The NeXus Data Format for muon Spectroscopy and Neutron or X-ray Scattering

Mark Könnecke

Paul Scherrer Institute
Switzerland

October 26, 2010

The Predicament of the Traveling Scientist

- A different data format wherever she goes

PAUL SCHERRER INSTITUT

PSI

The Predicament of the Traveling Scientist

- A different data format wherever she goes
- Spends lots of time converting formats or writing readers

PAUL SCHERRER INSTITUT

PSI

The Predicament of the Traveling Scientist

- A different data format wherever she goes
- Spends lots of time converting formats or writing readers
- Waits even longer to load data from inefficient data formats

PAUL SCHERRER INSTITUT

PSI

The Predicament of the Traveling Scientist

- A different data format wherever she goes
- Spends lots of time converting formats or writing readers
- Waits even longer to load data from inefficient data formats
- DA requires N files in different formats, notes, local knowledge

PAUL SCHERRER INSTITUT

The Predicament of the Traveling Scientist

- A different data format wherever she goes
- Spends lots of time converting formats or writing readers
- Waits even longer to load data from inefficient data formats
- DA requires N files in different formats, notes, local knowledge
- Cannot read her collaborators data

PAUL SCHERRER INSTITUT

The Predicament of the Traveling Scientist

- A different data format wherever she goes
- Spends lots of time converting formats or writing readers
- Waits even longer to load data from inefficient data formats
- DA requires N files in different formats, notes, local knowledge
- Cannot read her collaborators data
- Has to keep extra information in yet another form

PAUL SCHERRER INSTITUT

PSI

- Definition of a standard data format
  - Rules
  - Validation tools
- Promotion of NeXus
  - Documentation
  - NeXus API
  - Outreach to the scientific community

PAUL SCHERRER INSTITUT

PSI

NeXus Design

- Complete data for typical use
- Extendable, add additional data as you please
- Self describing
- Easy automatic plotting
- Platform independent, public domain, efficient
- Suitable for a wild variety of applications

PAUL SCHERRER INSTITUT

NeXus Levels

1. Physical file format and API for accessing files
2. Rules for storing data in files
3. Component and application definitions
4. NeXus Utilities

PAUL SCHERRER INSTITUT

Physical File Format

- Portable, self describing, extendable, public domain
- Hierarchical data format, NCSA, HDF-4, later HDF-5
- HDF-5:
  - grouping support
  - on the fly compression
  - reading/writing subsets
  - first dimension appendable
  - Public domain C, F77 access library
  - Used by: NASA, Boing, the weathermen, ...
- XML for those who wish to edit their data

PAUL SCHERRER INSTITUT

PSI

- NeXus-API hides complex HDF API
- Transparent access to all three supported physical file formats
- ANSI-C implementation
- Bindings: C++, F77, Java, python, IDL, SWIG
- January, 4, 2010: 1311217 files processed at PSI alone

PAUL SCHERRER INSTITUT

Threads Versus NeXus

- Planned: NeXus is threadsafe when each thread has its own NXhandle
- A little work needs to be done to arrive there
- BUT: HDF-5 serialises access, no performance gain!
- Parallel HDF5, PHDF, with a different API
- PHDF requires: MPI, MPI-IO, parallel file system
- A new NeXus file driver for PHDF would be required
- Will only be implemented when the community really wants it

PAUL SCHERRER INSTITUT

PSI

NeXus Objects

- Files
- Groups identified by name and a classname beginning with NX
- Scientific data sets
- Attributes
- Links

PAUL SCHERRER INSTITUT

PSI

- McStas Coordinate System
- Angle based polar coordinate system
- NEW: full mapping imageCIF - NeXus now possible
- NEW: General axis and transformations

PAUL SCHERRER INSTITUT

NeXus General Rules

- NeXus reserves the prefix NX for group names.
- Store as much as possible
- A NeXus file has one to many NXentry groups
- There are two types of entries: raw data and processed data
- Multiple different techniques in one file go into separate NXsubentries
- If there is only one entry, the preferred name is entry, else entry1, entry2... entryn
- If an entry or an NXsubentry conforms to an application definition, the application definitions must be stated in the entries definition field.

