

A data model of the CF metadata conventions

**David Hassell¹, Jonathan Gregory^{1,2}, Jon Blower³, Bryan Lawrence¹, Karl Taylor⁴
+ the CF community**

¹National Centre for Atmospheric Science (NCAS), University of Reading, UK

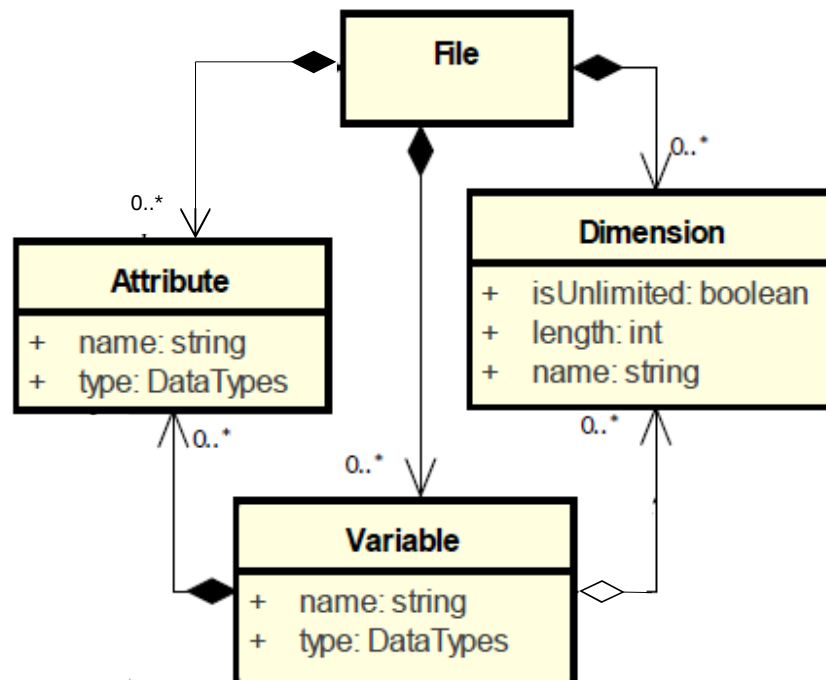
²Met Office Hadley Centre, U.K.

³Institute for Environmental Analytics, University of Reading, U.K.

⁴PCMDI, Lawrence Livermore National Laboratory, U.S.

- Evolved from work by Jonathan Gregory that was subsequently discussed in various CF trac tickets
 - #68 , #88 , #95 , #107
- The CF data model has been written up and is currently *under review* and openly available as a GMD discussion paper
 - www.geosci-model-dev-discuss.net/gmd-2017-154
 - key elements of the CF conventions are described and how they are encoded in netCDF files
 - the proposed CF data model
 - the CF data model is compared with other data models
 - a software implementation

