GDA AND EPICS: WORKING IN UNISON FOR SCIENCE DRIVEN DATA ACQUISITION AND CONTROL AT DIAMOND LIGHT SOURCE

E. P. Gibbons, M. T. Heron, N. P. Rees
Diamond Light Source, Oxfordshire, United Kingdom

Abstract

Diamond Light Source has recently received funding for an additional 10 photon beamlines, bringing the total to 32 beamlines and around 40 end-stations. These all use EPICS for the control of the underlying instrumentation associated with photon delivery, the experiment and most of the data acquisition hardware. For the scientific users Diamond has developed the Generic Data Acquisition (GDA) application framework to provide a consistent science interface across all beamlines. While each application is customised to the science of its beamline, all applications are built from the framework and predominantly interface to the underlying instrumentation through the EPICS abstraction. We will describe the complete system, illustrate how it can be configured for a specific beamline application, and how other synchrotrons are, and can, adapt these tools for their needs.

INTRODUCTION

Diamond Light Source has developed a beamline automation system that combines the low-level control system EPICS with the high-level data acquisition system GDA. This combination is used on the majority of beamlines at Diamond and has proven to be very successful at enabling scientists to maximize the quality and quantity of the wide range of science being performed. Use of the system results in a seamless experience for visiting scientists; from applying for beam time to having easily available data, be it raw, reduced, or partially analysed. The system design is such that changes in requirements can be accommodated in a timely and efficient manner. Both EPICS and GDA are open source projects allowing Diamond to benefit from development in both by third parties.

SYSTEM CONTEXT

The detailed description of the system begins with the generalised context diagram, shown in Fig. 1.

Scientist

Scientists interact with the system by reference to scientific properties of the experiment, such as the energy of the photons of the beamline. Scientists define experiments, run scripts and analyse data recorded. They do not have access to all low level details of the beamline.

Engineer

Engineers need access to all low level details of the beamline for commissioning and trouble shooting. The interaction is real-time in nature.

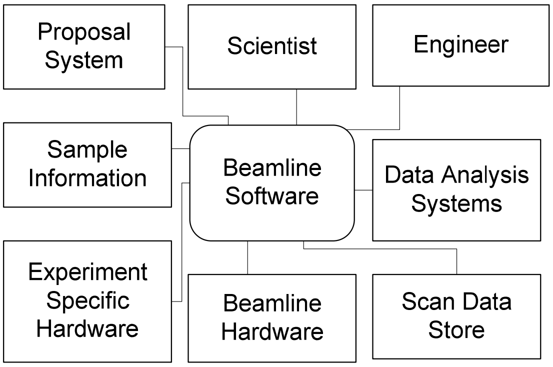

Figure 1: Context Diagram.

Data Analysis Systems

In order to aid scientists to make best use of time at Diamond great emphasis is given to providing on-site analysis of data. The aim is for this to be done automatically and the results fed back to the scientist as soon as possible.

Scan Data Store

The output of all scans is stored in a centrally available store. The store may also contain the results of any analysis performed on the data during the visit.

Beamline Hardware

This item encompasses all Diamond hardware on the beamline, as opposed to hardware brought by scientists for particular experiments. Hardware that is normally scanned or monitored during an experiment includes intelligent motor controllers, high speed 2d detectors and sample mounting robots.

It is important to note that for performance reasons some high speed detectors write directly to the Scan Data Store.

Experiment Specific Hardware

For some experiments scientists will make use of specialised equipment that is not normally on the beamline. Such equipment acts under control of the scanning mechanism of the beamline software. The beamline software has to be able to accommodate such additions in a timely and efficiently manner.

Sample Information

High level automation requires use of Sample Information systems containing details of samples and their locations in sample mounting robots. This information is presented to the scientist to aid automatic data collection, and linked to data to aid analysis.

Proposal System

This is Web based system used by potential users to request time on a beamline. The automation system uses this to check that a user is to be granted access to the beamline.

