P12R.5

ISSUES IN DESIGNING A NEW RADAR DATA EXCHANGE FORMAT

Dennis Flanigan[1], Joseph VanAndel,John Caron, Wen-Chau Lee

1. INTRODUCTION

Well defined and adhered to data formats are vital in the exchange and processing of scientific data. For over a decade, the National Center for Atmospheric Research (NCAR) has distributed meteorological radar data in data files defined by the Dorade format. This format was useful in its time, but now does not meet the needs of today's radar instruments. For example, Dorade can not accurately represent data from a radar instrument that uses more then one frequency.

As a first step to define a new data format for radar data, this paper presents a number of use cases and requirements. It will also look at several data access libraries that may be used to define the new format. The requirements discussed here should viewed as the first version of requirements, not the final version of the requirements.

The preliminary name of the new format will be Exchange Format. This paper will use that name.

2. USE CASES

Use cases define how a system is envisioned to be used. The exercise of developing use cases determines what are its most import design attributes. This is particularly important in the case when those design attributes come into conflict with each other. For example: use cases may show that the design of a data system should favor access speed over data compression.

There are two popular ways to document use cases: in Universal Modeling Language (UML) diagrams or just written descriptions. Written descriptions are used here.

2.1 *Distribution of radar data from a field experiment.*

Radar data collected during a field experiment will be distributed to the scientific investigators in the new Exchange Format. The investigators will be able to view the data using a new version of radar display and editing program solo. Investigators will also be be able to examine the data using data visualization programs such as IDL or MatLab.

2.2 *Archive of original field experiment data*

The Exchange Format will be able to be used for archiving of the original data. Corrections applied to the data for distribution will be stored in a separate meta-data file that could also be archived.

2.3 *Product generation and application and algorithm development.*

The Exchange Format will be able to encompass the original data as well as new data fields for products and algorithm output. Examples of product include particle identification and rainfall rates.

3. REQUIREMENTS

Simply stated, requirements are what the Exchange Format has to provide for. They are in many instances derived from the use cases. The requirements listed here are considered high level requirements, leaving out much of the details.

3.1 *Byte Order Independent*

The processing of the byte ordering will handled by the Exchange Format interface. Whether the data is ordered in Big Endian or Little Endian fashion will not make any difference to the application software using the Exchange Format interface.

[1] Corresponding author address: Dennis F. Flanigan Jr., National Center for Atmospheric Research, Earth Observing Laboratory, P.O. 3000, Box Boulder, Colorado 80307; e-mail: dennisf@ucar.edu

What this means: It does not matter what kind of compute wrote the original data. A Mac should be able to read data created on Windows machine.

3.2 *Multisensor*

The Exchange Format will be able to represent data from systems that use more then one sensing device. Examples: airborne radar system that use both forward and aft facing antennas; Ground based radar systems that use two transmitting frequencies.

The implications of this requirement are that all data from a single platform does not have to share the same geometry or attributes. In the case of the dual frequency radars data from the first frequency could have different gate spacing then the data from the second frequency.

3.3 *Corrections*

Corrections and edits made to the original data will be optionally recorded in the Exchange Format. Distributed data itself will always have the corrections already applied. The level of detail about what corrections were made may not enable a user to "undo" those corrections and recreate the original data.

Either the corrections are applied before the data is distributed or the corrections are distributed with the original data for the recipient of the data to apply. The second option adds complexity for the recipient, so the first option is the better path to take.

3.4 *New Fields*

The Exchange Format will accept the addition of new data fields to already existing Exchange Format data files. Examples of new data fields are products derived from other fields such as particle ID and rainfall rate. The new fields do not have to share the same geometry as the original source fields.

This requirement provides the same functionality found in previous data formats.

3.5 *Translation*

The Exchange Format will be able to fully represent radar data recored in these radar formats:

- Dorade
- WSR88D Archive II
- Universal Format

There are several note worthy implications of this requirement. First, the Exchange Format will need to handle radar volume scans found in WSR88D Archive II data. Second, the Exchange Format will need represent data values encoded in floating point as well as integer binary formats as is done in the case of the Dorade Format.

Also, and this is similar to the multisensor requirement, data has to be accurately represented even though all data fields may not share the same spatial attributes. This can be the case in sweeps of certain WSR88D scanning strategies where reflectivity data has different gate spacing then that of the velocity and spectral width data.

4. DATA ACCESS LIBRARIES

A definition of a new data format need not be specification of how data appears byte by byte in a data file. The availability of general purpose data access libraries can be used to define a format using simple high level abstractions defined by the library.

The major advantage to using a data access library is that the complicated low level tasks of how data is represented in files is done by the library. These tasks include byte swapping, data compression and random access. These functions are already designed, coded and tested in these libraries Ultimately, using a data access library leads to faster software development and robust software applications.

Given that using a data access library is desirable, the questions then becomes which library to use and how should the format be define using that library.

4.1 *netcdf-3*

The current version (version 3) of the netcdf libary is one the most popular data access libraries. It was developed and is supported by the UniData Program.

The netcdf paradigm is ideal for defining the Exchange Format. It is simple, popular and well tested data access library.

The problems with using it is that it is not technically advance enough to meet the requirements of the Exchange Format. For example, netcdf only allows for only one array dimension to be unlimited. This would be an issue for multisensor platforms.

4.2 *HDF5*

Hierarchical Data Format 5 (HDF5) is a data access library from the National Center for Supercomputing Applications (NCSA). It is a very powerfull data access libraray that can fulfill all of the Exchange Format requirements. It already is supported by many applications such Matlab and IDL.

Issues with HDF5 is that the interface can be complicated. It is also not supported by Unidata Applications such as IDV.

4.3 *netcdf-4*

Netcdf 4 is an implementation of the netcdf interface on top of HDF5 library. The advantage to this approach is that through a simple netcdf interface users get access to some of the powerful features of HDF5 such as support for large files and multiply unlimited dimensions.

Netcdf is supported by IDV and other unidata applications. While there exists software for netcdf access within Matlab and IDL, it is not directly support by those two applications.