The Unidata netCDF classic data model

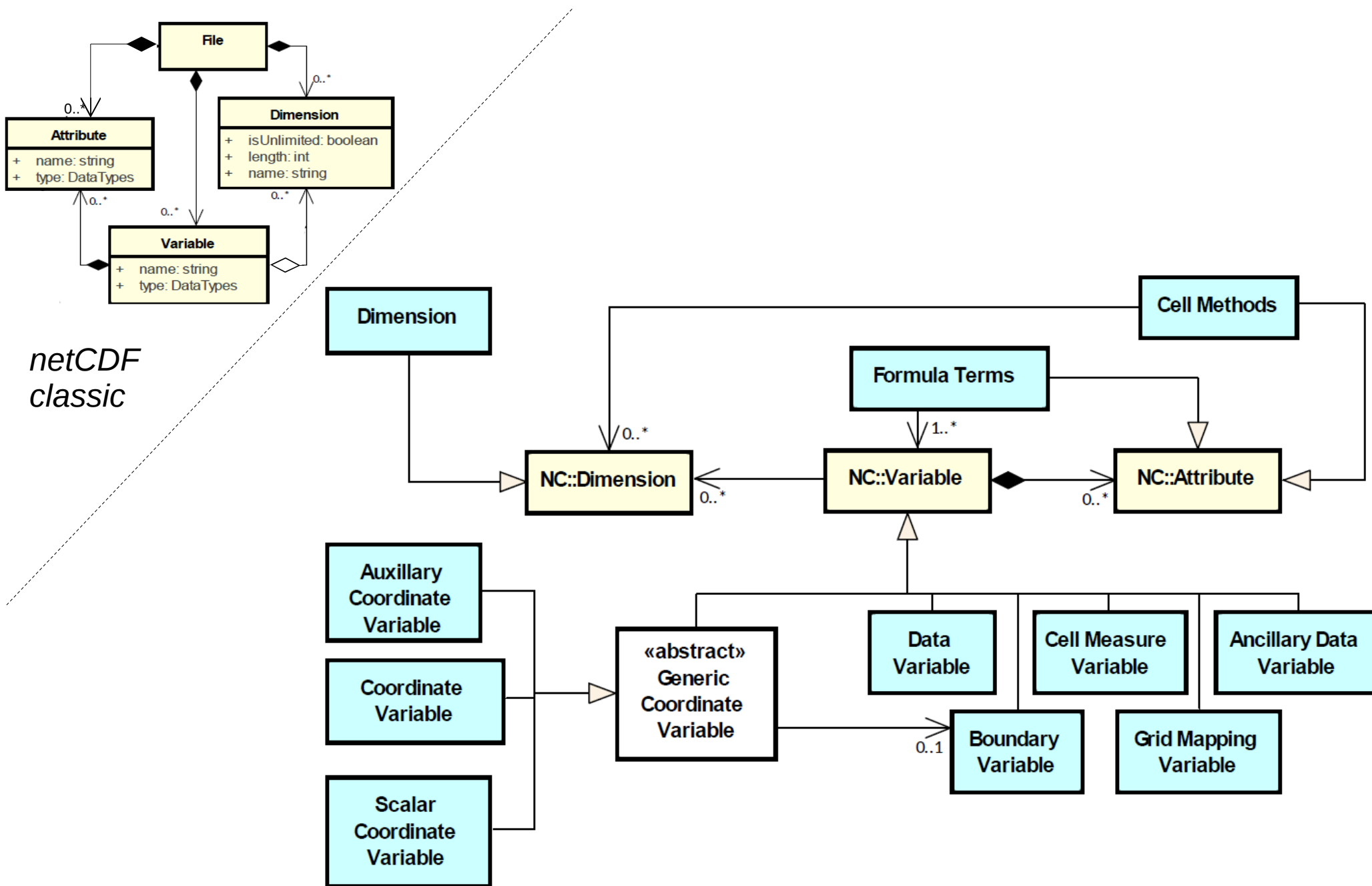




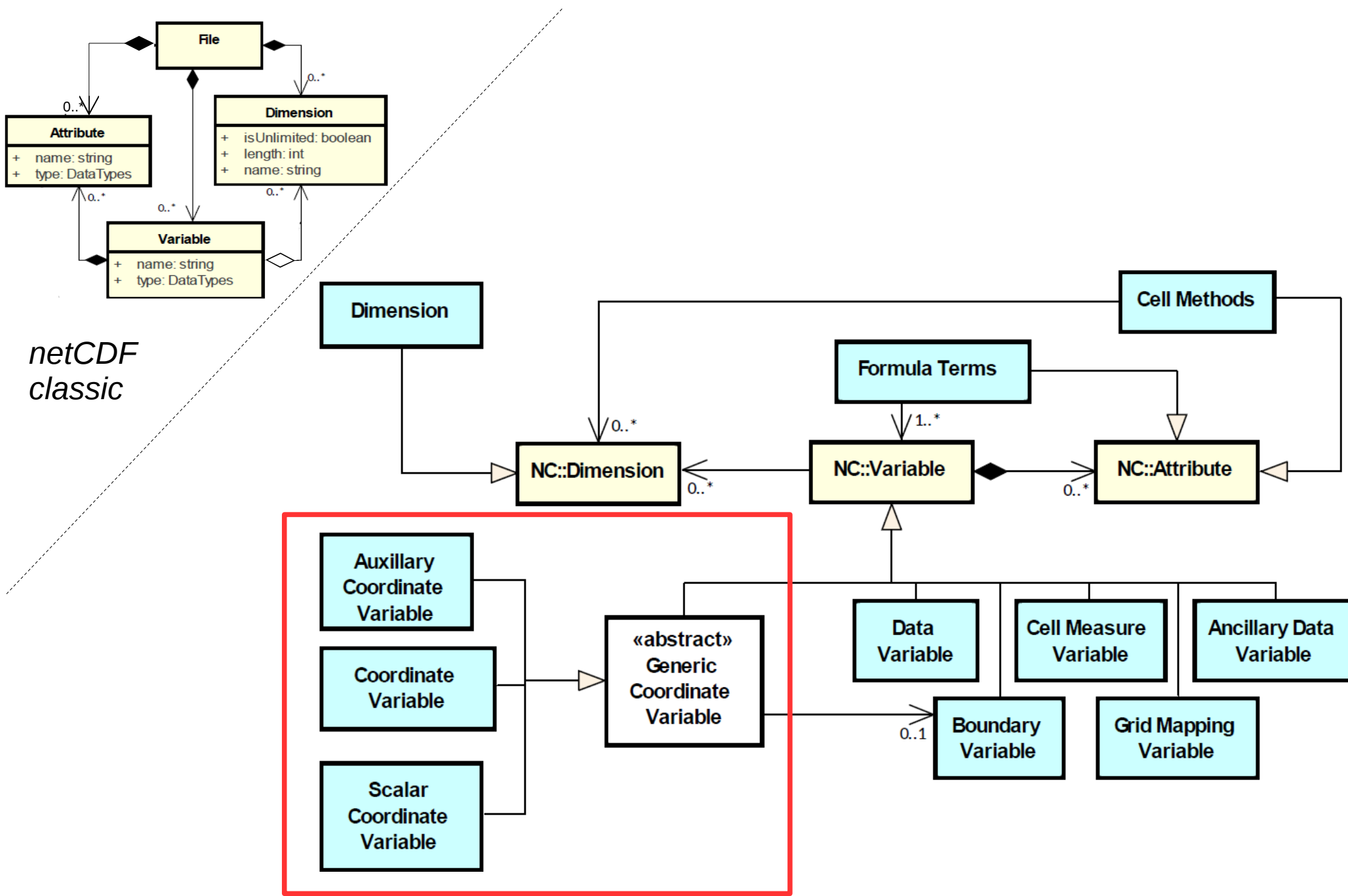
- To achieve a better understanding the CF
- To write better software
- To create better enhancements to the CF

- The data model should be for CF-1.6
- The data model should be what CF *is* rather than what CF ought to be
- The data model should be composed of a minimal set of elements that are sufficient for accommodating all aspects of the CF
- The data model should not introduce additional elements not presently needed or used by CF
- The data model should be independent of any encoding

