

QUICK EXAFS EXPERIMENTS USING A NEW GDA ECLIPSE RCP GUI WITH EPICS HARDWARE CONTROL

R. J. Woolliscroft, C. Coles, M. Gerring, M. Pearson, Diamond Light Source, Oxfordshire, UK

Abstract

The Generic Data Acquisition (GDA) framework [1] is an open source, Java and Eclipse RCP [2] based, data acquisition software framework for synchrotron and neutron facilities. A new implementation of the GDA on the B18 X-ray beamline at the Diamond synchrotron will be discussed. This beamline performs energy scanning X-ray Absorption Spectroscopy (XAS) experiments. It includes a continuous-scan mode of the monochromator synchronised with various detectors for Quick EXAFS (QEXAFS) experiments. XAS energy scans may now be performed in 30s, where the equivalent step scans take approximately 40 minutes.

A new generic perspective for the GDA client has been developed in which graphical editors are used to write XML files which hold experimental parameters. The XML files are marshalled by the GDA server to create Java beans used by Jython scripts which run on the GDA server. Underlying this, a new generic continuous scanning mechanism for the GDA has been developed. The new Eclipse RCP UI, the new continuous scanning mechanism, new hardware development and integration between the two systems shall be covered.

INTRODUCTION

XAS is a mature X-ray technique, with a wide range of applications across scientific disciplines. As a result, there is a need at modern synchrotrons to provide robust and

highly automated XAS beamlines, designed to be used by both experienced and occasional users.

B18 is a general purpose XAS photon beamline, complimentary to other XAS beamlines at Diamond. A new quick scanning mode has been developed in which EXAFS data can be collected with a continuous move of the monochromator.

The experimental hutch has two sample positions; an in-vacuum position for 2-4keV (soft X-ray) and an in air position for 4-36keV (hard X-ray). The hard X-ray position has a large experimental area for a range of sample environments (Fig. 1).

DETECTOR

Fluorescence data for EXAFS studies is collected using a 9 element XSPRESS Ge detector [3]. This detector can collect at high count rates whilst being placed close to the sample to maximise the fluorescence yield.

The detector is controlled by a Time Frame Generator (TFG2) [3] card which has a number of programmable input and output signals. The TFG2 card is used to receive and fan-out signals from the monochromator Bragg motor during a quick scanning data collection (Quick EXAFS or QEXAFS). It can be used to drive other detectors which may be used during the experiment. For the QEXAFS experiments, most commonly used are the ion chambers which provide a normalisation for the fluorescence yields.

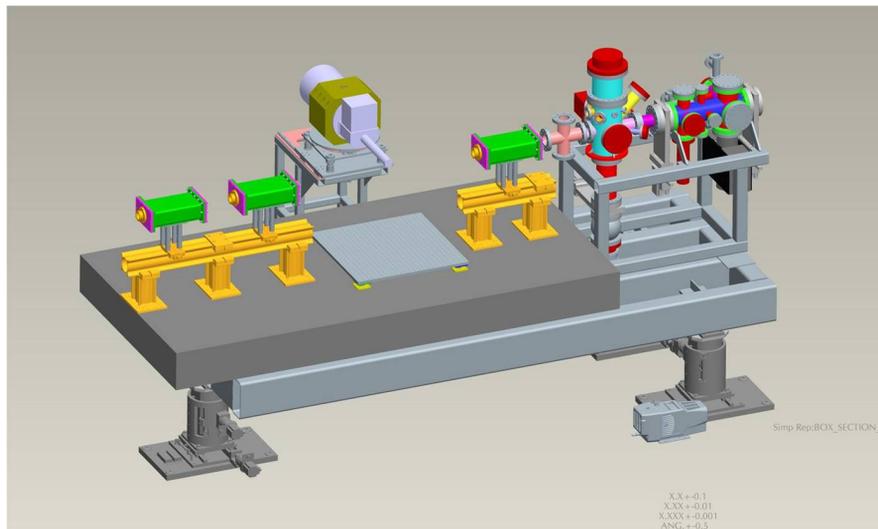


Figure 1: Diagram of the B18 experimental hutch. The in-air sample location is shown. This has a large area for a range of sample environments. The XSPRESS detector for fluorescence and the ion chambers for transmission measurements are shown.

