A NEW FLEXIBLE INTEGRATION OF NeXus DATASETS TO ANKA BY FUSE FILE SYSTEMS*

W. Mexner, D. Ressmann, T. Spangenberg, H.Pasic ANKA,
Karlsruhe Institute of Technology, Germany
E. Iurchenko Institute of Cybernetics, Tomsk Polytechnic University, Russia

Abstract

In the high data rate initiative (HDRI[1]) German accelerator and neutron facilities of the Helmholtz Association agreed to use NeXus[2] as a common data format. The synchrotron radiation source ANKA[3] decided in 2012 to introduce NeXus as common data format for all beam lines. Nevertheless it is a challenging work to integrate a new data format in existing data processing work flows. Scientists rely on existing data evaluation kits which require specific data formats. To solve this obstacle, for linux a file system in userspace (FUSE)[4] was developed, allowing to mount NeXus-Files as a file system. Easy in XML configurable filter rules allow a very flexible view to the data. Tomography data frames can be directly accessed as TIFF[5] files by any standard picture viewer or scan data can be presented as a virtual ASCII[6] file compatible to spec[7].

INTRODUCTION

ANKA is a synchrotron light source located at the Karlsruhe Institute of Technology, providing light from hard X-rays to the far-infrared for research and technology. It serves as a user facility for the national and international scientific community. Within the High Data Rate Initiative (HDRI) of the German Helmholtz research program Photons, Neutrons, and Ions (PNI) a coordinated and common approach was agreed towards developing and providing the appropriate hardware and software tools to cope with high data rates. Within this program ANKA and other PNI facilities agreed to introduce NeXuS based on the Hierarchical Data Format HDF5[8] as standard HDRI data format to store scientific data. As the NeXus API is not thread save, a re-write of the NeXus API developed by E.Wintersberger (DESY) has been released. Libpninx[9] provides an alternative C++ API for the NeXus file format. Actually only HDF5 is supported as backend file format. Based on this development it is planned to use NeXus as a primary file format for all scientific data sets written to persistent storage. As the first use case the tomography station at image beam line was selected to introduce NeXus at ANKA. For using NeXus at ANKA the main obstacle is, that for data processing some existing software can not handle HDF5 files. In case of some commercial products it depends on the company, if HDF5 support is added. Some companies like volumegraphics[10] or IgorPro[11], used for spectroscopic applications at ANKA, already support HDF5. Some other software is simply to old (e.g. fortran libraries) or even the source code is missing. An interesting effort to avoid such obstacles in the future is the CDMA project[12]. An actual work around is simply to extract the NeXus file in the required format. In case of small files here the additional effort is negligible, but a typical tomography dataset has a size of around 50 Gbyte, so saving and reading is doubling the network or disk traffic.

Instead of doubling the disk traffic it is more convenient to display a NeXuS file as a virtual file system with the expected data format for the existing application. In the linux community, file system in user space (FUSE) is a well known technology for writing virtual file systems. Unlike traditional file systems that essentially save data to and retrieve data from disk, virtual file systems do not actually store data themselves. They act as a view or translation of an existing file system or storage device. Some well known use cases are the NTFS or UDF file system drivers for linux. As a NeXus file is highly hierarchic structured, it is easy to display it as a file system with data files in the expected application format.

SOFTWARE DESCRIPTION

The NeXus FUSE application[13] is designed to get a NeXus file mounted as a file system (see Fig 1.) The software has been developed on an openSUSE Linux version 12.1(amd64) and tested on openSUSE 12.3 (amd64) as well as Debian 7 (amd64). The content of a NeXus file is represented in a user-friendly way. Filter rules based on an XML file are used. The XML file provides the options how the data within the NeXus object (nxobject) are presented (see Fig.2). Currently this can be done in two ways, either as ImageData or as TableData (see section about XML rules). The advantage of NeXus is the collection of several nxobjects within the same file. As such for every single nxobject a rule needs to be defined. The application has some default rules, which are overwritten by the XML file. If no valid rule for an nxobject is available, the program is terminated with an appropriate error message and detailed information is logged. On the other hand rules defined in the XML file, which do not correspond to any nxobject are not used and as such not validated. Every rule has a set of options, where the key is the name of the tag as it appears in the XML file (e.g. "fsobject type") and the value is the element value of its XML element (e.g. FOLDER). Since the rule is responsible for providing data to the user, the rule has a method to handle the data from nxobject in appliance with the XML file description. Hard coded rules are the following: display NXField as a file without extension

* Work supported by BMBF

ISBN 978-3-95450-139-7

containing raw data from NXField, and display NXGroup as a folder. Also each case of omitting a rule will be logged, so one can easily find the reason why the rule was omitted.

Format of a Typical NeXus Layout at ANKA

A typical structure of a NeXuS file system in use at ANKA is illustrated in Fig. 2. Whereby beneath *-entry/instrument* there might be several different detector information. The most prominent of these including their plottable information have a softlink to *entry/data*. Inside of the directory *entry/instrument/detector/data* all image-Data of this detector will be located.

