Read and Write Data

The structure of "Classic" NetCDF files

Thanks to all contributors:

Alison Pamment, the Unidata team.

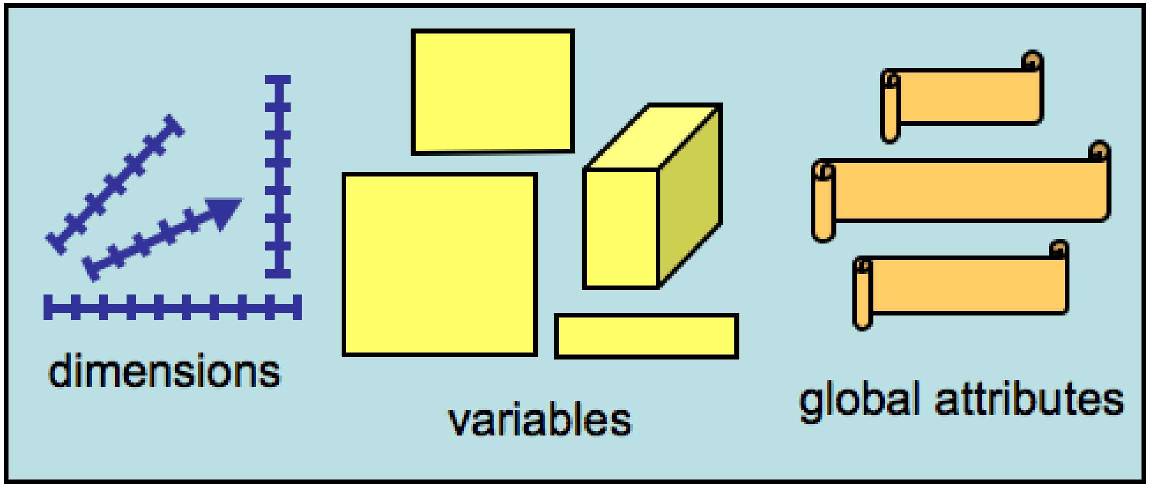
The "Classic" NetCDF Data Model

The netCDF classic data model associated with netCDF-3 is now (and will continue to be) widely used.

Understanding this simple and effective "classic" data model will be very beneficial in your use of NetCDF.

What's in a NetCDF file?

- NetCDF files are containers for Dimensions, Variables, and Global Attributes.



A netCDF file has a **path name** and possibly some **dimensions**, **variables**, **global** (file-level) **attributes**, and **data values** associated with the variables. Sometimes we refer to netCDF files more abstractly as *datasets*.

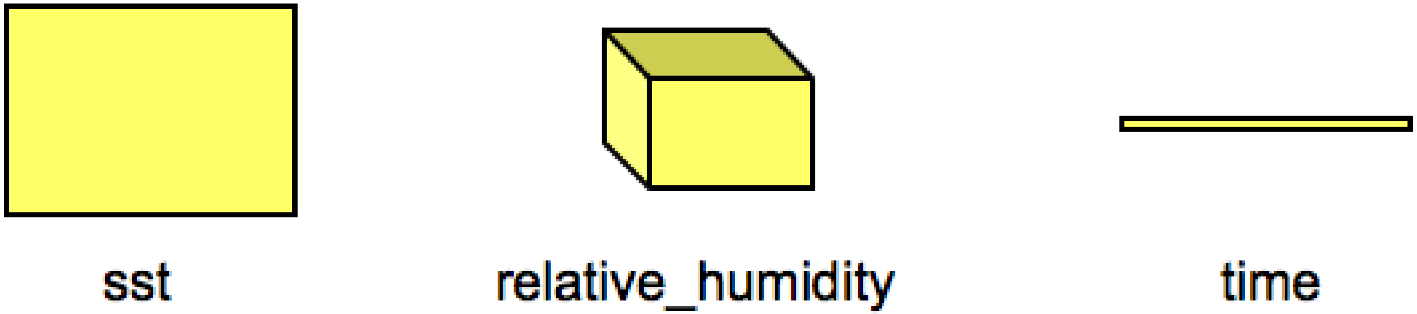
Operating on a NetCDF file

When working with a netCDF file you can:

- **Create** a new file, given its path name and whether to overwrite or not.
- **Open** an existing file for access, given dataset name and read or write intent.
- **Add** dimensions, variables, or attributes.
- **Close** a file, writing to disk if required.
- **Get** the number of dimensions, variables or global attributes.
- **Get** the unlimited dimension, if present.

