On the suitability of BUFR and GRIB for archiving data

John Caron
Unidata/UCAR
December 2011

Abstract

A major weakness of the BUFR and GRIB formats is their dependence on external tables. For GRIB, one must have the correct tables in order to understand the meaning of the data. For BUFR, the correct tables are needed to understand the data but also to even parse the binary file. An important design flaw in these formats is that there is no way for a reader to know for certain which tables the writer used. Compounding this are implementation practices which make this problem occur more often than might be expected. This makes BUFR and GRIB not suitable as long-term storage formats. In order to correct this problem, 1) there must be a foolproof way for software to know which tables were used when the data was written, and 2) there must be an authoritative registry of tables.

Introduction

The World Meteorological Organization (WMO) has made GRIB (GRIdded Binary) its standard format for gridded model output and BUFR (Binary Universal Form for the Representation of meteorological data) the standard format to encode meteorological observational data. Both are binary, portable, table-driven formats, highly specialized for meteorological data, and used operationally for weather forecasting within national centers and to transmit data between centers. Both were designed in the 1980s by members of the WMO, with an emphasis on data compression and controlled vocabularies.

GRIB and BUFR are major improvements over previous practices. GRIB replaces files that are application, operating system, and hardware dependent, with a well-defined format that can be read by any program on any machine. GRIB is especially good at compressing floating point data with a variety of algorithms, including JPEG-2000 wavelet compression. BUFR replaces older ASCII character-based and position-dependent encodings, also with an emphasis on efficient floating point storage.

In addition to these strengths, GRIB and BUFR have serious weaknesses that compromise their suitability as long-term archival formats:

1. There is no sure way for software to know which tables were used when the data was written
2. There is no reference software implementation, no conformance checking tools, nor an effective user community to answer questions or clarify ambiguities in the specification.

This paper will not attempt to describe GRIB and BUFR in detail, but only describe these problems in order to stimulate discussion on remedies.

Problems with Tables in Table-driven Formats

The WMO calls the GRIB and BUFR formats "Table-driven Code Forms". Data is encoded in discrete, self-contained *messages* suitable for real time transmission, which are decoded on the fly or collected into files. The meaning of the data is stored in tables that are maintained separately from the message. These tables comprise a controlled, shared vocabulary for meteorological data, which are the result of many years of hard work by the WMO, and are as much the real content of the GRIB and BUFR specifications as the actual file format itself.

Maintaining centralized, controlled vocabularies for many organizations, numerical models, instruments and observational types is a major challenge. Such tables must be constantly growing and changing to match ongoing developments in the science. Changes must be carefully managed and versioned to track incompatibility. In GRIB-1, a single byte represents the version number for all tables. GRIB-2 and BUFR (editions 2 and greater) have another byte for the local table version. In principle, this might be seen as adequate for table versioning, but in practice many other factors complicate the task of obtaining correct tables.

Non-Authoritative WMO tables

The WMO publishes the standard tables in Word and PDF format, neither of which are machine readable. (Recently the WMO has started to correct this problem, publishing XML and CSV versions for GRIB-2 and BUFR, but not GRIB-1.) Software that writes GRIB and BUFR therefore have their own versions of the standard tables, in various formats, with various lineages. Of the many versions of the WMO tables I have examined, no two agreed exactly, and although in most cases the differences are minor capitalization or punctuation, in some cases there are difference in units, parameter names, and (for BUFR) bit lengths. They appear to be maintained by hand, as to be expected when tables are not machine readable. The tables I have examined come from freely available software from national centers, and from personal requests to data writers for their tables. I believe they represent the actual practice of table maintenance.

The process of adding new entries to WMO standard tables can lead to incorrect entries. Member organizations propose new entries, which are assigned preliminary ids. Messages using these preliminary ids are sometimes generated before the ids are finalized. The ids sometimes are changed before the version becomes final. Even more difficult is that there have also been typographical errors in the published WMO standard tables.

The result is that there can be differences in the names of the standard parameters used by different decoding software and centers. More troubling is that there are also differences in the units of standard parameters in these packages. Do these represent parameters that were not used by these centers, and so can be ignored? Did the center use the correct units but misrepresent them in their tables? Was the unit later corrected in the table but the incorrect unit used when writing the files? Unless the original writing software and tables were stored, it is difficult now, and will become impossible in the future, to answer these questions.

For example, for GRIB-1, parameter 61 ('Total precipitation') and 90 ('Water runoff') have units of 'kg/m2', in all tables I have seen except for center 34 (Tokyo), which uses “mm/day”. Another example is parameter 37 ('Montgomery stream function') which has units of='m2/s2' except in ECMWF’s GRIBEX software tables where the units are listed as 'm2/s'.

