Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2017-154-RC2, 2017
© Author(s) 2017. This work is distributed under
the Creative Commons Attribution 4.0 License.

Geoscientific Model Development Discussions

Interactive comment on "A CF data model and implementation" *by* David Hassell et al.

B. E. Eaton (Referee)

eaton@ucar.edu

Received and published: 17 August 2017

The paper contains complete and concise descriptions of the NetCDF data model and the CF metadata conventions. It presents a generalized data model for CF and describes how this data model represents the CF metadata. The data model is compared with three others, and finally an implementation in python is presented.

In what follows the CF metadata conventions are referred to as CF-netCDF, and a dataset which conforms to those conventions as a CF-netCDF dataset.

This paper is very well written and is a significant contribution to the processing of CF-netCDF datasets. The presentation of cf-python as an implementation of the CF data model is on its own standing sufficient for me to recommend publication. However my point of view concerning how the CF data model relates to CF-netCDF differs in several respects from that expressed in the paper. I have made comments below that,

if addressed, I feel would help to clarify this important relationship.

I don't agree with the paper's contention that CF-netCDF lacks an explicit data model and that such a data model is necessary for processing a CF-netCDF dataset (L39, L42). In the introduction of Nativi et al. (2008) CF-netCDF is called out as an example of a "community standard data model". And later in that paper figure 3 presents the CF-netCDF data model. This suggests that claiming the new CF data model provides an explicit data model for CF-netCDF is not focussing on the most important difference between the two. In fact CF-netCDF "identifies the elements of the dataset and their scientific intent, and describes how they are related to one another and to the real or model world from which the data were derived" (L37). The difference is that CF-netCDF does this at the file level, hence the close connection to the elements of the netCDF data model are necessary. On the other hand, the CF data model describes the same features, but does so at a more general level and without referring to the netCDF data model.

L46 states that the CF data model enables CF-netCDF to be presented in a manner that's easier to understand. I think this is valid. But I don't see how "adherence to the data model will ensure the production of CF-compliant datasets". It is a correct mapping of the elements of the CF data model to the elements of the CF-netCDF data model that ensures production of CF-compliant datasets. If an application or library does this successfully, then users of the application or library API will produce CF-compliant datasets.

I feel that Figure 1 is misleading at best. To say that CF-netCDF allows multiple interpretations implies that the data can be interpreted in different ways. This would imply that the elements of CF-netCDF are not well defined. If there are ambiguous elements of CF-netCDF then it is not the responsibility of the CF data model to rectify this. The problems must be corrected in CF-netCDF.

This leads to what I consider to be the most important contribution made by this work.

The cf-python library is presented as an implementation of the CF data model, but in fact it could just as well be considered a reference implementation of CF-netCDF. This to my knowledge has not been previously accomplished, and is an extremely valuable contribution to the community. Having a reference implementation is a powerful tool to help in verifying that the CF-netCDF data model is well defined. It also has great potential to be used as a testbed which could facilitate future development of CF-netCDF.

At L529 readers not familliar with data models are encouraged to move on. I would suggest removing section 5 as it doesn't really add to the description of the CF data model. The Unidata CDM and the OGC CF-netCDF standards are both concerned with mapping parts of CF-netCDF to the relevent ISO standards. Since that is not a goal of the CF data model the comparison of various data model elements in this section seems extraneous.

Similary I would suggest that appendix B on data compression in CF-netCDF is extraneous information that is easily found in the CF Convention document.

If space is freed up by the above deletions, I would find it more useful to have a summary of the cf-python API. The paper does a nice job of illustrating how the library faithfully represents all features of an existing CF-netCDF file, but doesn't say anything about how the API would be used to create a new CF-netCDF file.

Finally, one issue I've found with cf-python (v2.0.3) is that it encounters an error when attempting to open a CF-netCDF dataset which contains only coordinate variables. The error message is "RuntimeError: No fields found from 1 files". This seems to be related to the CF data model's treatment of the field construct as a container for all other constructs. Having a file contain only domain information is useful in practice, particularly in the context of interpolating fields from one domain to another.

Minor points:

L119: CF uses the term "coordinate variable" exactly as it was defined by the NUG (as

per section 1.2 in the CF convention document). The motivation for the term "auxiliary coordinate variable" was to avoid the need for the unwieldy phrase "coordinate variable in the NUG sense", i.e., auxiliary coordinate variables contain coordinate data, but are not coordinate variables.

L143: There is nothing in CF that says a file can't just contain coordinate variables, i.e., it's possible that N=0 and M>0. So it's not necessary that M<=N.

L149, L154: t is not a coordinate variable of temp. It's a scalar coordinate variable. There is no t dimension.

L175: I think it's misleading to describe auxiliary coordinate variables as providing "additional or alternative" coordinate information in the sense that that description makes the information sound optional. I would say that the most important use of auxiliary coordinate variables is to provide *required* coordinate information, as discussed starting in line 182.

Figure 8: I'm not sure what this diagram adds. L346 states it could be interpreted as a data model for CF, but I don't see how. For example there is no way from the diagram to know that a coordinate variable is a one dimensional array with a dimension name that matches the variable name and contains strictly monotonic data with no missing values. Figure 3. from Nativi et al. (2008) does a better job of expressing the CF-netCDF data model, though it too is incomplete.

L366: Note that the "units" attribute is not optional in most cases.

Figure 9: Note that "Auxiliary" is misspelled in the «construct» box. Also in Figure 10.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-154, 2017.