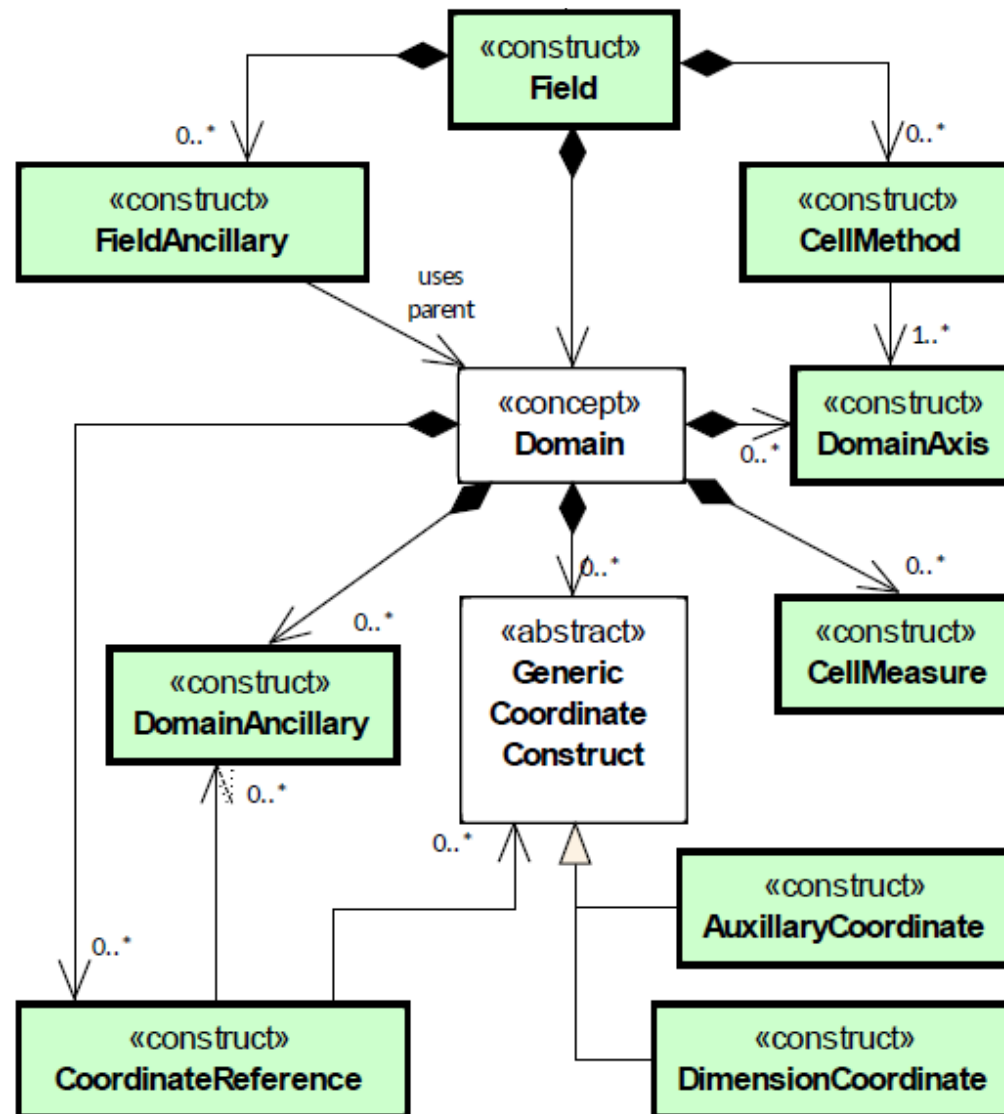
Elements of CF-netCDF



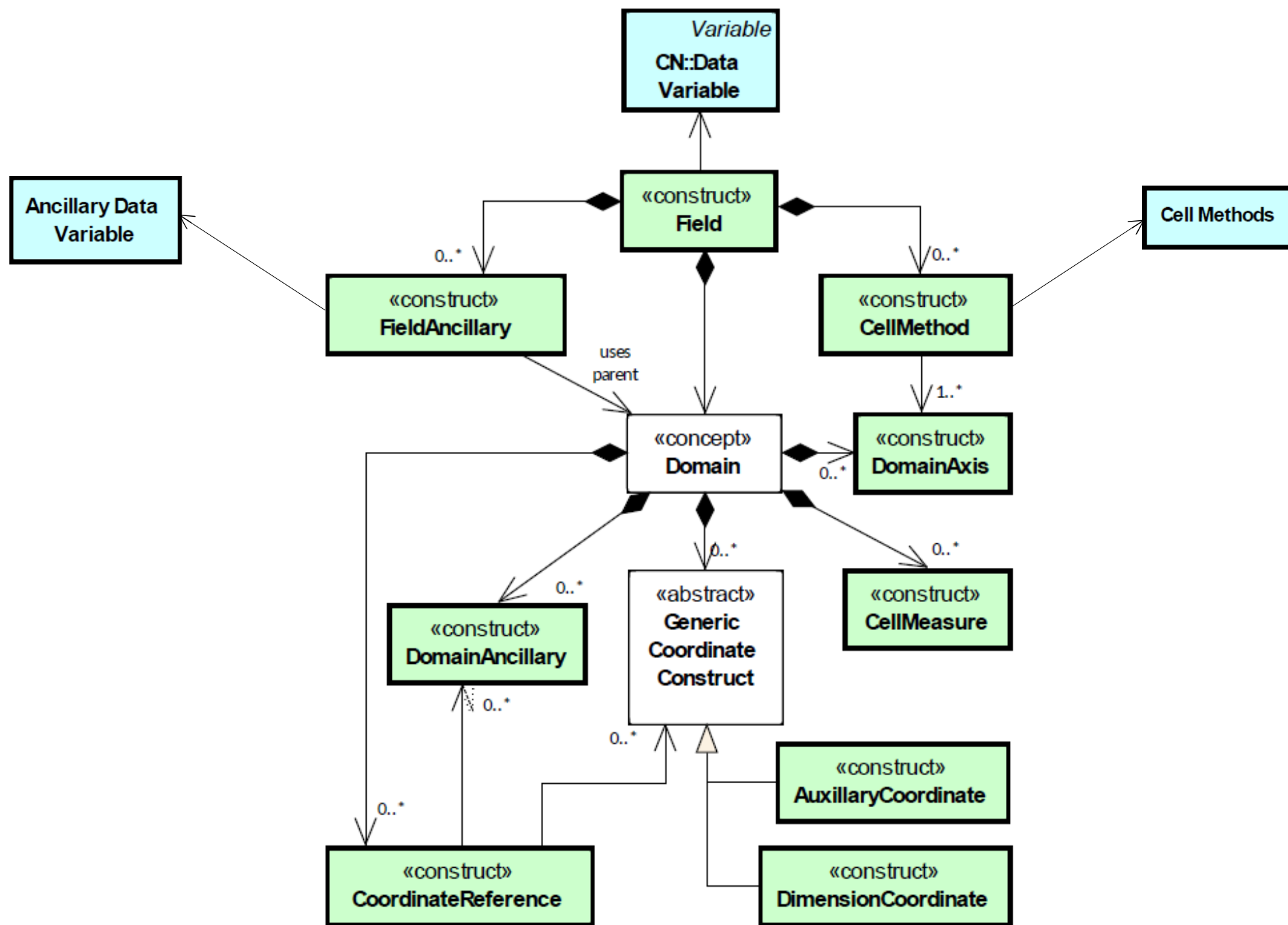
Elements of CF-netCDF



