NetCDF Data Model Issues

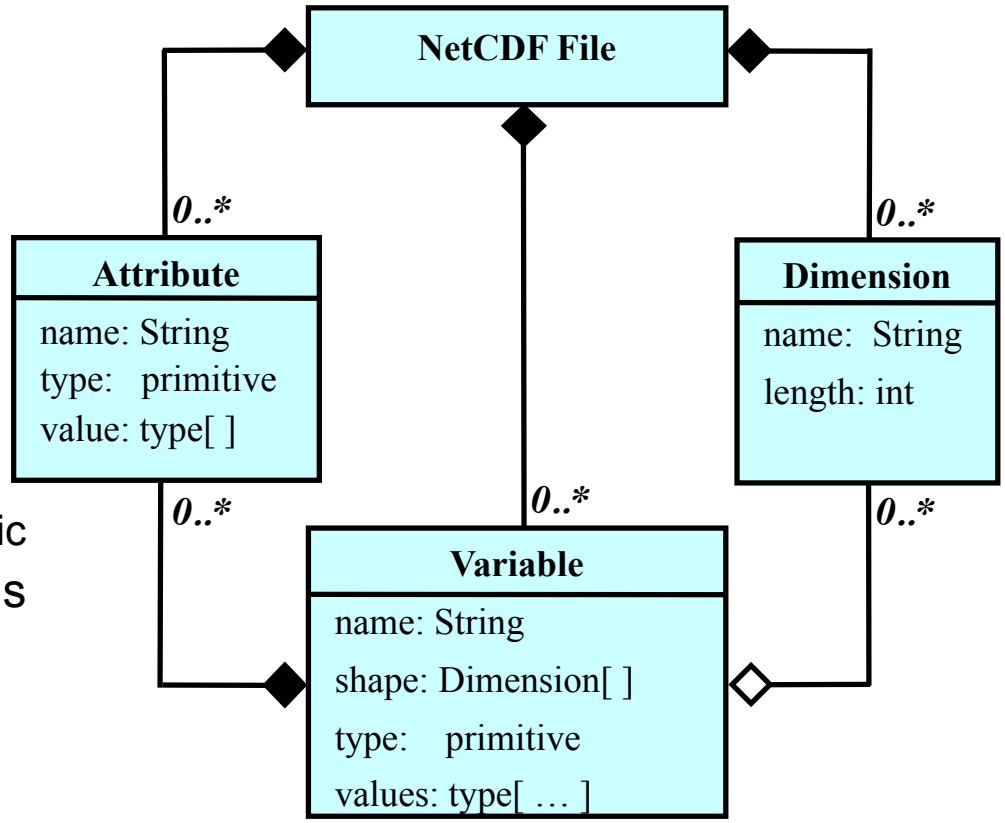
Russ Rew, UCAR Unidata

NetCDF 2010 Workshop

2010-10-25

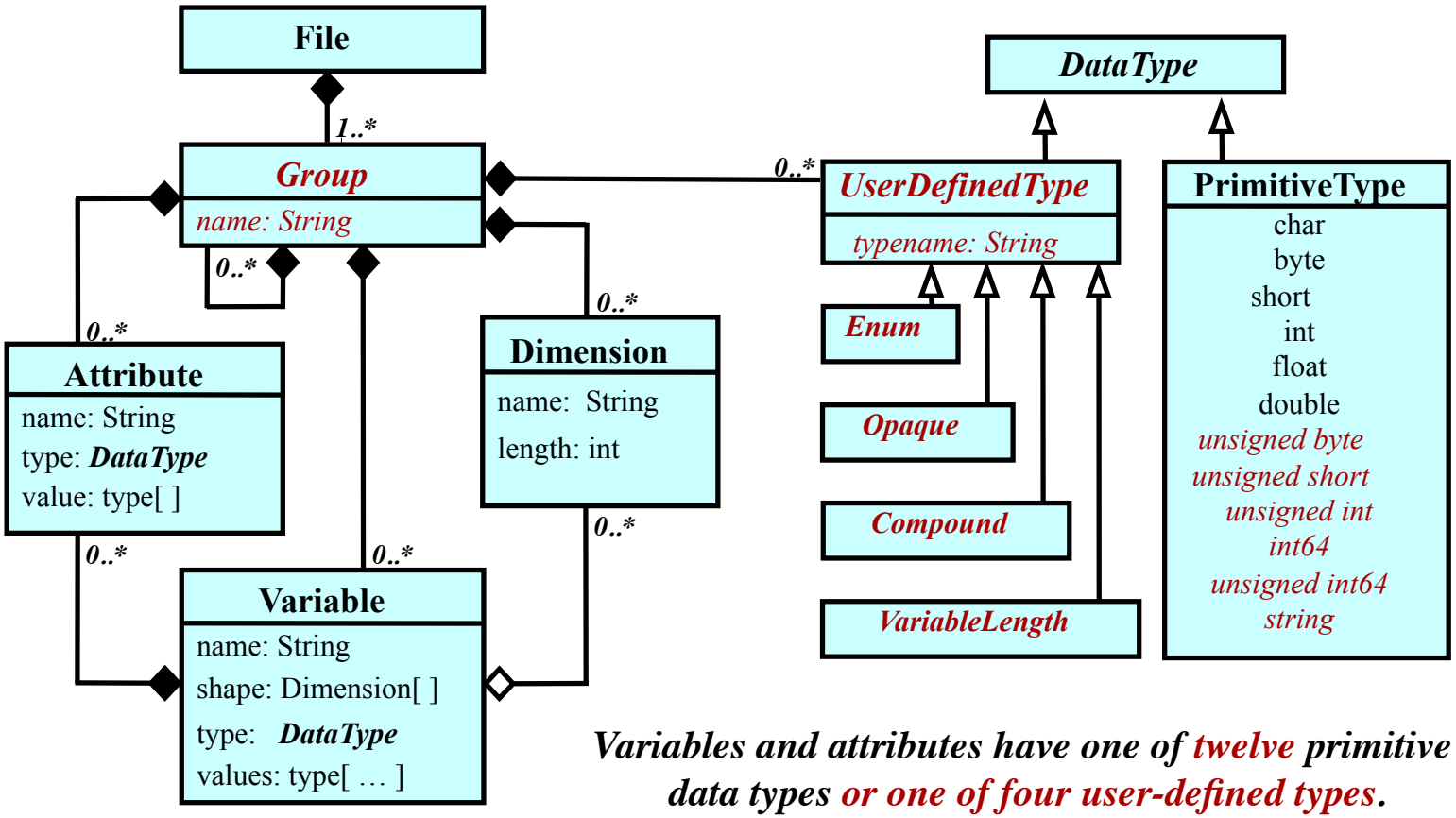
The netCDF classic data model

- A netCDF **File** has
  - **Variables**
  - **Dimensions**
  - **Attributes**
- Variables have
  - Name, shape, type, values
  - Associated attributes
- Dimensions have
  - Name, length
  - One dimension may be dynamic
