Progress towards connecting Global Phasing's workflow with MXCuBE

Gerard Bricogne, **Rasmus Fogh,**
Global Phasing

MXCuBE meeting, Grenoble
January 2017

GΦL

Contents

- **Preamble**
- Workflow architecture and MXCuBE connection
- Workflow demonstration

GΦL

Introducing myself

- Coming from NMR, CCPN project
  - Data models and standards
  - Python application development
  - Software integration
  - Structural biology
- I am not a crystallographer
  - but I can learn.
  - Please be patient

GΦL

Contents

- Preamble
- **Workflow architecture and MXCuBE connection**
- Workflow demonstration

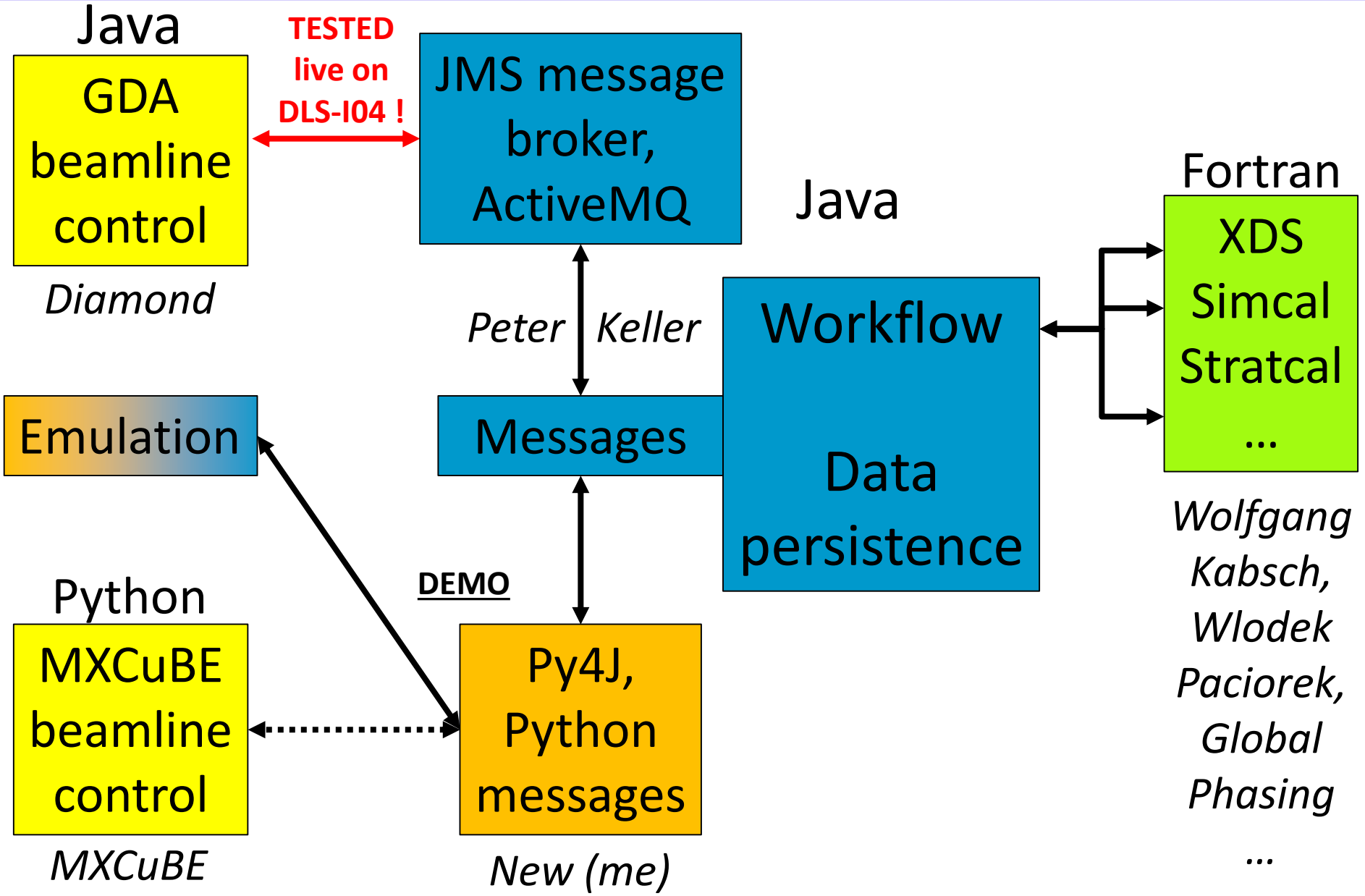
GΦL

Workflow operation

- Drives the beamline, data collection and (eventually) processing
- Beamline-independent core; messages using Abstract Beamline Interface
- Works through beamline control system
- Input, instructions and output through beamline user interface (or emulator)