XML Rules

To meet all expectations about the representation of NeXus objects, there are two sections in the XML file defined. First section are default rules containing defined options for each type of representation (imageData, or tableData). Another section are specific rules containing sets of options to be applied for certain NeXus objects. A specific rule for a NeXus object might contain some options that redefine default rules. So it provides a flexible tuning of representation for each NeXus object.

For **imageData** typically Tagged Image File Format (TIFF) files are produced by topo tomo cameras. A TIFF file by itself might be a container of several images itself, however in order to avoid compatibility problems baseline TIFF files should be used. A Baseline TIFF does not include layers.

A tool like hdfview[14] would display an image within an HDF5 file as a table with n x 1024 * 1024 uint 16 bit where n is the amount of tif images. In order to display such an image with an regular image viewer, header information have to be added. It is also possible to store an image as binary array, however in this case the header information would be missing. In order to add these header information the missing information have to be defined in the XML file as shown in the example of table 1. Since a TIFF image might have different photometric interpretation (colored, grayscale, etc.) additional settings are required to be defined in the XML file. These tags are ”bits” and ”colorscheme”. The values in these tags should be written in accordance with Adobe TIFF 6.0 specification.

Table 1: XML Definition for Image Data

XML tag	Example value
<fsobject_type >	FOLDER
<mode >	image
<colorscheme >	MINISBLACK
<bit >	16
<extension >	.tif
<path >	/entry/instrument/detector/data

TableData are objects interpreted as table. The headline of each row are defined inside of the XML file and as such can be restructured and changed easily on the fly. In addition the application allows to merge data from several NeXus objects of one group in one table. Within the XML file, the headline of each column, including the file containing the values for this column, is defined. Furthermore the column order can be defined.

In table 2 the major rules of an XML file are presented.

Table 2: XML Definition for Table Data

XML tag	Example / value
<fsobject_type >	FILE
<mode >	table_csv
<separator >	;
<extension >	.csv
<column_count >	3
<title >	x resolution
<content >	introducing a new section
<content><path >	path to the NeXus field with content
<column_order >	<item1 >column2 </item1 > <item2 >column1 </item2 >

PERFORMANCE MEASUREMENT

Opening 1500 TIFF images in a native file system takes 1 minute and 1.367s as shown in Fig. 3. On the other hand opening 1500 images included in one HDF5 file mounted via this software only takes: 46.379s as shown in Fig. 4. As a result we can conclude, that our tool improves the usability of a NeXus file and even allows faster access. The environment for these tests: a virtual machine running OpenSuSe 12.3 64bit, on an Intel core i5-3210M with an OCZ Agility 4 SSD and 2 GB DDR3 RAM.

```
time read_tiffs
1500imgs processed
real     1m1.367s
user     0m9.068s
sys      0m22.341s
```

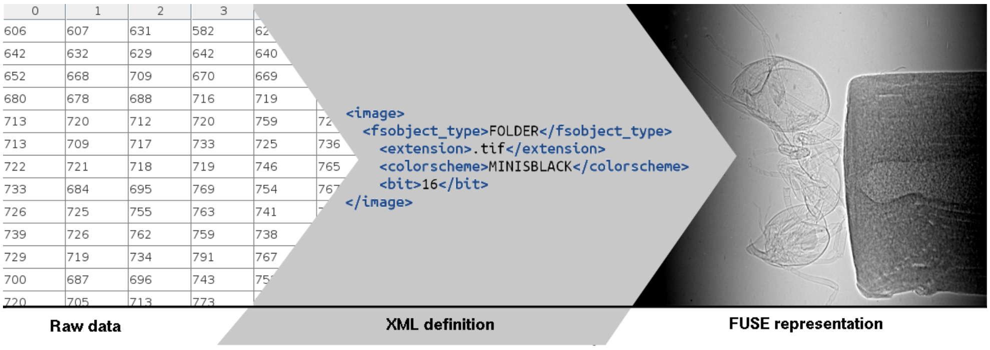
Figure 3: Opening 1500 native TIFF images.

```
time read_tiffs
1500imgs processed
real     0m46.379s
user     0m0.754s
sys      0m2.502s
```

Figure 4: Opening 1500 TIFF images included inside a single HDF5 file.

