NeXus Project Plan
DRAFT

May 11, 2009

NeXus is a data exchange format for scientists working in the field of neutron or xray scattering and muSR spectroscopy. More information about NeXus can be found at the NeXus web site at: www.nexusformat.org

The intended audience for this document are the members of the NIAC and other parties interested in the development of NeXus.

This document is a kind of whitepaper which describes where NeXus is going to move to in the future. Projects in the various categories are ordered by priority. In addition, the NeXus developers will focus on application definitions and NXvalidate. As NeXus ist still dependent on voluntary contributions of effort, progress is hard to predict. There is a suggestion for this project plan to be valid until NOBUGS 2010.

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 derviate imgCIF are a competion 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 bnuch 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.

2 The Actual Project Plan

2.1 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 choosen 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.

2.2 NeXus Software

While the NeXus-API is largely mature, the NeXus effort could benefit from some additional support software. Three applications are currently under discussion:

NXvalidate a NeXus file validator

NXplot a generalized NeXus file plotting tool.

NXfiletool a anything to NeXus and neXus to anything converter.

Of these two NXvalidate is the most important one and should be addressed as soon as possible.

2.2.1 NXvalidate

NXvalidate takes as input a NeXus file and an application definition as an NXDL file. The program then proceeds to test the NeXus file for validity against the application definition and complains about any misfits. There are two user groups for NXvalidate. Data file producers will want to use NXvalidate in order to verify that the files they write adhere to a given application definition. Data analysis software providers can use NXvalidate in order to verify if an incoming NeXus file is valid for their application. A protoype NXvalidate based on the old meta-DTD description format already exists and has to be modified or rewritten to work with NXDL. The validation will most certainly be done in a two step process:

1. The NXDL will be XLST transformed to a XML schema and the NeXus file will then be validated using schema tools after conversion to XML.

2. A second application will test for all those validations which cannot be done with XML schema.

As NXvalidate really helps to make the application definitions fly its development has the highest priority.

2.2.2 NXplot

Since long, people have dreamt off an application which can load any NeXus compliant file and display a default plot of the data in it. For the time being this application is called NXplot. Several attempts have been made to create such an application but none reached maturity. The most promising of the attempts is NXvalid (not to be confused with NXvalidate described above). In order to arrive at NXplot, the following steps have to performed:

1. Review past attempts to create an NXplot.

2. Write a whitepaper about the desired feauture set for NXplot. Questions to be answered include:

   (a) Shall it be restricted to well formed data: minimum requirement: a data set with the attribute signal=1 in the group or shall it allow to plot arbitrary data?

   (b) Shall it be extendable by plugins in order to form the basis for a more general data analysis application?

3. Acually write the code.

2.2.3 NXfiletool

The current situation in scientific software concerning file formats and conversions can be characterized by Figure 1.

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 as described in Figure 2 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. But NeXus as an output format is also interesting. Uses may be the rewriting of NeXus files or splitting multi entry NeXus files into separate ones. 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.