SOFTWARE DESCRIPTION

Figure 2 is a first level breakdown of the Beamline Software.

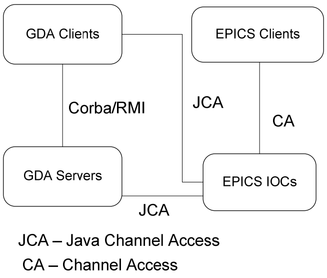

Figure 2: First level breakdown of the Beamline Software.

GDA Clients

Scientists interact with the beamline using the GDA Client program. This GUI is a Java application built using the Eclipse Rich Client Platform. It contains panels allowing the user to:

- Issue individual commands to perform actions. The command set is extremely diverse. A simple command would be to request the value of a variable, such as temperature readout. A complex command would be to execute a multi-dimensional scan of physical properties of the beamline resulting in measurements of multiple detectors and creation of a Scan Data file.
- Issue commands to execute a script file containing multiple commands. The script file can contain a simple list of commands or Jython programming language statements allowing the scientist to create complex sequences.
- Edit script files.
- Define scans using technique or even beamline specific GUI panels.
- View the results of scans and perform rudimentary analysis.
- Initiate science specific analysis of data.
- View results of automatic analysis of data.
- Perform technique specific actions using specific GUI elements, e.g. sample alignment.
- View the state of currently running measurements.

EPICS Clients

Engineers interact with the beamline using the EPICS clients, such as graphical user interfaces, which at Diamond are predominately EDM screens. EDM allows the beamline scientist or engineer to set and monitor values in the IOCs. EPICS archival and alarm handling systems are also used.

GDA Servers

The GDA Servers have several important features. Firstly the conversion between the engineering and scientific aspects of the beamline; this conversion is being represented by a set of Scannables and Detectors. Secondly a comprehensive and extensible scan mechanism for scanning such Scannables and Detectors and creating scan data files. Thirdly scriptable control of server objects via a Jython interpreter. The server can contain any additional Java objects one may want for a particular beamline; such as interfaces to hardware not supported by EPICS.

EPICS IOC

The EPICS IOC provide the majority of the hardware abstraction layer in the system, and specialised modules that are used by the GDA to provide important data acquisition tasks.

GDA AND EPICS IN UNISON

Given the very generic nature of GDA and EPICS there are many ways in which the two can be combined to create powerful, flexible and user friendly data acquisition systems. In the following sections we will describe two such systems in use at Diamond.

RECIPROCAL SPACE TRAJECTORY SCANNING

Beamline I16 at Diamond Light Source contains a six circle diffractometer which allows the orientation of a sample and detector relative to the incident x-ray beam to be set with few constraints. Such an arrangement permits measurements to be made over a wide range of reciprocal space.

GDA currently allows for three types of scans; a simple step scan, a constant velocity scan or a trajectory scan. In a simple step scan a measurement of a detector is made at different fixed motor, or pseudo-motor, positions. In a constant velocity scan the motors move at a constant velocity and the measurements are triggered by an additional time frame generator. In a trajectory scan the motors move along a series of points and the measurements are triggered to coincide with when each point is reached.

As well as performing scans of the six independent axes of the diffractometer, scientists need to scan in other space coordinates such as reciprocal space. A simple step scan through a number of points in reciprocal space is performed by the general purpose GDA scan command given in Fig. 3.

```
scan hkl [0 0 1] [0 0 1.1] [0 0 0.01] det1 0.1 peak2d
```

Figure 3: GDA command to perform a simple step scan through reciprocal space.

In the command the term ‘scan’ is the name of the command; ’hkl’ is the name of the object that represents a position in 3 dimensional reciprocal space; the 3 variables following ‘hkl’ represent the start, end and step size values of ‘hkl’ in the scan. The step size variable is followed by the name of the detector to be measured at each point in the scan and the exposure time per point. In this example the detector generates a 2d image; the final argument in this example is a pseudo-detector that analyses the reading from the 2d image and reports back the parameters of the analysis.