- Variables may share dimensions
  - Indicates common grid
  - Scalar variables have no dimensions
- Primitive types
  - Numeric: byte. short, int, float, double
  - Character arrays for text



The netCDF-4 *enhanced* data 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.



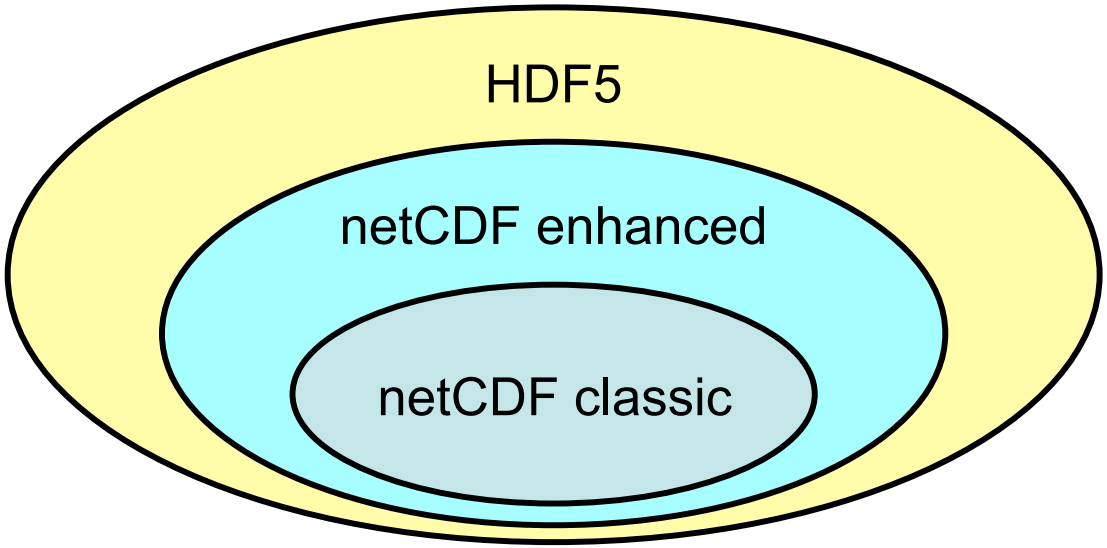
Variables and attributes have one of twelve primitive data types or one of four user-defined types.

