

1 NeXus Project Plan

NeXus is a data exchange format for scientists working in the field of neutron or x-ray scattering and muSR spectroscopy. This document is meant for the eyes of the NIAC and people intimate with NeXus only. More information about NeXus can be found at the NeXus web site at: www.nexusformat.org

This document is a kind of whitepaper which describes where NeXus is going to move to in the next few years.

1.1 Current State

NeXus is currently the only broad scoped data format in the field of neutron and x-ray scattering and muSR spectroscopy. The real competition to NeXus are the numerous old home grown data formats still in use. CIF and its derivate imgCIF are a competition in some fields of application. But CIF never tried to address such a broad scope of instrument as NeXus does and there is no information that it will be developed to such extent. NeXus aims to replace the old incompatible file formats in order to facilitate data exchange and reuse of data analysis software.

This examination of the current state of NeXus is based on data available in early 2009. The current state can best be discussed following the five levels of NeXus.

1. The physical file format. With HDF-5 as physical file format NeXus includes an efficient state of the art binary and compressed file format suitable for large data sets. XML is a solution for those users dealing with smaller data sets or who have the requirement to touch up their data with text editors like emacs. Thus NeXus is well set up concerning the physical file format.
2. The NeXus API. The NeXus-API abstracts from the complexity of the HDF API and hides the multiple physical file formats from the user. It provides bindings for C, C++, Fortran, Java, IDL, python and other scripting languages. The NeXus-API can be considered to be complete, mature and to require maintainance only.
3. Rules for structuring information in the file. This is largely agreed upon and stable since a couple of years.
4. Rules for storing individual data items. This is agreed upon and stable since years.
5. NeXus base classes and application definitions. This is the definition of what actually has to be in a NeXus file for a given instrument raw file format or a processed data exchange format. In this area more work is required. The NIAC recently decided to simplify the definition standardization process and to modify the format in which to write definitions to an XML dialect called NXDL. The switch to NXDL is mostly done. What is mostly lacking is application definitions.

A general observation is that new projects, new facilities or new data analysis projects, use NeXus as a file format. Due to the lack of application definitions

many NeXus files currently written do not conform to standards. It seems to be difficult to make inroads at established sites with working data acquisition to data analysis pipelines. For such sites, the adoption of NeXus requires effort. And due to the general difficulty to get funding for scientific software construction the man power to make such changes is simply not there. NeXus suffers somewhat from this general problem. But eventually the pressure to overhaul the 30+ years old software base in many fields of application of NeXus will grow strong enough that something happens. NeXus must be ready then to jump the train.

There are some comments from the scientific community that NeXus does not deliver, mostly because the application definitions do not move forward. This has to be taken seriously. On the other hand a bunch of ~15 volunteers meeting once a year cannot solve the problems of all instruments and applications out there. The NIAC has promised too much. This has partially be remedied with the new application definition process decided upon at NIAC 2009.

1.2 Organisation

The development of NeXus is currently being overseen by the NeXus International Advisory Committee (NIAC). The NIAC strives to include members from any interested facility or party. Typically the NIAC meets once a year either in the US or Europe. The NIAC in its current form suffers from some problems:

- All its members work for NeXus part time, besides demanding obligations to the organisations they represent. This slows NeXus development considerably.
- The knowledge about NeXus details in the full NIAC varies wildly. There are the long time NeXus veterans who know everything very well and there are newcomers who know little. This leads to the fact that many members seem not to be able to follow discussions in the NIAC or ask to revisit former decisions because they are not aware of them and the background from which decisions originated. All this makes decision finding in the NIAC difficult and slow. The attempt to solve this through workgroups at NIAC meetings has just pushed the problem to the workgroup level.

