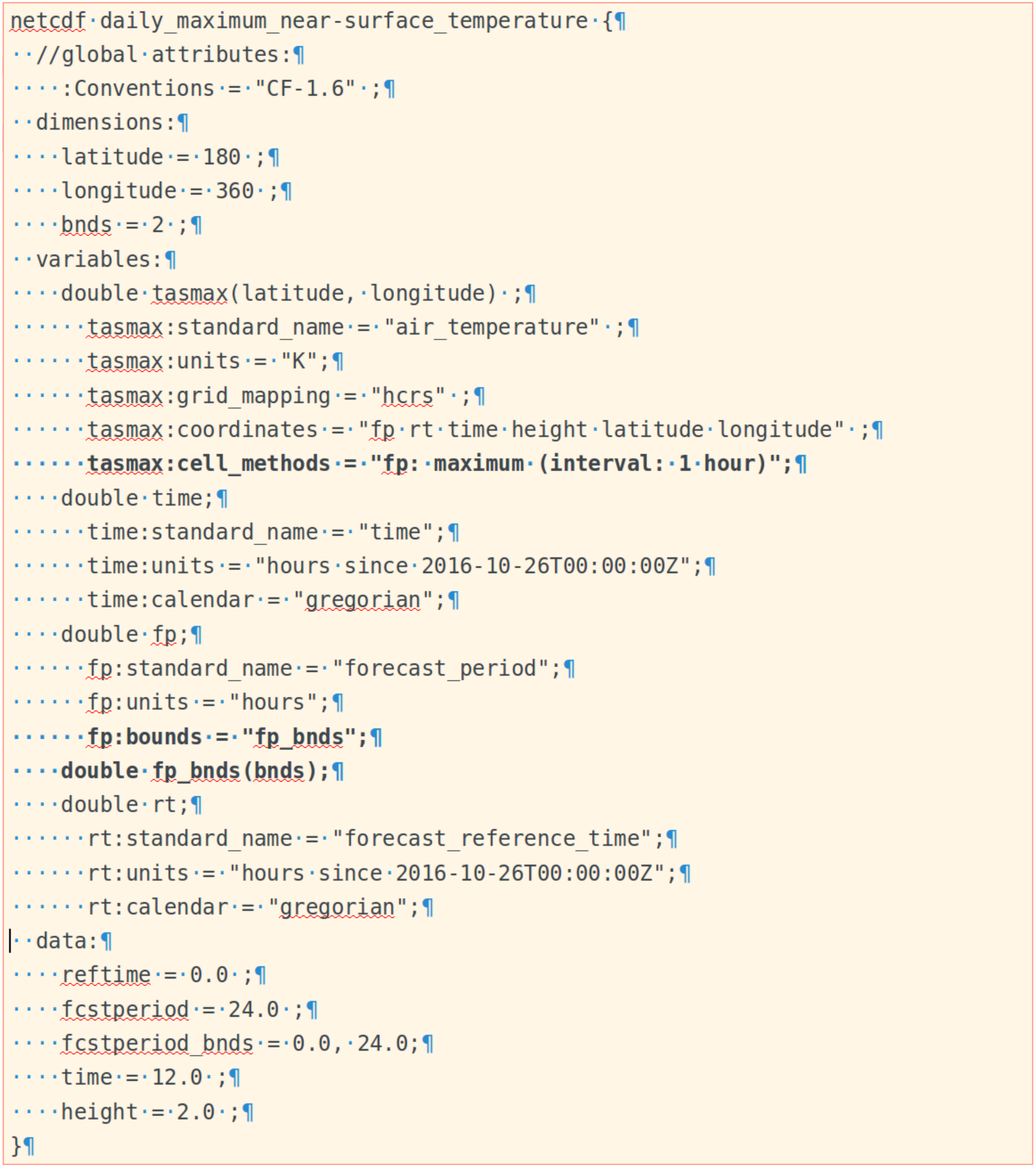
`institution`: An auxiliary coordinate variable with a standard name of *institution* contains string values which specify where the original data, with which the coordinate variable is associated, were produced. The use of *institution* as the standard name for an auxiliary coordinate variable permits the **aggregation** of data from **multiple institutions** within a single data file.

