

# Data Analysis Workbench

- An open source not for profit project*
- On GitHub **'DawnScience'***
- Diamond Light Source Ltd. and the ESRF are largely publically funded research facilities*

# Disclaimer

**AKA - who says that?**

- **A Java Software Developer (not a Scientist) worked for 16 years with various Java based applications in science and engineering**
- I will attempt to explain a bit of the science for your enjoyment (hopefully not *schadenfreude*).
- Talk biased towards how Diamond and the ESRF are using Ptolemy 2
- An Eclipse/RCP fan

Matthew Gerring



# Synchrotron

**AKA – cool word, but what does it mean?**

**syn·chro·tron**/'siNGkrə, trän/

Noun: A cyclotron in which the magnetic field strength increases with the energy of the particles to keep their orbital radius constant.

**“They are machines which produce very strong light used for many different type of scientific experiments and sometimes other things.”**



ESRF  
(Experimental Facility)

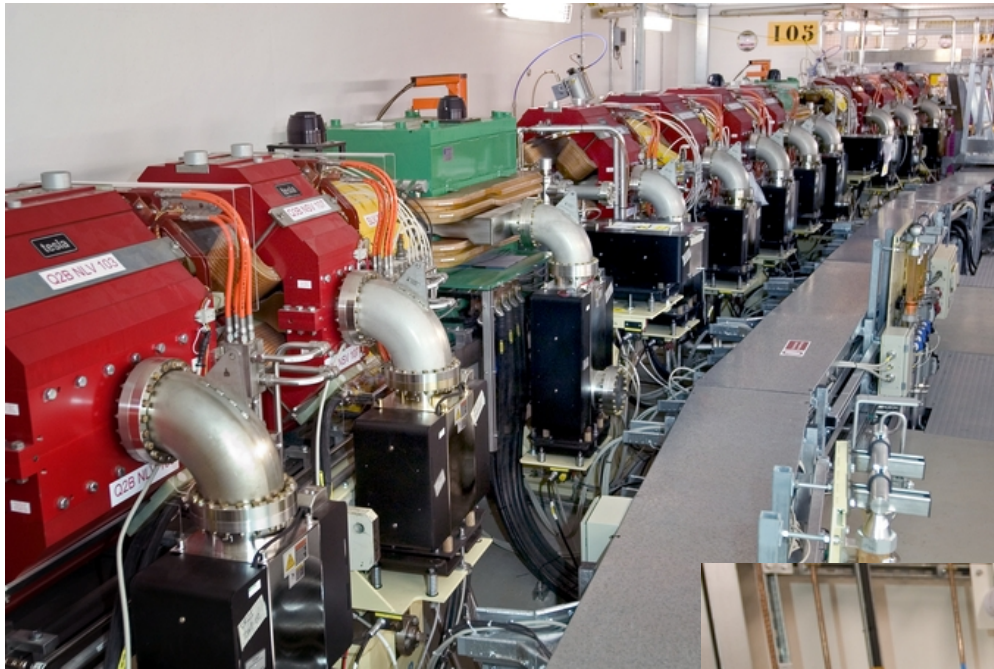
Diamond  
(User Facility)







The Queen and Duke of Edinburgh  
at the official opening of DLS, 19<sup>th</sup>  
October 2007

