Creating, as a community, a customisable Java software framework to operate experiments on synchrotron facilities

Paul Gibbons PhD.
Presented at WP10 workshop - Beamline instrumentation software and possible collaboration between synchrotrons in Europe, ESRF Jan 2011

A Data Acquisition System has a wide range of use

- large range of science techniques - surface science, MX, tomography, spectroscopy,
- large range of equipment throughout the facility
- high throughput industrial type measurements by users with little need or desire to understand how the measurements are being taken
- large data volumes
- 24x7 data collection with little support from engineering staff
- Remote Access

A Data Acquisition System must have many different features

- ease of use
- automatic data reduction and visualisation
- tools to aid analysis to allow users to get confidence in quality of data or to direct the current experiment
- use of a common data format to allow automatic interpretation by visualisation and analysis programs
- connection to user databases used to direct and record the measurement/experiment

A Data Acquisition system must be adaptable

- flexibility to allow rapid adaptation of techniques by beamline staff rather than by engineers
- a common tool set, structure and user interface to allow users, beamline staff and engineers to easily move from one beamline to another
- expectations of GUI's are always going up and so GUI's must be easy to change without impacting the beamline operation

A Data Acquisition system must work with other systems

- connection to archival systems
- connection to user databases for authorisation
- logging facilities
- error reporting systems

A Data Acquisition must be cost effective

- The skills required to maintain and enhance it must be readily available
- The technology must be relevant for a very long time
- The system must be maintainability and remain able to keep up with future demands

What is GDA?

- One or more ObjectServers each holding collection of Java objects.The objects may themselves provided a connection to external hardware controllers e.g. EPICS devices
- A mechanism for one object to access another across ObjectServers
- A mechanism for objects to notify other objects of changes of state
- One ObjectServer contains a Jython Interpreter.This has access to the other objects in the system.

What is GDA ? Part 2

- Rich GUI clients that also have access to the other objects in the system, including the Jython Interpreter. The Rich GUI is constructed using Eclipse plugin framework.
- Eclipse Plugins exist to give access to the objects in the servers, the Jython interpreter and provide data visualisation. Top level plugins also give users the ability to define, execute and monitor measurements.

Architecture

Local users

Remote users

CORBA

CORBA

Scripting Engine

GDA servers

Object Servers

JCA/CAJ

EPICS IOCs

Socket

Detectors

How does GDA satisfy the demands placed upon it -(1)

- highly flexible
  - The objects are constructed using Spring Framework config files which places little restrictions on their form or function
- integration - Jython language used to integrate the objects is very powerful
- little engineering support - the system is stable and well tested
- very powerful common scan mechanism that can easily be used to scan any combination of real or virtual devices at any level of nesting

How does GDA satisfy the demands placed upon it (2)

- easy to use command line access to a Jython interpreter to allow low level control of the system,
- ability to execute user written Jython scripts to automate lengthy or complex procedures.
- CAJ/JCA built in and GDA helper classes allow easy creation of devices to interface to EPICS pvs. Tango plugin exists.
- DiffCalc provides scanning in hkl space by diffractometer in various modes. Diffcalc objects fit perfectly into the scan mechanism

How does GDA satisfy the demands placed upon it (3)