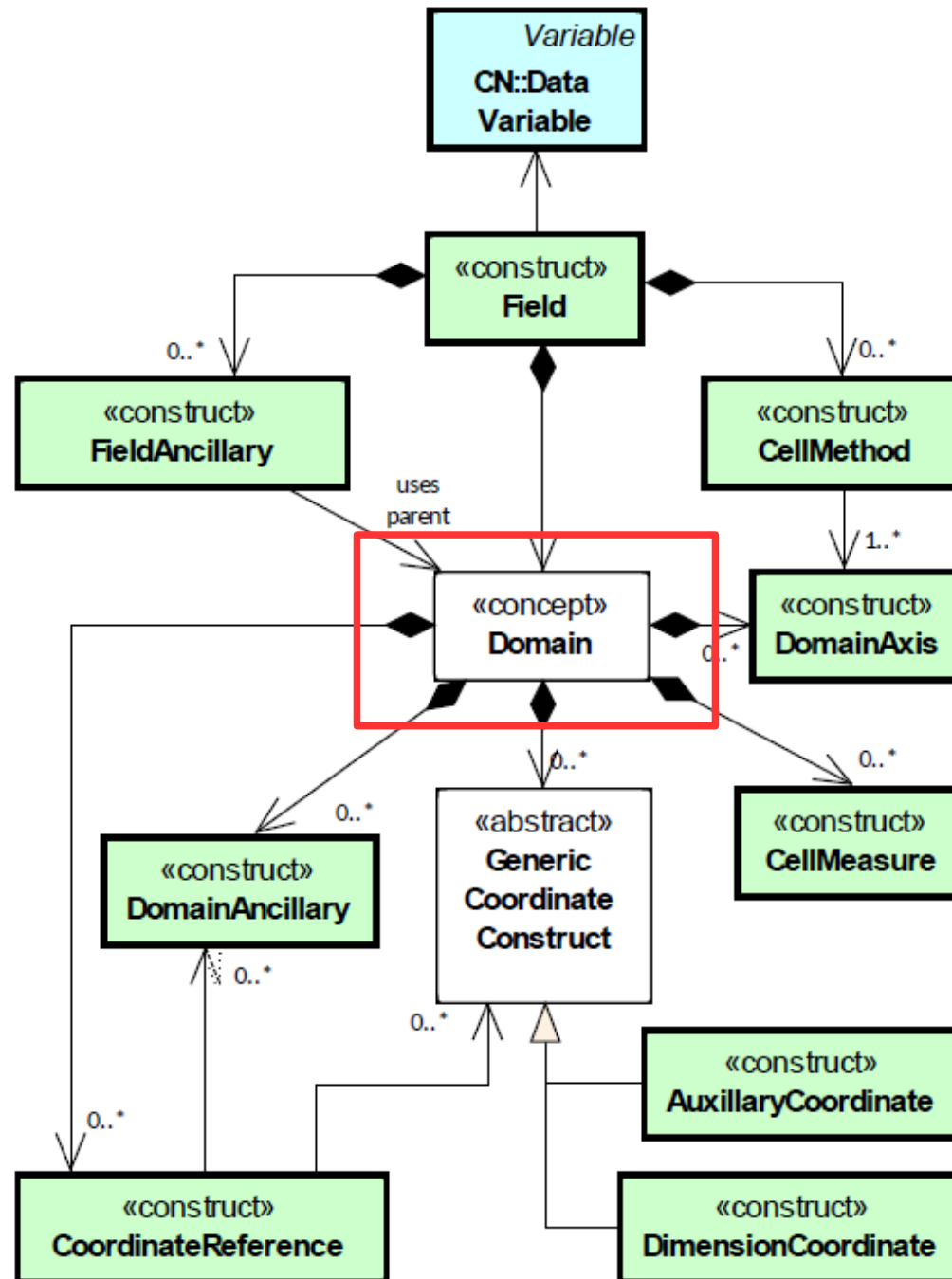
netCDF classic



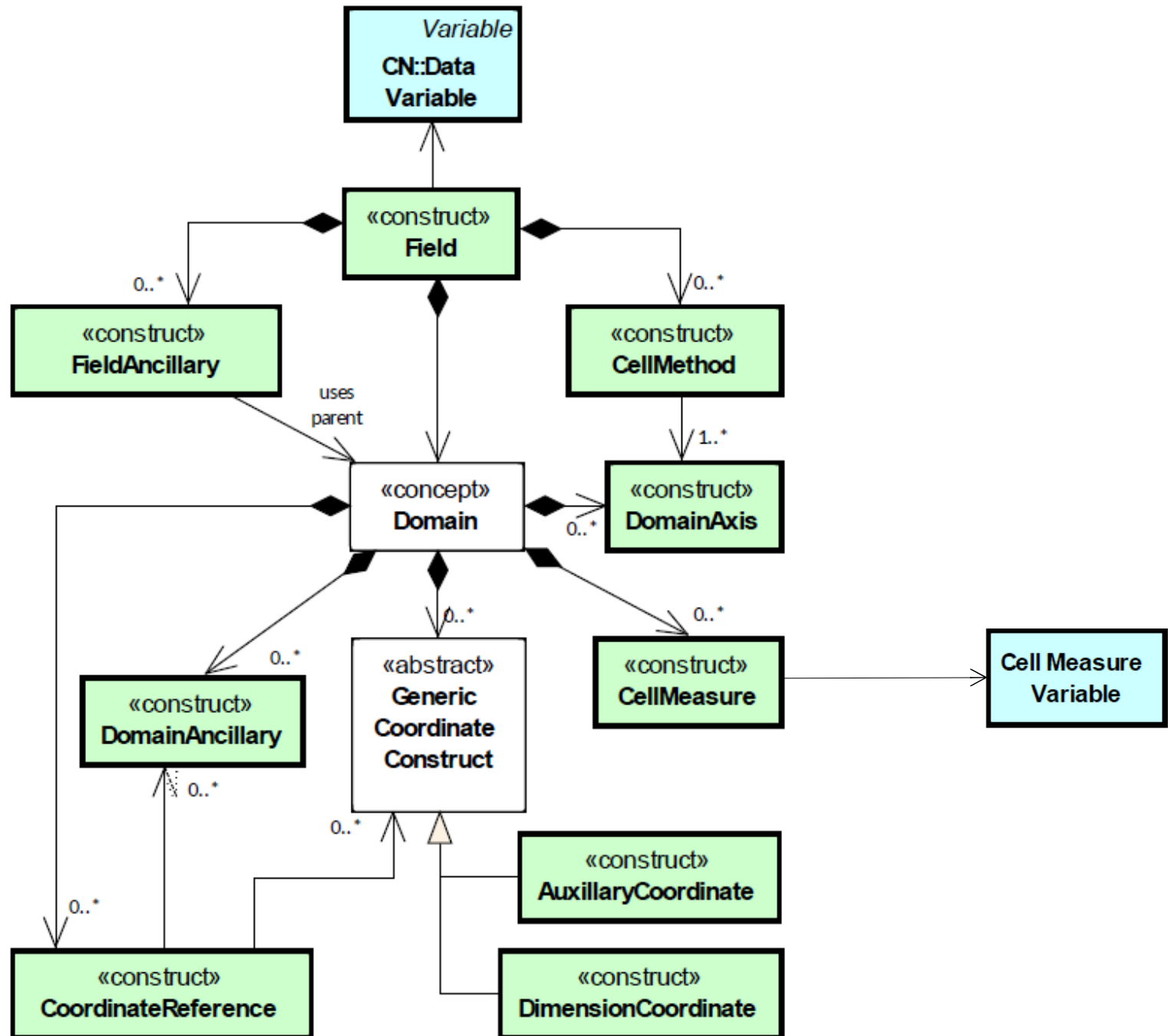
The CF data model: field ancillary and cell method



