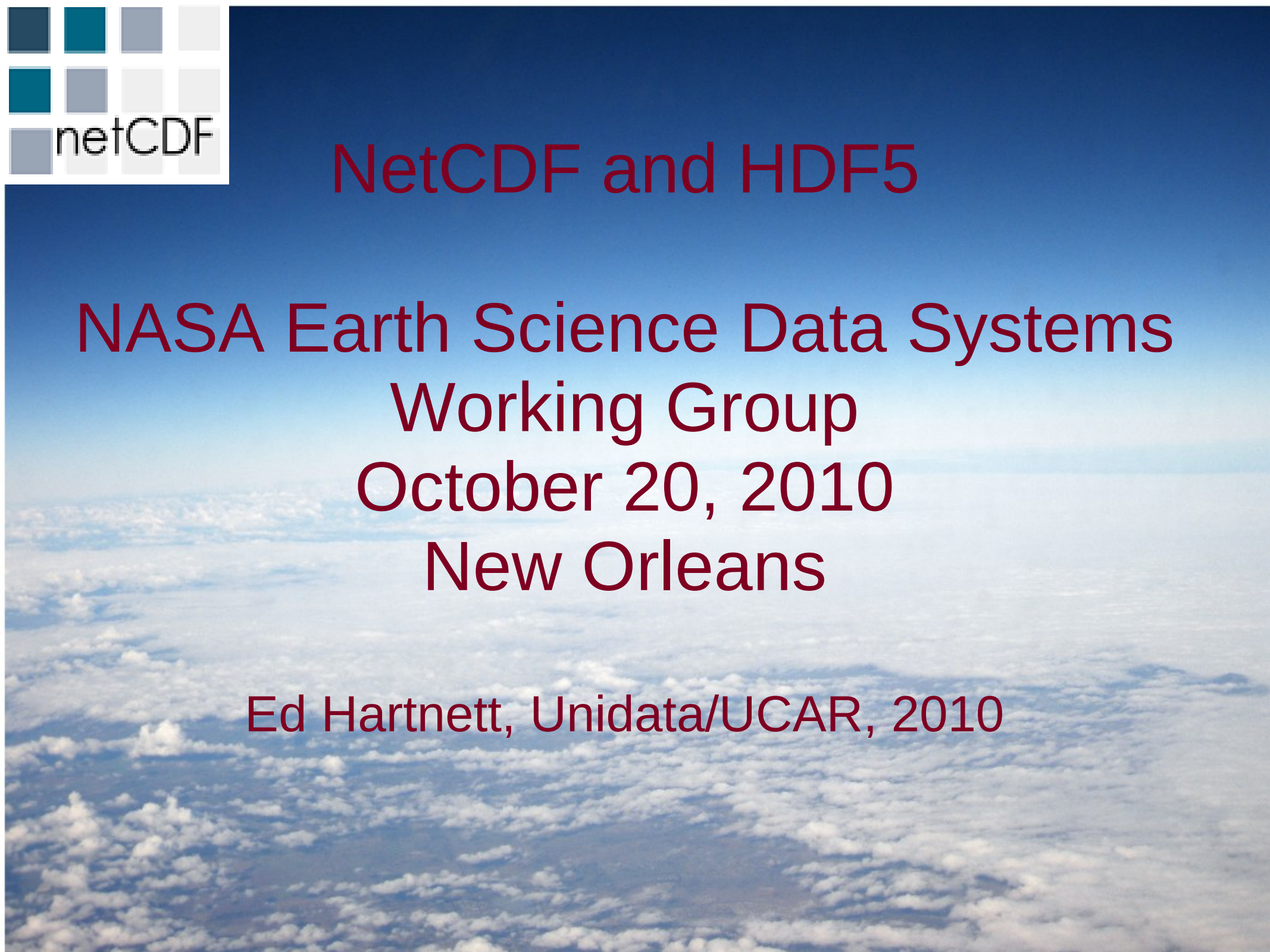


NetCDF and HDF5

NASA Earth Science Data Systems Working Group
October 20, 2010
New Orleans

Ed Hartnett, Unidata/UCAR, 2010

Unidata

- Mission: To provide the data services, tools, and cyberinfrastructure leadership that advance Earth system science, enhance educational opportunities, and broaden participation.