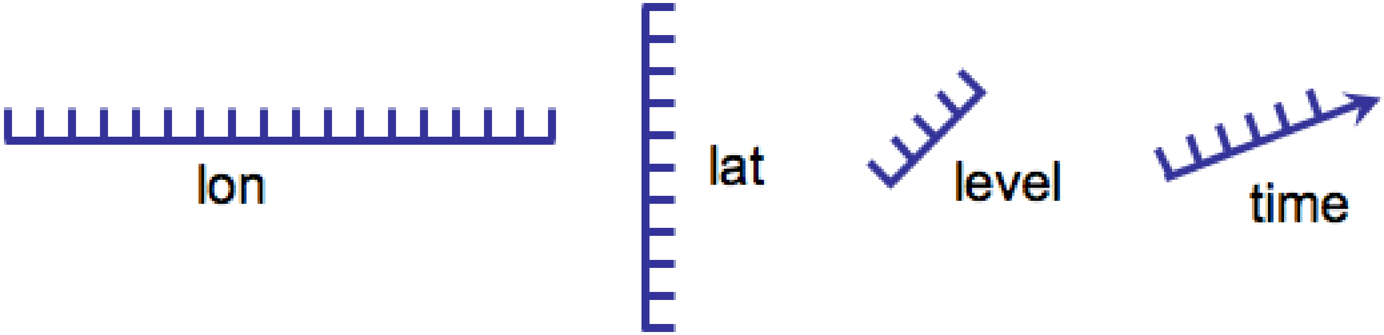
Variables

Variables hold data values. In the classic netCDF data model, a variable can hold a multidimensional array of values of the same type.



Dimensions

Dimensions are used to specify variable shapes, common grids, and coordinate systems.



A dimension has a name and a length. Dimensions are used to define the shape of one or more variables in a netCDF file.

In the classic netCDF data model, at most one dimension can have the *unlimited* length, which means variables can grow along that dimension. *Record dimension* is another term for an unlimited dimension.

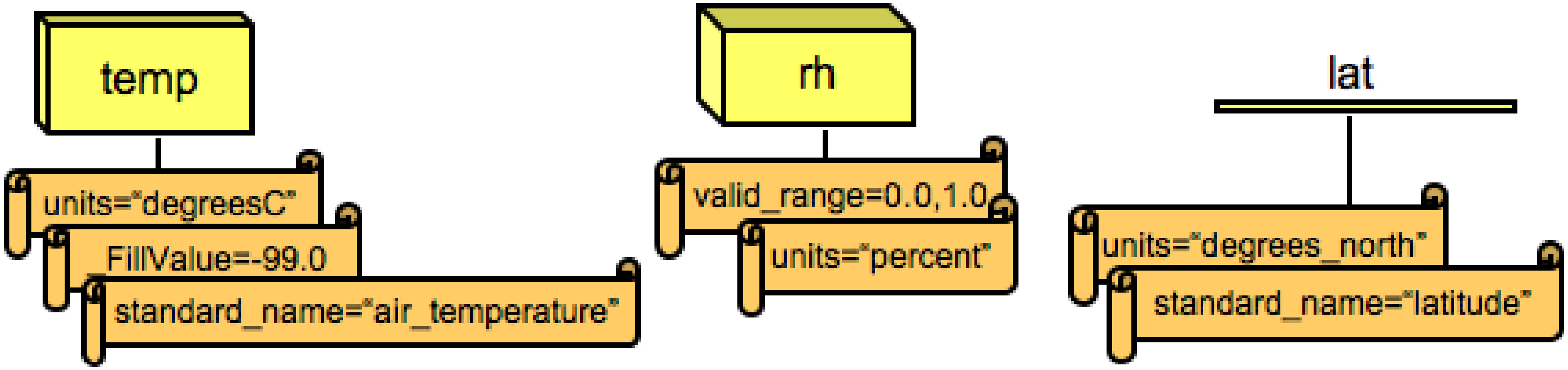
NetCDF Variables

NetCDF Variables have:

- A **type**, e.g. char (text character), byte (8 bits) or float (32 bits)
- A **shape**, specified by a list of dimensions, e.g.:
  - 1 dimension: a 1-D (vector) variable, such as time
  - 2 dimensions: a 2-D (grid or matrix) variable, such as surface_pressure
- **Attributes** (optionally) – specifying properties such as long name and units.
- **Values** – the actual data values.

Attributes

Attributes hold metadata (data about data). An attribute contains information about properties of a variable or dataset.



Attributes can be "global" (applying to the whole file) or "variable attributes" (applying only to a specified variable).