From recent experience, NIAC 2008, it can be concluded that we have more productive NIAC meeting when fewer people are involved. Summing it up, it might be necessary to review the NIAC structure. When extrapolating the way the NIAC currently works a division into two entities suggests itself:

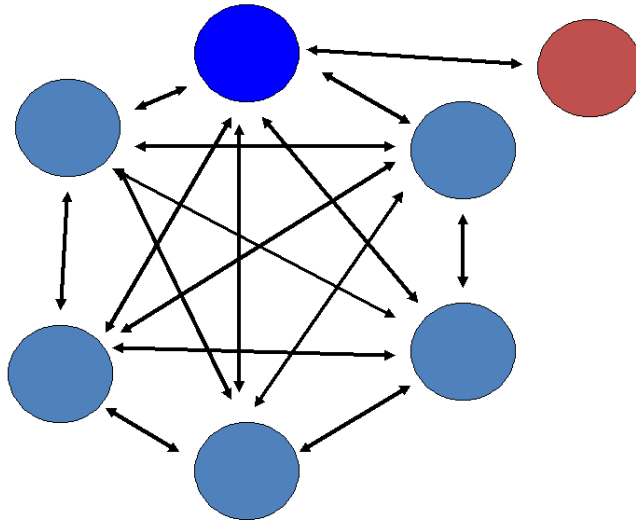
1. A NeXus-developers group which haggles out all the mostly technical details, base classes, API, definitions and possible object hierarchies. Members of this group will be chosen from NIAC members based on merit. Merit is based on constructive and useful input to either API or application definition development.
2. A larger NIAC with all people which decides upon policies, priorities, projects plans, who actually is a NeXus developer and such.

The first event where such a restructuring can be discussed and decided upon is the full NIAC meeting 2010.

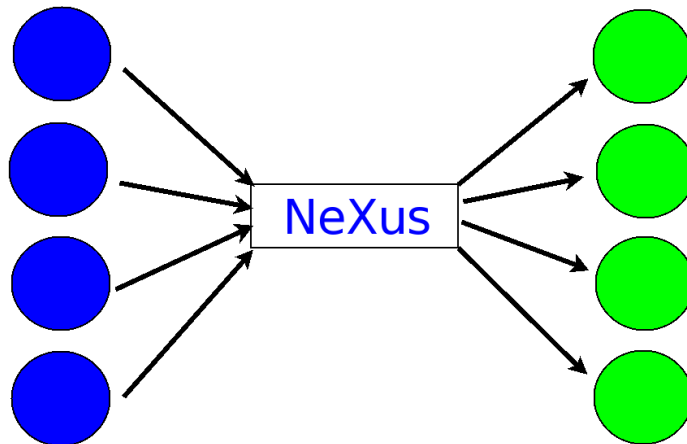
Every chance to get funding for NeXus development must be used. Experience shows that NeXus benefits much when someone is working fulltime at it.

1.3 NeXus Software

The current situation in scientific software concerning file formats and conversions can be characterized by this image:



All have to convert or read everything. But converting all software to be able to read NeXus files is a lot of work for which there is no funding as of now. Another use of NeXus may be more feasible:



In this scheme NeXus is used as an intermediate file format. Producers convert to NeXus and consumers convert from NeXus to whatever they need.

This is a single piece of software which could go a long way to solve a lot of data conversion problems. Thus I will call this the swiss army knife of file conversions, suggested name NXfiletool. The left half of the diagram is more or less covered by NXtranslate, the anything to NeXus converter. For the right hand side a configurable converter to ASCII would go a long way as many older file formats are ASCII. The project plan for NXfiletool looks like this:

1. A decision needs to be made by the NIAC if the development of NXfiletool shall be started.
2. A design document has to be drafted and agreed upon. Any design should at least reuse the existing plugins from NXtranslate.
3. Implementation of NXfiletool

1.4 Instrument Definitions

Concerning instrument definitions the NIAC to often got side tracked by the complicated, one of a kind instruments, and forgot to get the simple cases out the way. Clearly, there is an order of importance of applications of NeXus. The criteria for this ordering are:

1. Applications which occur both at x-ray and neutron sources in a similar form
2. Applications which occur at most sources of either kind, for example most neutron sources do triple axis, synchrotrons tend to have EXAFS.
3. Applications where people actually collaborate on data and share data analysis software

Clearly, instruments which are one-of-kind are encouraged to store their data in NeXus, but as little data exchange happens for such instruments, an application definition is less important.