NetCDF and HDF5 Data Models

- The netCDF classic data model: simple and flat
  - Dimensions
  - Variables
  - Attributes
- The netCDF enhanced data model added
  - More primitive types
  - Multiple unlimited dimensions
  - Hierarchical groups
  - User-defined data types
- The HDF5 data model has even more features
  - Non-hierarchical groups
  - User-defined primitive data types
  - Hard- and soft-links (providing multiple names for Groups, variables)
  - References (pointers to objects and data regions in a file)
  - Attributes attached to user-defined types

The Enhanced NetCDF Data Model

- Additions to classic netCDF data model
- Still a subset of HDF5 data model (**with shared dimensions workaround*)
- Made possible by adding a few things to HDF5 so netCDF classic data model could fit within it
- Criteria for additions: handling identified classic limitations, simplicity
- Is netCDF enhanced data model the right balance of simplicity and power?



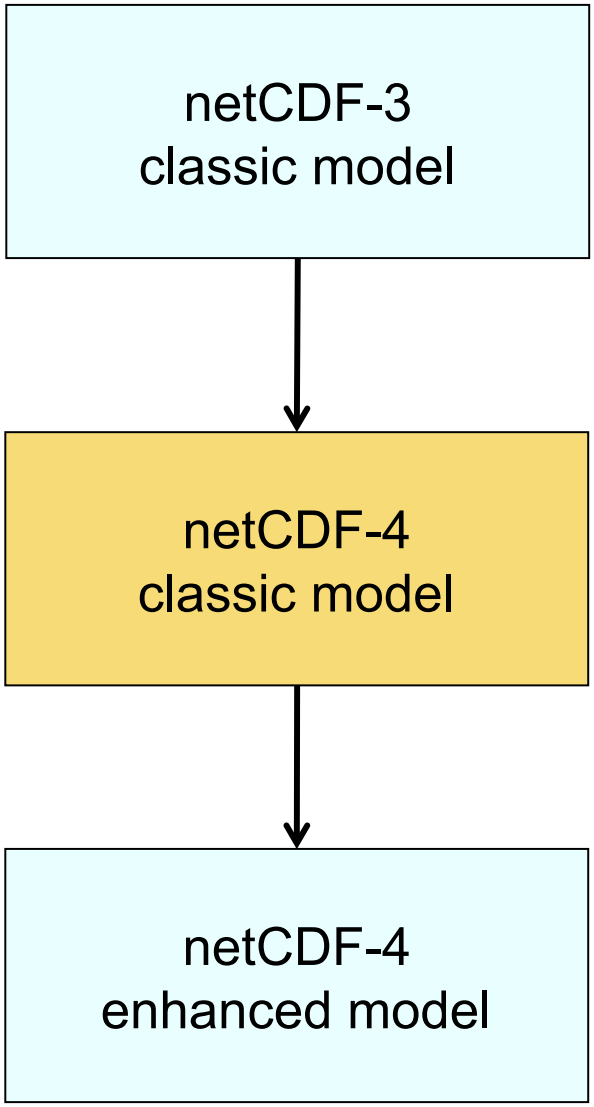
Evaluation: netCDF enhanced data model

- Strengths
  - Simpler than HDF5, with similar representational power
  - Compatible with existing data, software, conventions
  - Efficient reference implementation
  - Orthogonal features permit incremental adoption
- Limitations
  - More complex than classic data model
  - More challenging to develop general software tools
  - Comprehensive conventions still lacking
  - Not yet widely adopted

Why Is Adoption of Enhanced Data Model Slow?

