COMMON DATA MODEL ACCESS; A UNIFIED LAYER TO ACCESS DATA FROM DATA ANALYSIS POINT OF VIEW

S. Poirier, A. Buteau, M. Ounsy, C. Rodriguez,
Synchrotron SOLEIL[1], France
N. Hauser, T. Lam, N. Xiong, ANSTO[2], Australia

Abstract

For almost 20 years, the scientific community of neutron and synchrotron institutes have been dreaming of a common data format for exchanging experimental results and applications for reducing and analyzing the data. Using HDF5 as a data container has become the standard in many facilities. The big issue is the standardization of the data organization (schema) within the HDF5 container. By introducing a new level of indirection for data access, the CommonDataModelAccess (CDMA) framework proposes a solution and allows separation of responsibilities between data reduction developers and the institute. Data reduction developers are responsible for data reduction code; the institute provides a plug-in to access the data.

The CDMA is a core API that accesses data through a data format plug-in mechanism and scientific application definitions (sets of keywords) coming from a consensus between scientists and institutes. Using a innovative "mapping" system between application definitions and physical data organizations, the CDMA allows data reduction application development independent of the data file container AND schema. Each institute develops a data access plug-in for its own data file formats along with the mapping between application definitions and its data files. Thus data reduction applications can be developed from a strictly scientific point of view and are immediately able to process data acquired from several institutes.

THE GENESIS OF THE PROJECT

Working independently, ESRF, SOLEIL, DESY and ANSTO software development has focused on the design of frameworks for data processing, operating on top of a NeXus [1] data storage layer.

The central issue for collaboration was related to using the same tool independent of the data container and schema. Work at ANSTO on the GumTree Data Model [2] abstracted data file access of the underlying NeXus files (the standard data format at ANSTO) by designing a data model with a set of Java interfaces. This seemed to be a very promising development to share.

SOLEIL became interested in the concept as it was coincidentally looking for a unified data access layer based on NeXus (the standard data format used at SOLEIL) to build on top of it its COMETE [3] project, a Java framework that aims to ease data visualization and data analysis applications programming.

The collaboration started between ANSTO and SOLEIL in January 2010, after a meeting between the authors at ICALEPCS 2009. The work started from the data access layer of ANSTO's GumTree project [4], written in Java.

In Q4 2011, DESY will join the collaboration and help us developing the C++ port of CDMA.

BACK TO OUR MOTIVATIONS

The important thing we must understand is that the data format is not an issue. The issue is how to allow users of our institutes to use the same tools regardless the origin of the data?

We notice that even in the same synchrotron or neutron facility, data schemas are not the same across the beamlines. The NeXus standard is useful, but not strong enough to ensure uniformity. Interpretation of the standard allows two identical beamlines to have different data schemas.

The aim of the CDMA is to offer an abstract data access layer in order to build analysis/reduction applications regardless of the data schema.

HDF-like formats allow the recording of any kind of data using an API; the physical file organization is abstracted. The NeXus-like specification is a set of logical data organizations in a standardization effort. This kind of standardization may be applicable to various tree-oriented data formats (such as HDF or XML). The problem of this approach is that facilities must produce data files that strictly comply with the specification. This is a huge challenge because each facility's staff (mostly scientists & engineers) has its own view on acquisition systems and experimental and contextual data. Also the hardware and the acquisition process are rarely driven by the data recording system.

The idea behind the CDMA is to reverse the data point of view. Rather than desperately try to standardize the data files across institutes, is it not easier to introduce a layer able to hide the different way the data is stored? We think the answer is *yes*!

We encourage the use of the NeXus standard. We believe that in the future, as the standard matures, the NeXus API may replace CDMA. We encourage facilities to take part in the NeXus standardization process. However, CDMA allows now to deal with the various standard interpretation.

[1] http://www.synchrotron-soleil.fr
[2] http://www.ansto.gov.au/

Each institute will continue to produce data using the most suitable format. There is no need to wait for the ultimate data organization specification before running acquisitions, and sharing data and applications!