Cell aggregations

When data does not represent the **point values** of a field, but instead represents some characteristic of the **field within cells** of finite "volume", a complete description of the variable should include metadata that describes the domain or extent of each cell, and the characteristic of the field that the cell values represent.

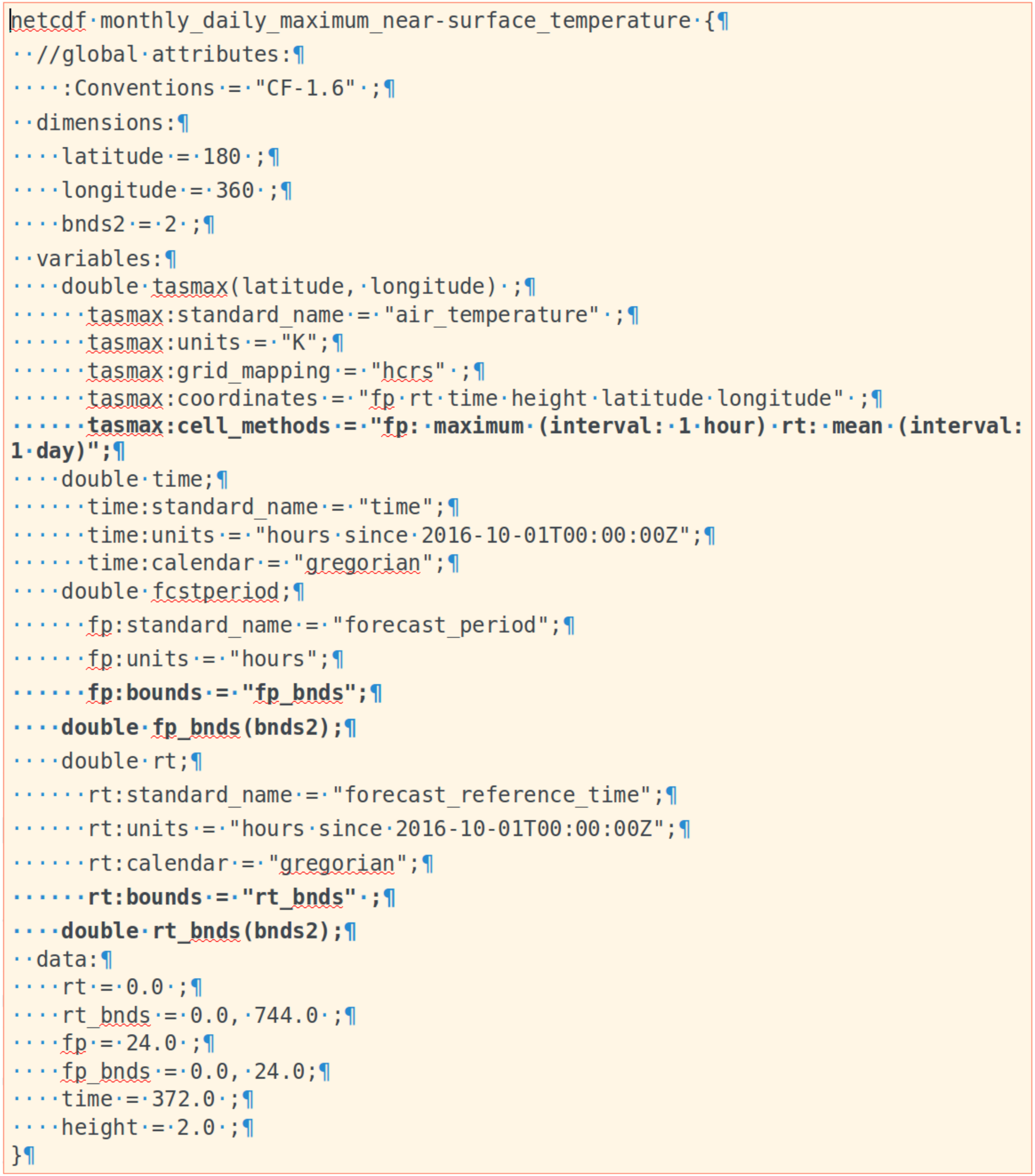
- `cell_measures` attribute represents information about the size, shape or location of the cells that cannot be deduced from the coordinates.
- `cell_methods` attribute describe the characteristic of a field that is represented by cell values:

```
cell_methods="time: mean (interval: 1 day) longitude:
maximum (interval: 1 degree_north)"
```



```
netcdf daily_maximum_near-surface_temperature {
  //global attributes:
    :Conventions = "CF-1.6" ;
  dimensions:
    latitude = 180 ;
    longitude = 360 ;
    bnds = 2 ;
  variables:
    double tasmax(latitude, longitude) ;
      tasmax:standard_name = "air_temperature" ;
      tasmax:units = "K";
      tasmax:grid_mapping = "hcrs" ;
      tasmax:coordinates = "fp rt time height latitude longitude" ;
      tasmax:cell_methods = "fp: maximum (interval: 1 hour)";
    double time;
      time:standard_name = "time";
      time:units = "hours since 2016-10-26T00:00:00Z";
      time:calendar = "gregorian";
    double fp;
      fp:standard_name = "forecast_period";
      fp:units = "hours";
      fp:bounds = "fp_bnds";
    double fp_bnds(bnds);
    double rt;
      rt:standard_name = "forecast_reference_time";
      rt:units = "hours since 2016-10-26T00:00:00Z";
      rt:calendar = "gregorian";
  data:
    reftime = 0.0 ;
    fcstperiod = 24.0 ;
    fcstperiod_bnds = 0.0, 24.0;
    time = 12.0 ;
    height = 2.0 ;
}
```

