



Australian Government

Ansto

Nuclear-based science benefiting all Australians

Common Data Model

Nick Hauser

NOBUGS 2010

Gatlinburg, Tennessee

October 11-13

Common data model



It's not a new idea –

just an implementation of an old one.

Fig 1. Picture of Nick getting off the flight from Sydney to Knoxville

Common data model



As of this year, this is a collaboration between ANSTO and Soleil

Common data model

NetCDF

(network Common Data Form) is a set of **software libraries** and machine-independent data formats that support the creation, access, and sharing of **array-oriented scientific data**.



Common data model



So – what is different?

(between this CDM and the netCDF CDM)

Common data model

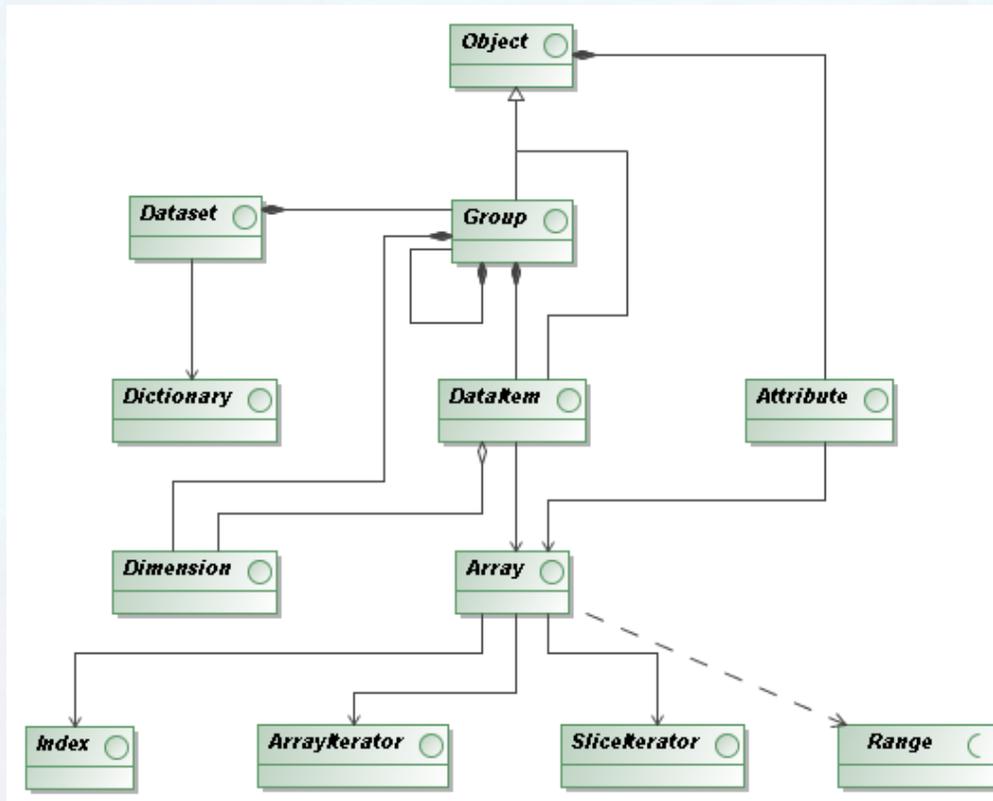


Nexus dictionary

L1=\$entry/instrument/L1

- I can write my data access code without dependency on the location of the data in your file
- You provide me with a dictionary