There is a strong trend towards more and bigger area detectors; this implies that the NIAC should rather neglect single detector definitions.

The following list of application definitions constitute a project plan sorted by priority according to the criteria given above.

1.4.1 Application Definitions Shared By Neutron and X-Ray Sources

This is the most important set.

SANS/SAX small angle scattering is common at both x-ray and neutron facilities. The SAS community is actively trying to establish a new data format in their canSAS meetings. Thus SAS is a prime concern.

Monochromatic beam powder diffraction Another commonality. Covered by the NXmonopd definition.

Strain scanning is powder diffraction with additional data on the sample position.

Monochromatic reflectometry A draft definition exists and must be reviewed.

Tomography This is an application of NXscan with the sample rotation as scanned parameter and an area detector. The polar_angle of the area detector is unimportant but the distance to the sample matters.

Monochromatic single crystal diffraction and PX There are various popular geometries out there: rotation camera, eulerian cradle diffractometer, normal beam and kappa geometry. But basically these are applications of NXscan with different angles to be recorded.

General Scanning instrument covers many instruments. An application definition exists but needs additional documentation in order to be applicable.

1.4.2 X-Ray Special Application Definitions

As the number of synchrotron sources is higher than the number of neutron sources, these must be higher on the priority list.

EXAFS is an energy scan on four detectors. This can be quickly derived from NXscan. Extensions to the NXcrystal base class may be required to cover double crystal monochromators.

IR Microspectroscopy not enough information to judge. May be beyond the scope of NeXus.

1.4.3 Neutron Specific Application Definitions

Triple axis The triple axis community is extraordinarily conservative. It will be difficult to establish a new file format here. Except where new style triple axis machines with multiple analysers and area detectors are being built. Here we may just standardize on the RITA format used at PSI. Which is an application of NXscan.

Time-of-flight reflectometer available at most neutron facilities. A draft definition exists and requires review.

Time-of-flight powder diffraction another workhorse, mostly covered by NXTofRaw.

Time-of-flight SANS another workhorse.

Time-of-flight instruments in general There is a definition for a general time-of-flight instrument named NXTofRaw. TOF instruments these days tend to be built with an ever increasing array of detectors, often arranged in weird shapes. Currently NeXus differentiates between various detector shapes. But this may not be incredibly useful for sharing software. Software reuse may happen if all can agree to use the lowest common nominator, the description of each detector element in McStas coordinates. At this time, TOF data analysis is being redefined in the mantid and DANSE projects. A meeting should be called ASAP with the authors of mantid, DANSE and possibly DAVE to arrive at a common general format (or sets of formats) for time-of-flight instrumentation.

1.4.4 muSR Applications

Relatively few instruments and sources exist. An application definition by Stephen Cotrell exists and is being adapted at other facilities.

1.4.5 Processed Data Applications

SANS/SAX-Reflectometry the methods are different but the processed data in both cases is $I/1/Q$.

Powder Diffraction same as raw data.

Inelastic scattering energy transfer against counts.

Inelastic scattering 2 survey instrument may store $S(Q,e)$ which is three dimensions for Q and another for energy transfer.

Single crystal diffraction The result of a data reduction in single crystal crystallography is a list of miller indices with structure factors and errors on the structure factors. Typically this is stored in ASCII. But with the thousands of reflections processed in PX, there may be a market for an efficient binary format.

The NIAC is not entirely clear about how processed data shall be stored. Surely there will be an NXentry group which has enough groups to contain the meta data and the sample information. The NIAC also agreed upon a NXprocess group which holds information about the programs and parameters which resulted in this processed data set. The NXinstrument group provides further groups and fields if additional information about the instrument needs to be carried along. But processed data is not really detector data. It is derived of that. So it must be discussed if such data should not just simply be stored in NXdata groups.