Unidata

Data:

Over 30 data streams provided

Data collection, cataloging, and distribution

Both push and pull technologies are used

Software:

Data Distribution: LDM

Remote Data Access: THREDDS, ADDE, and RAMADDA

Data Management: netCDF and UDUNITS

Analysis and Visualization: GEMPAK, McIDAS and IDV

GIS support via TDS (WCS, WMS) and KML and Shapefiles

User Support & Training:

Direct email support

Community mailing lists

Annual Training Workshops, Triennial Users Workshops, and Regional Workshops as needed.

Community:

Equipment Awards to universities; Seminars; Information Commons; Advocacy;

Unidata Software

- NetCDF – data format and libraries.
- NetCDF-Java/common data model – reads many data formants (HDF5, HDF4, GRIB, BUFR, many more).
- THREDDS – Data server for cataloging and serving data.
- IDV – Integrated Data Viewer
- IDD/LDM – Peer to peer data distribution.
- UDUNITS – Unit conversions.

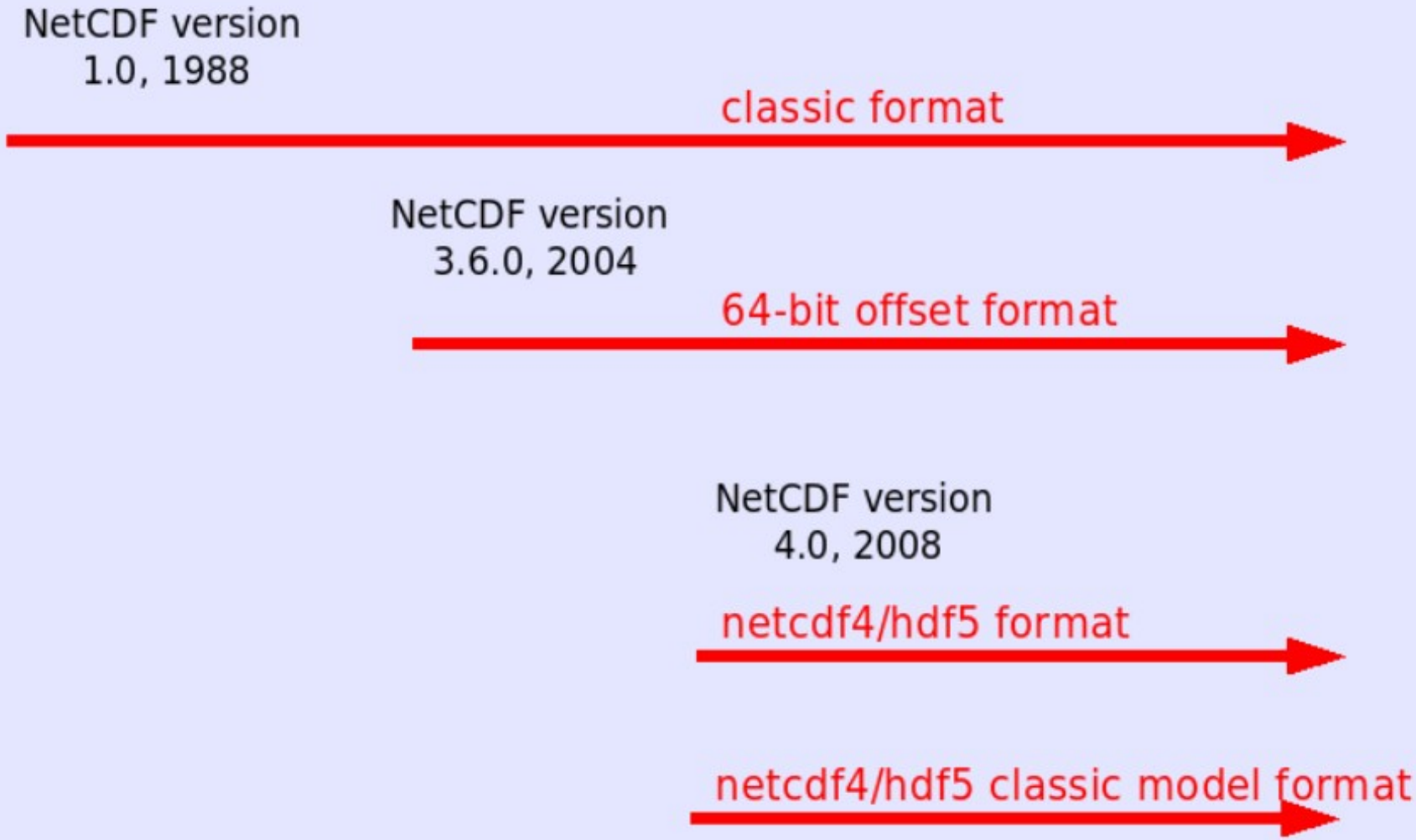
What is NetCDF?