GΦL

Architecture



MXCuBE messaging

- Synchronous RPC calls from workflow
  - Beamline side py4j server
  - Java message objects
  - Python access via py4j
  - Conversion to pure Python message objects matching **Abstract Beamline Interface**
  - Self-contained, beamline-independent, can be replaced
- Messages trigger processMessage calls
  - Hook for beamline emulator, or beamline control
  - Intended to hook into MXCuBE queue

GΦL

Messages from Workflow

- RequestConfiguration
- RequestCentring
- ChooseLattice
- GeometricStrategy (requests re-centring)
- CollectionProposal (requests data)
- Information messages:
  - SubprocessStarted
  - SubprocessStopped
  - Text (for echoing)
- Workflow Ready/ Aborted / Completed / Failed

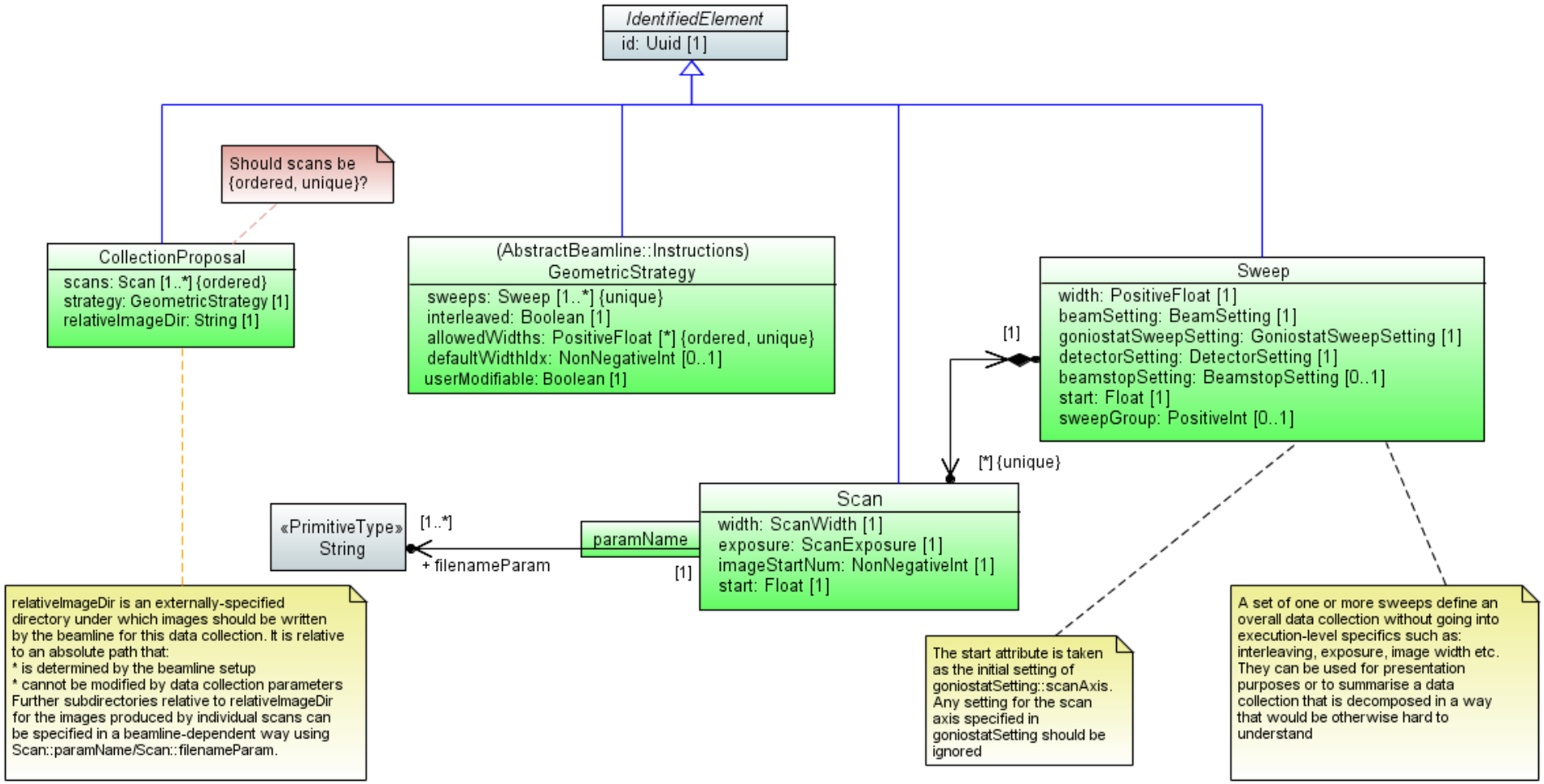
GΦL

Messages to Workflow

- ConfigurationData
- PriorInformation (Sample initial data)
- SampleCentred
- SelectedLattice
- CollectionDone
- BeamlineAbort
- CentringDone (for calibration workflow)

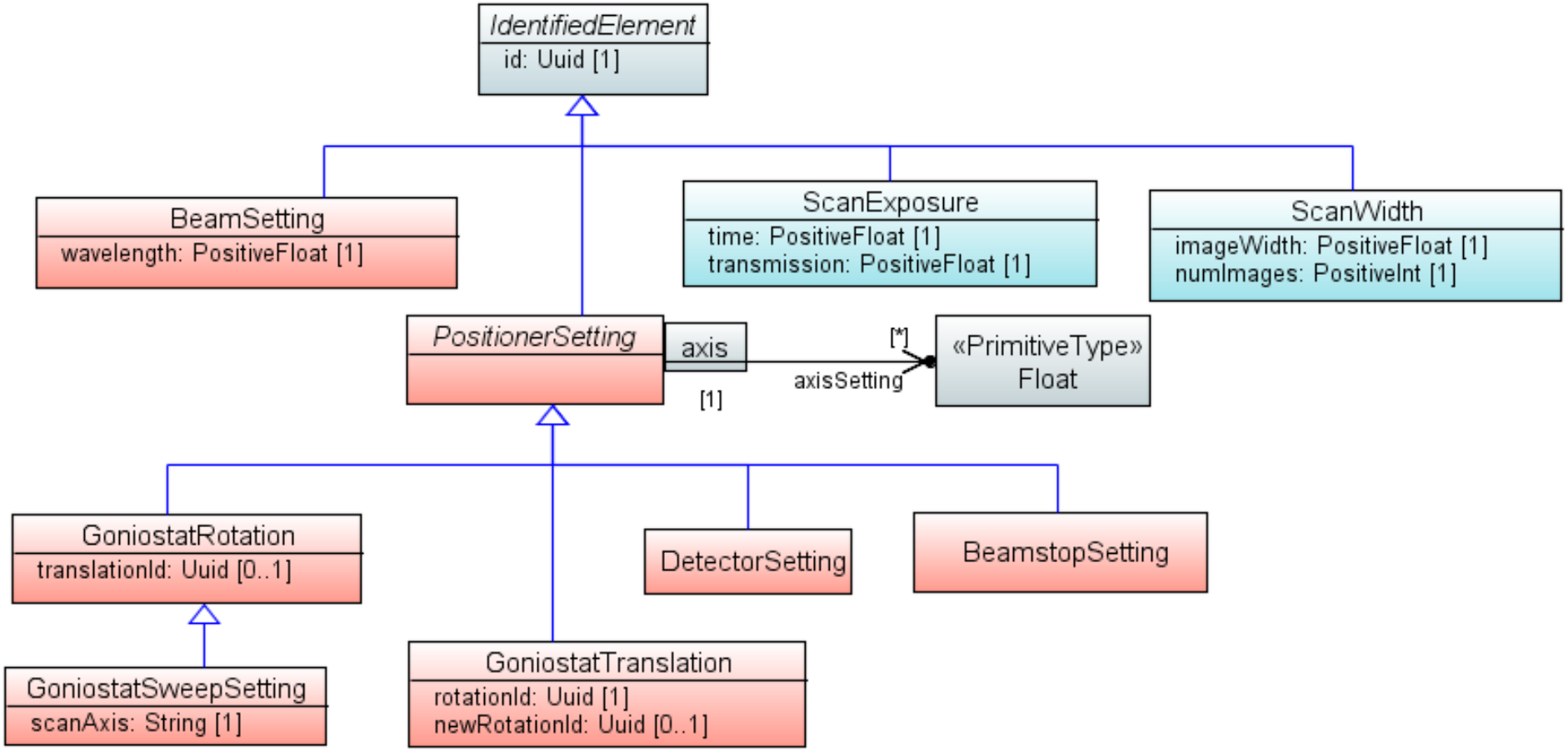
GΦL

Data Collection UML



GΦL

Instrument Setting UML



GΦL

Future issues

- Messaging protocol
  - Synchronous or asynchronous?
  - Is py4J OK?
  - XML-RPC server?
- Handshaking and control
  - Who triggers whom?
  - Processes, aborts, and restarts
- User interface
  - Need for changes or dedicated modules?
- Calibration and configuration
  - High standard required
  - Bespoke calibration workflows