HARDWARE FOR QEXAFS EXPERIMENTS

The monochromator Bragg axis is controlled using a DC brushless in-vacuum motor, with four encoders placed at equal distance around the stage. There is also an encoder on the back of the motor. A Delta Tau Geobrick Low Voltage Integrated Motion System (LV IMS) is used as the motor controller, and this is the Diamond standard controller for beamlines currently being constructed. The Bragg axis and crystal distance axis are controlled together to provide an ‘energy’ combined axis, and these calculations are performed directly on the controller. This provides us with true synchronised motion of Bragg and crystal distance. Using this control method a very smooth continuous motion between two energy points can be achieved, at speeds of up to 1deg/s.

The Experimental Physics and Industrial Control System (EPICS) [4] is used to control and set up the motor controller. Each axis controlled by the Geobrick has an EPICS Motor Record interface, which provides a consistent ‘motor’ interface for the GDA. The same interface is used for the virtual ‘energy’ axis, which means the GDA can treat the energy axis as any other axis, and scan it in the same way.

The EPICS Input Output Controller (IOC) is also responsible for setting up the ‘position compare’ trigger signals for the Bragg axis. These are TTL pulses produced by the Geobrick as the Bragg axis encoders increment in position. The triggers are fed directly into the TFG2 card, which is controlled by the GDA. The readout of the ion chambers is performed using a Ge-Fanuc 110BL VME ADC card. This is a 24-bit sigma-delta ADC running at up to 108KHz, which is read out through a new driver used by an IOC running VxWorks.

The trigger signals that are fed into the TFG2 card are used to provide a ‘gate’ signal. This signal is read into one of the ADC channels, and is used to provide a means by which the IOC can calculate mean and standard deviation of the samples collected during the gate period. Gates as long as several seconds and as short as 1ms are used. Buffering of this data is provided by the EPICS IOC in arrays, so that they can be read out asynchronously by the GDA. However, if required, raw ADC count data is also provided by the EPICS IOC.

NEW CONTINUOUS SCAN

The GDA has a flexible and powerful step scan mechanism which has been applied to the full range of techniques used at the Diamond synchrotron. A new generic continuous scan mechanism was developed for implementation on the B18 beamline, but which could easily be applied to other beamlines with suitable hardware.

Existing Scanning Mechanism

The generic scanning mechanism is available through the *scan* command in the GDA Jython environment.

Behind this command is a single class (*ConcurrentScan*) which operates both hardware and detectors in n-dimensional step scans. This operates the hardware through two high-level interfaces: *Scannable* and *Detector*.

The *Scannable* interface enables the scanning mechanism to operate those objects to be incremented during the scan, and to know what information is supplied by that object at each point. *Scannables* can represent a wide variety of objects; simple motors, compound motions or even offline data processes which return the results of their calculations asynchronously.

The *Detector* interface is an extension to the *Scannable* interface. At each point in the scan *Detector* objects are used to collect data after the *Scannable* objects have completed their step movement.

Working with the *ConcurrentScan* class is a system to broadcast the data from each step in the scan to parts of the UI for live data display to the user and to file writers to record the data in various data formats. The performance of this system is fast enough to cope with large and frequent data points at the speed expected in B18 QEXAFS scans.

Extension to the Scanning Mechanism

The new continuous scan mechanism is designed to work within the existing step scan mechanism. It incorporates many aspects of that system to maximise code reuse. Its design has analogous interfaces: *ContinuouslyScannable* and *BufferedDetector*, which are extensions of the *Scannable* and *Detector* interfaces respectively.