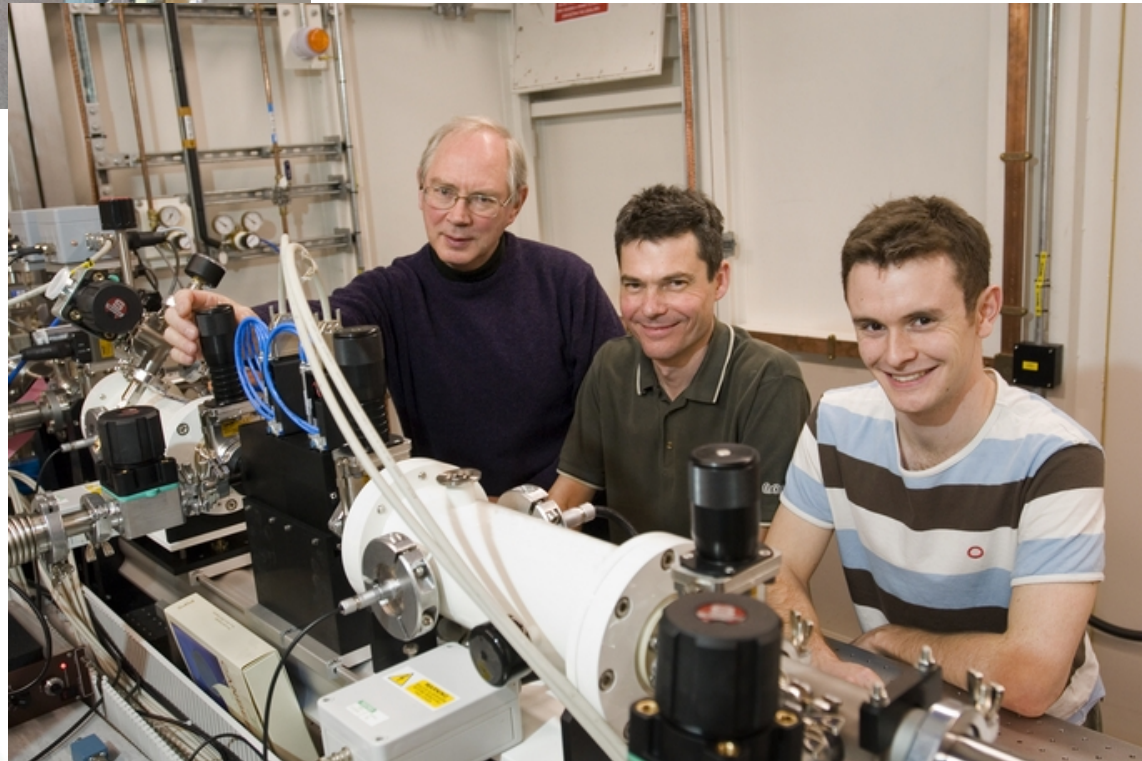


Scientists with some  
of the hardware used  
in their research