PAUL SCHERRER INSTITUT

PSI

entry:NXentry
- sample:NXsample
- instrument:NXinstrument
  - source:NXsource
  - velocity_selector:NXvelocity_selector
  - detector:NXdetector
    - data[xsize,ysize], signal=1 (1)
- control:NXmonitor
  - data
- data:NXdata
  - link to (1)

PAUL SCHERRER INSTITUT

NeXus Processed Data File Structure

entry:NXentry
- sample:NXsample
- processing_name:NXprocess
  - program
  - version
  - parameters:NXparameter
    - raw_file
- data:NXdata
  - data[nx,ny,nz], signal=1

PAUL SCHERRER INSTITUT

NEW: NXsubentry

entry:NXentry
- sample:NXsample
- instrument:NXinstrument
- .....
- sas:NXsubentry
  - sample:NXsample
  - instrument:NXinstrument
    - source:NXsource
    - velocity_selector:NXvelocity_selector
    - detector:NXdetector
      - data[xsize,ysize], signal=1 (1)
  - control:NXmonitor
    - data
  - data:NXdata
    - link to (1)

PAUL SCHERRER INSTITUT PSI

NEW: NXcollection

entry,NXentry
- measurement:NXcollection
  - positions:NXcollection
    - om
    - two_theta
  - scalars:NXcollection
    - title
    - wavelength
  - data:NXdata
    - detector1
    - mca5

Why the Hierarchy??

- Supports self description and allows short names in components

PAUL SCHERRER INSTITUT

Why the Hierarchy??

- Supports self description and allows short names in components
- Name, classname pair allows for multiple components of the same type

PAUL SCHERRER INSTITUT

PSI

Why the Hierarchy??

- Supports self description and allows short names in components
- Name, classname pair allows for multiple components of the same type
- NXentry allows for multiple datasets in the same file

PAUL SCHERRER INSTITUT

Why the Hierarchy??

- Supports self description and allows short names in components
- Name, classname pair allows for multiple components of the same type
- NXentry allows for multiple datasets in the same file
- NXdata supports automatic plotting

PAUL SCHERRER INSTITUT

Why the Hierarchy??

- Supports self description and allows short names in components
- Name, classname pair allows for multiple components of the same type
- NXentry allows for multiple datasets in the same file
- NXdata supports automatic plotting
- Take care once when writing, use n times

PAUL SCHERRER INSTITUT

PSI

Rules for Storing Data Items

- Store physical values in C storage order
- Use NeXus components and dictionary names
- Missing names will be quickly accepted by the NIAC
- Names: full words separated by _
- Specify units in same format as used by UDunits
- Application definitions may restrict units

PAUL SCHERRER INSTITUT

NEW: Data Storage Exceptions

- There are situations were data has to be dumped as fast as possible in order to keep up with a high data rate. Or to save disk space.
- **Data not in C storage order:** use attributes stride and offset to describe the memory layout of the data.
- **Data needs scaling:**Use a NXformula group to specify a formula in muParser notations plus the parameters and data necessary to do the scaling.
- Details on both methods will be in the NeXus manual

PAUL SCHERRER INSTITUT

PSI

Associating Axis and Data, Preferred method

entry:NXentry
    data:NXdata
        data[nx,ny,nz], signal=1, axes=x_axis,y_axis,z_axis
        x_axis[nx]
        y_axis[ny]
        z_axis[nz]

PAUL SCHERRER INSTITUT

- Preserve original dimensionality of detector, if possible
- Time-of-flight becomes last dimension
- Highly irregular detectors:

```
entry:NXentry
        instrument:NXinstrument
                detector:NXdetector
                        data[ndet], signal=1
                        polar_angle[ndet], axis=1
                        azimuthal_angle[ndet]
                        distance[ndet]
```

PAUL SCHERRER INSTITUT

- Come in all shapes and sizes
- Captured by rules:
  - Store all varied parameters as arrays of length NP at the appropriate place in the NeXus hierarchy
  - For multi detectors, NP, number of scan points is always the first dimension
  - In NXdata: create links to counts and varied variables

