

NeXus at ESRF

Panosc WP4 Meeting

euXFEL

27th June 2019

T. Vincent, V. Armando SOLÉ
ESRF – Data Analysis Unit

- ESRF NeXus interpretation for raw data
- ESRF NeXus interpretation for processed data
- Other ESRF NeXus Uses: Metadata storage
- Status

HDF5/NeXus – ESRF Interpretation for Raw Data

NXroot

Top level. One per file.

NXentry

One group per measurement

NXinstrument

Describe the instrument.

Only one per NXentry

measurement (@NXcollection)

Flattened view of everything measured

Only one per NXentry

sample (@NXsample)

Define the physical state of the sample during the scan

NXdata

The default data to be plotted.

One NXdata group per plot

user (@NXuser)

Details of a user, i.e., name, affiliation, email address, *etc*

NXsubentry

Data or links to data for particular analysis

Exclusive **Acquisition** Domain

Almost exclusive **Acquisition** Domain

User/Scientist Domain

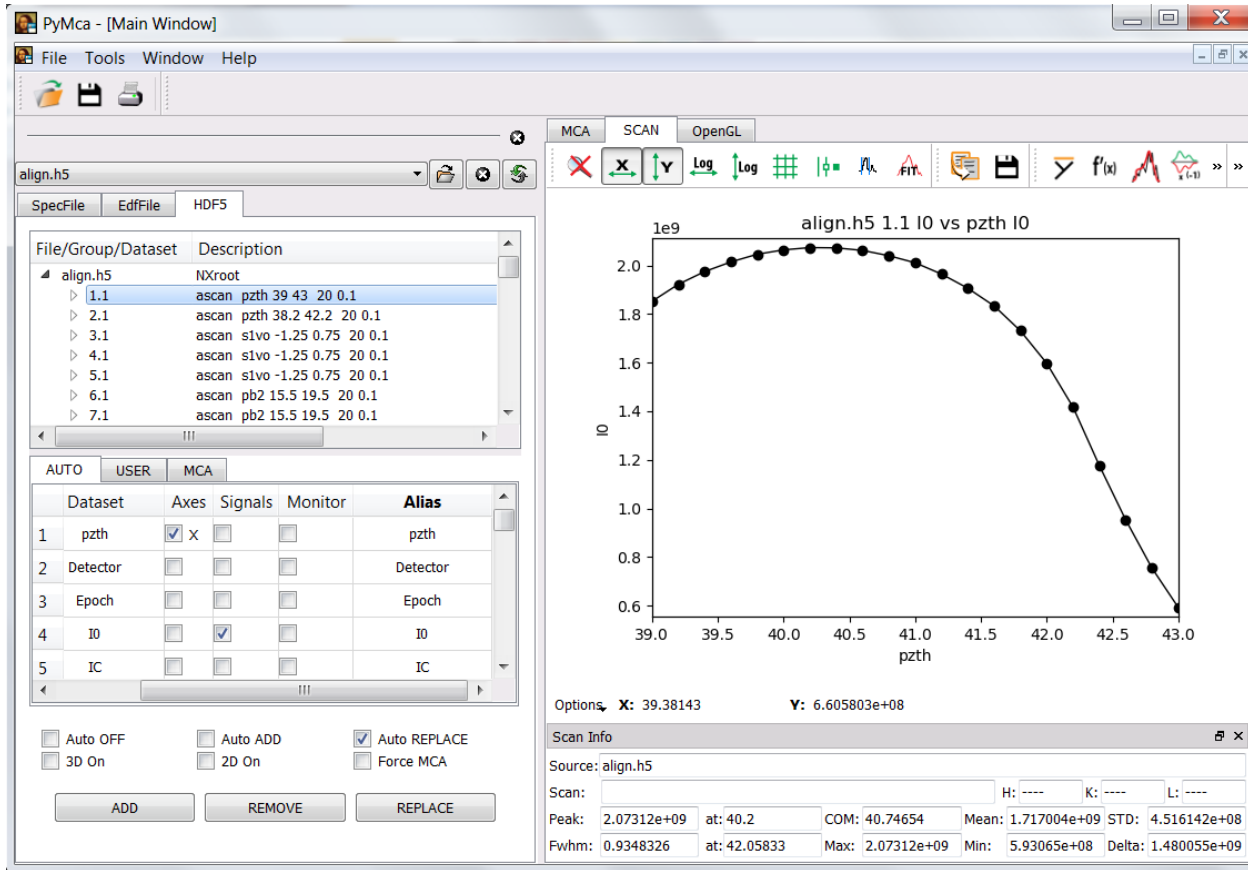
User/Scientist Domain

Administrative Domain (GDPR? DOI?)

Analysis Domain

Measurement Group Convention

- Name-based convention followed by ESRF and Sardana (MAX IV, ALBA...)



- Targets interactive use
- Applications can profit

Application Definitions

- Their goal is to enable interpretation and analysis of the data
- Unfortunately great ideas can be badly implemented

NXxas

entry

definition="NXxas"

start_time

title

instrument@NXinstrument

monochromator@NXmonochromator

energy

incoming_beam@NXdetector

data

absorbed_beam@NXdetector

data

.....