- Combination of classic data model with netCDF-4 adequate for many uses
  - Only requires relinking instead of modifying software
  - Performance benefits: compression, multi-dimensional chunking, larger variables
- Data using enhanced data model features not common yet
- Best practices and conventions not yet developed for enhanced data model
- NetCDF-4 enhanced data model not endorsed as a standard yet
- Developer perceptions
  - Must upgrade to features of enhanced model all at once
  - Handling potentially infinite number of user-defined types is difficult

NetCDF-4 classic-model: a transitional format



- Compatible with existing applications
- Simplest data model and API

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

- Not compatible with some many existing applications
- Enhanced data model and API more complex and powerful

Experience so far: Adapting to netCDF-4

Features	NCAR's NCL	NetCDF Operators (NCO)	netCDF-Java	Python API	CCFE's C++ API for netCDF-4	ncdump ncgen nccopy
Performance features: compression, chunking, …	yes	yes	read-only	yes	yes	yes
New primitive types	yes	yes	read-only	yes	yes	yes
Multiple unlimited dimensions	read-only	read-only	read-only	yes	yes	yes
Groups	not yet	not yet	read-only	yes	yes	yes
Compound types, variable-length types	not yet	not yet	read-only	flat	yes	yes

Experience developing nccopy utility

- Shows developing generic netCDF-4 software is practical
- Provides measure of difficulty of developing for enhanced data model
  - Classic data model: 500 lines of C
  - Enhanced data model: 1000 lines of C
- Shows usefulness of higher-level APIs for tool developers
  - Iterator APIs for uniform data access in nccopy
  - Comparing two values of a user-defined type for equality
  - Getting group IDs of all descendents of a group

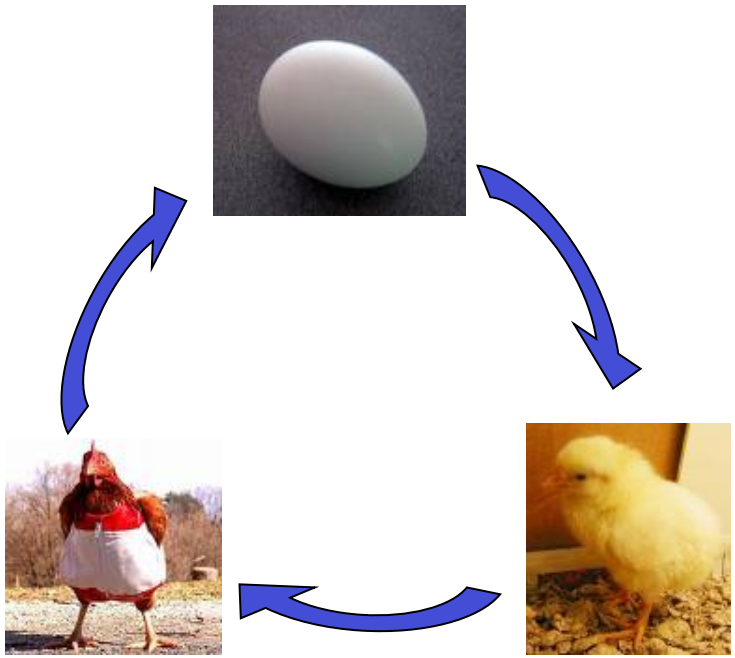
Recommendation for Developers

- Add support for netCDF enhanced data model features incrementally
  - new primitive types: unsigned numeric types and strings
  - opaque types (easy, no nesting)
  - enumeration types (easy, no nesting)
  - nested Groups (simple recursion or Group iterator)
  - compound types with only primitive members
  - variable-length arrays of primitives
  - compound types with members of user-defined type
  - variable-length arrays of user-defined types

Benefits and Costs of Adapting Tools to Enhanced Model

- Benefits:
  - NetCDF-4's enhanced data model adds representational power
  - Data providers can use more natural representation of complex data semantics
  - More natural conventions become possible
  - Generality provides improved interoperability with other formats, with access to more types of data through netCDF-like APIs
- Costs:
  - Development resources, opportunity costs, risk of adding functionality not proven useful yet

Game of chicken: Who goes first?



- Data producers
  - Waiting until netCDF enhanced data model features are supported by more software, development of conventions
- Developers
  - Waiting for netCDF data that requires enhanced model and for development of conventions
- Convention creators
  - Waiting for data providers and software developers to identify needs for new conventions based on usage experience
- Result: “chicken-and-egg logjam”
  - *Delays effective use of advances in scientific data models for large and complex collections*

Concluding remarks

- Serious use of netCDF-4 enhanced data model just beginning
- Future adjustments to model, if any, will be made by addition, not modification or deletion of existing features
- Will one data model “win” the hearts and minds of data producers, developers, users?
  - netCDF-4 classic model, netCDF-4 enhanced model, HDF5 model, or something else?