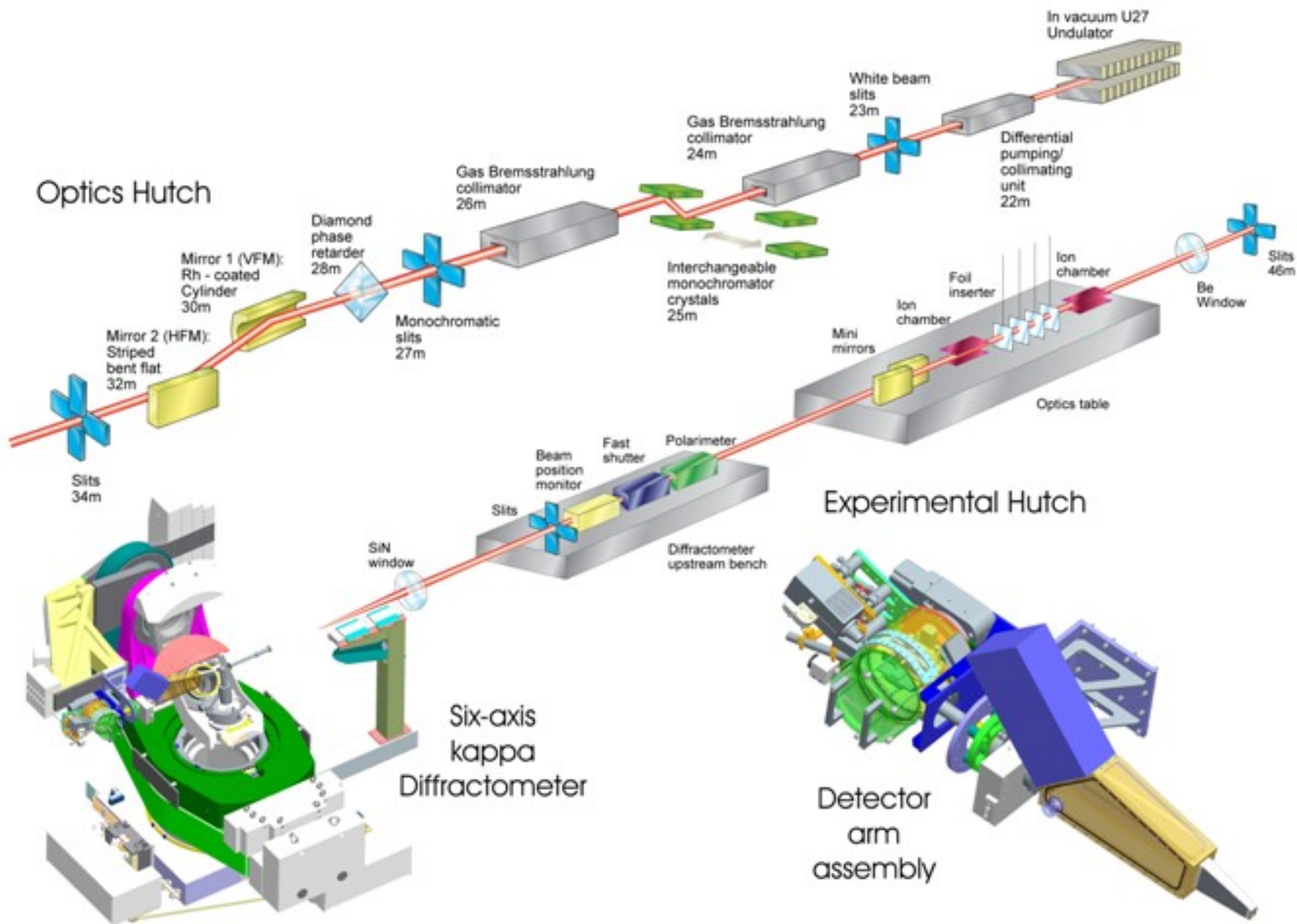
Inside the storage  
ring [not star-trek  
“conduit”...]

Video of Diamond...