Application Definitions

- In 2010 it was communicated to the NIAC that multiple techniques were quite common (SAXS/WAXS, FLUO/DIFF,...)
- It was decided to create a new field NXsubentry containing the relevant information for each technique

ESRF only considers application definitions in subentries

- But the NIAC kept imposing all the rest of the structure in the subentry (see nexusformat.org documentation on NXsubentry)

The actual analysis applications do not need the structure !!!!!

Application Definitions as Understood by the ESRF

- Only the relevant part for the analysis required
- If they are actual items or links to items is irrelevant
- If there are no programs using the definitions the later are useless
- They should come from developers or communities (not just NIAC)

NXxas

entry

```
whatever_name@NXsubentry  
  definition="NXxas"  
  energy  
  i0  
  it
```

DISCLAIMER

I'm not advocating the use of NXxas (in any of the shown forms)

HDF5/NeXus: Requirements for Processed Data

- NeXus conventions are fairly clear in what concerns raw data
- How to store processed data in HDF5 files?
 - Needs
 - Program used
 - Configuration parameters
 - Results
 - Minimize file creation
 - More than one data treatment step into the file
 - Describe data treatment sequence

NeXus: ESRF Implementation for Processed Data (v1)

- Goals can be achieved with “extended” NXprocess groups

entry

start_time

end_time

title

process_name@NXprocess

program_name

version

date

sequence_index

configuration@NXcollection if HDF5 supported by program

configuration@format=“ini” or “json” or ... if string

results@NXcollection or NXdata if plot

Just a name based convention added to NXprocess

NeXus: ESRF Implementation for Processed Data (v2)

- A 100% pure NeXus way to specify the configuration: NXnote entry

`process_name@NXprocess`

`program_name`

`version`

`date`

`sequence_index`

`configuration@NXnote`

`file_name`

`type`

`data`

`results@NXcollection` or `NXdata` if just a plot

**The key point is that the configuration can be used back.
We have to be able to feed the original program with it.**



NeXus

ICAT

- Clear mapping from existing NeXus conventions to ICAT
 - ICAT key = Class1Class2Class3_dataset@attribute
 - NeXus current and mode in class Source inside class Instrument:
 - InstrumentSource_current
 - InstrumentSource_mode
- Technique or beamline specific information as NXsubentry based keys

```
<group NX_class="NXsubentry" groupName="EM">
  <protein_acronym ESRF_description="Protein acronym" NAPIttype="NX_CHAR">${EM_protein_acronym}</protein_acronym>
  <voltage ESRF_description="Voltage" NAPIttype="NX_CHAR">${EM_voltage}</voltage>
  <magnification ESRF_description="Magnification" NAPIttype="NX_CHAR">${EM_magnification}</magnification>
  <images_count ESRF_description="Number of images in movie" NAPIttype="NX_CHAR">${EM_images_count}</images_count>
  <position_x ESRF_description="Position X" NAPIttype="NX_CHAR">${EM_position_x}</position_x>
  <position_y ESRF_description="Position Y" NAPIttype="NX_CHAR">${EM_position_y}</position_y>
  <dose_initial ESRF_description="Dose initial" NAPIttype="NX_CHAR">${EM_dose_initial}</dose_initial>
  <dose_per_frame ESRF_description="Dose per frame" NAPIttype="NX_CHAR">${EM_dose_per_frame}</dose_per_frame>
  <spherical_aberration ESRF_description="Spherical aberration" NAPIttype="NX_CHAR">${EM_spherical_aberration}</spherical_aberration>
  <amplitude_contrast ESRF_description="Amplitude contrast" NAPIttype="NX_CHAR">${EM_amplitude_contrast}</amplitude_contrast>
  <sampling_rate ESRF_description="samplingRate" NAPIttype="NX_CHAR">${EM_sampling_rate}</sampling_rate>
</group>
```

- Acquisition
 - SPEC
 - Not worth native output. Use *silx convert* if desired
 - Bliss
 - NeXus native output operational but concurrent access issues
 - Studying to externalize via REDIS + memcached
 - Data Analysis and not Control responsible of data writing?

Status of NeXus @ ESRF – Data Analysis

- Data Analysis
- ✓• Capability to read HDF5 files (preferred data analysis I/O format)
- ✓• Unified API to access all data formats
- ✓• Support of NeXus NXdata I/O in viewers and analysis codes
- ✓• Provide provenance via NXprocess (pyFAI, PyMca, PyNX,...)
- Only one NeXus application definition supported (NXcxi)

- Data Policy and NeXus
 - Mirror ICAT and NeXus master file done
 - External links between master file and raw HDF5 files desirable

Ideally one should aim at processing a dataset from its master file

The Weight of Legacy

Adoption of HDF5/NeXus has been slower at the ESRF than at other synchrotrons due to the raw data being acquired in different formats. Detector output in HDF5 and the deployment of Bliss are speeding things up.

User experience with HDF5 files has to be better than with legacy formats HDF5 should not be the question but the answer.

Concerning data analysis, ESRF started to provide HDF5 support in 2009. Currently making convenient **use of the NeXus formalism as output and as integral part of the ESRF data policy.**

**Thank you for your
attention!**

