



ICANS-XV  
15<sup>th</sup> Meeting of the International Collaboration on Advanced Neutron Sources  
November 6-9, 2000  
Tsukuba, Japan

## 16.1 New Software for Neutron Data Reduction and Visualization

T. Worlton<sup>1\*</sup>, A. Chatterjee<sup>1</sup>, D. Mikkelson<sup>2</sup>, R. Mikkelson<sup>2</sup>, J. Hammonds<sup>1</sup>, D. Chen<sup>1,3</sup>, C.-K. Loong<sup>1</sup>

<sup>1</sup> Argonne National Laboratory, Argonne, IL, USA

<sup>2</sup>University of Wisconsin-Stout, Menomonie, WI, USA

<sup>3</sup>Current Address: China Institute of Atomic Energy, Beijing, China

\* E-Mail: tworlton@anl.gov

### Abstract

Development of advanced neutron sources and instruments has necessitated corresponding advances in software for neutron scattering data reduction and visualization. New sources produce datasets more rapidly, and new instruments produce large numbers of spectra. Because of the shorter collection times, users are able to make more measurements on a given sample. This rapid production of datasets requires that users be able to reduce and analyze data quickly to prevent a data bottleneck. In addition, the new sources and instruments are accommodating more users with less neutron-scattering specific expertise, which requires software that is easy to use and freely available. We have developed an Integrated Spectral Analysis Workbench (ISAW) software package to permit the rapid reduction and visualization of neutron data. It can handle large numbers of spectra and merge data from separate measurements. The data can be sorted according to any attribute and transformed in numerous ways. ISAW provides several views of the data that enable users to compare spectra and observe trends in the data. A command interpreter, which is now part of ISAW, allows scientists to easily set up a series of instrument-specific operations to reduce and visualize data automatically. ISAW is written entirely in Java to permit portability to different computer platforms and easy distribution of the software. The software was constructed using modern computer design methods to allow easy customization and improvement. ISAW currently only reads data from IPNS "run" files, but work is underway to provide input of NeXus files.

### 1. Introduction

New sources under construction such as the Spallation Neutron Source (SNS) at Oak Ridge will produce data much more rapidly than older sources due to a combination of higher flux and more detector elements per instrument. Even at some current sources, improvements in instruments are resulting in more and larger data sets. For instance, the GEM diffractometer at ISIS can produce more than four thousand spectra at once and can produce a full data set in about a minute [1]. The next generation powder diffractometer proposed for SNS (POW-GEN3) will have as many as 237,000 detector elements that will each produce a time-of-flight spectrum of thousands of channels. Through time-focusing it is hoped to combine all that data into one large spectrum, but prior to this step instrument diagnostics will be needed which can examine large numbers of spectra from different detector elements. These

diagnostics allow scientists to see noisy detectors, dead detectors and extraneous signals affecting many detector elements. Due to the high neutron fluxes now achievable, collection times have dropped and more measurements are being made per sample. This enables scientists to measure spectra as a function of temperature, pressure, composition, etc. Effective display of multiple spectra obtained under different sample conditions permits scientists to quickly recognize transitions or other effects occurring during study. Such knowledge enables more efficient use of the very expensive facilities at these advanced sources.

Another problem faced by software designers for neutron scattering data collection and analysis is the need to support different types of computers with different operating systems. A few decades ago most analysis was done on centralized computers so software could be developed for a specific computer, but today scientists have different kinds of personal computers they would like to use to analyze their data. The growth of the Internet and introduction of the World Wide Web have raised expectations of users regarding the ease of use of programs and ease of access to data.

At IPNS we are in the midst of a project to design software to meet some of these needs. The Java language enables programs to be written, compiled once and then run on one of several common computer platforms. Java also is designed for use on the Internet; so it has built-in facilities for network communication, network security, etc. Java also includes components for easy design of menu-driven software to facilitate use by occasional users who are not familiar with specialized command languages. New capabilities are added to Java regularly and updated versions are distributed to the community free of charge. Therefore, Java is a language with a bright future.

## 2. Software Overview

In order to be useful and widely accepted, software for neutron scattering data reduction and visualization needs to be network-accessible, portable to different platforms, easy to use, powerful, and easily adapted to different types of instruments. When we began this project we chose to write all the programs in Java because it has all the features needed to write software meeting these requirements [2]. Java is written to be network-accessible and portable to different platforms. It includes powerful libraries (Swing) for developing user-friendly interfaces [3] and it is object-oriented, making it easy to reuse software components and adapt them to different types of instruments. We have been very pleased with the Java language and its associated libraries for program development.