There is a new single generic continuous scan class (*ContinuousScan*) which uses those generic interfaces. This class extends the *ConcurrentScan* class so the continuous scan may be used as the innermost dimension of an n-dimensional step scan. For example, from a single command, the user may loop over a series of samples in an automated changer and perform a QEXAFS scan on each one.

B18 implementation

An implementation of the *ContinuouslyScannable* interface was written for B18 which operates the Bragg motor via EPICS and controls the position compare parameters. The XSPRESS detector is operated by an implementation of the *BufferedDetector* interface from which the amount of data stored and the data itself can be retrieved asynchronously by the scan class.

In the B18 QEXAFS scan, the *ContinuousScan* class is given the experimental parameters and references to the *Bragg ContinuouslyScannable* and *XSPRESS BufferedDetector* objects. The hardware is prepared for the scan, and once the Bragg motor motion has been started the scan class polls the *XSPRESS BufferedDetector* for new data. As new data becomes available, then the existing step scan’s mechanism for recording and visualizing the data is used.

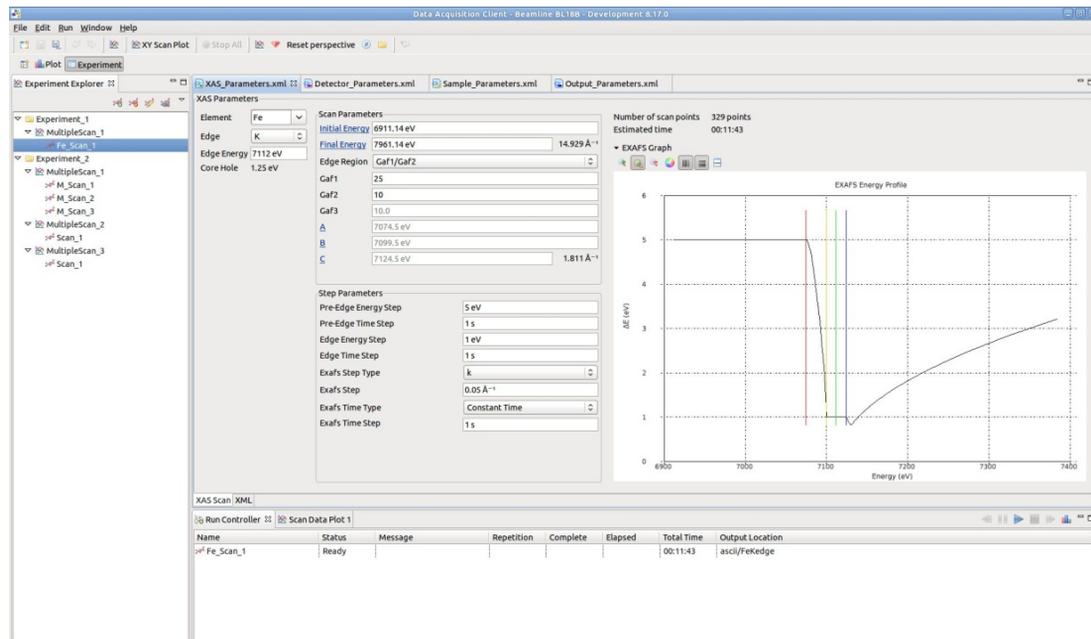


Figure 2: New Experiment perspective for users to design experiments. In B18's implementation four graphical editors are used to write the four XML files which define each experiment. The visible editor is the EXAFS step scan editor.

NEW RCP UI

B18 was the first newly commissioned Diamond beamline to use a GDA client based on the Eclipse RCP platform. A range of generic perspectives have been developed for tasks such as visualisation of data files and Jython scripting.