PAUL SCHERRER INSTITUT

Scan Example 1: rotating sample

entry:NXentry
- sample:NXsample
  - rotation_angle[NP], axis=1 (1)
- instrument:NXinstrument
  - detector:NXdetector
    - data[NP],signal=1 (2)
- control:NXmonitor
  - data[NP]
- data:NXdata
  - link to (1)
  - link to (2)

PAUL SCHERRER INSTITUT

Scan Example 2: complex scan in Q

entry:NXentry
- sample:NXsample
  - rotation_angle[NP], axis=1 (1)
  - phi[NP], axis=1 (2)
  - chi[NP], axis=1 (3)
  - h[NP], axis=1 (4), primary=1
  - k[NP], axis=1 (5)
  - l[NP], axis=1 (6)
- instrument:NXinstrument
  - detector:NXdetector
    - data[NP],signal=1 (7)
    - polar_angle[NP],signal=1 (8)
- data:NXdata
  - link to (1)
  - link to (2)
  - link to (...)
  - link to (8)

PAUL SCHERRER INSTITUT

Scan Example 3: sample rotation, area detector

```
entry:NXentry
        sample:NXsample
                rotation_angle[NP], axis=1 (1)
        instrument:NXinstrument
                detector:NXdetector
                        data[NP,xsize,ysize],signal=1 (2)
        control:NXmonitor
                data[NP]
        data:NXdata
                link to (1)
                link to (2)
```

PAUL SCHERRER INSTITUT

PSI

- This is rastering a sample at different wavelengths, positions etc.
- Same treatment as scans, NP replaced by NR number of raster points
- For the common case of rastering on a 2D grid one can store [nx,ny,detdim]. Be aware, though, that this causes problems if the rasterisation is aborted in mid operation.

PAUL SCHERRER INSTITUT

NeXus Component and Application Definitions

- Component definitions: dictionaries of allowed field names for the various NeXus groups
- Application Definitions
  - Define what has to be in a NeXus file for a certain application
  - Defines standards
  - Another view: Contract between file producers and users about what has to be in a NeXus file for a well defined purpose
  - Validation by NXvalidate
- Written in NeXus Definition Language, NXDL

PAUL SCHERRER INSTITUT PSI

All Base Classes

NXaperture
NXattenuator
NXbeam_stop
NXbeam
NXbending_magnet
NXcharacterization
NXcollimator
NXcrystal
NXdata
NXdetector
NXdisk_chopper
NXentry
NXenvironment
NXevent_data
NXfermi_chopper
NXfilter
NXflipper
NXgeometry
NXguide
NXinsertion_device
NXinstrument
NXlog
NXmirror
NXmoderator
NXmonitor
NXmonochromator
NXnote
NXorientation
NXparameters
NXpolarizer
NXprocess
NXsample
NXsensor
NXshape
NXsource
NXtranslation
NXuser
NXvelocity_selector

PAUL SCHERRER INSTITUT

Application Definition Process

1. Construct an application definition with advice from the NIAC
2. You can also inherit from and extend an existing definition
3. Cure for a year; data should be produced in the new format in this time
4. After curation and review: this is the standard for this application type.

- No promises, but the NIAC may do it for you
  - Description of experiment
  - Minimum set of data items necessary form common use
  - Example data

PAUL SCHERRER INSTITUT

Four Steps

1. Think! what ought to go into the file
2. Map this into the NeXus file structure
3. Cast this mapping into a NXDL file
4. Standardize your application definition together with the NIAC

PAUL SCHERRER INSTITUT

Think!