- NetCDF is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.
- First released in 1989.
- NetCDF-4.1.1 (2010) maintains full code and data compatibility with all previous releases.

The NetCDF-4 Project

- Uses HDF5 as data storage layer.
- Also provides read-only access to some HDF4, HDF5 archives.
- Parallel I/O for high performance computing.
- Does not indicate any lack of commitment or compatibility for classic formats.

NetCDF Disk Formats



Commitment to Backward Compatibility

Because preserving access to archived data for future generations is sacrosanct:

- NetCDF-4 provides both read and write access to all earlier forms of netCDF data.
- Existing C, Fortran, and Java netCDF programs will continue to work after recompiling and relinking.
- Future versions of netCDF will continue to support both data access compatibility and API compatibility.

Who Uses NetCDF?

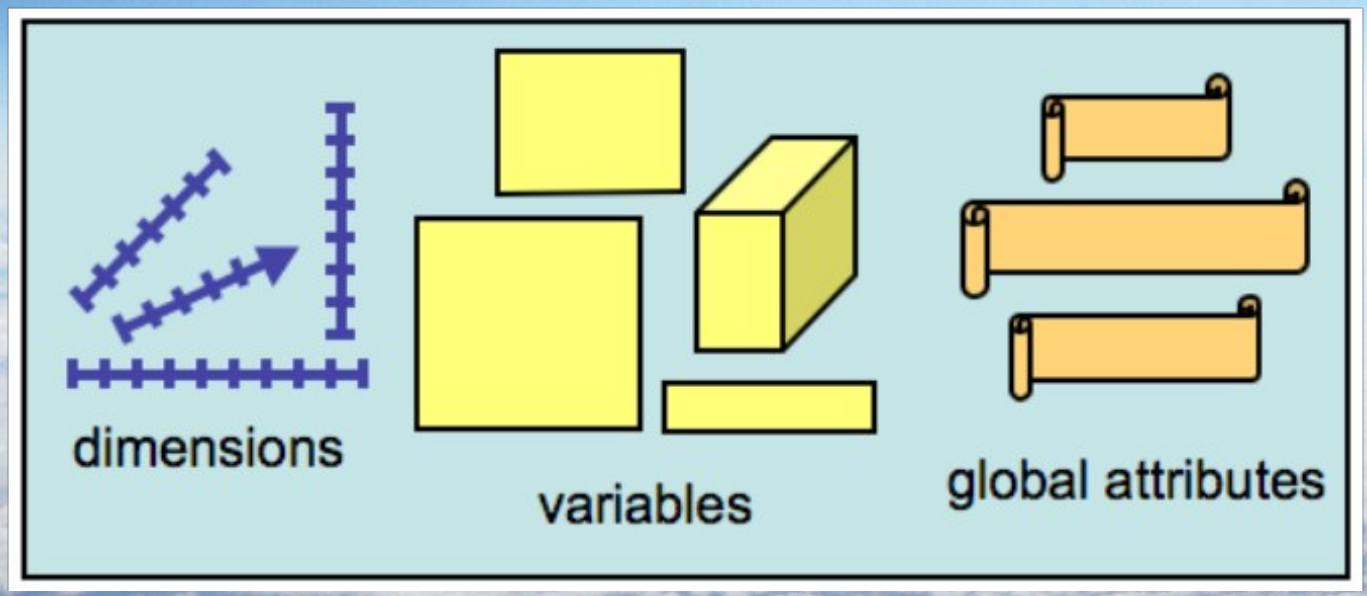
- NetCDF is widely used in University Earth Science community.
- Used for IPCC data sets.
- Used by NASA GMAO and other large data producers.