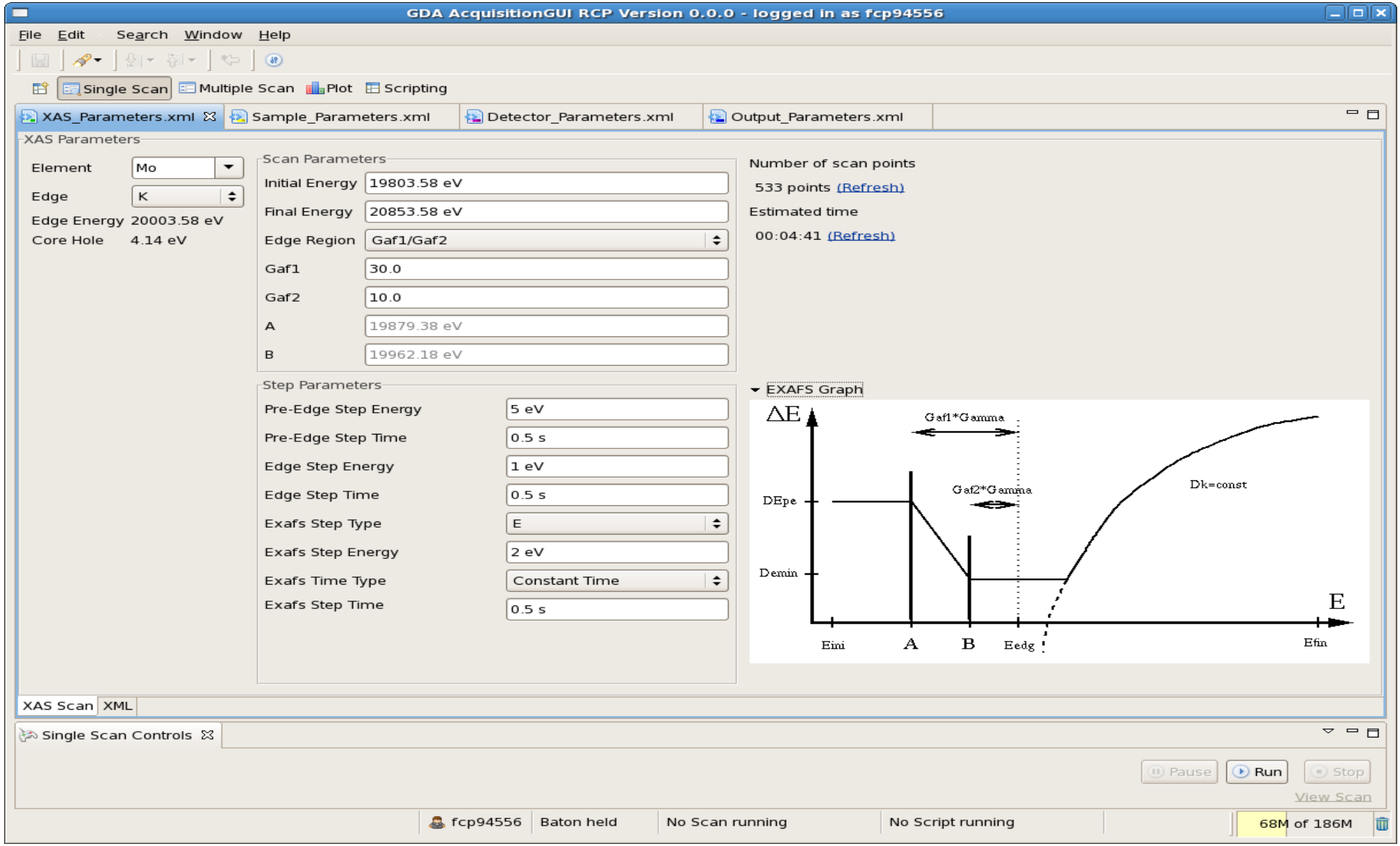
- Client server design to allow multiple users to access the system at once without endangering the data collection process.
- A baton system to control who has ability to change the beamline operation.
- An authorisation system to allow different user levels of control.