- What has to go into the file?
- Minimum data necessary for common usage scenarios
- Haggle it out with your community
- Coverage ratio: > 80 % of use cases

PAUL SCHERRER INSTITUT

PSI

Map to NeXus

- Consider into which NeXus group an item might belong
- Look in the base class for a suitable data field
- Link the data items required for the default plot into NXdata

PAUL SCHERRER INSTITUT

From an application definition to a real NeXus file

- The structure defined by the application definition is the minimum

PAUL SCHERRER INSTITUT

From an application definition to a real NeXus file

- The structure defined by the application definition is the minimum
- Practical files strive to capture much more

PAUL SCHERRER INSTITUT

From an application definition to a real NeXus file

- The structure defined by the application definition is the minimum
- Practical files strive to capture much more
- Suggested procedure:
  - Look at each of your instruments components and the matching NeXus base class
  - Add whatever you feel like adding or the instrument scientists wants to have
  - Add whatever management wants to have (may be not in a NeXus group)

PAUL SCHERRER INSTITUT

From an application definition to a real NeXus file

- The structure defined by the application definition is the minimum
- Practical files strive to capture much more
- Suggested procedure:
  - Look at each of your instruments components and the matching NeXus base class
  - Add whatever you feel like adding or the instrument scientists wants to have
  - Add whatever management wants to have (may be not in a NeXus group)
- Remember: Adding more fields does not break application definition compliance!

PAUL SCHERRER INSTITUT

NEW: Available NeXus Application Definitions

- NXarchive
- NXreftof
- NXtas
- NXtomophase
- NXxnb
- NXtomoproc
- NXindirectof
- NXsastof
- NXtofsingle
- NXmonopd
- NXsas
- NXtofraw
- NXxeuler
- NXxrot
- NXtofsingle
- NXiqproc
- NXsqom
- NXxas
- NXrefscan
- NXscan
- NXtomo
- NXxkappa
- NXiqproc
- NXdirectof
- NXlauetof
- NXtofraw
- NXxasproc

PAUL SCHERRER INSTITUT

Challenge 1 in science you are supposed to do new, non standard, things. These of course cannot be easily cast into a standard.

Data Format Challenges

Challenge 1 in science you are supposed to do new, non standard, things. These of course cannot be easily cast into a standard.

Challenge 2 in order to establish a standard a lot of people need to agree

PAUL SCHERRER INSTITUT

Data Format Challenges

Challenge 1 in science you are supposed to do new, non standard, things. These of course cannot be easily cast into a standard.

Challenge 2 in order to establish a standard a lot of people need to agree

Challenge 3 a standard requires scarce scientific programming resources for adoption

PAUL SCHERRER INSTITUT

Data Format Benefits

Benefit 1 By using a discoverable data format like NeXus, XML, HDF-5, people can at least figure out what is in the data file.

PAUL SCHERRER INSTITUT

PSI

Data Format Benefits

Benefit 1 By using a discoverable data format like NeXus, XML, HDF-5, people can at least figure out what is in the data file.

Benefit 2 Using predefined names from a dictionary gives meaning to the data in a file.

PAUL SCHERRER INSTITUT

Data Format Benefits

Benefit 1 By using a discoverable data format like NeXus, XML, HDF-5, people can at least figure out what is in the data file.

Benefit 2 Using predefined names from a dictionary gives meaning to the data in a file.

Benefit 3 Using a shared API reduces learning costs and increases application stability.

PAUL SCHERRER INSTITUT

PSI

Data Format Benefits

Benefit 1 By using a discoverable data format like NeXus, XML, HDF-5, people can at least figure out what is in the data file.

Benefit 2 Using predefined names from a dictionary gives meaning to the data in a file.

Benefit 3 Using a shared API reduces learning costs and increases application stability.

Benefit 4 With NeXus, HDF-5 plus professional programming techniques a DA application can read any file which contains the required data.

PAUL SCHERRER INSTITUT

PSI

Data Format Benefits

Benefit 1 By using a discoverable data format like NeXus, XML, HDF-5, people can at least figure out what is in the data file.

Benefit 2 Using predefined names from a dictionary gives meaning to the data in a file.

Benefit 3 Using a shared API reduces learning costs and increases application stability.

Benefit 4 With NeXus, HDF-5 plus professional programming techniques a DA application can read any file which contains the required data.

Benefit 5 Storing as much data as possible increases the likelihood that the needed data is actually on file, even for unforeseen uses.

PAUL SCHERRER INSTITUT

PSI

Data Format Benefits

Benefit 1 By using a discoverable data format like NeXus, XML, HDF-5, people can at least figure out what is in the data file.