NetCDF Data Models

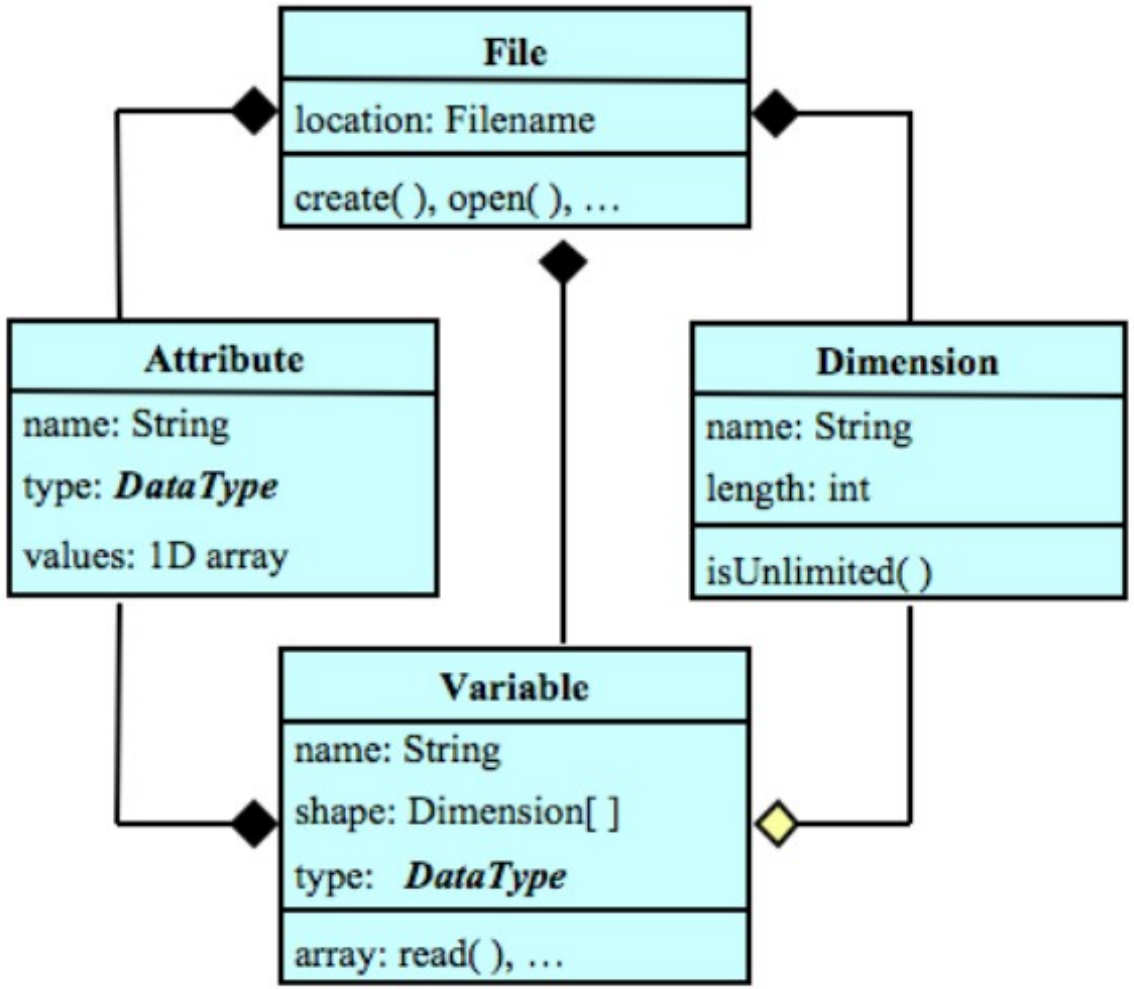
- The netCDF data model, consisting of variables, dimensions, and attributes (the classic model), has been expanded in version 4.0.
- The enhanced 4.0 model adds expandable dimensions, strings, 64-bit integers, unsigned integers, groups and user-defined types.
- The 4.0 release also adds some features that need not use the enhanced model, like compression, chunking, endianness control, checksums, parallel I/O.

NetCDF Classic Model

- Contains dimensions, variables, and attributes.