The result of this confusion is that, even assuming that the data provider was not intending to override the standard WMO parameters, they may have inadvertently used an incorrect parameter or unit in their encoding. While mistakes can and do happen in all software and formats, the nature of table-driven formats and the lack of systematic and authoritative table capture makes GRIB and BUFR especially vulnerable and difficult to recognize after the fact, except for experts who know what the data means and what the values should be, and can therefore recognize problems.

Standard vs. local tables

GRIB-1 allocates a single byte for parameter ids, with the range 0-127 reserved for WMO standard parameters, and 128-254 for local parameters. These limits were quickly reached once GRIB-1 became widely used. Different centers reacted to this problem in differing ways. It appears that larger centers such as ECMWF and NCEP adopted internally consistent practices for dealing with versioning. Other centers did not adopt any policy at all, did not version their tables, and did not maintain a record of which tables were used in the past. This may be because personnel were focused on making changes to their data and software to keep their operational systems working, and not concerned with the problems of data archival.

There is no central repository for local tables for GRIB or BUFR. No center that I know publishes their tables in an authoritative, centralized location in machine readable form. Many publish their tables in HTML, which requires human intervention to parse. Some centers make their encoding and decoding software available, with tables that may or may not be up-to-date with respect to the operational systems that produce the data. Common practice is for decoding software to make their best effort to identify the correct table, but put the onus on the user to know which tables should really be used. In one case that I know, the original data producer, using a standard software package with incorrect tables, misidentified a parameter from their own dataset.

There is also confusion as to whether a local table can override entries in the “reserved for WMO” section of the parameter table, and how this possibility interacts with the version number. The WMO Manual on Codes describes the version byte in GRIB-1:

> *4: GRIB tables Version No. (currently 3 for international exchange) – Version numbers 128–254 are reserved for local use*

Thus a version number > 127 tells the reading software to use a special local table, but doesn’t explicitly say what is in the local table. ECMWF local tables include entries in the entire range from 1-254. NCEP local tables with versions > 127 use only entries in the range 128-254. However, NCEP also publishes a “Table 2” that has the WMO standard table in 0-127, and local parameters in 128-254, which is to be used when the message version equals 2. ECMWF publishes standard WMO tables only in the range 0-

127. Other centers use variants on these practices, for example, the Japanese (center 34) appear to use the full range of entries, but not use versions > 127.

The question of overriding WMO standard parameters is therefore less clear than it might seem. They are sometimes overridden in local tables, and the local table is sometimes mixed with the standard table. The question is not should they be, but are they in practice? If they are not overridden, then when the param id < 128 and table version < 128, the reading software can assume that the standard WMO table was used. If they are overridden, then the user must contact the data provider to get the correct tables, no matter what version or parameter range is used, adding considerable effort to the process of reading GRIB and BUFR.

The situation with GRIB-2 tables is better than GRIB-1 for the following reasons:

- There is not a crisis (yet) in running out of parameter numbers, since there are additional bytes for category and discipline.
- A separate byte for the local table version allows the master and the local tables to evolve separately.
- There is more clarity that local tables may not override master table entries: *"Local tables shall define those parts of the Master table which are reserved for local use "* (from WMO Manual on Codes)
- As of 2011, the WMO is publishing the standard tables in a machine readable form.

Nonetheless, there remain issues when reading GRIB-2 messages:

- Prior to 2011, there were no machine-readable version of GRIB-2 tables from WMO, only Word and PDF documents. So the same practice of maintaining local, non-authoritative versions of the tables, often specific to a center or software package, remains in place.
- Unversioned local tables are used at some centers. It's not clear if the data producers understand the need for versioning, especially at the smaller centers.
- The process of adding entries to the standard table can take a long time; provisional entries are changed or moved; there is pressure on operational systems to "get something working". So the question of which table was actually used in a GRIB-2 message whose provenance in unclear remains.
- Some local tables may still be overriding entries in the master table. For example Korea (center 40) has for discipline 0, category 19, parameter 0 : "VISIBILITY_AT_1.5M" with units of "%", whereas the WMO standard table has "Visibility" with units "m".

BUFR has some advantages over GRIB:

- Tables can be encoded in BUFR records and stored in the same file as the BUFR data records.
- As of 2011, the WMO is publishing the standard tables in a machine readable form.

However, most of the problems described above also apply to BUFR, with non-canonical versions of the WMO tables, typographical errors in the official tables, a proliferation of local tables, and a relaxed

attitude toward versioning. More complex is the fact that, unlike GRIB, the packing parameters for BUFR data are stored in external tables, and so the ability to even parse a BUFR message depends on having the correct table. It is common for centers to override the standard packing parameters in order to store more decimal places of accuracy in their data than the standard tables permit. If these overrides are not known to the reading software, then incorrect data values are computed, with no indication that there is a problem ("silent failure"). This can be partly mitigated if the reading software counts the expected bits of data and compares that to the actual data count and so can flag problem BUFR records.