GDA
software for science

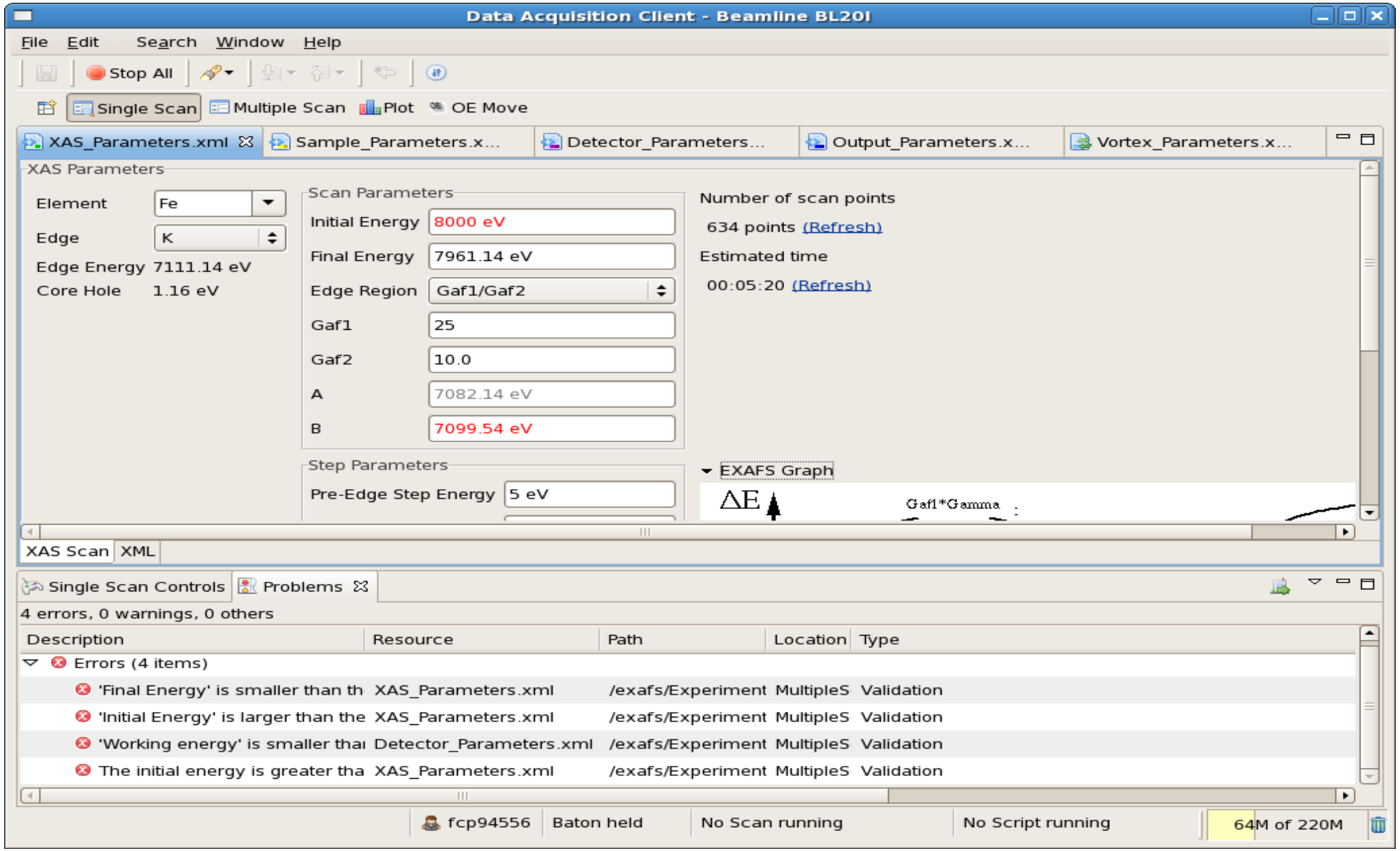
How does GDA satisfy the demands placed upon it (4)

- The Eclipse Rich Client Platform plugin framework to allow software engineers to create either technique specific client gui's and associated server components, data visualisation components or data reduction pipelines.
- A basic set of GUI panels that allow users to write scripts, enter commands, scan devices and visual results.
- A proven design pattern for interfacing GUI aspects of experiment definition to server script.