INTRODUCTION TO THE CDMA

The CDMA comprises

- a client layer API for writing data reduction and analysis applications.
- a developer API to build the data access plug-in.

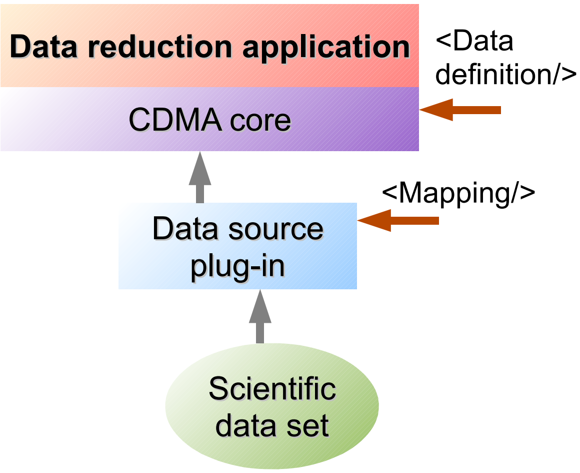


Figure 1: CDMA general usage schema.

Client API

Using the client API, the data reduction applications developer doesn't have to know anything about the file formats because the API uses an abstract data access layer that hides the data file specification.

Data Source Plug-In

The data file specification is embedded through a plug-in mechanism. It is therefore the institutes' responsibility to develop their data access plug-ins in order to open datasets acquired in their institutes through the CDMA.

Classical Navigation API

Nevertheless, using the navigation API, the data reduction developer has to know precisely the data schema in each data file their application will access. We think it's a useless waste of time. This is a roadblock to these projects being used outside of the organization where they are being developed.

Dictionary Mechanism

To solve this issue the CDMA introduces a new, innovative way of accessing data. This is the dictionary mechanism.

Using this mechanism, the data reduction application developer no longer has to know the data schema. The data is accessed using keywords. A keyword is a short character string naming a scientific measurement or a generic technical data item. Thus the data access part of data reduction applications' source code is simpler and (this is the most important) more stable.

Considering this mechanism, scientists have to agree on key names, regardless of the way the data is physically organized. Moreover, the data may have different units (wavelength vs. energy for example) for the same measurement; the CDMA provides a mechanism to perform conversions at run-time.

Please note that the old-style navigation API is still available but we strongly recommend considering the dictionary API.

CLIENT API

The client API defines interfaces that abstract the data sources. There are three levels of abstraction:

- The top level is the *IDataset* interface which represent a handle all the data of an experiment.
- *IGroup* (or *ILogicalGroup* if using the dictionary mechanism, see below) defines a group of related data. There is at least one root group for each dataset; a group may contains sub groups and data items.
- *IDataItem* defines a single value or measurement which can be a scalar or a multidimensional array.

Along with the data access layer, the Array class allows an efficient manipulating of multidimensional arrays. It is is more than a primitive Java or C++ array. It is a scientific data object allowing you to slice and dice arrays and to do math.

The CDMA provide also an class that allows error propagation, the *Error* object, that provides propagation of count uncertainties based on Poisson statistics with every math operation. This is extensible to other uncertainty calculations.

DATA SOURCE PLUG-IN SYSTEM

Using the CDMA library, the data reduction developer doesn't care about data schemas. The plug-in mechanism dynamically loads the plug-in (a dynamic library) that accesses data from the data file. Thus this mechanism allows a user to open files acquired from different institutes, in the same session.

DICTIONARY MECHANISM

The dictionary mechanism relies on two XML documents.

Data Definitions

The first document is a set a keywords matching scientific or technical data items. For instance a key named current should refer to the effective current in a storage ring at acquisition time. These keywords may be just listed by this document or organized through a tree hierarchy. In the latter case, this document describes a particular a view on the data, like a NeXus application definition.

