

GDA: Open Source Data Acquisition

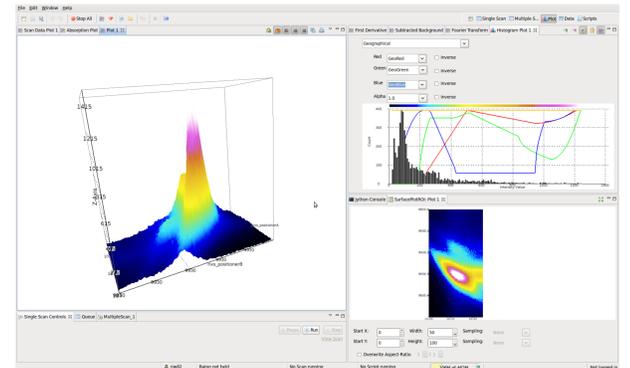
Richard Woolliscroft, Diamond Light Source on behalf of the GDA development team

The Generic Data Acquisition system (GDA) is a Java-based data acquisition system developed at Diamond Light Source Ltd., UK. It has been used on all the Diamond beamlines since the facility opened to users in 2007. The core of the software has been Open Sourced under the GPL license since December 2009 and is available at www.opengda.org.

This poster outlines the GDA and the major changes in the last couple of years.

Introduction

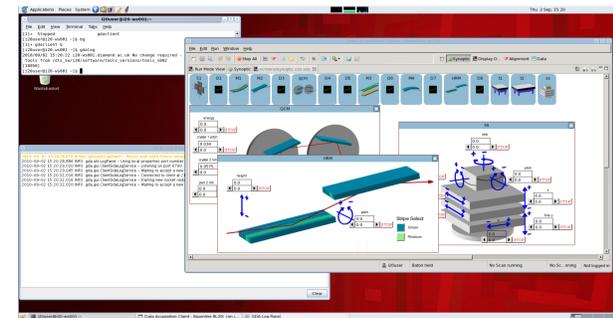
- Main technologies:
 - core written in Java
 - embedded Jython interpreter
 - code managed using the Eclipse plugin framework
 - client program uses Eclipse RCP (previously Swing)
 - server-client communication by CORBA
 - Nexus is the main data format
- Used on all Diamond beamlines across all different disciplines
 - Also at DUBBLE (BM26, ESRF)
- Provides the user-level software in a single client
- A powerful, flexible scanning mechanism at its core
 - Scans operated through the Jython environment
 - Customised Jython syntax to provide simple commands
 - GUIs drive Jython scripts
 - Training courses in Jython and GDA scripting given to beamline staff
- Spectrometer calculations included in the Jython environment
 - DiffCalc (<http://www.opengda.org/OpenGDA/About/Diffcalc.html>)
- Hardware control either direct or via a separate controls system
 - EPICS at Diamond
 - TANGO at ESRF



Example of the 2D surface plotting capabilities. Shown is an XES scan – a 2D energy scan over monochromator energy and secondary analyser crystal energy to get high resolution XAS data.

Open source

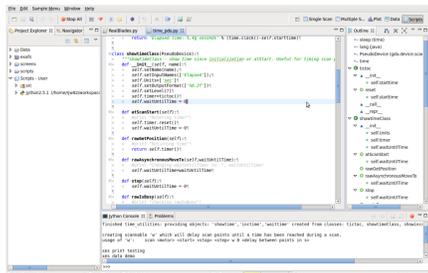
- Enabled by breaking up code base using Eclipse plugin framework
 - A core set of plugins has been identified
- Specific functionality in separate plugins which implement interfaces in the core
 - EPICS
 - Tango
 - Diamond site-specific code
 - Technique-specific code
- Download from www.opengda.org
- Used at DUBBLE (BM26, ESRF)
 - Where the control system is Tango



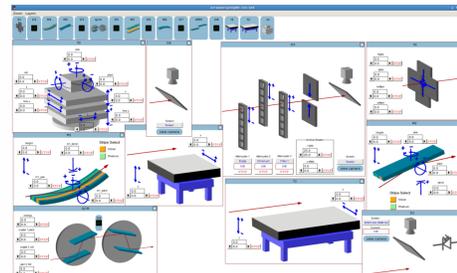
Example of the GDA client running within an NX session. The NX performance is good enough remote running of the GDA client or support by beamline staff.

