Mapping Geometry to the data model

Geometry Container scalar variable - doesn’t need to go anywhere in the data model

Coordinates fit in the existing green boxes, leaving out details of, e.g., contiguous ragged arrays, to the specific implementation.Polygons are just types of bounds, which already are defined on the data model coordinate constructs.

Data model doesn’t currently specify how to store ancillary information necessary for properly interpreting coordinates, e.g., am_i_a_polygon_hole. A similar case to this is the definition of climatological time bounds (section 7.4 in CF). Is this an omission in the data model? Could, perhaps, be easily resolved by adding (to the definitive descriptive text) that “coordinate constructs contain properties to define how their bounds are to be interpreted”.

Not all coordinate will have the same number of vertices (cf. bounds). Is this an issue for the data model? Following the lead of DSG ragged arrays, the answers would be “no”, so long as you’re happy with padding your bounds array with missing data …..

Chris Little -- Longer term strategy is to separate data model into semantics of topology and container. In present implementation, the edges are blurred. See W3C data cube.

Next Steps for Data Model

How to represent corridors

Generalize to plume

As data get bigger, subsetting becomes more important

Respond to paper reviews and get the paper published.

Bring data model up to date with CF-1.7 (no problems envisaged - all recent changes appear straight forward …..?)

Where to put the data model in the CF conventions. Appendix? Commitment to keep it current? Require that future changes consider the data model.

It seems difficult to imagine how the data model might fair with new features without detailed proposals on the table, probably because the issues may not be sufficiently well defined otherwise?

Ontologize the data model -- need someone to lead this effort (Simon Cox?)

- Prior work in this space https://w3id.org/seas/ClimateAndForecastOntology by **Maxime Lefrançois**
- **The netCDF-LD working group have defined an ontology for Binary Array data models but that is more about the containers data model rather than CF per se (potentially complementary) - http://binary-array-ld.net/_latest?classView=true. Some examples on the ncld website http://tinyurl.com/netcdf-ld**

Geometry Proposal

Having separate categories for polygons and multipolygons is flawed -- Chris Little. SVG is example where they've already solved this problem. See how graphics software has solved this.
Chris's types:

- Polylines (multilines)
- Polygon sets (multipolygons)
- not paths (way of connecting points with splines, etc., or straight lines)

Geometries are currently tied to data variables. May want to rethink that in future revisions.

Shapefile polygon types are deciphered on the fly.
Geodatabase polygons are flagged with type. Topology wrt neighbors also stored. Prototype for topology in netCDF is in the works.
GeoPackage - stores vector or raster in SQLite.

Connectivity is an interesting case. The data model assumes that connectivity is defined from coordinate/bounds values, but this may not be the case for polygons? I don't know if connectivity on UGRID is just nice

Kevin Butler (Esri) - Implemented early version of geometry work in a couple of release of their software. Interested in geometries themselves, not data-first-and-then-geometry. Concerned with topological relationships. Internally look at polygons, polygons with holes, or multipart polygons differently. Performance of the implementation in the software is very good.

Next Steps for Geometry

Iterate on trac/GitHub - Chris's comments must be addressed