NetCDF Classic Model

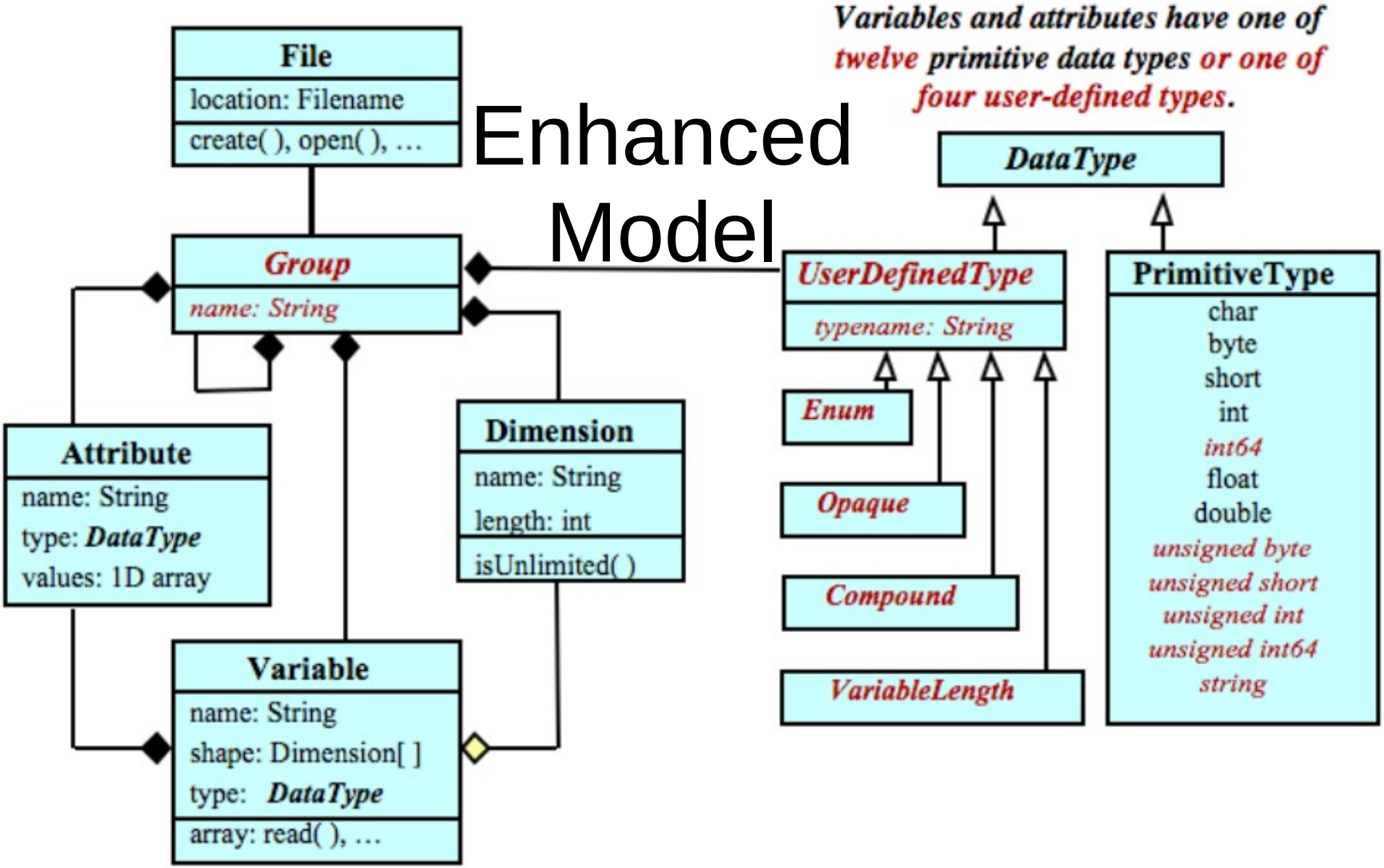


Variables and attributes have one of six primitive data types.