GΦL

Contents

- Preamble
- Workflow architecture and MXCuBE connection
- **Workflow demonstration**

GΦL

Demonstration

- I show running demo, interspersed with screenshots from live DLS-I04 test
- Workflow program runs in full production mode
- Client-server communication fully functional
- Beamline control system (BCS) is emulated by a plug-in replacement (mock-BCS)
- Mock-BCS runs as separate server process

GΦL

Beamline emulation

- Mock-BCS based on
  - Configuration files
  - Sample description files
  - Dummy centring values
  - User input
  - Copies of pre-acquired characterisation images

- Full emulation (existing, not demonstrated)
  - Images calculated from structure model, mosaicity value, specific crystal orientation, …

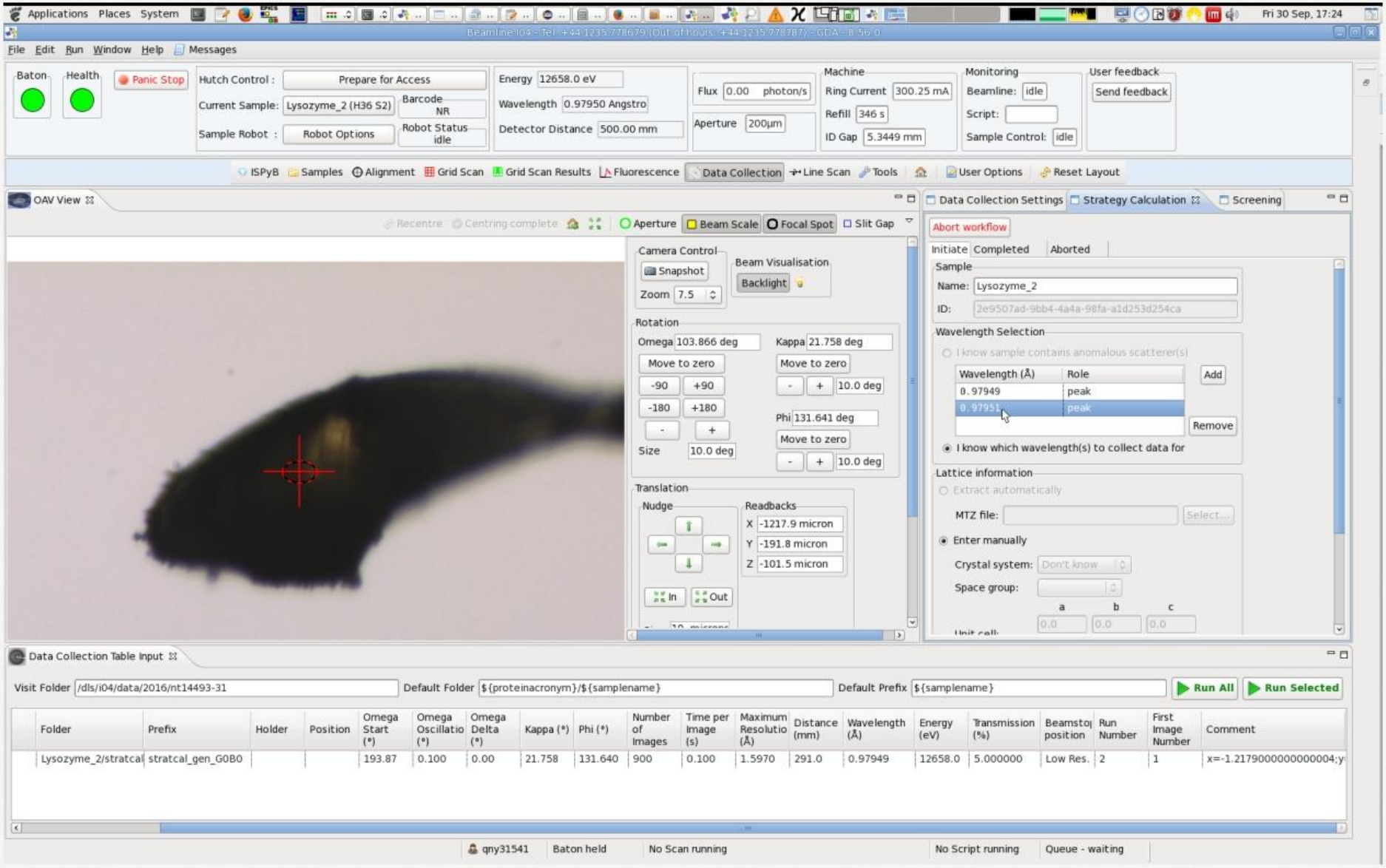
GΦL

Workflow *demo* 1

- Start mock-BCS and workflow
- Load sample information and parameters
- Centre sample

GΦL

DLS-I04 *live* - image 1



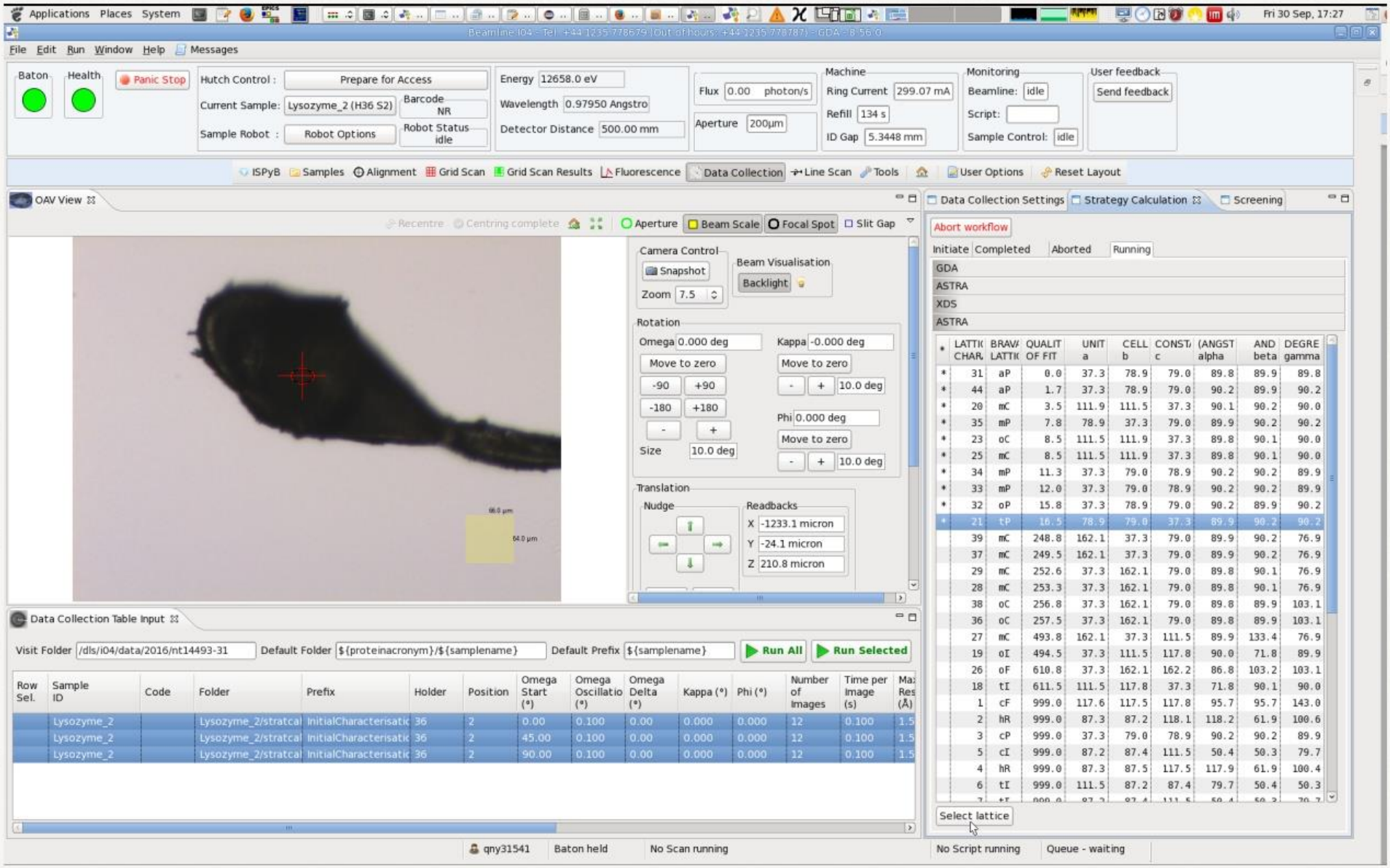
Setting the workflow's initial parameters