A new generic perspective, shown in Fig. 2, was written which uses graphical editors to allow users to edit XML files holding the experimental parameters. The perspective has views to manage the XML files and to run a single or a group of experiments (*multiscans*). Multiscans allow several experiments to run sequentially. Multiscans may be stored in different file system folders to keep the files organised. The UI provides a tree view of the multiscans and functionality such as drag and drop to move individual experiments between multiscans. There is a view to run the selected multiscans and display progress.

The perspective is configured using Eclipse extension points and may be applied to other beamlines which could use XML files to define each experiment.

For B18 there are four XML files to describe each experiment. These hold the scan parameters, detector options, sample environment setup and readout options. There are three different editors which write scan parameters to reflect the different experiments run on the beamline: EXAFS, XANES and QEXAFS (detail shown in Fig. 3).

Design

To collect data the UI unmarshalls the XML files into Java beans. These beans are used in a Jython script running on the GDA server which uses the information in

those beans to set up the hardware and run appropriate scan commands.

Data is returned to the UI for display and analysis in the same manner whether the scan is a quick or regular EXAFS.

A custom perspective has been written to view experiments in progress, as shown in Fig. 4. This contains views to display the current experiment progress; XSPRESS count rates; visualisation of the data as it becomes available and online data analysis. Graphs show standard XAS analyses: background subtraction, first derivative of the data and the Fourier transform.

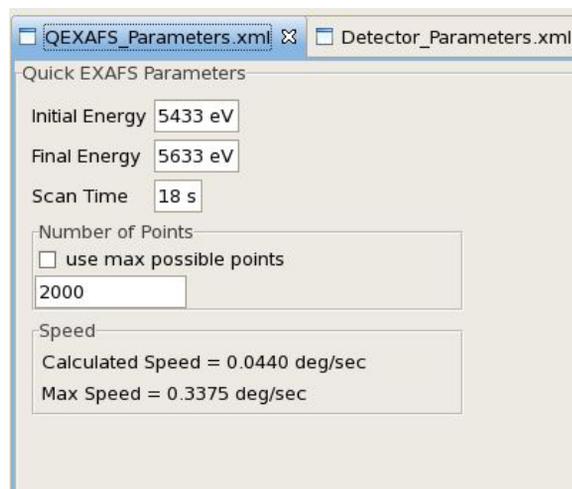


Figure 3: Detail of the Experiment perspective showing the QEXAFS editor.