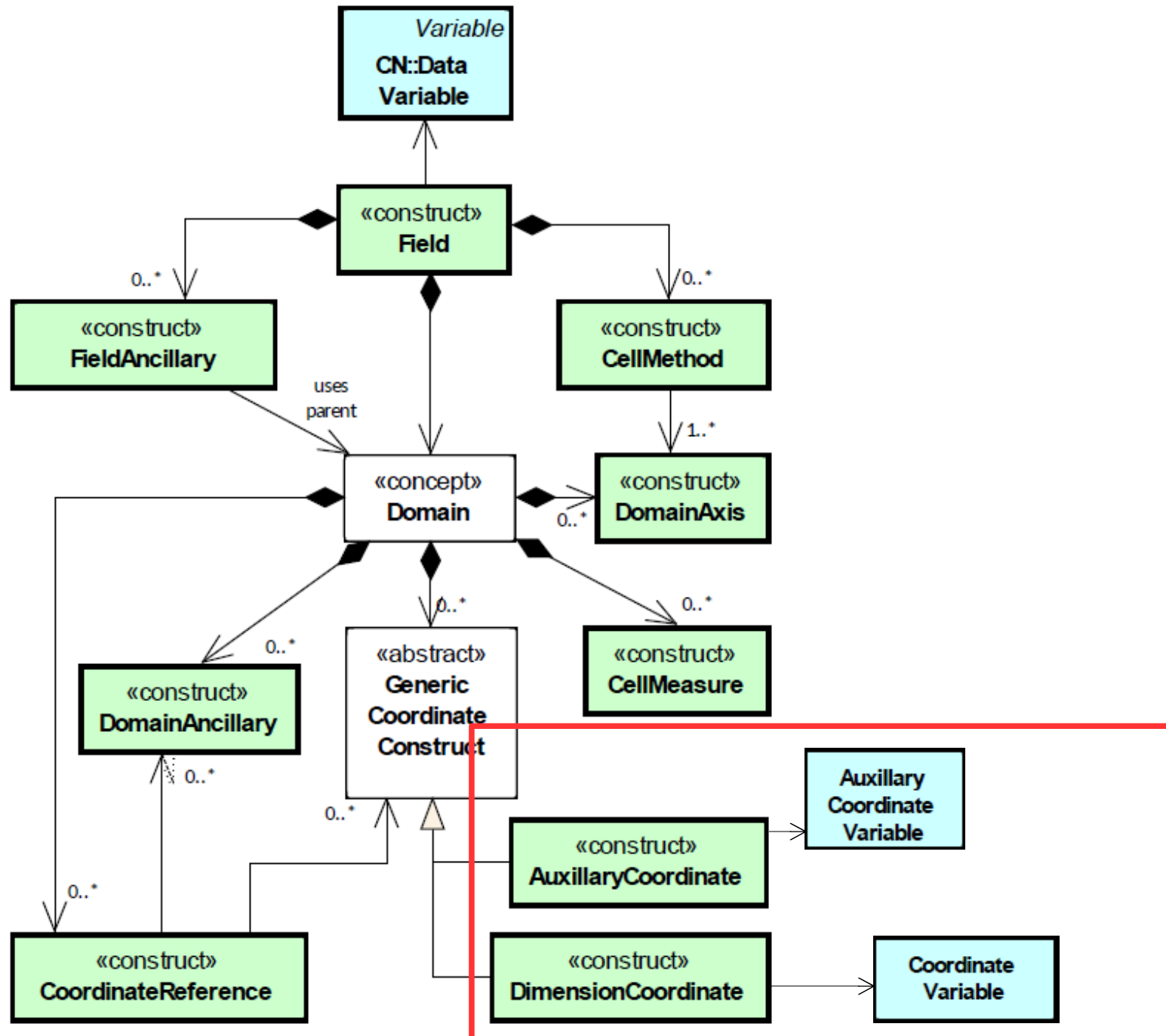
The CF data model: domain



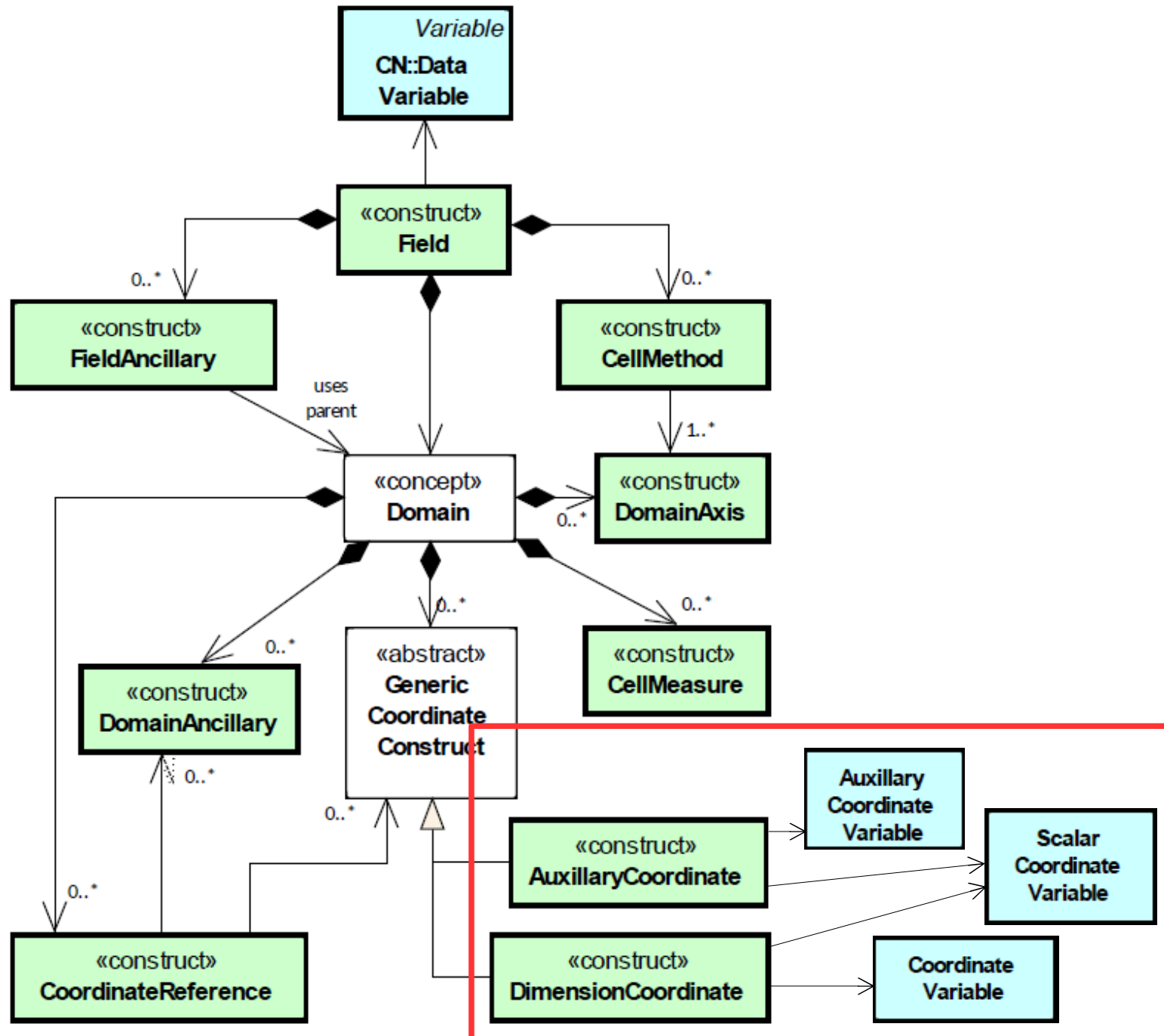