Common data model

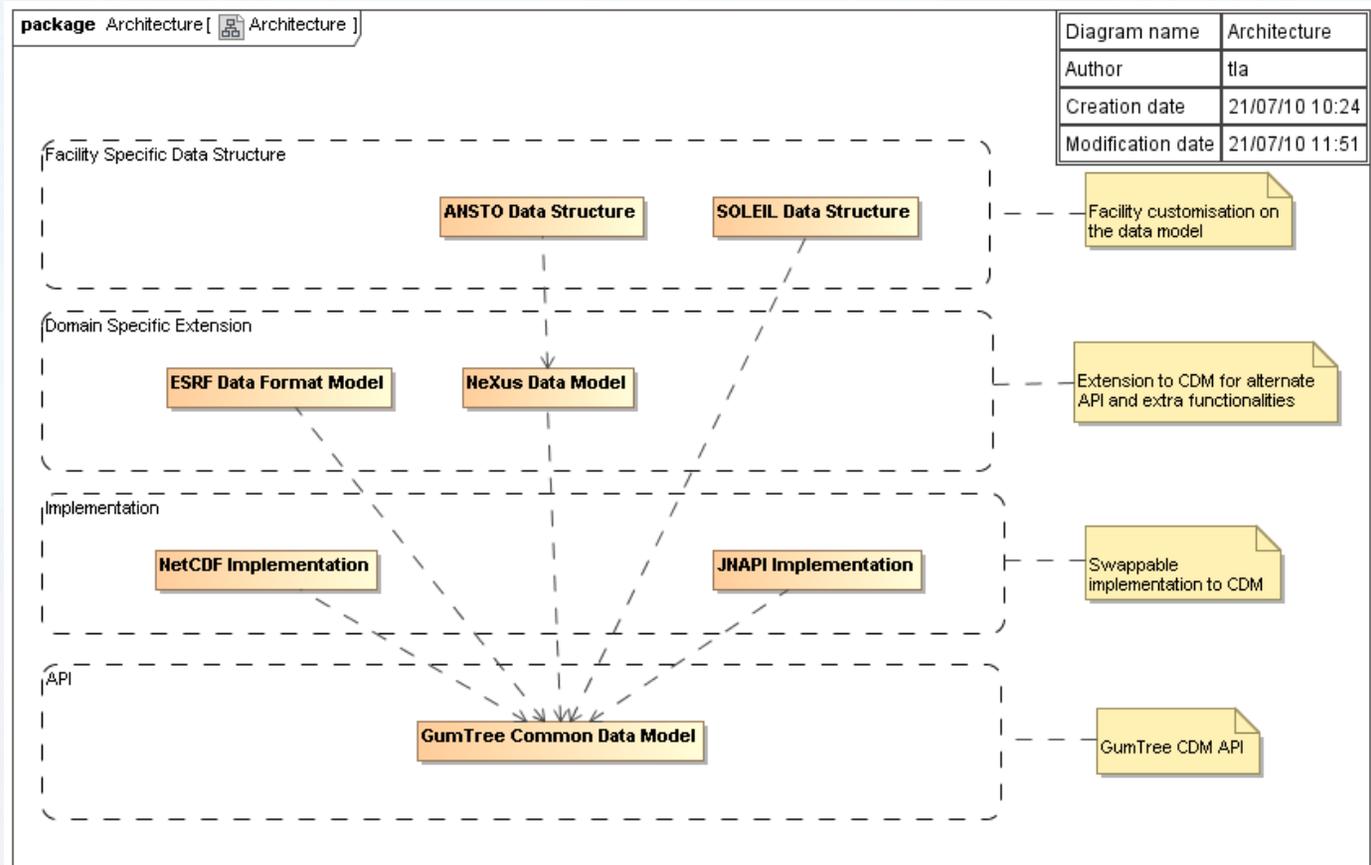


CDM interfaces

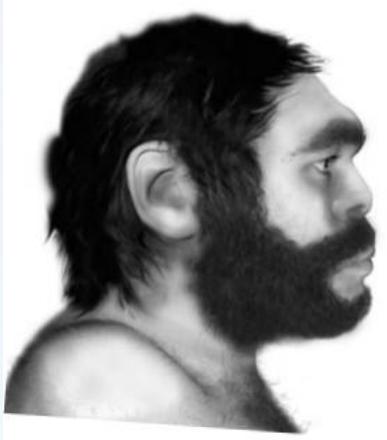
But you don't use these.