Benefit 2 Using predefined names from a dictionary gives meaning to the data in a file.

Benefit 3 Using a shared API reduces learning costs and increases application stability.

Benefit 4 With NeXus, HDF-5 plus professional programming techniques a DA application can read any file which contains the required data.

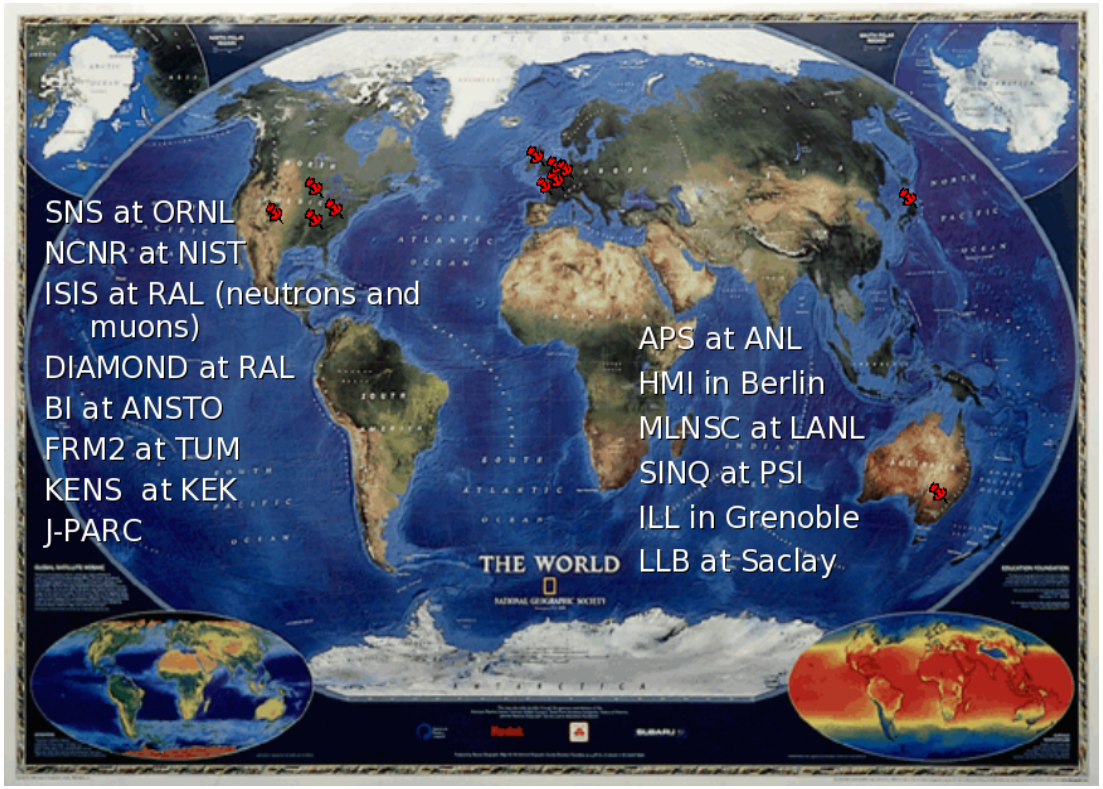
Benefit 5 Storing as much data as possible increases the likelihood that the needed data is actually on file, even for unforeseen uses.

Benefit 6 Application Definitions

PAUL SCHERRER INSTITUT

PSI

Who commits to NeXus?



PAUL SCHERRER INSTITUT

PSI

What You Can Do with NeXus

1. Store and archive data from a wild variety of instruments
2. Store processed data
3. Store a complete workflow from raw data to publication ready data in several NXentries in one file
4. Store a set of related experiments in one file
5. Define strict and validatable standards

PAUL SCHERRER INSTITUT

PSI

NeXus Outlook

- New systems tend to use NeXus
- No competitor for a general purpose data format
- Planned:
  - Refine application definitions together with communities
  - Release application definitions and NXvalidate
  - Update manuals and the NeXus WWW-site
  - www.nexusformat.org, embarrasingly outdated, download manual

PAUL SCHERRER INSTITUT

PSI