The system that relates hkl to diffractometer angle is called DiffCalc. Although built to work with the GDA scanning system DiffCalc can be used independently of GDA [10].

At each point in the scan the hkl object is asked to move to a new position represented by the tuple [h, k, l]. From these values the hkl object calculates the positions needed for the six axes and sends them as a single move command to the individual EPICS motor records in the appropriate IOC. Once all motors have moved the scan command then instructs the detector to read for the given exposure. After the exposure the value of hkl and the data from the detector are written to a scan file.

Trajectory scans are preferred to step scans as the former involve less time starting and stopping of motors. However trajectory scans need to be performed using low level motor controllers and involve hardware triggering of detectors and some form of data storage system. On the face of it, making a system for scientists to perform either scan from very similar generic commands would seem a daunting task; but given the design of the GDA scan mechanism this has been relatively straightforward.

To perform the same measurement as a trajectory scan the user simply replaces the term ‘scan’ in the command shown in Fig. 3 with ‘trajscan’.

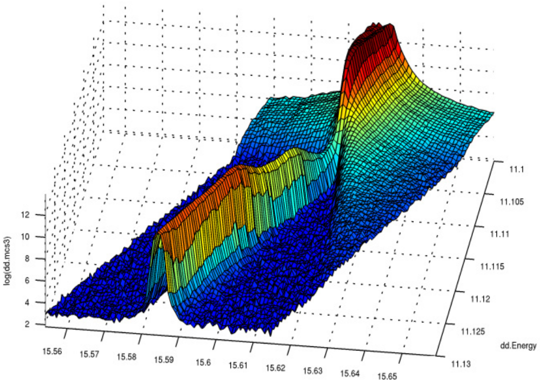

Figure 4: Detector readings from a trajectory scan of angle eta repeated for different incident photon energies.

In this case, rather than sending move commands to the six Epics motor records during the scan, the whole trajectory is calculated and sent to the related Epics TrajectoryScan [11] interface at the start. The Epics Trajectory Scan interface relays the trajectory to a program in the underlying intelligent motor controller. Once the scan is started the motor controller is responsible for sending the trigger pulses to the detector; GDA simply monitors the state of the detector as the scan proceeds. Whenever a new reading from the detector has been made the pseudo-detector peak2d fits a 2d Gaussian to the image and reports the size, position and sum over a region of interest. All of this data for each point is written to the scan file. Certain parts of the data are displayed on an XY plot in the GDA client.

Figure 4 displays actual results from using a trajectory scan of angle eta repeated for different photon energies. In this case the single channel detector was connected to a multichannel scaler with the motor controller’s at position signal connected to the channel advance source. GDA scanning is multi-dimensional which allows for the whole dataset to be taken by a single trajscan command.

Note that the trajscan command works with any scannables and detectors that implement particular Java interfaces. Also the list of detectors that can be read at each point is only limited by the particular hardware configuration. In addition the format of the data and how that is written is not limited. On I16 the Pilatus detector writes its image directly to the Scan Data Store and GDA simply records the filename in the scan data file along with the hkl values and the values from the analysis pseudo-detectors. On other detectors the whole image could be read by GDA and written directly into the scan data files. The scan data file format at Diamond is currently either a GDA specific ASCII format or, more commonly, Hierarchical Data Format (HDF5) compatible with the Nexus standard.

TIME RESOLVED 2D SCAN

Beamline I22 is a non-crystalline diffraction beamline at Diamond Light Source. A typical experiment on I22 involves recording a sequence of scattering patterns from a collection of detectors. The timing of the individual detector exposures is controlled by a time frame generator (TFG). For further complexity the whole sequence is repeated over a grid of positions of a sample.

Most of these detectors on I22 are connected to GDA directly. However a recent addition is a Pilatus detector that is interfaced to GDA using the EPICS AreaDetector system [12].