An easier way to view NetCDF: CDL

CDL (network Common Data form Language) is a human-readable notation for netCDF objects and data.

```
netcdf  example { // example of CDL notation
dimensions:
    lon = 3 ;
    lat = 8 ;
variables:
    float rh(lon, lat) ;
        rh:units = "percent" ;
        rh:long_name = "Relative humidity" ;
// global attributes
        :title = "Simple example, lacks some conventions" ;
data:
  rh =
    2, 3, 5, 7, 11, 13, 17, 19,
    23, 29, 31, 37, 41, 43, 47,
    53, 59, 61, 67, 71, 73, 79, 83, 89 ;
}
```

```
netcdf  example { // example of CDL notation
dimensions:
   lon = 3 ;
   lat = 8 ;

variables:
   float rh(lon, lat) ;
      rh:units = "percent" ;
      rh:long_name = "Relative humidity" ;

// global attributes
      :title = "Simple
conventions" ;

data:
  rh =
    2, 3, 5, 7, 11, 13
    23, 29, 31, 37, 41
    53, 59, 61, 67, 71
}
```

This example specifies a netCDF dataset with two dimensions (**lon** and **lat**), one variable (**rh**), two variable attributes (**units** and **long_name**), one global attribute (**title**), and some data values for the variable.

Acknowledgement

The material presented here was primarily taken from the Unidata NetCDF workshop notes at:

https://www.unidata.ucar.edu/software/netcdf/workshops/2012

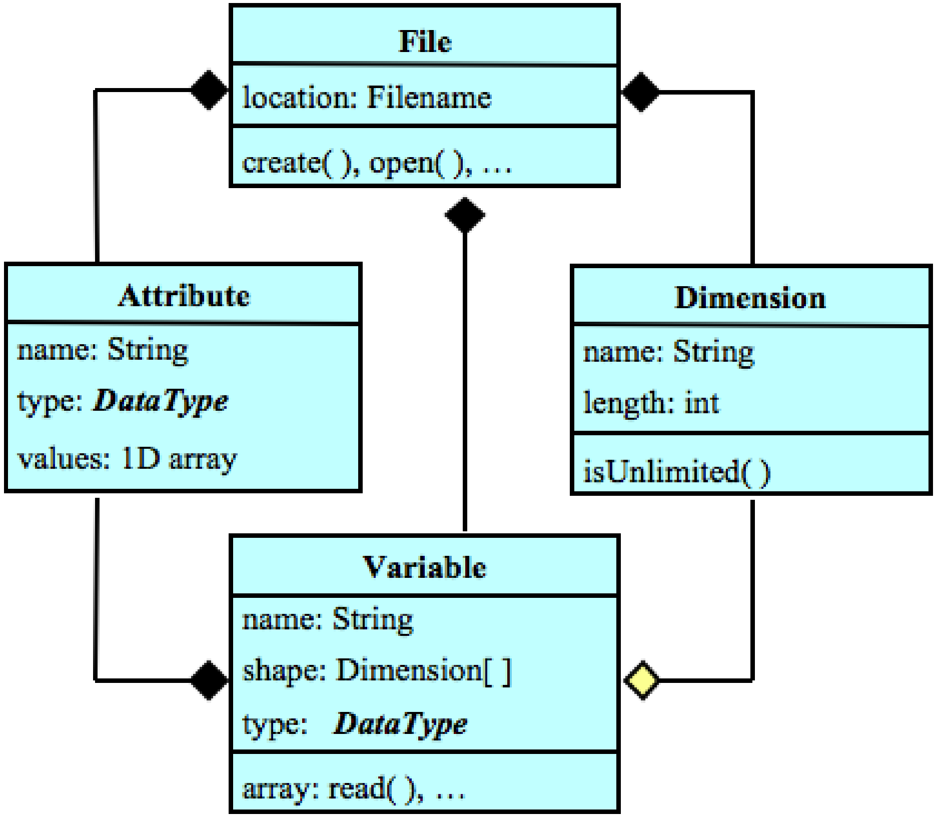
The "Classic" Data Model

The classic netCDF data model uses *dimensions*, *variables*, and *attributes*, to capture the meaning of array-oriented scientific data.

The following diagram represents the "classic" data model visually. Each box contains:

- the name of a class of objects
- characteristics of object in the class
- operations (methods) for that class of objects

The "Classic" Data Model



File

location: Filename

create( ), open( ), …

Attribute

name: String

type: ***DataType***

values: 1D array

Dimension

name: String

length: int

isUnlimited( )

Variable

name: String

shape: Dimension[ ]

type: ***DataType***

array: read( ), …

Variables and attributes have one of six primitive data types.

DataType

char
byte
short
int
float
double

A file has named variables, dimensions, and attributes. Variables also have attributes. Variables may share dimensions, indicating a common grid. One dimension may be of unlimited length.