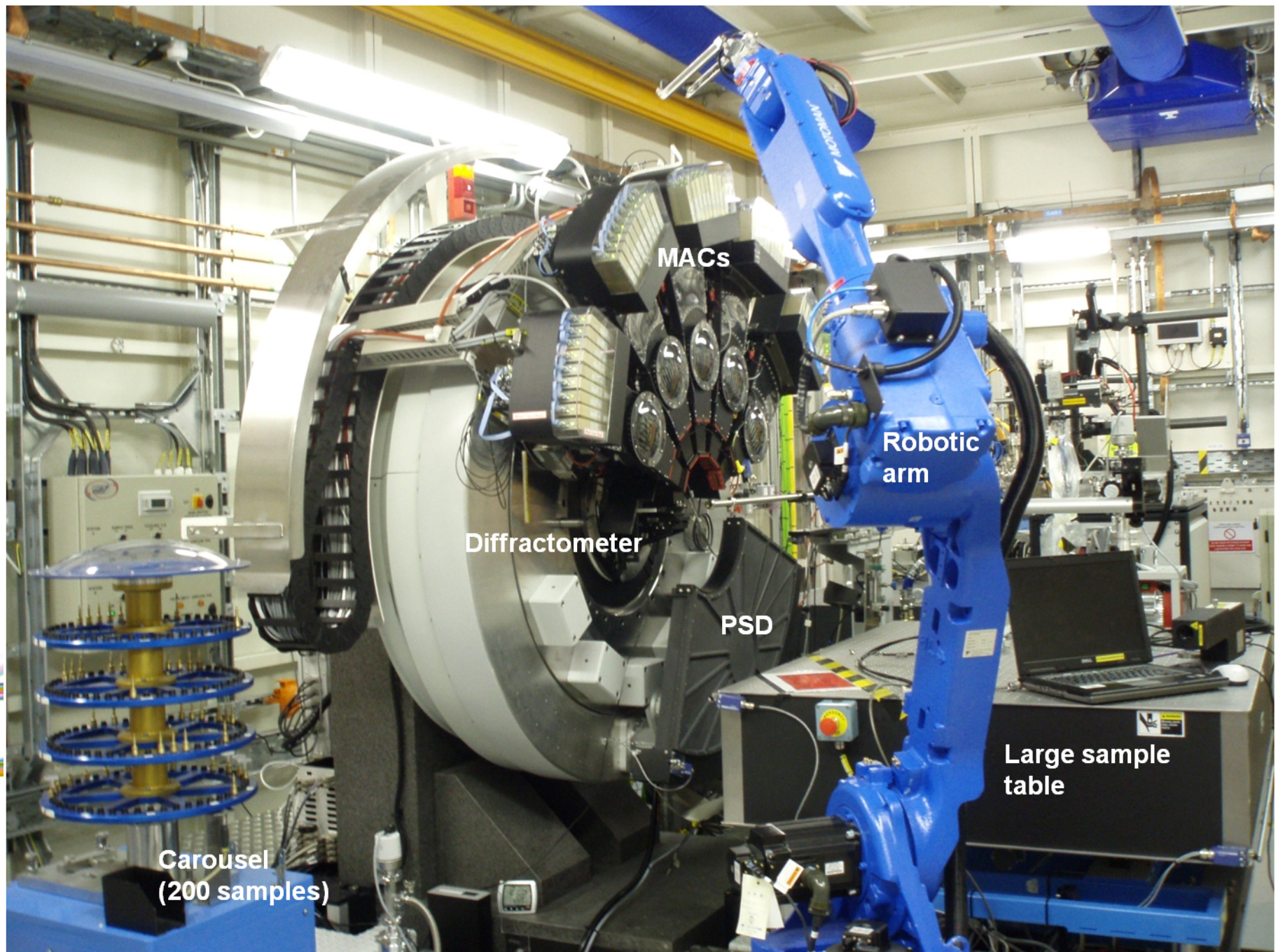




## Optics Hutch







MACs

Robotic  
arm

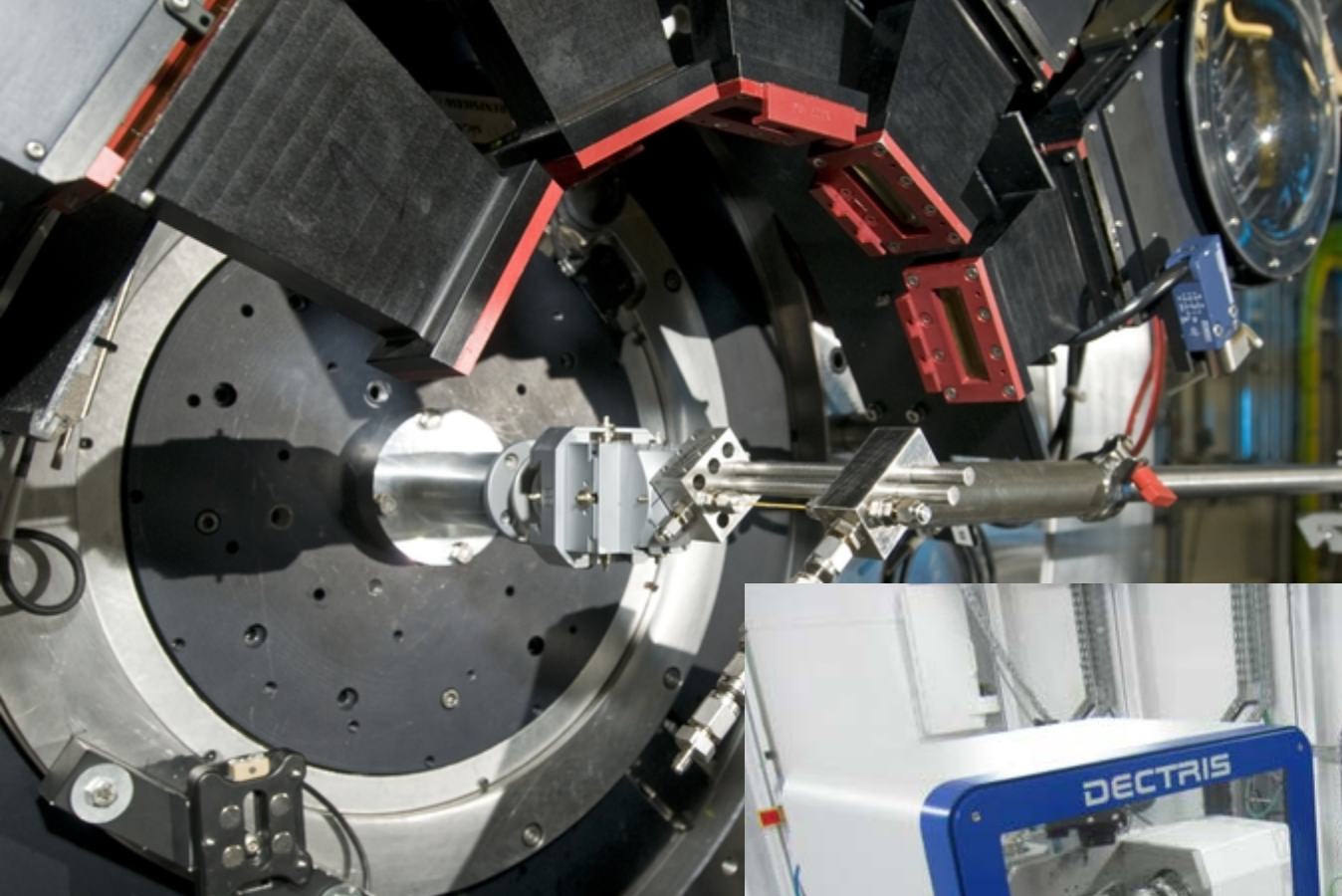
Diffractometer

PSD

Large sample  
table

Carousel  
(200 samples)





*Detectors of  
Various  
flavours*



11/07/2011 09:54

# Responsibilities

**AKA – what developers do at Diamond...**

- **Software for controlling experiments**

- Motors, detectors, configuration.
- A high quality and flexible GUI.
- Ptolemy 2 not currently used.
- Data collection scripts

- **Software for data**

- Ability to visually interact with n-dimensional data (i.e. graphs and slices).
- Ability to write scripts to interact with data.
- Custom user interface and forms for specific experiments.

## **Software for running analysis pipelines**

- Hard coded and/or user configurable options.
- Real time visualization of analysed results.
- Ptolemy 2



# Integration Tools

## AKA – how we are getting it done

- Eclipse IDE - around 20 developers, 8 in scientific software
  - Controls currently in process of migrating to RCP (~15 more developers)
- Eclipse RCP product built using *Buckminster* (previously PDE)
- Usage of *Jenkins* for continuous integration.
- Unit tests using *Squish UI Testing*, *Junit* and *Junit plugin* tests.
- We do not currently do code walkthroughs or pair programming. Agile practices being used where otherwise possible.
- We document our designs and code using *confluence*.
- We use **Cheat Sheets** for tutorials and testing guides.
- Source code control using **Git/eGit** (which has a pure Java client)