The CF data model: cell measure



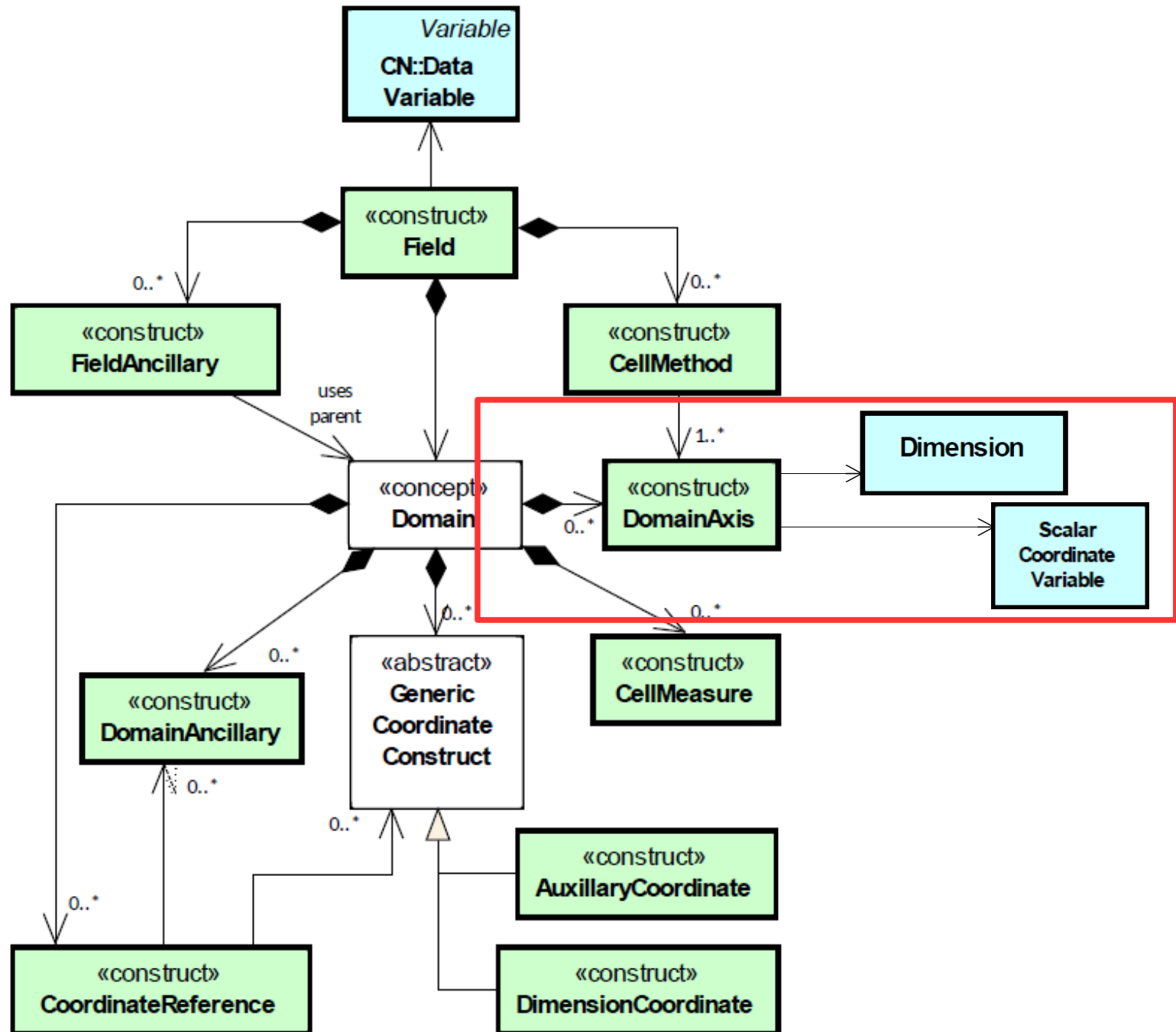
The CF data model: domain axis, coordinates