Daily maximum near-surface temperature



```
netcdf monthly_daily_maximum_near-surface_temperature {
  //global attributes:
    :Conventions = "CF-1.6" ;
  dimensions:
    latitude = 180 ;
    longitude = 360 ;
    bnds2 = 2 ;
  variables:
    double tasmax(latitude, longitude) ;
      tasmax:standard_name = "air_temperature" ;
      tasmax:units = "K";
      tasmax:grid_mapping = "hcrs" ;
      tasmax:coordinates = "fp rt time height latitude longitude" ;
      tasmax:cell_methods = "fp: maximum (interval: 1 hour) rt: mean (interval: 1 day)";
    double time;
      time:standard_name = "time";
      time:units = "hours since 2016-10-01T00:00:00Z";
      time:calendar = "gregorian";
    double fcstperiod;
      fp:standard_name = "forecast_period";
      fp:units = "hours";
      fp:bounds = "fp_bnds";
    double fp_bnds(bnds2);
    double rt;
      rt:standard_name = "forecast_reference_time";
      rt:units = "hours since 2016-10-01T00:00:00Z";
      rt:calendar = "gregorian";
      rt:bounds = "rt_bnds" ;
    double rt_bnds(bnds2);
  data:
    rt = 0.0 ;
    rt_bnds = 0.0, 744.0 ;
    fp = 24.0 ;
    fp_bnds = 0.0, 24.0;
    time = 372.0 ;
    height = 2.0 ;
}
```

Monthly mean of daily maximum near-surface temperature

Summary

The objective is to provide some **recommendations** (and examples) for the **encoding of metadata** and data in a form suitable for archiving.

The aim is to be explicit (as possible), to provide: values for file/record specific attributes (and not for overall collections), richer metadata and conventions. Description metadata and provenance it's another key aspect

The **intention** is to be minimalist to allow downstream data **re-use beyond the original intent**, product development, scientific quality control and provision of long-term preservation. Library of tools will be required.

This means that this recommendation **is not intended to provide metadata to specific project,** experiment or simulation, like attributes for data discovery, or special characteristics. Each project/experiment should define its own Data Management Plan (i.e. guide for C3S seasonal data providers).

This recommendation is **based on CF-1.6 Convention Document** and the **Standard Name Table**. The CF it's community driven. This summer next version of CF-1.7.

The **encoding reference** used is the **netCDF-classic data model**, but extension to other encoding formats should be possible. D. Hassell et al. (2017). "A CF data model and implementation". To be appear in Geosci. Model Dev.

Questions?

Antonio S. Cofiño[1], Manuel Fuentes[2], Kevin Marsh[2], Sebastien Villaume[2], Richard Mladek[2], Eduardo Penabad[3], Cedric Bergeron[2] and Baudoin Raoult[2]

[1] Dep. of Applied Maths and Computational Sciences. Univ. de Cantabria (ES)

[2] Forecast Department, Products Team. ECMWF

[3] C3S, Seasonal Forecast. ECMWF