NeXus Application Definitions Tutorial

Mark Könnecke

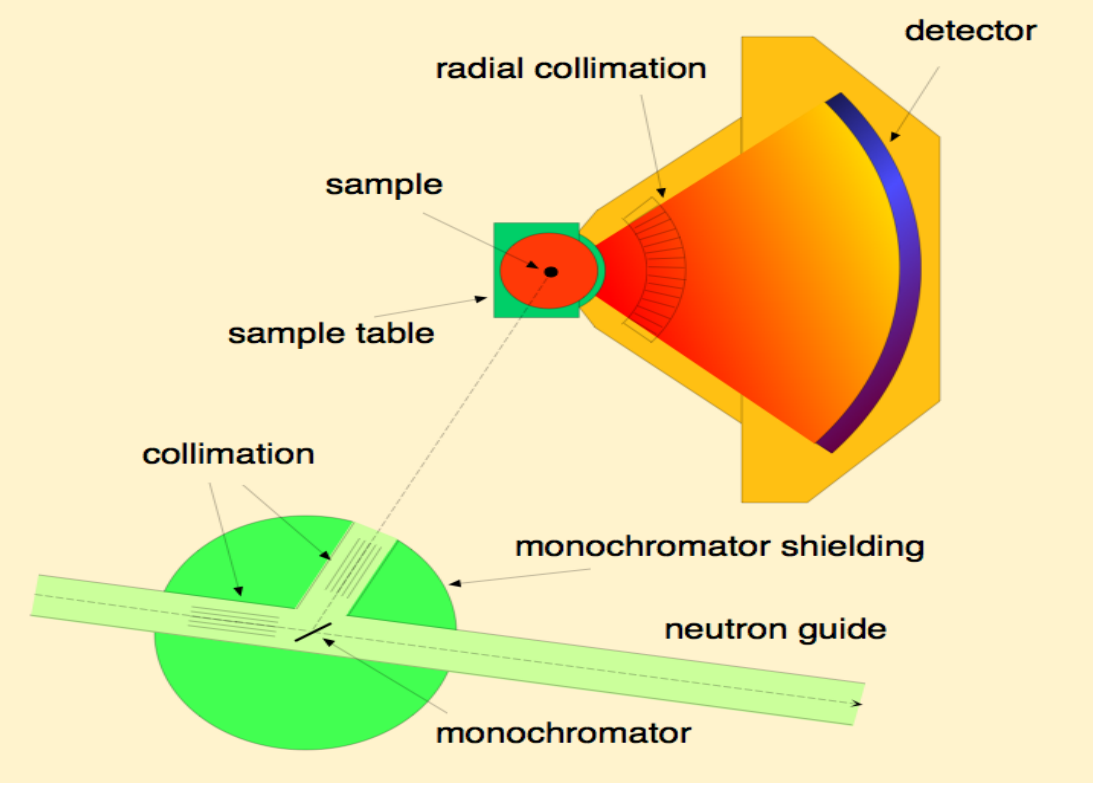
Paul Scherrer Institute
Switzerland

May 7, 2010

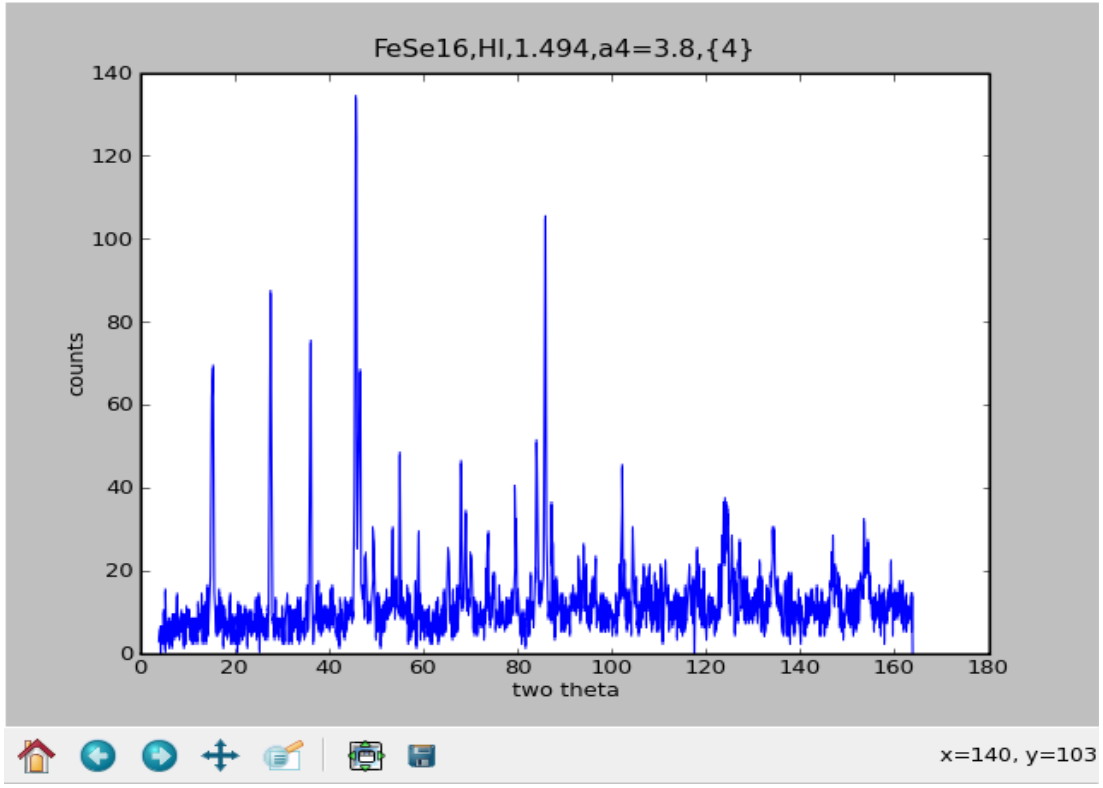
Your Task:

- Develop an application definition for WONI at INIS
- WONI = WOnderful New Instrument
- INIS = INsanely Intense Source

WONI Design


detector
radial collimation
sample
sample table
collimation
monochromator shielding
neutron guide
monochromator

WONI Plot


FeSe16,HI,1.494,a4=3.8,{4}
counts
two theta
0
20
40
60
80
100
120
140
160
180
x=140, y=103

What You Need

- A copy of the current NeXus base class definitions
- A XML editor

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

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

NeXus

- Common usage is Rietveld analysis or profile analysis
- Data required:
  - Title
  - Sample name
  - Wavelength
  - Counts versus two_theta
  - Monitor, for normalisation

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

entry:NXentry
- title
- definition

Mappings for WONI

entry:NXentry
- title
- definition
- sample:NXsample
  - name

Mappings for WONI

entry:NXentry
- title
- definition
- sample:NXsample
  - name
- instrument:NXinstrument
  - monochromator:NXmonochromator
    - wavelength

Mappings for WONI

entry:NXentry
- title
- definition
- sample:NXsample
  - name
- instrument:NXinstrument
  - monochromator:NXmonochromator
    - wavelength
  - detector:NXdetector
    - data[ndet], signal=1 (1)
    - polar_angle[ndet], axis=1 (2)

Mappings for WONI

entry:NXentry
- title
- definition
- sample:NXsample
  - name
- instrument:NXinstrument
  - monochromator:NXmonochromator
    - wavelength
  - detector:NXdetector
    - data[ndet], signal=1 (1)
    - polar_angle[ndet], axis=1 (2)
- control:NXmonitor
  - data

Mappings for WONI

entry:NXentry
- title
- definition
- sample:NXsample
  - name
- instrument:NXinstrument
  - monochromator:NXmonochromator
    - wavelength
  - detector:NXdetector
    - data[ndet], signal=1 (1)
    - polar_angle[ndet], axis=1 (2)
- control:NXmonitor
  - data
- data:NXdata
  - link to (1)
  - link to (2)

Casting into NXDL: groups

```
<group type="NXsource" name=''source''>

</group>
```

Casting into NXDL: fields

```
<field name="data" type="NX_INT" signal="1">
     <doc>
       Some blabla
     </doc>
     <dimensions size="3">
         <dim index="1" value="np" />
         <dim index="2" value="number of x pixels" />
         <dim index="3" value="number of y pixels" />
     </dimensions>
     <attribute name="signal" type="NX_CHAR">
         <enumeration>
            <item value="1" />
          </enumeration>
      </attribute>
</field>
```

NeXus

- There is a XML schema for NXDL, the editor helps you!

- Forward you application defintion to the NIAC for review
- Correct you definition according to NIAC comments, if any
- Cure and use the definition for a year, data should be written and analysed in this year
- After a final review, this is the standard for that application. Period.

NeXus

From an application definition to a real NeXus file

- The structure defined by the application definition is the minimum

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

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)

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!

What about ONOKI?

- ONOKI: ONe Of a Kind Instrument

What about ONOKI?

- ONOKI: ONe Of a Kind Instrument
- Think! what you want to store

NeXus

What about ONOKI?

- ONOKI: ONe Of a Kind Instrument
- Think! what you want to store
- Map the list into the appropriate NeXus base classes. It helps to look at each of the components ONOKI is assembled from and to decide what you wish to store for each of them.

What about ONOKI?

- ONOKI: ONe Of a Kind Instrument
- Think! what you want to store
- Map the list into the appropriate NeXus base classes. It helps to look at each of the components ONOKI is assembled from and to decide what you wish to store for each of them.
- The next one to copy ONOKI is well advised to copy what you did NeXus file wise, otherwise she will not be able to reuse your software!

Validation

- Not quite ready yet
- nxvalidate nexusfile.hdf