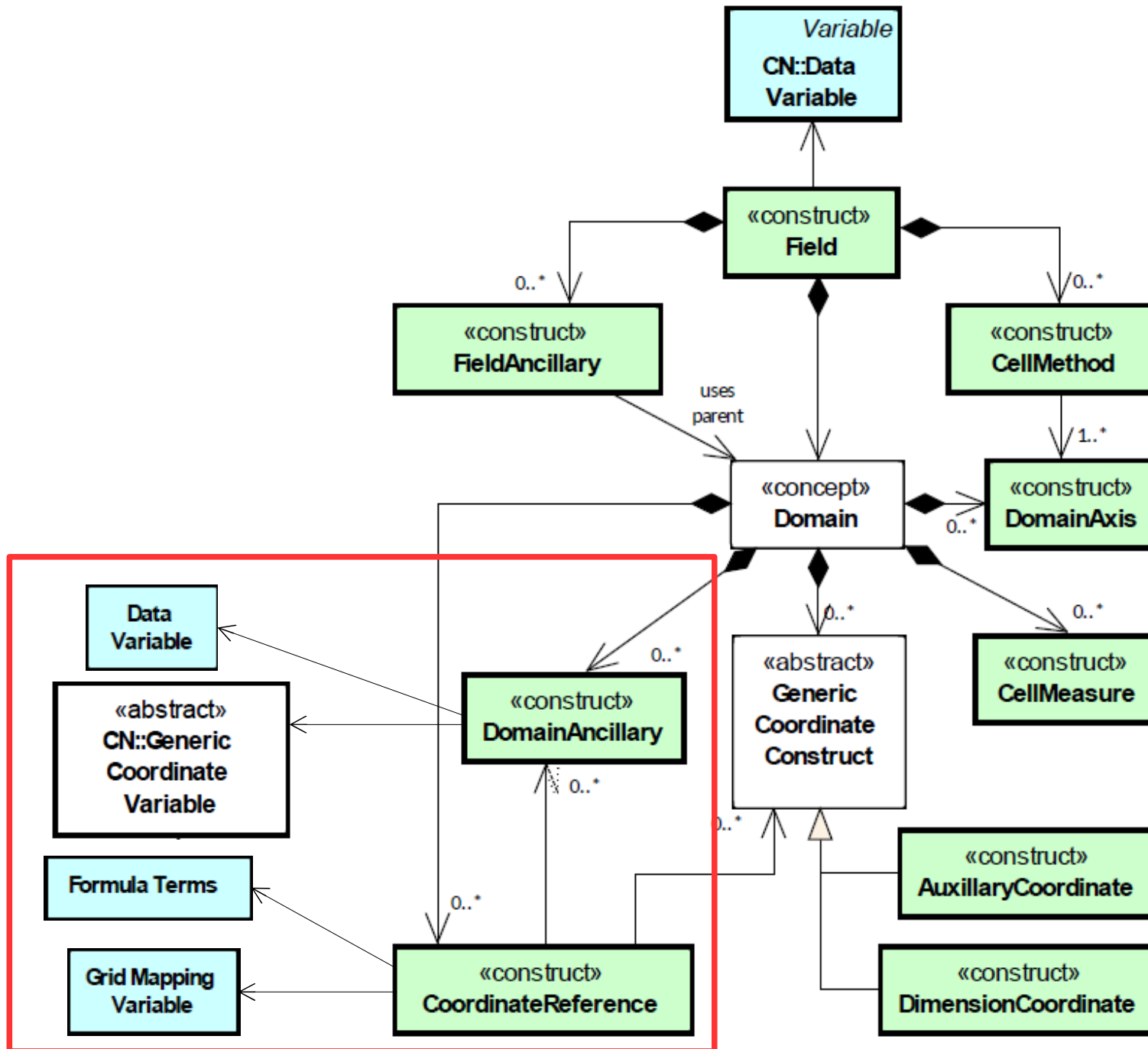
The CF data model: domain axis, coordinates



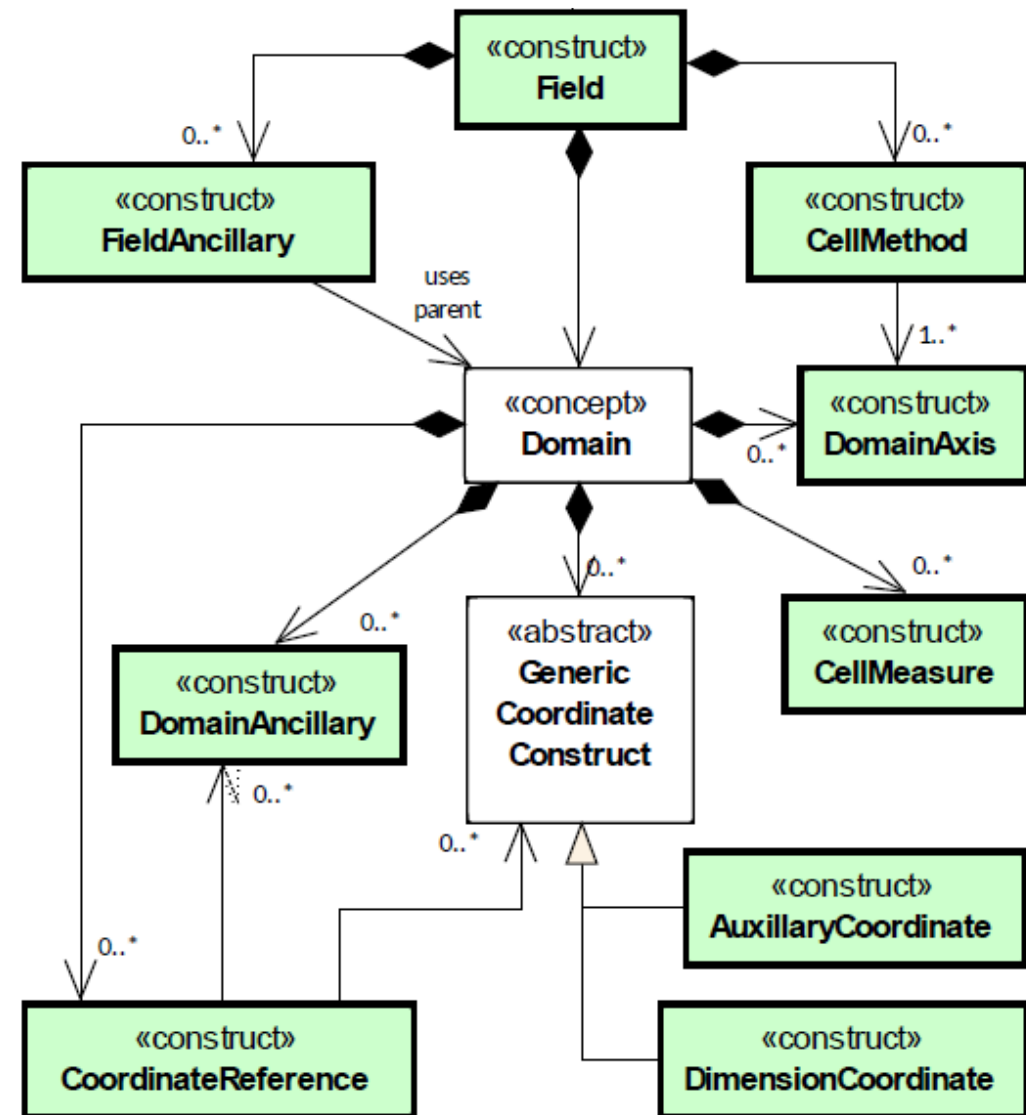
The CF data model: domain axis, coordinates