One of the philosophies of object-oriented programming is that "everything is an object" [4] and that if your program is too complicated, you probably need to make more objects [5]. In Fortran programming (prior to Fortran 90), the only types of data structures were numbers or characters and arrays of these primitive data types. In an object-oriented approach, you can create your own definitions of objects that include data and methods, then create an arbitrary number of objects of that type. In Java, a "class" is the description of methods and data that instances (objects) of that class will possess. For example, Bruce\_Brown and Maria\_Heinig could be instances of the employee class, having different salaries, employee numbers, and tasks they can perform.

The most popular "design pattern" [6] of object-oriented software is the "Model-View-Controller" (MVC) architecture, first introduced in Smalltalk. When using MVC, you divide

your software design into three parts: 1) the model or data, 2) the view of the model, 3) the controller that defines the way the user interacts with the model and produces views of the model.

At the heart of our software is the "model". The first part of the model is a spectrum or "Data" class that contains counts or intensity, an axis, standard deviations, attributes, etc. The second part of the model is the "DataSet" (DS) class that contains one or more spectrum objects and methods for transforming axes and performing operations on the spectra. Careful design of the Data and DataSet classes results in a powerful, easily extensible foundation for neutron scattering software. Because our software package is used to combine, view, and transform these spectra, we named it "Integrated Spectral Analysis Workbench" (ISAW) software.

An important part of the MVC concept is that there may be different "views" of the model that respond to any changes in the model through communication from one software component to another. These changes are initiated through the "controller" portion of the software. The user uses the mouse, the mouse buttons, and the keyboard to interact with the "control" portion of the program. The response of the views to the control actions is what makes interactive visualization a useful and pleasant experience. For example, our viewers respond to the cursor position by displaying attributes of the spectrum pointed at by the cursor and displaying the x and y coordinate values at the cursor position when it is dragged over or clicked on one of the graphs or images. Also, whenever a menu selection is used to cause an operator to modify a DataSet, all viewers of that DataSet are updated to reflect the changes. The viewers do this by registering themselves as "observers" of the DataSet objects.

The command or script panel is also part of the control portion of the software that can perform operations on DataSets as well as load files and create DataSets. A command interpreter is essential for automating data reduction. Manually entering commands for hundreds of operations to be performed to combine and reduce a large set of measurements is cumbersome and prone to error. Scripts with loops and symbolic logic make long repetitive procedures easy and simple.

### **3. Data Reduction Requirements**

In order to compare experimental data to theoretical models, all neutron scattering data must be corrected for various factors affecting the observed intensity spectrum. Some of these factors are the incident neutron spectrum, background, detector efficiency, etc. In ISAW, these corrections are implemented either through DataSet operators, or sequences of operations. It is important during any such corrections to maintain information about the precision of the measurements, so estimated errors are maintained with the DataSet and propagated correctly when transformations are performed on the DataSet. One operator provides a method to remove bad (noisy or dead) detectors.

An early goal of this software project was to provide the ability of merging, sorting, and viewing data from different measurements. A measured spectrum may have an arbitrary number of attributes of interest, such as sample temperature, pressure, sample composition, etc. We found it very useful to allow spectra to be sorted by any of up to three attributes, in order to visualize the dependence of the data on the specified attributes. Some of the attributes such as starting time, detector position, etc. are retrieved automatically from the

IPNS run files. Other attributes that are not available in the run files can be added interactively.

All the IPNS raw data consists of histograms of counts versus time-of-flight. Transformations to other units such as d-spacing, wavelength, wave vector, energy, etc. are provided. Some transforms are appropriate to diffractometers, and others are appropriate to spectrometers. DataSet operations that are appropriate for the type of instrument producing the data are included when each DataSet object is created. Operations that are available in the selected DataSet object automatically appear in the menus. This automatic population of the menus simplifies maintenance and improvement of the software, and reduces confusion for the user by not displaying inappropriate choices for operations to be performed. Table 1 shows some of the operations that have been implemented for making corrections to DataSets.

Operation	Arg1	Arg2	Result
DataSet Scalar Math (+, -, *, /, &)	DS1	Scalar	DS
DataSet Math (+, -, *, /)	DS1	DS2	DS
DataSet Transform (x-axis conversion)	DS1	Range	DS
Data sort, select, extract, delete	DS1	Attribute	DS
Calculate Moments	DS1	Range	Float

One of our design goals not yet implemented is the ability to find and fit peaks automatically. A preliminary version of a peak class has been included to fit the elastic peak from spectrometers, but the more general function class to fit a diffractometer spectrum containing an arbitrary number of peaks has not yet been written. We expect to include this in the next version of the ISAW software. At this time, we do not intend to include complete analysis capabilities such as Rietveld analysis in ISAW. However, ISAW can generate standard formatted datasets (e.g. GSAS compatible) for Rietveld refinement.