You use the 'NeXus' api that sits on top.

Common data model



Common data model



More than a primitive data type

The CDM Array class is more than a primitive Java or C++ array. It is a **scientific data object with file IO**, allowing you to slice and dice arrays and to do math. These methods are the same as found in the default interface implementation provided by netcdf

Common data model

The CDM API has an **error** object, that provides propagation of count uncertainties based on Poisson statistics with every math operation. This is extensible to other uncertainty calculations

Note that an uncertainties property has been added to NXdata at the 2010 NIAC, hence providing a standard way to store uncertainties to file

Common data model

File IO using either the hdf java library or the jnapi

Data can be saved in NeXus hdf5, XML (either NeXus or cansas), 3 column ascii

Common data model

Scripting interface

Data can be scripted using python – either CPython or jython

Supports operator overloading

e.g. if array is a cdm array, then in python

```
array * 2 * math.pi
```

```
array.sin()
```

```
array is a java object
```

Common data model

Scripting interface

numpy support - yes, but...

array can be cast into a numpy array object. It is no longer a cdm object, and we don't have a way to transform a numpy object to a cdm object

NeXus browse and vis

The screenshot displays the Scientific Workbench interface with the following components:

- Project Explorer:** Shows a tree view of the 'echidna' project containing a 'result' folder and numerous '.hdf' files.
- Dataset Viewer:** Shows the hierarchical structure of the selected dataset: 'entry1' > 'data' > 'instrument' > 'detector' > 'total_counts'.
- Data Table:** A table with 12 rows and 2 columns (Index, Value).

Index	Value
0	36805
1	37022
2	37188
3	37190
4	37507
5	37475
6	36735
7	37249
8	36827
9	36843
10	36714
11	36696
- Line Graph:** A plot of 'total_counts' vs index, showing a fluctuating red line with a peak around index 4.
- Properties Panel:** Lists metadata for the 'total_counts' dataset, including 'units: count', 'long_name: total_counts', 'target: /entry1/instrument/detector/total_c...', and '_lastModified: 2010-09-08T23:02:02Z'.
- Command Line:** Displays engine information: 'Engine: Java Embedded Python', 'Engine Version: 2.x', 'Language: CPython', and 'Language Version: Whatever you compiled with'. A 'Command >' prompt is visible.
- Python History:** A panel for tracking Python execution, with tabs for 'Variables', 'Functions', and 'Modules'.

NeXus browse and vis

The screenshot displays the Scientific Workbench interface for NeXus data. The main window is titled "Scientific Workbench - data/experiments/echidna/ECH0004911.nx.hdf - GumTree Workbench 1.5.5".

Project Explorer: Shows a tree view of the "echidna" project with subfolders like "result" and "data".

Dataset Viewer: Shows the "entry1" dataset structure. The selected data item is "entry1/data/hmm".

Data Item Properties: Shows "Data item: entry1/data/hmm", "Frame: 1 / 50", and "Layer: 1 / 1".

Data Table: A table with 9 rows and 3 columns. The data is as follows:

	0	1
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	1
7	0	0
8	1	0

Data Visualization: A heatmap plot showing the data from the table. The x-axis ranges from 0 to 125, and the y-axis ranges from 0 to 125. The color scale ranges from 0.0 (blue) to 150.0 (red).

Command Line: Shows the engine information: "Engine: Java Embedded Python", "Engine Version: 2.x", "Language: CPython", and "Language Version: Whatever you compiled with".

Python History: Shows a table with columns "Name", "Type", and "Value".