# ‘Shoulders of Giants’

- **RCP** many of the core features, editors, toolbars, views, projects
- **Ptolemy 2** (a version known as ‘**Passerelle**’) workflow and pipelining
- **GEF** for visualization of pipeline graphs
- **Draw2D** for 1D and 2D plotting (SWT XY Graph)
- **Pydev** for python/jython scripting layer used by the scientists
- **HDF5** libraries for storing large data sets
- **SWT/Jface** – lazy viewers being used extensively for large trees and tables
- Apache, Eclipse-WST, springsource, JDK, and many more of course...



## Scientific Data Analysis

<b>PEEMA</b>  Dawn APIs Visualization Slicing	<b>MX Live</b>  Dawn APIs Visualization Pipeline Connectivity	<b>NCD Data</b> Reduction / Calibration Dawn APIs Visualization Workflows Algorithms Maths APIs	<b>ARPES Data</b> Reduction  Dawn APIs <u>Workflows</u> Visualization Algorithms Maths APIs	<b>Tomography</b> Visualization Pipelines Python HDF5/Nexus Algorithms Maths APIs
---	--	--	--	---

### Visualization Layer

- 1D 'X vs. Y' Plots
- 2D 'Image' viewing
- 3D surfaces and stacks
- Slicing of data

### Scripting Layer

- Custom Data Analysis
- Integration existing scripts
- Plot Data

### Workflows

Pipelines and  
automated  
analysis

### DAWN APIs

Maths	Plotting and Tools	Loaders	HDF5 Nexus	File Conversion	Persistence
-------	-----------------------	---------	---------------	--------------------	-------------

### Third Party Open Source

Pydev

Passerelle

Others...