DataType
char
byte
short
int
float
double

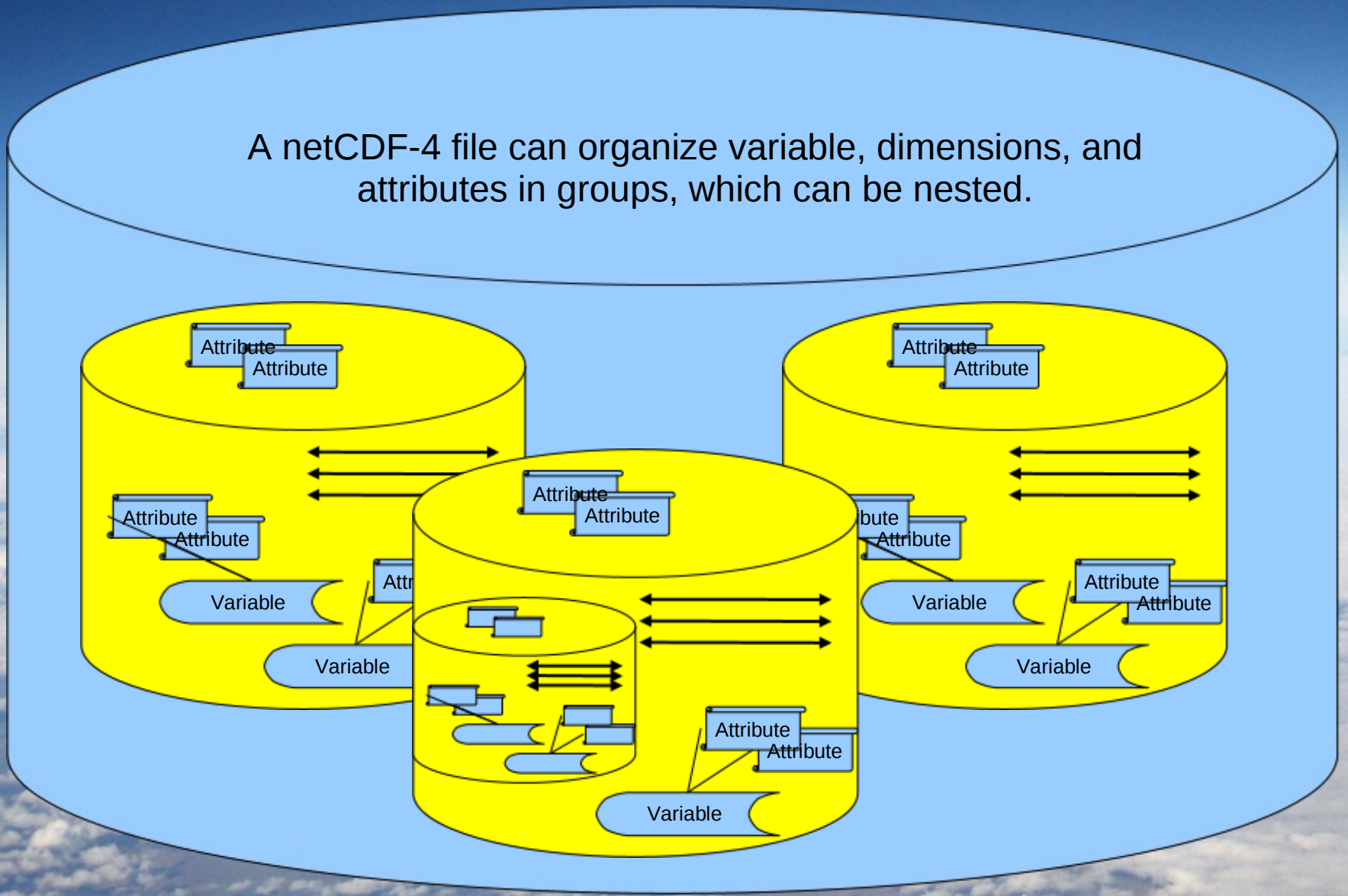
A file has named variables, dimensions, and attributes. Variables also have attributes. Variables may share dimensions, indicating a common grid. One dimension may be of unlimited length.

Enhanced Model



A file has a top-level unnamed group. Each group may contain one or more named subgroups, user-defined types, variables, dimensions, and attributes. Variables also have attributes. Variables may share dimensions, indicating a common grid. One or more dimensions may be of unlimited length.

NetCDF Enhanced Model



Reasons to Use Classic Model

- Provides compatibility with existing netCDF programs.
- Still possible to use chunking, parallel I/O, compression, endianness control.
- Simple and powerful data model.

Accessing HDF5 Data with NetCDF

- NetCDF (starting with version 4.1) provides read-only access to existing HDF5 files if they do not violate some rules:
- Must not use circular group structure.
- HDF5 reference type (and some other more obscure types) are not understood.
- Write access still only possible with netCDF-4/HDF5 files.

NetCDF APIs

- The netCDF core libraries are written in C and Java.
- Fortran 77 is “faked” when netCDF is built – actually C functions are called by Fortran 77 API.
- A C++ API also calls the C API, a new C++ API us under development to support netCDF-4 more fully.

Tools

- ncdump – ASCII or NcML dump of data file.
- ncgen – Take ASCII or NcML and create data file.
- nccopy – Copy a file, changing format, compression, chunking, etc.