Common data model



It's not a new idea –

The code has been in use at ANSTO since 2005, as is part of the Gumtree ecosystem to combine acquisition with reduction and visualisation. Seems like a few of us had the same idea at around the same time e.g. GDA

Soleil will put into production late 2010

Soleil also coding the CDM in C++ and using it to read and manage ESRF datafiles within their C++ data analysis application

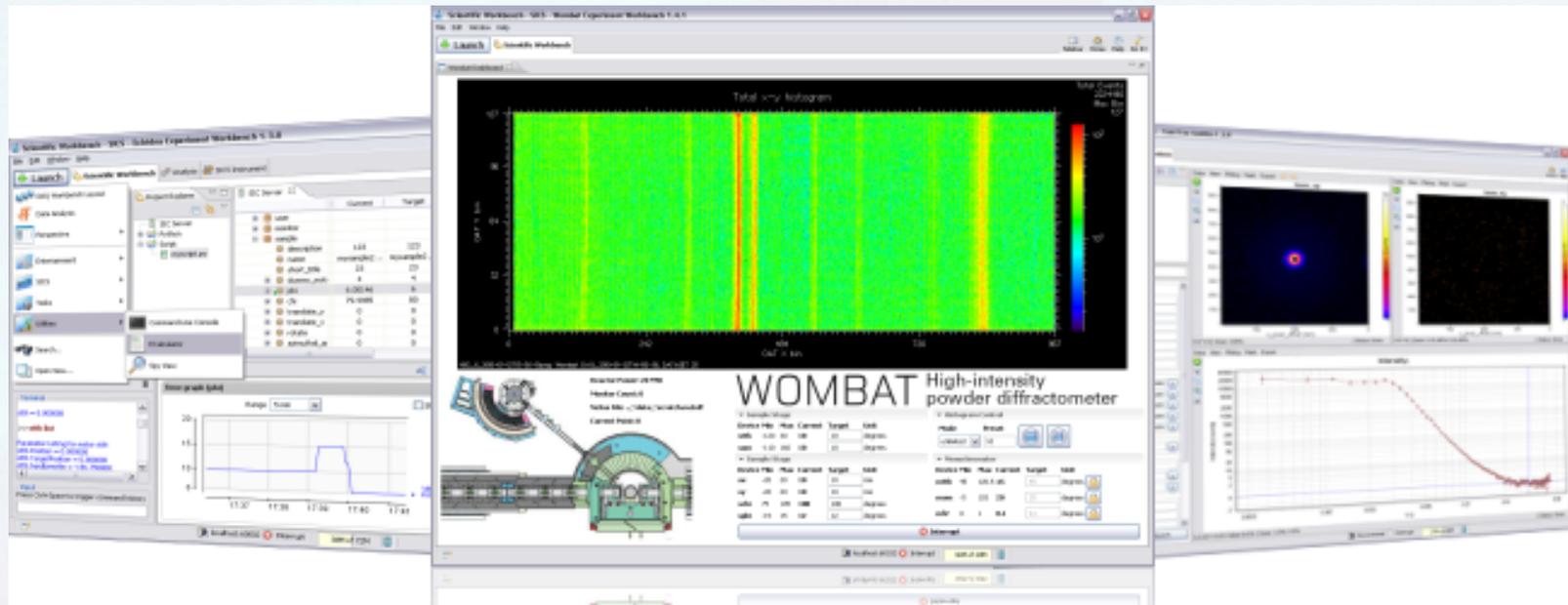
Common data model

Java plug-in available at

gumtree.codehaus.org

Or contact nick.hauser@ansto.gov.au

Common data model



Common data model

Credits

netCDF team

NeXus tech committee

ANSTO team - Paul Hathaway, Darren Kelly, Tony Lam,
Norman Xiong

Soleil team - Majid Ounsy, Clement Rodriguez, Stephane
Poirier, Alain Buteau

The ANSTO logo features a stylized white 'a' inside a circle, followed by the letters 'nsto' in a bold, sans-serif font. The background is a vibrant blue with abstract, flowing light trails that create a sense of motion and energy.

ansto

Nuclear-based science benefiting all Australians