Incorrect Versions

In principle with correct table version information and correct tables, the mismatch between data providers and data readers would not occur. In practice, the table version numbers are often encoded incorrectly in the message. This is especially true in BUFR, which currently has sixteen versions of their tables, many in simultaneous use. One often sees standard parameters that were not added until a later version than the one encoded in the message. Until BUFR version 14, the tables were backwards-compatible, which likely fostered a casual attitude towards encoding the correct version.

Local versioning is sometimes not done, that is, local tables are changed in incompatible ways, but the version is not changed. As previously mentioned, this is likely due to a focus on operational needs and an incomplete understanding of the requirements for archiving data. Typically the center itself does not store the historical tables and so cannot correctly decode their own data.

Another common practice is for a message to use another center's tables, and so the center and subcenter ids in the message must be set to a different center than the "originating/generating center". Besides misrepresenting the source of the data, this also may create other incorrect metadata, such as the generating process id which depends on the correct center id for interpretation.

Lack of reference library / user community

There is no reference software or official conformance testing tools for GRIB or BUFR. The complexity of the formats prevents all but the most ambitious centers from creating their own encoding or decoding software. Much of the existing, non-reference software relies on legacy code, often in Fortran, which is not a modern language amenable to good encapsulation or other software engineering techniques.

Without reference software or conformance testing, interoperability is difficult to test. Major centers appear to use "locally consistent" practices, and test their data against their own software. Examining the tables that are released with these packages gives you an idea of which data they have probably tested against. It appears that centers test reading the data that they produced, but not a broad range from other centers. In short, there may be subtle problems if one tries to use software from center A to read datasets written from center B, especially with BUFR, due to the complexity of the encoding.

The semantics of the data and metadata is a much harder issue to get right than the actual format. It is not uncommon for groups to miscode their data due to incomplete understanding of the specification. An active user group which can help resolve ambiguities is needed.

Current Solutions

Storing BUFR tables with the BUFR data in an archive file is currently best practice. Encoding the tables in BUFR format itself is one possible choice; other formats are possible as long as the reading software understands whatever convention is being used. Therefore this solution could also be used for GRIB, without necessarily having to encode GRIB tables in the GRIB format.

There are some problems with this approach, however. The main one is that the burden of getting the correct tables into the archive file has now been shifted to whoever creates the file. But really that task must be the job of the data writer itself, but in some cases the archive file is not written by the data writer. For example, GRIB and BUFR messages are transmitted routinely on WMO's Global Telecommunications System (GTS) and also on Unidata's Internet Data Distribution (IDD) message passing systems. In these cases, typically each GRIB or BUFR record is a separate message with an identification header used for routing. This makes these systems suitable for low-latency real-time data movement. But storing this data into a file is done by the receiver, not the sender. Obviously one can think of any number of conventions that could solve this problem, but none of them are as robust as storing an unambiguous reference to the tables directly in the GRIB or BUFR messages.

Proposed Solutions

The following are proposed steps to resolve these problems:

1. The WMO or its delegate should establish a web service for registering tables. Authorized users can submit tables to the service, which are stored permanently. The service returns a unique hashcode, (e.g. the 16-byte MD5 checksum) for the table to the user.

   The hashcode for the table is encoded into the BUFR/GRIB message when it is written. This may require a new version of the encoding, but could be retrofitted into the "Local Use" section of both GRIB and BUFR.

   Anyone can submit a hashcode to the web service and retrieve the table associated with it.

2. The WMO or its delegate should create reference software that can be used to validate that a BUFR/GRIB message is well-formed, and parses the message and applies the registered tables, returning a result that can be used to validate other software.

   This could also be a web service, in addition to being an open-source library. The reference software can be written in any language and need not be high performance.

3. An international GRIB and BUFR users group should be established which can share knowledge in an informal way, where users can ask questions and get timely answers.

Conclusion

Whether the cause is design flaws or user mistakes, the reality is that it is not possible for unsupervised software to correctly determine the correct tables for GRIB and BUFR decoding. Instead, a human must personally contact each center to find out which tables were used. This doesn't scale to the tasks of

searching or processing large, heterogeneous collections of data. It essentially allows only experts to reliably process the data, and often only the data providers themselves can read their data with complete assurance that the metadata (especially the parameter name and units) is correct, or (for BUFR) that the data values are correct.

This is a serious problem for operational data processing, but it's a disaster for the long-term archival of data. As time passes, the software that wrote the data becomes unusable and then unavailable, and the people that understand the data and might spot incorrect tables retire.

The current situation is unacceptable both for the tremendous waste of time for current users, and the loss of reliable archives for future users.