This document is intended to be independent of the way the data is physically organized. There are (at least) two ways of writing this document:

- it may be written for a specific data analysis application which already exists and is adapted to use the CDMA,
- or it may be written independently of any application, like the NeXus application definitions.

```
<data-def name="Example">
  <group key="detectors">
    <group key="detector">
      <item key="camera"/>
      <item key="distance"/>
      <item key="exposureTime"/>
      <item key="shutterCloseDelay"/>
      <item key="xBin"/>
      <item key="yBin"/>
    </group>
  </group>
  <group key="data">
    <item key="images"/>
    <item key="spectrums"/>
  </group>
</data-def>
```

Figure 2: Data definition example.

Keywords Mapping

The second document is the dictionary itself. It's the mapping between keywords and real paths..It needs an exact knowledge of the data schema in the file.

```
<map-def name="Example" version="1.0.0">
 <item key="distance">
  <path>
   /{NXentry}/{NXinstrument}/detector/distance
  </path>
 </item>
 <item key="exposureTime">
  <path>
   /{NXentry}/{NXinstrument}/detector/Exposure
  </path>
 </item>
 <item key="shutterCloseDelay">
  <path>
   /{NXentry}/{NXinstrument}/detector/CloseDelay
  </path>
 </item>
 <item key="xBin">
  <path>
   /{NXentry}/{NXinstrument}/detector/Xbin
  </path>
 </item>
 <item key="yBin">
  <path>
   /{NXentry}/{NXinstrument}/detector/Ybin
  </path>
 </item>
<map-def>
```

Figure 3: NeXus file mapping example.

Responsibilities

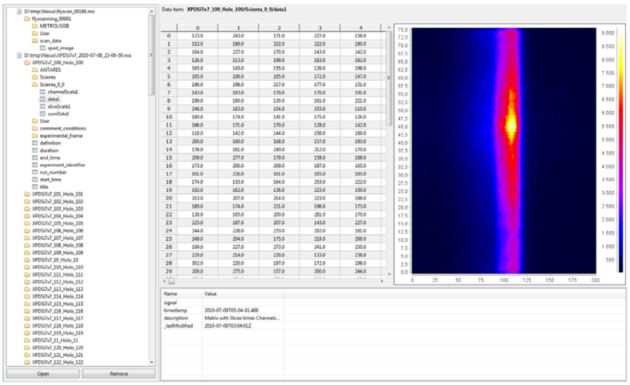
Given a keywords list, the institutes that produce experimental data must write the mapping document corresponding to their data files organization.

CURRENT STATUS

There are two implementations of the CDMA, in Java and C++.

The Java implementation is mature (in operation) and already available on the SVN Codehaus repository [5].

ANSTO has developed a data browser based on the Java version of the CDMA, and uses CDMA on 4 neutron beam instruments as part of the Gumtree ecosystem.



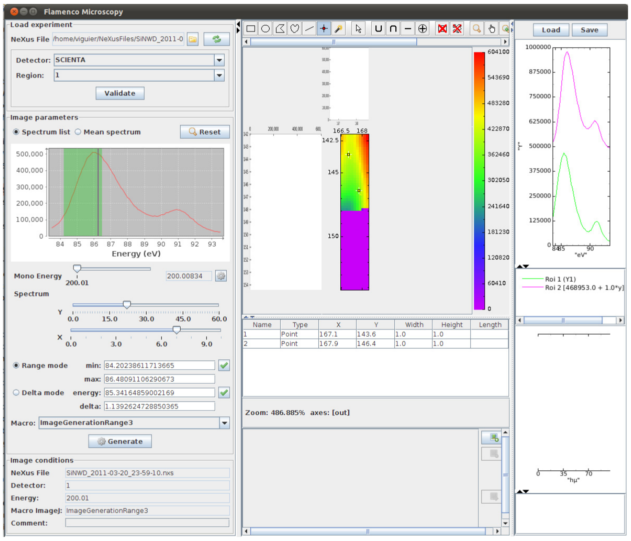
Data item: XPDSI7x7_100_Holo_100/Scienta_0_0/data1