Once the TFG has been programmed via the GDA GUI the whole time resolved grid scan is performed by the single command shown in Fig 5.

```
scan px 0. 10. 1. py 0 10. 1. ncd
```

Figure 5: GDA command to perform a 2d scan of two motors, px and py, taking data from the detector system, ncd, at each point.

In the command in Fig. 5 px and py are ScannableMotors that control the x and y position of the sample and ncd is the detector system. Note that the ncd object starts the TFG for each value of px and py, and reads all the frames that are triggered by it.

The output of this command is a combination of 2 files. The first file, the so called master, is created by GDA, is compatible to the Nexus standard and is in HDF5 format. It contains the metadata for the scan, the positions of all scanned motors and the data from the directly connected detectors. The second file contains the Pilatus image data written by the HDF5 writer AreaDetector plugin; it too is in HDF5 format. The HDF5 writer plugin writes the data with a dimensionality that matches that of the data in the master file. By putting an external link to the second file in place of the Pilatus data in the first, Nexus compatible readers read the data in the second as if it were within the first.

HDF5 AREA DETECTOR PLUGIN

The use of the EPICS HDF5 AreaDetector plugin is proving very useful at Diamond, not only can much higher frame rates be achieved when writing to a single HDF5 file than the traditional approach of a single file per image, but the read rate is also improved.

CONCLUSION

The combination of GDA and EPICS has proven to be highly successful at providing flexible, powerful and user friendly data acquisition systems for beamlines at Diamond Light Source.

GDA is an open source project that is under continual development and welcomes collaborators. The www.opengda.org website provides downloads and documentation to aid evaluation by those considering its adoption.

ACKNOWLEDGEMENTS

GDA is primarily developed and supported by the Diamond Light Source Data Acquisition and Scientific Computing group. The EPICS system at Diamond is developed and supported by the Diamond Light Source Controls group. The following members of the two groups are acknowledged particularly: Mr R Walton and Dr M Pearson for the I16 trajectory scan; Dr T Richter for the I22 time resolved 2d scan; and Mr U Pedersen for the HDF5 AreaDetector plugin.

REFERENCES

[1] http://www.aps.anl.gov/epics/

[2] http://www.opengda.org/

[3] N.P. Rees, E.P. Gibbons, Diamond Controls Group, Diamond Data Acquisition Group. “The Diamond Beamline Controls and Data Acquisition Software Architecture”. SRI09. AIP Conf. Proc. (2010), Vol. 1234, pp. 736-739

[4] J.A. Thompson, I. Horswell, J. Marchal, U.K. Pedersen, S.R. Burge, J.D. Lipp and T.C. Nicholls “Controlling the EXCALIBUR Detector” ICALEPCS 2011.

[5] T.M. Cobb. “Integrating Gigabit Ethernet Cameras into EPICS at Diamond Light Source” ICALEPCS 2011.

[6] B.J. Nutter, T.M. Cobb, M.R. Pearson, N.P. Rees, F. Yuan. “Recent Developments in Synchronised Motion Control at Diamond Light Source” ICALEPCS 2011.

[7] R.J. Woolliscroft, C. Coles, M.R. Pearson. “Quick EXAFS Experiments Using a New GDA Based Eclipse RCP GUI with EPICS Hardware Control” ICALEPCS 2011.

[8] R. Mercado, I.J. Gillingham, J. Rowland, K.G. Wilkinson. “Integrating EtherCAT Based Remote IO into EPICS at Diamond” ICALEPCS 2011.

[9] M.G. Abbott, T.M. Cobb. “An EPICS IOC Builder” ICALEPCS 2011.

[10] R. Walton, “DiffCalc User Guide”; http://www.opengda.org/documentation/manuals/Diffcalc_User_Guide/trunk/contents.html

[11] M. Rivers, “EPICS trajectory Scan”; http://cars9.uchicago.edu/software/epics/trajectoryScan.html

[12] M. Rivers, “EPICS AreaDetector”; http://cars9.uchicago.edu/software/epics/areaDetector.html