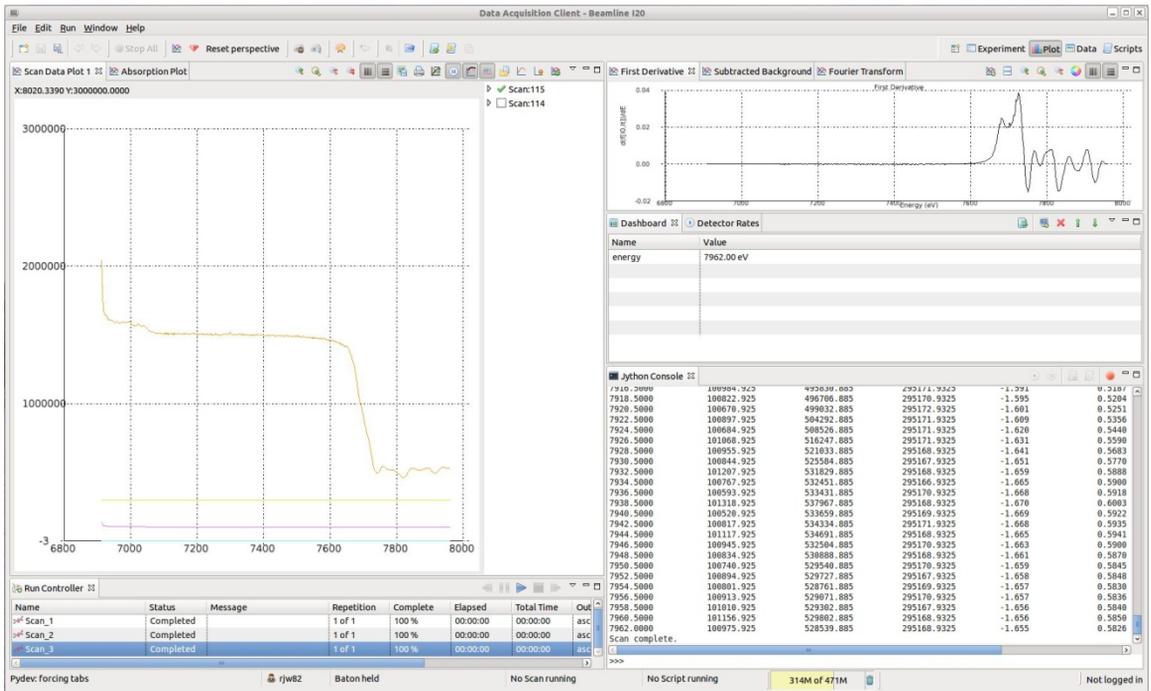


Figure 4: The Plot perspective for Diamond XAS beamlines. This shows data from the experiment in progress. Live plots of the absorption, first derivative of the data after background subtraction, the subtracted background and FFT are shown in other views. The Run Controller view from the Experiment perspective is also shown so experiments may be started from within this perspective.

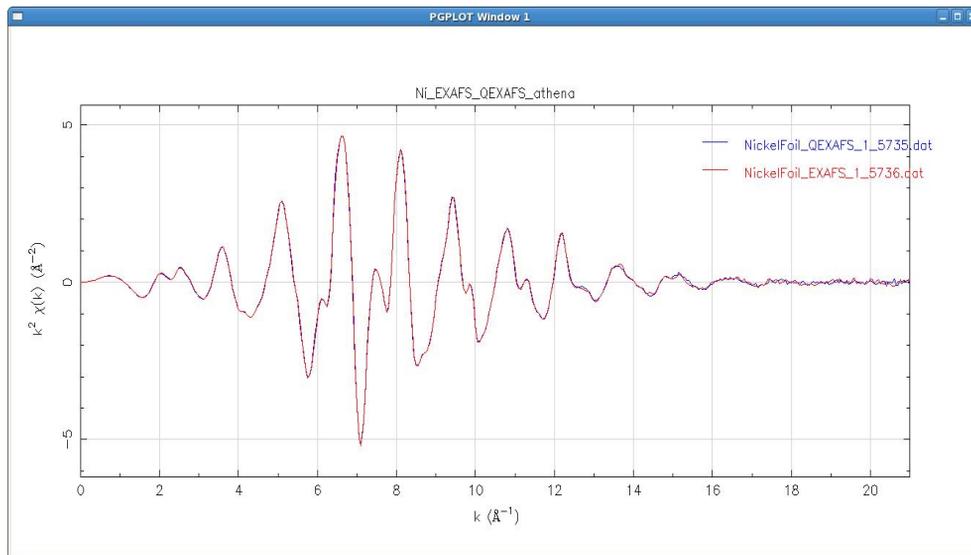


Figure 5: Figure shows the first Quick EXAFS spectrum taken in 60s (25ms per point) (red curve), compared to a step scan running for 50 minutes (1-5s per point) (blue curve).

PERFORMANCE

A typical time for a QEXAFS scan is about 30s where typical step scans take 20 to 40 minutes. The QEXAFS results have been shown to be comparable to step scan data (see Fig. 5).

The QEXAFS mode has been available since Spring 2011 and is currently requested by about a third of proposals to the beamline.

REFERENCES

- [1] <http://www.opengda.org>
- [2] <http://www.eclipse.org>
- [3] R. Farrow et al., "XSPRESS: A new generation of detector system for EXAFS studies", Physics B 208 & 209, pp. 256-8 (1995).
- [4] <http://www.aps.anl.gov/epics/>

Copyright © 2011 by the respective authors — cc Creative Commons Attribution 3.0 (CC BY 3.0)