Workflow *demo* 2

- Collect characterisation data
- Run indexing (XDS, COLSPOT, IDXREF)
- Select indexing solution

GΦL

DLS-I04 *live* - image 2



Choosing an indexing solution

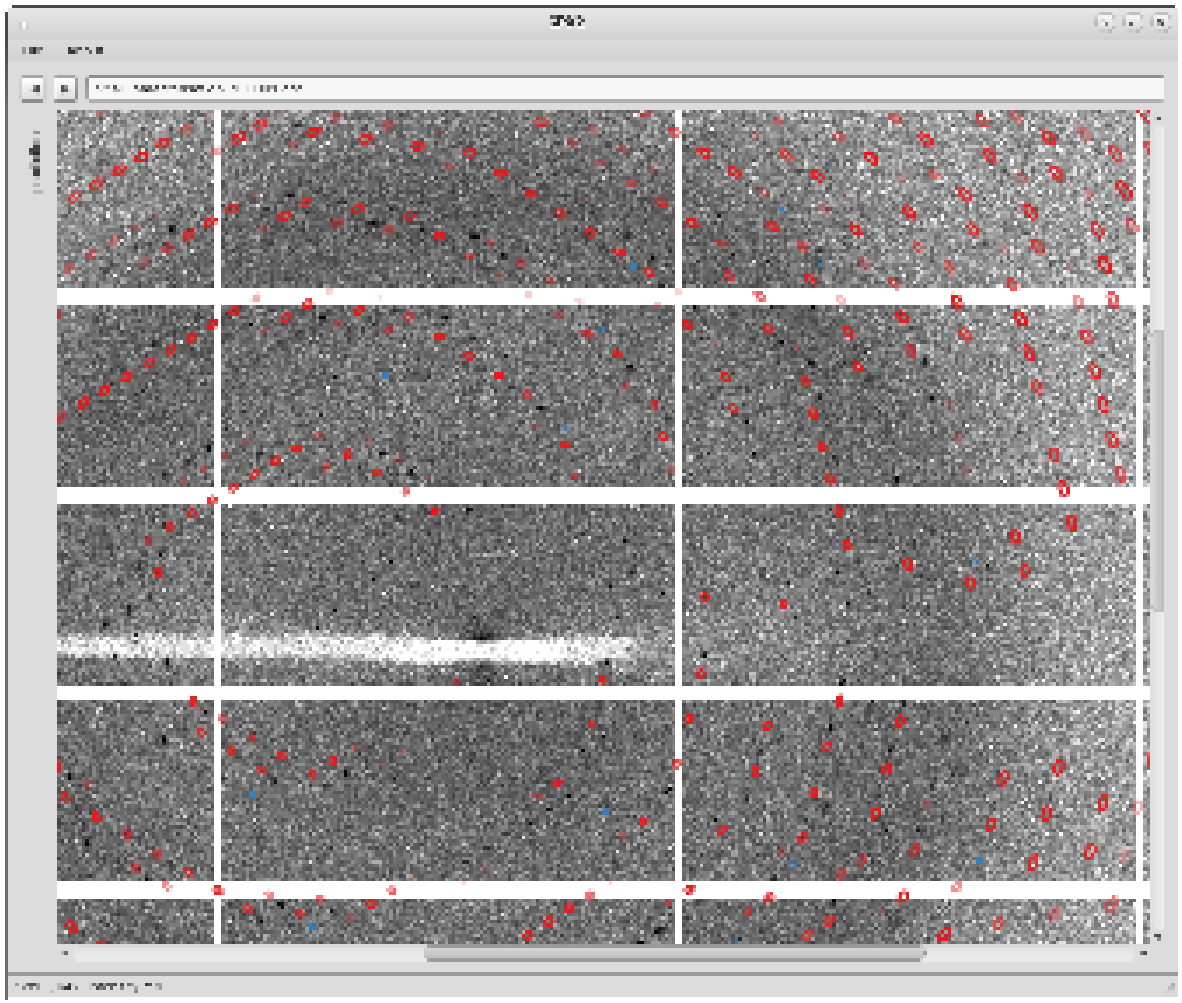
Workflow *demo* 3

- Calculate spot predictions from indexing solution for characterisation images (XDS, simcal_predict)

- User check of indexing solution (GPX2)

GΦL

DLS-I04 *live - image 3*
GPX2 display of characterisation images



The spots used for indexing (faint blue crosses) come from the SPOTS.XDS file. Since these characterisation images are fine-sliced (0.1^{o}) the spots are all partial, so that many of their blue crosses will be found on adjacent images.