#### 4. Visualization Requirements

As mentioned in the introduction, new neutron scattering instruments are producing large numbers of spectra. For example, the proposed POWGEN3 instrument at SNS expects to produce up to 237,000 spectra of thousands of channels each [7]. IPNS time-of-flight instruments now typically produce a few hundred spectra with up to about 10,000 channels per spectrum. It was an early goal of ISAW to be able to manipulate and visualize large numbers of spectra.

Current display devices are limited to resolutions on the order of 1600 horizontal by 1200 vertical pixels, although most people use displays with lower resolution. Since most spectra we deal with have more channels than the number of pixels per line on a screen, it is necessary to either re-bin the spectrum to fit within the screen resolution, or to allow horizontal scrolling. In ISAW, the default is to re-bin the data to fit the resolution of the display panel, but the user is given the option of using horizontal scrolling to allow all data points to be seen.

The traditional way to view spectra is to draw lines with the y-axis amplitude representing counts or intensity and the x-axis value representing the independent variable, time-of-flight, wavelength, etc. This works well for displaying a single spectrum or a few spectra, but for more than about a dozen spectra another method is needed. Mikkelson, et al. [8] devised a method of displaying many spectra by using color to represent counts or intensity so that an image consisting of many spectral lines stacked on one another allows a person to view many spectra at once. If there are fewer spectra than vertical resolution elements, each spectrum will cover a band of horizontal lines. This proved to be a very useful way to compare large numbers of spectra. Figure 2 shows an image view produced by merging spectra from 130 runs taken at different sample conditions.