Reading HDF5 with NetCDF

- Before netCDF-4.1, HDF5 files had to use creation ordering and dimension scales in order to be understood by netCDF-4.
- Starting with netCDF-4.1, read-only access is possible to HDF5 files with alphabetical ordering and no dimension scales. (Created by HDF5 1.6 perhaps.)
- HDF5 may have dimension scales for all dimensions, or for no dimensions (not for just some of them).

Accessing HDF4 Data with NetCDF

- Starting with version 4.1.1, netCDF is able to read HDF4 files created with the “Scientific Dataset” (SD) API.
- This is read-only: NetCDF can't write HDF4!
- The intention is to make netCDF software work automatically with important HDF4 scientific data collections.

Using HDF4

- You don't need to identify the file as HDF4 when opening it with netCDF, but you do have to open it read-only.
- The HDF4 SD API provides a named, shared dimension, which fits easily into the netCDF model.
- The HDF4 SD API uses other HDF4 APIs, (like vgroups) to store metadata. This can be confusing when using the HDF4 data dumping tool hdp.

Confusing: HDF4 Includes NetCDF v2 API

- A netCDF V2 API is provided with HDF4 which writes SD data files.
- This must be turned off at HDF4 install-time if netCDF and HDF4 are to be linked in the same application.
- There is no easy way to use both HDF4 with netCDF API and netCDF with HDF4 read capability in the same program.

Building NetCDF for HDF5/HDF4 Access

- This is only available for those who also build netCDF with HDF5.
- HDF4, HDF5, zlib, and other compression libraries must exist before netCDF is built.
- Build like this:

./configure –with-hdf5=/home/ed –enable-hdf4

HDF4 MODIS File ncdumped

../ncdump/ncdump -h MOD29.A2000055.0005.005.2006267200024.hdf
netcdf MOD29.A2000055.0005.005.2006267200024 {
dimensions:
Coarse_swath_lines_5km\:MOD_Swath_Sea_Ice = 406 ;
Coarse_swath_pixels_5km\:MOD_Swath_Sea_Ice = 271 ;
Along_swath_lines_1km\:MOD_Swath_Sea_Ice = 2030 ;
Cross_swath_pixels_1km\:MOD_Swath_Sea_Ice = 1354 ;
variables:
float Latitude(Coarse_swath_lines_5km\:MOD_Swath_Sea_Ice,
    Coarse_swath_pixels_5km\:MOD_Swath_Sea_Ice) ;
Latitude:long_name = "Coarse 5 km resolution latitude" ;
Latitude:units = "degrees" ;
...

Accessing HDF4-EOS Data with NetCDF

- Data can be read, but netCDF does not (yet) understand how to break down the StructMetadata attribute into useful information.

```
// global attributes:
:HDFEOSVersion = "HDFEOS_V2.9" ;
:StructMetadata.0 =
    "GROUP=SwathStructure\n\tGROUP=SWATH_1\n\t\tSwathName
    =\"MOD_Swath_Sea_Ice\"\n\t\tGROUP=Dimension\n\t\t\\tOBJEC
    T=Dimension_1\n\t\t\t\tDimensionName=\"Coarse_swath_lines_5
    km\"\n\t\t\t\tSize=406\n\t\t\tEND_OBJECT=Dimension_1\n\t\t\tOBJ
    ECT=Dimension_2\n\t\t\t\tDimensionName=\"Coarse_swath_pixel
    s_5km\"\n\t\t\t\tSize=271\n\t\t\t...
```

The OPeNDAP Client

- OPenDAP (http://www.opendap.org/) is a widely supported protocol for access to remote data
- Defined and maintained by the OPenDAP organization
- Designed to serve as intermediate format for accessing a wide variety of data sources.
- Client is now built into netCDF C library.

Using OPeNDAP Client

- In order to access DAP data sources, you need a special format URL:

http://test.opendap.org/dods/dts/test.32.X

- Location of data source and its part, where X is one of "dds", "das", or "dods"
- Constraints on what part of the data source is to be sent.

Parallel I/O with NetCDF

- Parallel I/O allows many processes to read/write netCDF data at the same time.
- Used properly, parallel I/O allows users to overcome I/O bottlenecks in high performance computing environments.
- A parallel I/O file system is required for much improvement in I/O throughput.
- NetCDF-4 can use parallel I/O with netCDF-4/HDF5 files, or netCDF classic files (with pnetcdf library).