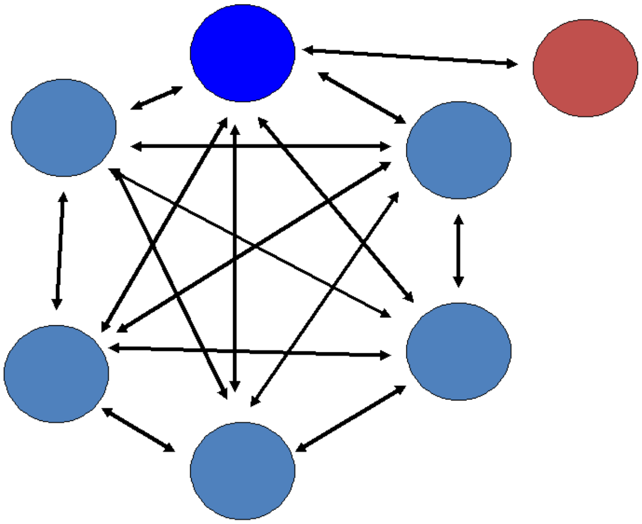

Figure 1: The data format world without NXfiletool

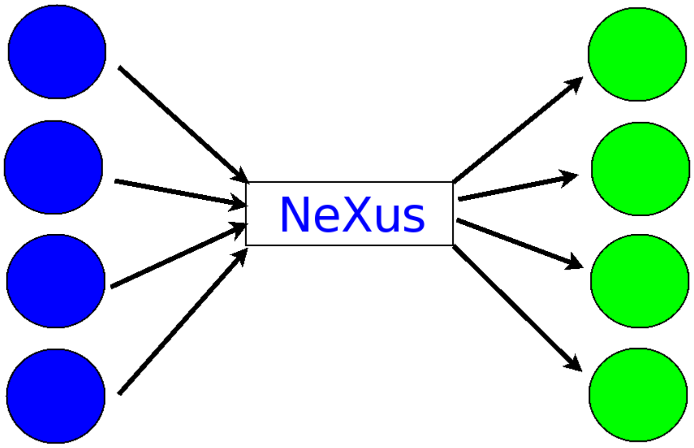

Figure 2: The data format world with NXfiletool

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

2.3 Application Definitions

A NeXus application definition describes the content of a NeXus file for raw or processed data from a certain kind of instrument or technique. NeXus application definitions are described through an XML file, a NeXus Definition Language File (NXDL).

A NeXus application definition is really a contract between a NeXus file producer, like a facility writing raw data, and a NeXus file consumer, like a data reduction or data analysis software author. The contracts content is that the NeXus file contains enough information to make a certain kind of processing possible. The contract also contains details about where to find the requested information in the NeXus file.

A given NeXus file can easily contain enough data to satisfy two or more NeXus application definitions. A powder diffraction data file, for example, may adhere to the application definition for powder diffraction but also a definition covering the requirements of a file indexing and archiving software.

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
4. User groups which actively ask for a NeXus application definition in order to adopt it shall be given precedence.

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 off application definitions constitute a project plan sorted by priority according to the criteria given above.

2.3.1 Application Definitions for Raw Data Files

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.

PX and monochromatic single crystal diffraction 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. The PX community is currently using imgCIF. However, with the advent of even larger and faster area detectors the current usage where each image is stored in a separate file becomes unworkable. Here NeXus can come to the rescue.

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

X-ray Absorption Spectroscopy is an 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.

Time-Of-Flight A general raw file format for time-of-flight neutron scattering. This covers a lot of diverse instruments like TOF-SANS, TOF-reflectometry, inelastic TOF instruments and TOF-powder and single crystal diffraction.

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

X-ray photoemission spectroscopy At first glance another application of NXscan for an area detector.

2.3.2 muSR Applications

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

2.4 Application Definitions for Processed Data

Powder diffraction This happens to be the same as the raw file format for monochromatic powder diffraction, for TOF it similar to the monochromatic powder diffraction format but replacing two theta of the detector against d.

Protein Cystallography The result from a protein crystallography experiment is typically a reflection list stored in CIF format. Now, CIF is ASCII and in PX often thousands of reflections neend to be processed. If a better performing file format then ASCII is required here, NeXus comes to the rescue.

SANS/SAX, Reflectometry The methods are different but after data reduction the data is similar: Intensity versus 1/Q basically.

Inelastic neutron scattering Many instrument deliver in the end data in the form of Intensity versus energy transfer.

Inelastic survey instruments Such instruments survey a larger volume of Q and energy transfer space. This is typically a 4D volume of data.

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.

3 Opportunities and Challenges

The data reduction process for time of flight neutron scattering is currently being redefined within the mantid, DANSE and DAVE projects. The NIAC is well advised to seek active collaboration with these groups in order to unify their data storage requirements.

In x-ray scattering experiments are frequently performed by storing individual images of a scan in separate files. With the advent of bigger and faster new detectors, most notably the Pilatus detectors, this data storage scheme falls over. Storing indiviudal files is slow compared to a bulk transfer or a series of images and moreover the many small files pose a handling and organisation problem. Here NeXus with its efficient and flexible HDF-5 based storage model can take over.

Slowly but steadily the pressure is building up to revise and reimplement the 30+ years old software used in many facilities for data analysis. There is even a rumour of a EU funding round for sceintific software. If this comes true NeXus should be ready to solve the data storage requirements.