Experiment Definition GUI design - common widgets



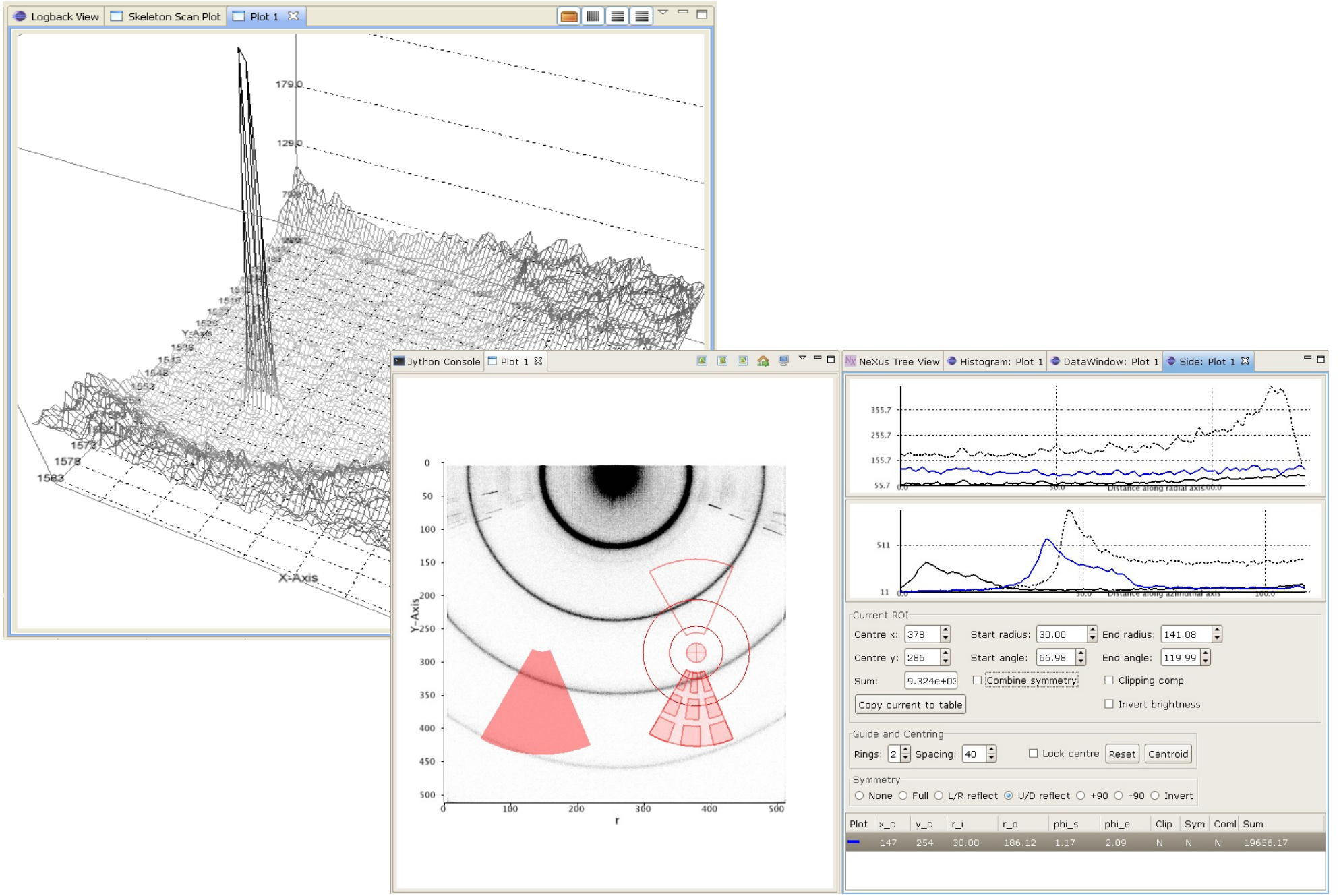
Experiment Definition GUI design - built in validation



How does GDA satisfy the demands placed upon it (5)

- Use of SDA Components
- A growing set of data visualisation tools using OpenGL to make use of hardware capabilities of the system graphics card.
- A set of FileIO tools to read data files of various formats into a common data format (DataSet) for access by the rest of the analysis tools
- The DataSet class aims to emulate some of the functionality of NumPy's ndarray class.