Conventions

- Conventions are published agreements about how data of a particular type should be represented to foster interoperability.
- Most conventions use attributes.
- Use of an existing convention is highly recommended. Use the CF Conventions, if applicable.
- A netCDF file should use the global "Conventions" attribute to identify which conventions it uses.

Climate and Forecast Conventions

- The CF Conventions are becoming a widely used standard for atmospheric, ocean, and climate data.
- The NetCDF Climate and Forecast (CF) Metadata Conventions, Version 1.4, describes consensus representations for climate and forecast data using the netCDF-3 data model.

LibCF

- The NetCDF CF Library supports the creation of scientific data files conforming to the CF conventions, using the netCDF API.
- Now distributed with netCDF.

UDUNITS

- The Unidata units library, udunits, supports conversion of unit specifications between formatted and binary forms, arithmetic manipulation of unit specifications, and conversion of values between compatible scales of measurement.
- Now being distributed with netCDF.

NetCDF 4.1.2 Release

- Coming soon!
- Performance improvements: much faster file opens (factor of 200 speedup).
- Better memory handling, much better testing for leaks and memory errors in netCDF and HDF5.
- nccopy now can compress and re-chunk data.
- Refactoring of dispatch layer (invisible to user).

NetCDF Future Plans

- By “plans” we really mean “aspirations.”
- We use agile programming, with aggressive refactoring, and heavy reliance on automatic testing.

Plans: Fortran Refactor

- We plan a complete Fortran re-factor within the next year.
- Fortan 90 and Fortran 77 backward compatibility will be preserved. No user code will need to be rewritten.
- Fortan 90 compilers will be required (even for F77 API code). Fortran 77 compilers will not work with netCDF releases after the refactor.
- Fortran 90 API will be rewritten with Fortran 2003 C interoperability features. Fortran 77 API will be rewritten in terms of Fortran 90 API.

Plans: Windows Port

- Recent refactoring of netCDF architecture requires (yet another) Windows port. This is planned for the end of 2010.
- Windows ports are not too hard, but require a detailed knowledge of Microsoft's latest changes and developments of the Windows platform.
- I invite collaboration with any Windows programmer who would like to help with the Windows port.

Plans: Virtual Files

- There are some uses (including LibCF/GRIDSPEC) for disk-less netCDF files – that is, files which exist only in memory.
- I am experimenting with this now – interested users should contact me at: ed@unidata.ucar.edu

Plans: More Formats

- The NetCDF Java library can read many formats that are a mystery to the C-based library.
- Recent refactoring of the netCDF architecture makes it easier to support additional formats.
- We would like to support GRIB and BUFR next. We seek collaboration with interested users.

NetCDF Team – Russ Rew

- Vision.
- ncdump, nccopy
- classic library

NetCDF Team – John Caron

- NetCDF-Java
- Common Data Model

NetCDF Team – Ed Hartnett

- NetCDF-4
- Release engineering
- Parallel I/O
- LibCF
- Fortran libraries

NetCDF Team – Dennis Heimbigner

- Opendap client.
- New ncgen
- Some netCDF-Java

Snapshot Releases and Daily Testing

- Automatic daily test runs at Unidata ensure that our changes don't break netCDF.
- Test results available on-line at NetCDF web site.
- Daily snapshot release provided so users can get latest code, and iterate fixes with netCDF developers.

NetCDF Workshop

- Annual netCDF workshop is a good place to learn the latest developments in netCDF, and talk to netCDF developers.
- October 28-29, 2010, and swanky Mesa Lab at NCAR – great views, mountain trails, without the usual riffraff.
- Preceded by data format summit.

Support

- Send bug reports to: support-netcdf@unidata.ucar.edu
- Your support email will enter a support tracking system which will ensure that it does not get lost.

Suggestions for Satellite Data Producers

- Do not use complex structures (nested groups, enumerations, etc).
- Keep the structures "flat" if possible; data variables should be 2D.
- Define correct units for all variables.
- Use CF conventions for defining variables.
- Use a standard projection; if not possible use lat-lon-per-pixel.
- Geolocation data could be in a separate file as long as it can be accessed via NcML to make it look like it is in the file.
- Include a "time" variable that is the nominal time of the data.

CF Satellite Conventions Mailing List

The cf-satellite@unidata.ucar.edu mailing list is devoted to discussion of developing CF conventions for satellite products.

Join here:

http://www.unidata.ucar.edu/support/mailinglist/mailing-list-form.html#subscribe