SUMMARY

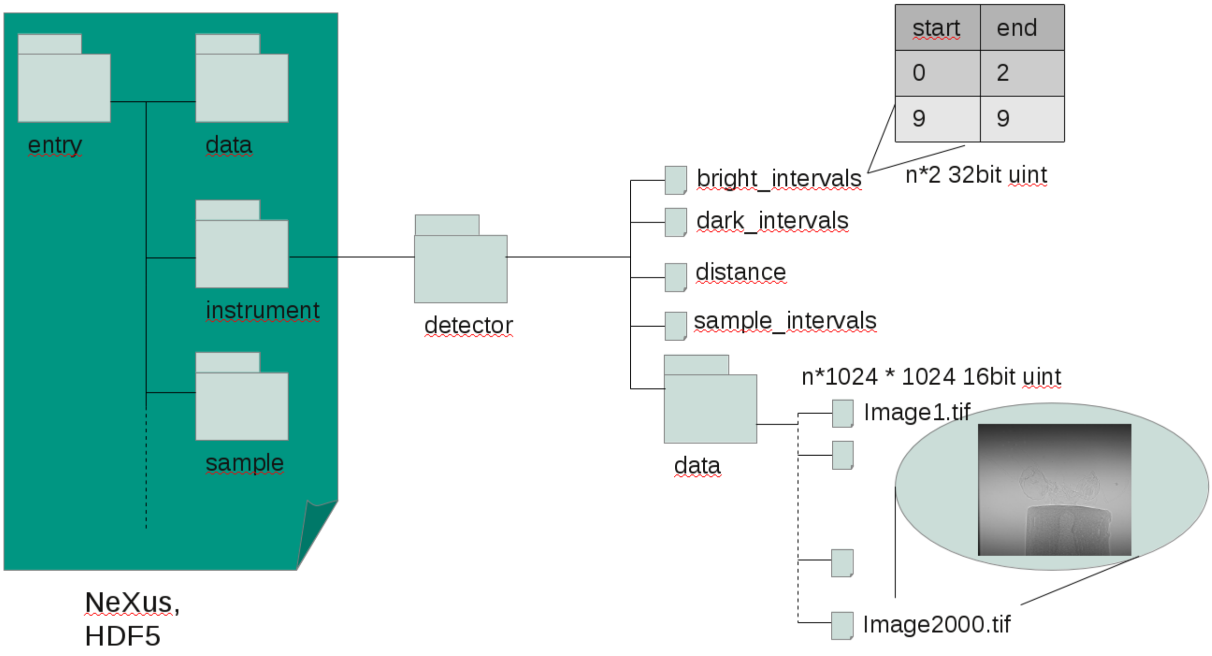
This project is still under development and currently at a beta state. We have proven to mount a NeXus file as a file



0	1	2	3		
606	607	631	582	62	
642	632	629	642	640	
652	668	709	670	669	
680	678	688	716	719	
713	720	712	720	759	72
713	709	717	733	725	736
722	721	718	719	746	765
733	684	695	769	754	767
726	725	755	763	741	
739	726	762	759	738	
729	719	734	791	767	
700	687	696	743	75	
720	705	713	773		

```
<image>
  <fsobject_type>FOLDER</fsobject_type>
    <extension>.tif</extension>
    <colorscheme>MINISBLACK</colorscheme>
    <bit>16</bit>
</image>
```

Figure 1: From RAW data to FUSE mounted image.



start	end
0	2
9	9

Figure 2: Typical file system layout.

system. Within this file system it is feasible to open image files with the use of an image tool like imagej[15]. Furthermore it will be possible to combine several files to a comma separated file (CSV), which can be easily accessed with any spreadsheet program. The code is available at google code. Developers in the team are welcome, just drop us a note, to be added to the development team.

REFERENCES

[1] High Data Rate Initiative. `http:www.pni-hdri.de`

[2] NeXus A common data format for neutron, x-ray and muon science `www.nexusformat.org`

[3] ANKA the Synchrotron Radiation Facility at Karlsruhe Institute of Technology `http://www.anka.kit.edu`

[4] FUSE a File system in Userspace, an operating system mechanism that lets non-privileged users create their own file systems without editing kernel code. `http://en.wikipedia.org/wiki/Filesystem_in_Userspace`

[5] Tagged Image File Format, TIFF Revision 6.0 Final June 3,1992, Adobe Developers Association, Adobe Systems Incorporated 1585 Charleston Road, P.O. Box 7900, Mountain View, CA 94039-7900 `http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf`

[6] American Standard Code for Information Interchange (ASCII) `http://en.wikipedia.org/wiki/ASCII`

[7] SPEC Software for instrument control and data acquisition `http://www.certif.com`

[8] HDF5 is a data model, library, and file format for storing and managing data. `http://www.hdfgroup.org/HDF5`

[9] Eugen Wintersberger `http://code.google.com/p/pni-libraries`

[10] volume graphics studio max, Volume Graphics GmbH, intelligent tools for data manipulation and more efficient ways. http://www.volumegraphics.com

[11] IGOR Pro is an extraordinarily powerful and extensible scientific graphing, data analysis, image processing and programming software tool for scientists and engineers. http://www.wavemetrics.com

[12] Common Data Model Access, A unified layer to access data from data analysis point of view, S. Poirier, A. Buteau, M. Ounsy, C. Rodriguez, Synchrotron SOLEIL1, France, N. Hauser, T. Lam, N.Xiong, ANSTO2, Australia, THCHAUST03 Proceedings of ICALEPSCS2011, Grenoble, France http://code.google.com/p/cdma

[13] E. Iurchenko, E.Wintersberger and D. Ressmann "Fuse-for-NeXus", The download page to the source code of this Application. http://code.google.com/p/fuse-for-nexus

[14] hdfview
http://www.hdfgroup.org/hdf-java-html/hdfview

[15] imagej http://imagej.en.softonic.com

ISBN 978-3-95450-139-7