GDA
software for science

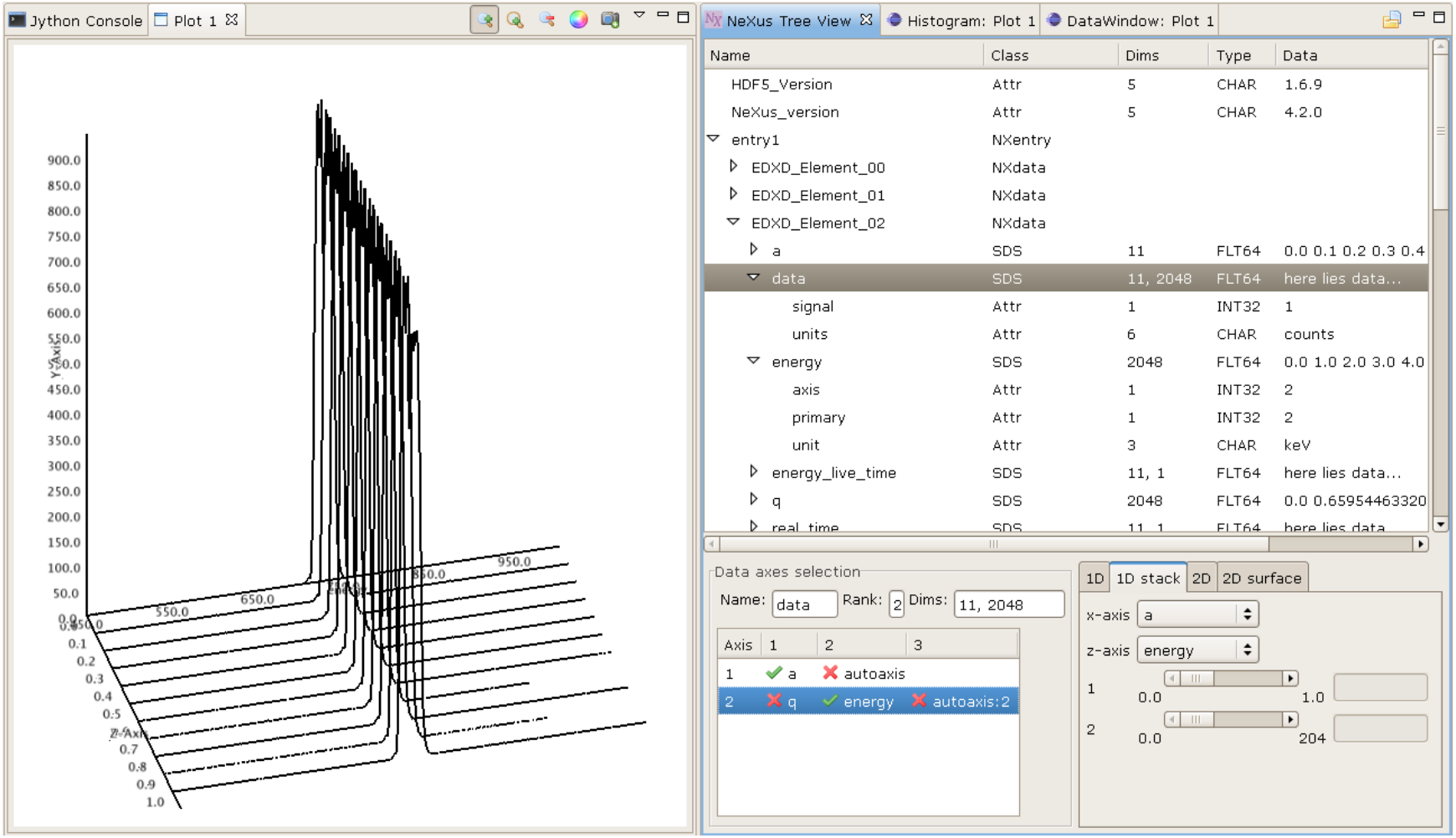


GDA
software for science

How does GDA satisfy the demands placed upon it (7)

- Being Eclipse based the GUI can easily be augmented by other plugins that are not aware of GDA. Some of these are already built in such as:
- PyDev python editor - http://pydev.org/
- CSS Synoptic Display Studio which allows non software experts to create bespoke panels to talk to both GDA native objects or directly to EPICS - http://css.desy.de/content/index_eng.html
- EDNA - analysis pipeline - http://scripts.iucr.org/cgi-bin/paper?wa5014

Full support for Nexus file creation and visualisation.



How does GDA satisfy the demands placed upon it (8)

- Being Java based it is easier to find software engineers with required skill set.
- There is a wealth of good Java development tools.
- Use of widely used Java technologies , e.g. Spring Framework(http://www.springsource.org/) , Jython (http://www.jython.org/) , Eclipse (http://www.eclipse.org/)
- Jython is easy for beamline staff and 'expert' users to learn.

Collaboration