This example shows some mismatch between predictions (red ellipses) and measured spots, probably due to a lack of up-to-date calibration data for the goniostat.

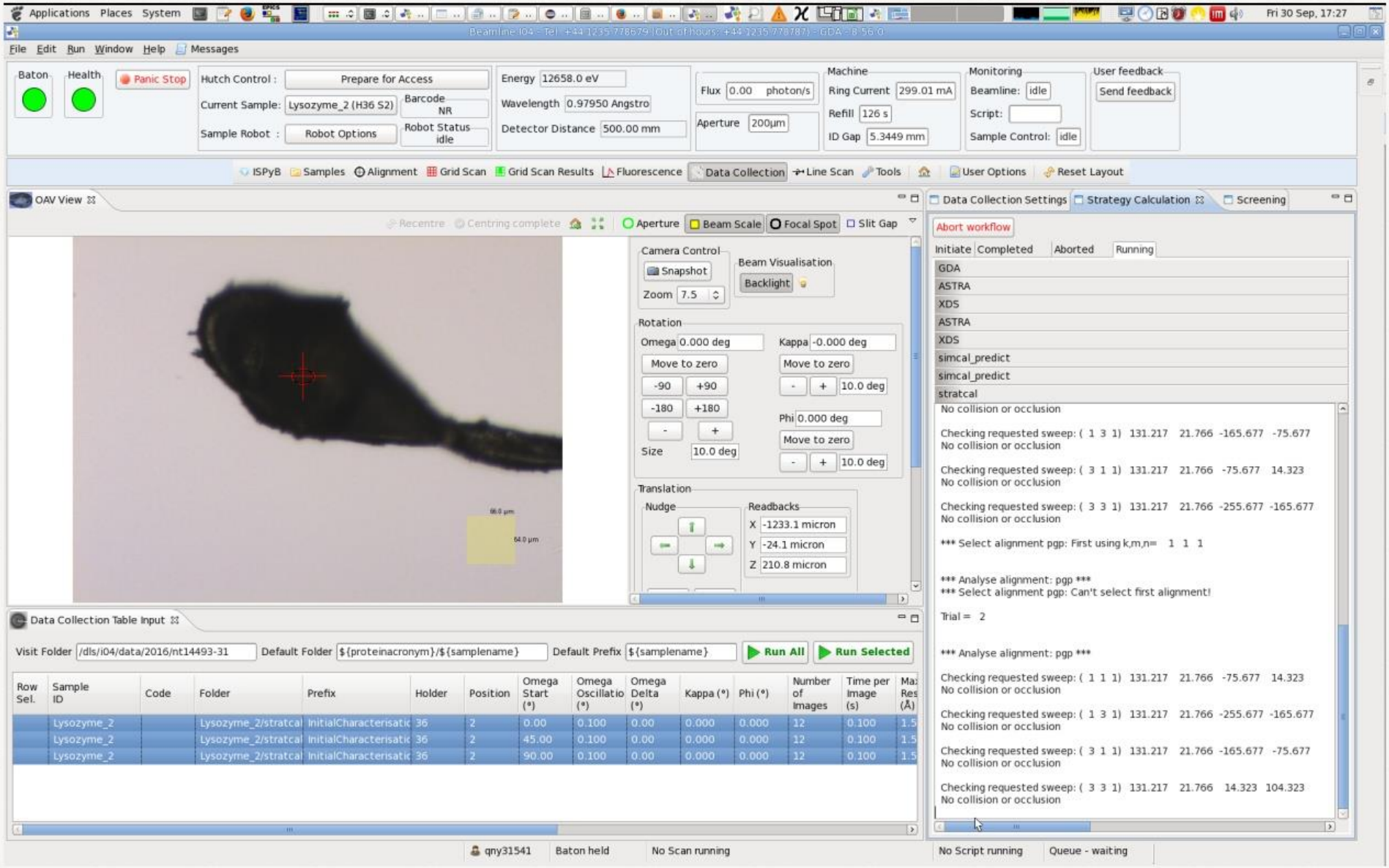
GΦL

Workflow *demo* 4

- Calculate acquisition strategy (stratcal)