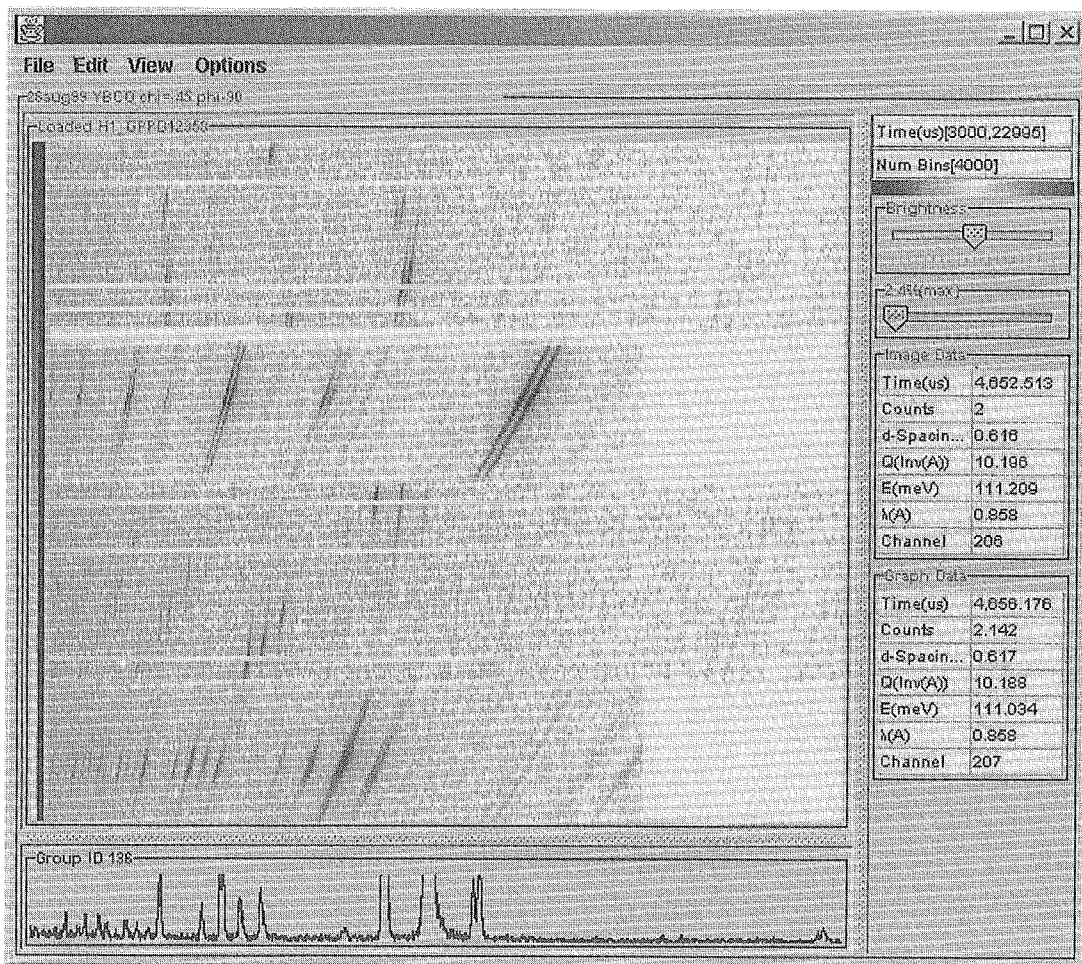
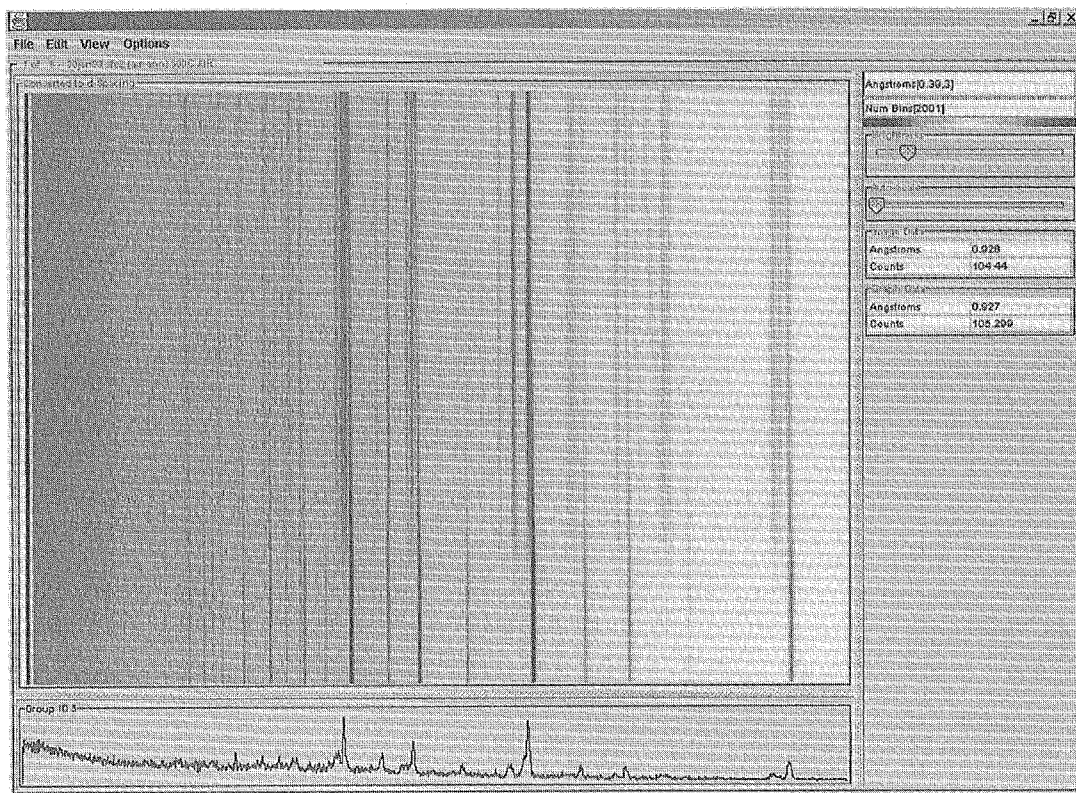


Figure 1. Image view of unfocused GPPD DataSet. The large area of the image view shows a bit-map with color representing intensity, and horizontal position representing time-of-flight. Each row of the bit-map represents the spectrum from one detector group. The small graph at the bottom is a conventional graph of the detector group corresponding to the cursor position. The tables at the right list properties of the spectrum corresponding to the position of the cursor on the image and/or graph.

With this type display, you can see the variation in intensity due to sample texture. Dead or noisy detector elements appear as light or dark horizontal lines. In ISAW we have included this "image view" and a scrolled "graph view" as ways of displaying large numbers of spectra. ISAW includes vertical scrolling of the image view to allow viewing of more spectra than allowed by the resolution. Whenever the number of spectra in the image exceeds the

number of lines of vertical resolution, scroll bars appear to allow viewing all spectra. The GLAD instrument at IPNS produces more than one thousand spectra at once and we are able to view them all in one image view. For a more traditional view of spectra, we designed a "scrolled graph" view that allows us to see a few individual spectra at once and easily scroll vertically to change which spectra are seen.