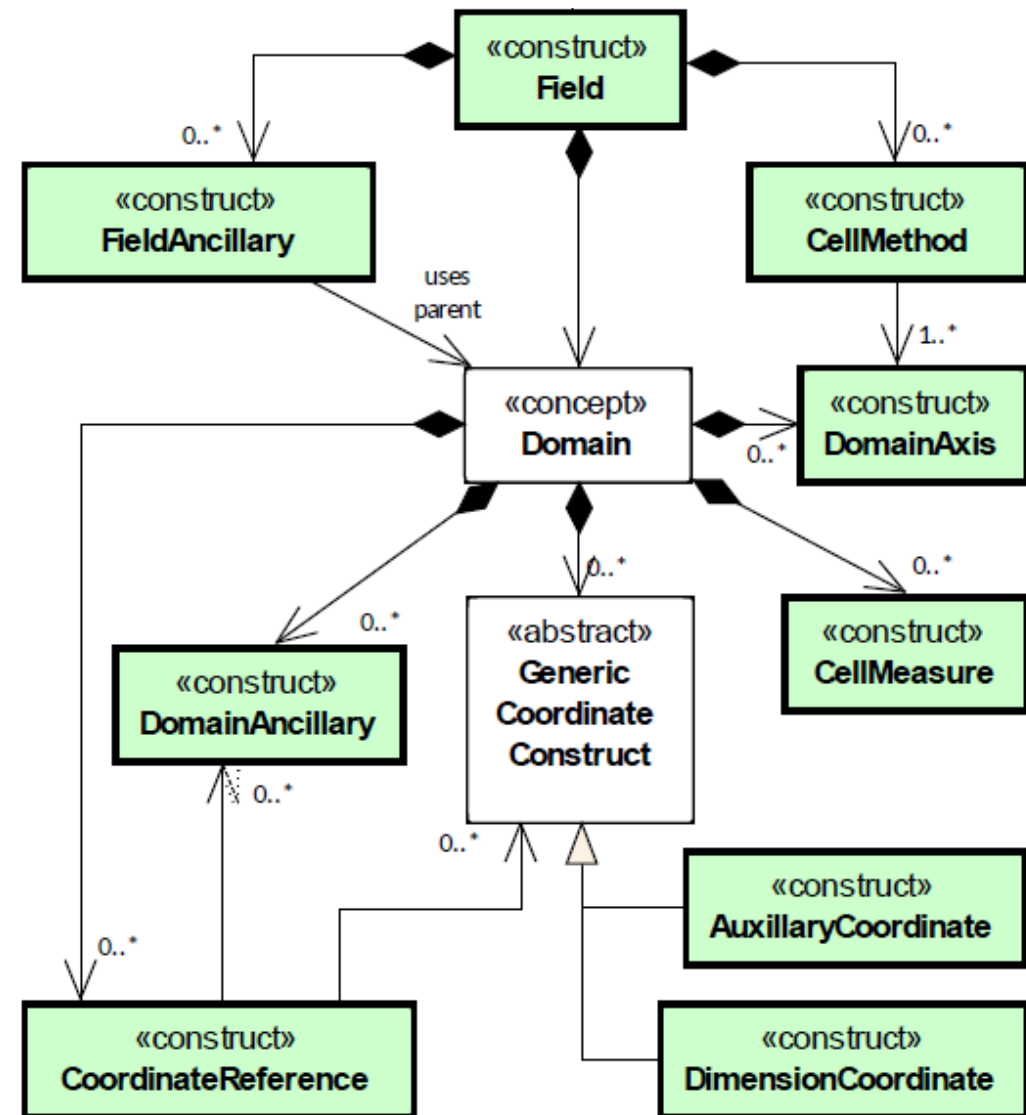
The CF data model: coordinate reference



- To be useful, a data model needs to be accepted as part of CF



- A CF data model can present the CF conventions in a manner that will lead to their being better understood*



- *A CF data model can allow software developers to design CF-compliant data processing applications*
 - Using the CF data model for the software's internal data storage ought to guarantee CF-compliance
- Data model implementation: cf-python
 - <https://cfpython.bitbucket.io>
 - described in chapter 6 in the paper
 - The actual data model implementation is currently embedded in an API with higher-level functionality, but is in the process of being pulled out as a stand alone library

- *A CF data model can provide guidance during the development of future extensions to the CF conventions*
- A CF data model can be used to ensure that an enhancement to CF fits in a logically, rather than just pragmatically
- If it can not be represented by the CF data model, then
 - the enhancement could be modified so it does
 - the data model could be extended/generalized (backwards compatible)
 - the data model could be changed in backwards incompatible ways