GΦL

DLS-I04 *live* - image 4



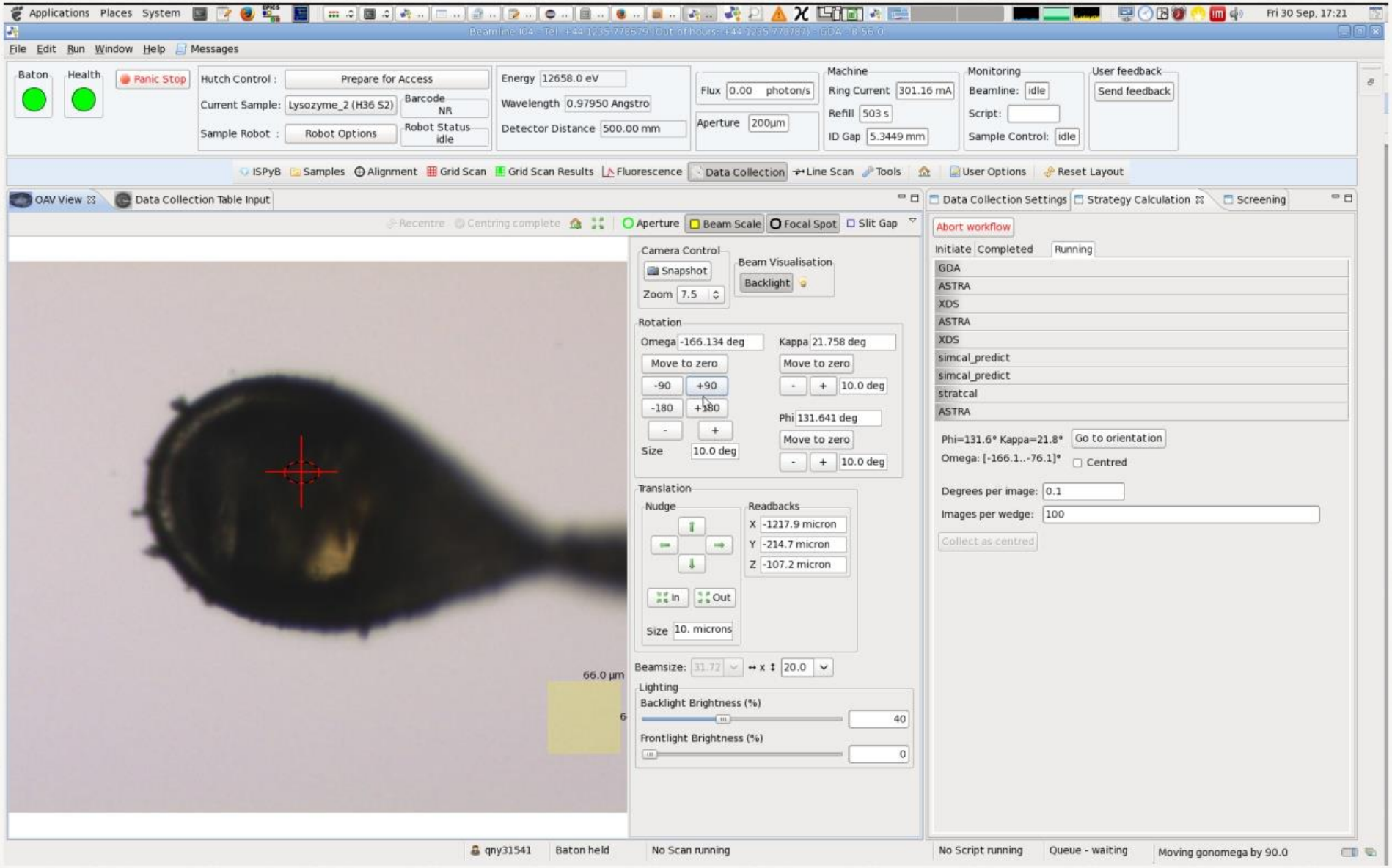
stractcal in mid-run

Workflow *demo* 5

- Check sample re-centring for each sweep of stratcal acquisition strategy

GΦL

DLS-I04 *live* - image 5



Centring the crystal for the strategy output by stratcal

Workflow *demo* 5

- Collect data

- Demo ends here

GΦL

Where we are

- Workflow engine fully operational
  - Tested and working on live beamline (GDA/ASTRA, DLS-I04)
- Message transfer to Python fully operational
  - Running under demo
  - Current (prototype) implementation could be replaced
- Next step is connecting to MXCuBE queue

GΦL

END

GΦL