For an instrument producing as many spectra as POW-GEN3, it would probably not be possible to handle all the individual spectra at one time due mainly to computer memory limits. On such an instrument one would probably view spectra by banks of one thousand detector elements or less. Separate spectra from all detector elements would only be viewed when doing instrument diagnostics or when looking for sample texture. Normally the data from the different detector elements would be focused and combined to produce only a few spectra, or perhaps only one combined spectrum per sample setting. Even if all spectra from a given measurement are combined into one composite spectrum, there is still a need to view many spectra at once to see the effects of sample environment or sample composition. For IPNS data, this involves reading run files for each sample condition, transforming and correcting the data, and merging all the spectra into one DataSet for comparison. Figure 2 shows an image view produced by merging 130 spectra taken at different sample conditions.



**Figure 2.** An ISAW view showing GPPD measurements on 130 sample settings merged into one image. Note the streaks which appear or disappear when the sample undergoes a phase transition. The spectra have been converted to d-spacing, so the tables in the right panel give the d-spacing and counts corresponding to the cursor position. The graph at the bottom and the tables are updated as rapidly as the cursor is moved across the image and/or graph.

## 5. I/O Requirements

IPNS is in the process of designing a new Data Acquisition System to replace the old VAX-based system that has been in use for 20 years. Data collection will be controlled through a computer running Linux, but the binary run file data format will remain the same because of existing software. This meant that ISAW needed to support reading IPNS run files. At the same time, it was recognized that using ISAW for working with spectra from any other facility would probably require a common data file format. Fortunately, the NeXus format [9] is starting to be widely accepted and we have plans to add NeXus support to ISAW, possibly through the "NeXus Data Server" (NDS) developed by Mark Koennecke of PSI [10]. The NDS consists of a server that can read the NeXus files and a client that can communicate with the server over the network using sockets.

## 6. ISAW Status and Plans

ISAW is currently used as the primary data reduction and visualization tool for the HRMECS instrument and is being used on several other IPNS instruments. Work to improve ISAW and add features needed by specific instruments is ongoing. Much of the customization for different instruments is being done through scripts to perform the series of operations normally done by a particular instrument. Development of a function class to include the peak class and methods for finding and fitting peaks in experimental data is underway. Work on NeXus input and possibly output is also underway. There are also plans to create a "table" view of the data that will allow us to copy data from ISAW to other Windows applications for production of publication quality graphs. We already have the capability to copy data from our current tables to other applications.

## Summary

The World Wide Web and Java have given us the tools for easy access to and analysis of information and data on the Internet to make neutron scattering accessible to more users. At IPNS we are using these tools to develop new neutron scattering software that is easier to distribute and use and will not require any licensing or fees. The Integrated Spectral Analysis Software (ISAW) we are developing is beginning to be adopted for IPNS instruments and should soon be able to handle data from other institutions. ISAW allows us to merge, view, and reduce the large numbers of experimental spectra that are being produced by new neutron sources and instruments.

## References

- [1] See, for example the GEM instrument at ISIS, [http://www.isis.rl.ac.uk/disordered/gem/gem\\_home.htm](http://www.isis.rl.ac.uk/disordered/gem/gem_home.htm)
- [2] Core Java 2, Vol. 1—Fundamentals, C. S. Horstmann and G. Cornell, Sun Microsystems Press.
- [3] See <http://developer.java.sun.com/developer/onlineTraining/Programming/BasicJava1/front.html>
- [4] Thinking in Java, 2<sup>nd</sup> Edition, Bruce Eckel, p. 31, 101, Prentice Hall, Inc. 2000.
- [5] Thinking in Patterns with Java, Bruce Eckel, p. 140, see <http://www.bruceeckel.com/TIPatterns/>.
- [6] Design Patterns, Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, Addison-Wesley, 1995.

[7] See the POW GEN3 powder diffractometer instrument description at <http://www.sns.anl.gov/>

[8] Visualization of time-of-flight neutron diffraction data, D. J. Mikkelsen, A.J.G. Ellison, D. L. Price, and T. G. Worlton, Nuclear Instruments & Methods in Physics Research, A 354 (1995) 112-120.

[9] See <http://www.neutron.anl.gov/nexus/>

[10] M. Koennecke, NOBUGS 2000. See <http://nobugs.dl.ac.uk/dataformats.htm>, and <http://nobugs.dl.ac.uk/presentations/koennecke/nexusATpsi/nds.html>. The NDS software can be downloaded from <http://lns00.psi.ch/nndb/>.