Switch from Swing to Eclipse RCP client

- SWT gives a more feature-rich environment than Swing for building complex GUIs
 - Many more APIs for tasks such as job management, event management, editors, actions
- Each beamline builds its own GUIs built on a common set of 'views'
- Views can be shared between different beamlines and facilities using the Eclipse plugin model
 - Makes collaboration easier
 - Third-party plugins can be integrated more cleanly e.g. the PyDev Jython editor, CSS Studio



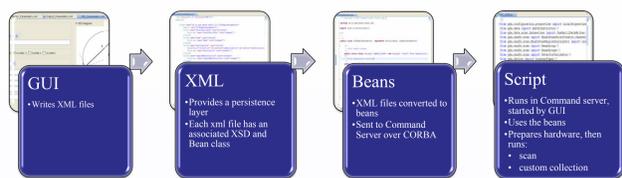
PyDev (pydev.org) Jython editor embedded in a GDA client. It has been extended to handle the GDAs extension of the Jython syntax



Example of the DESY Control System Studio used within the GDA (css.desy.de)

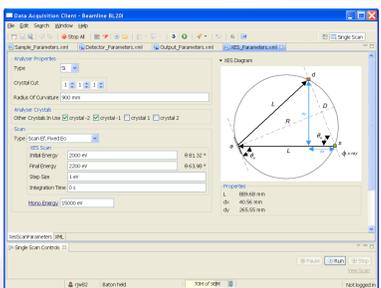
Role-based access control

- Use Eclipse editor framework to create graphical experiment editors which write xml files as users type
 - Persists experiment options
 - This makes it easier to re-run experiments or to graphically set up longer experiments

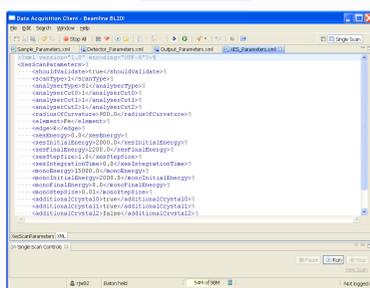


Client Number	User	Visit	Holding Baton
3	i20user*	cm1903-5	
2	rjw82	cm1903-5	

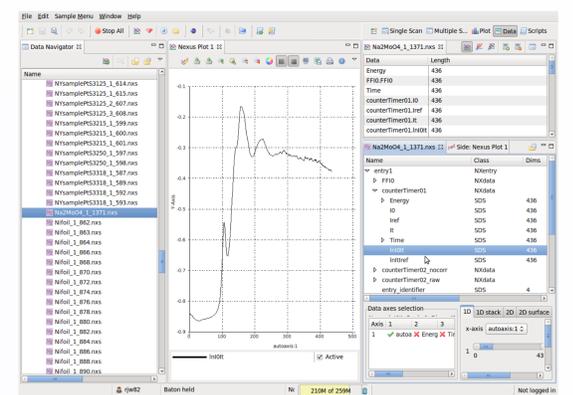
The view in the client which lists all the clients on that beamline, who is running each client and which experiment they are running as.



Example of a graphical editor on the I20 spectroscopy beamline. Each aspect of the experiment is defined by a different xml file with its own editor. The one shown defines an X-ray Emission Spectroscopy (XES) experiment.



The XML file for an XES experiment. The information in the XML editor and graphical editor are kept synchronised so users may edit either to define their experiment.



The GDA Nexus file viewer. 1D stacks, 2D plots and surface plots are available.

Future

- External Git repository
- Stand alone data analysis program (SDA)
 - By repackaging existing code
- New MX GUI
 - Database-centric, not hardware-centric
 - ISPyB is the underlying sample-tracking database
- Continue to refactor code base
 - New technique-specific plugins which are not Diamond-specific e.g. NCD, XAS