	0	1	2	3	4
0	123.0	243.0	171.0	227.0	158.0
1	152.0	189.0	232.0	222.0	180.0
2	164.0	227.0	170.0	143.0	142.0
3	126.0	113.0	199.0	183.0	182.0
4	165.0	145.0	155.0	126.0	196.0
5	185.0	198.0	185.0	172.0	147.0
6	196.0	198.0	217.0	177.0	151.0
7	143.0	183.0	170.0	170.0	191.0
8	199.0	180.0	130.0	161.0	221.0
9	246.0	183.0	154.0	153.0	110.0
10	180.0	174.0	141.0	175.0	126.0
11	166.0	171.0	170.0	128.0	142.0
12	118.0	142.0	144.0	158.0	189.0
13	200.0	160.0	168.0	157.0	193.0
14	176.0	161.0	249.0	212.0	170.0
15	209.0	277.0	179.0	159.0	189.0
16	173.0	200.0	209.0	187.0	165.0
17	161.0	226.0	161.0	185.0	185.0
18	174.0	133.0	164.0	253.0	222.0
19	192.0	162.0	136.0	223.0	159.0
20	213.0	207.0	214.0	223.0	188.0
21	189.0	174.0	221.0	196.0	173.0
22	138.0	165.0	200.0	281.0	170.0
23	225.0	187.0	207.0	143.0	227.0
24	244.0	226.0	233.0	202.0	161.0
25	249.0	204.0	175.0	219.0	206.0
26	199.0	227.0	273.0	241.0	230.0
27	229.0	214.0	220.0	133.0	238.0
28	302.0	220.0	197.0	172.0	198.0
29	209.0	275.0	157.0	200.0	244.0

Name	Value
signal	
timestamp	2010-07-09T05-04-01.466
description	Matrix with Slices times Channels...
_lastModified	2010-07-09T03:04:01Z

Figure 4: CDMA based data browser at ANSTO.

On SOLEIL side, two data reduction applications written in Java and based on the CDMA are currently in production, on SWING (SAX acquisitions) and ANTARES (EXAFS, Photo-emission measurements).



Flamenco Microscopy

Load experiment

NeXus File: /home/viguier/NeXusFiles/SiNWD_2011-0

Detector: SCIENTA

Region: 1

Image parameters

● Spectrum list ○ Mean spectrum

Mono Energy: 200.01 | 200.00834

● Range mode min: 84.20238611713665 max: 86.48091106290673

○ Delta mode energy: 85.34164859002169 delta: 1.1392624728850365

Macro: ImageGenerationRange3

Image conditions

NeXus File: SiNWD_2011-03-20_23-59-10.nxs

Detector: 1

Energy: 200.01

Macro ImageJ: ImageGenerationRange3

Comment:

Name	Type	X	Y	Width	Height	Length
1	Point	167.1	143.6	1.0	1.0	
2	Point	167.9	146.4	1.0	1.0	

Zoom: 486.885% axes: [out]

Figure 5: Flamenco, a data reduction application using CDMA at SOLEIL.

We are working on the C++ implementation which is scheduled to be available in early 2012. Then a Python port will be developed based on this C++ implementation.

CONCLUSION

The first results on using the CDMA are very promising. We have found that the time required to develop a new data reduction application has been significantly reduced, because the developers naturally no longer have to take care of the data format.

On the technical side , with the arrival of new participants (DESY, ANKA, ...), we have a community that will be able to quickly evolve this project, by enlarging the number of data reduction applications and data source plug-ins to allow cross institutes experimental files exchanges.

Last but not least, our conviction is that the CDMA project is a valuable technical answer to the Data Management issues that European projects like PANDATA, HDRI, NFFA, etc are willing to address, as CDMA provides a solution now to the old dream of able exchanging in a transparent way data files and data analysis applications between institutes.

REFERENCES

[1] NeXus format, http://www.nexusformat.org/
[2] Gumtree framework, http://gumtree.codehaus.org/
[3] COMETE framework, ICALEPCS 2011, WEMAU012 , http://comete.sourceforge.net/,
[4] Gumtree data access layer, http://gumtree.codehaus.org/GumTree+Data+Model
[5] https://svn.codehaus.org/gumtree/datamodel/trunk/