HDF5

Jama

Apache

### Eclipse RCP

Perspectives  
Editors, Views

Draw2D  
GEF

JFace

SWT

Java™ Virtual Machine

J  
y  
t  
h  
o  
n

P  
y  
t  
h  
o  
n

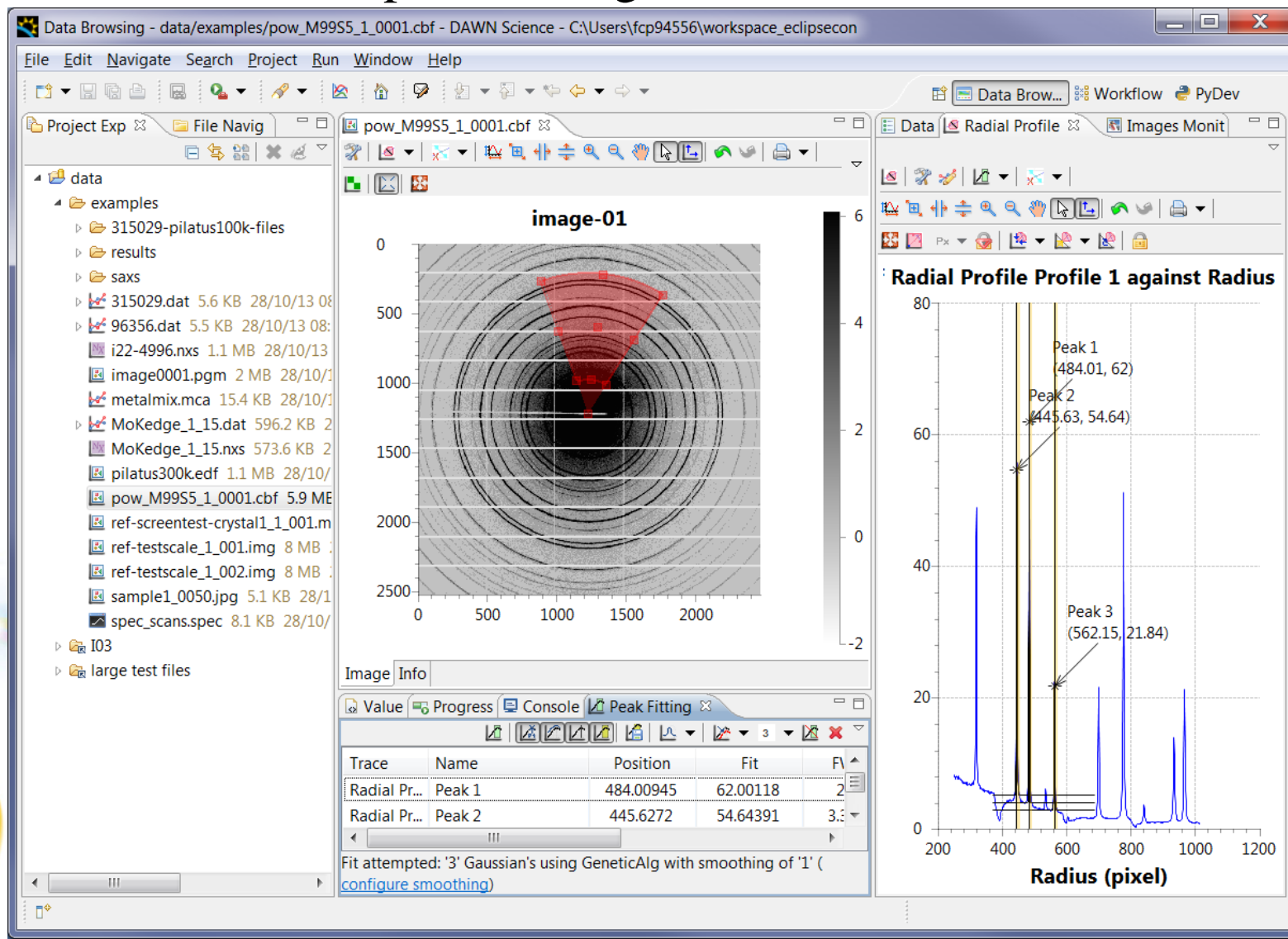
# Lots of Visual Tools

- For images
  - Line, Box, Sector integration
  - Diffraction image interpretation, line profile for ‘D-spacing’
  - Color mapping / Histogramming
  - Pixel Information and region control
- For XY Graphs
  - Peak Fitting and Line Fitting
  - Derivative and other functions, including user defined
  - Scientific tools
    - XAFS Analysis Tool
    - SAXS
- Use of eclipse architecture, extension points and pages inside PageBookView.



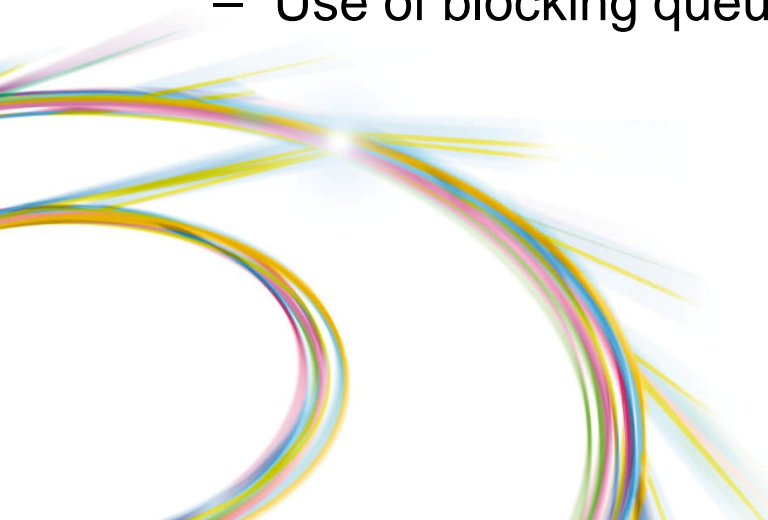
# Demonstration — Visual Tools

Example showing various visual tools



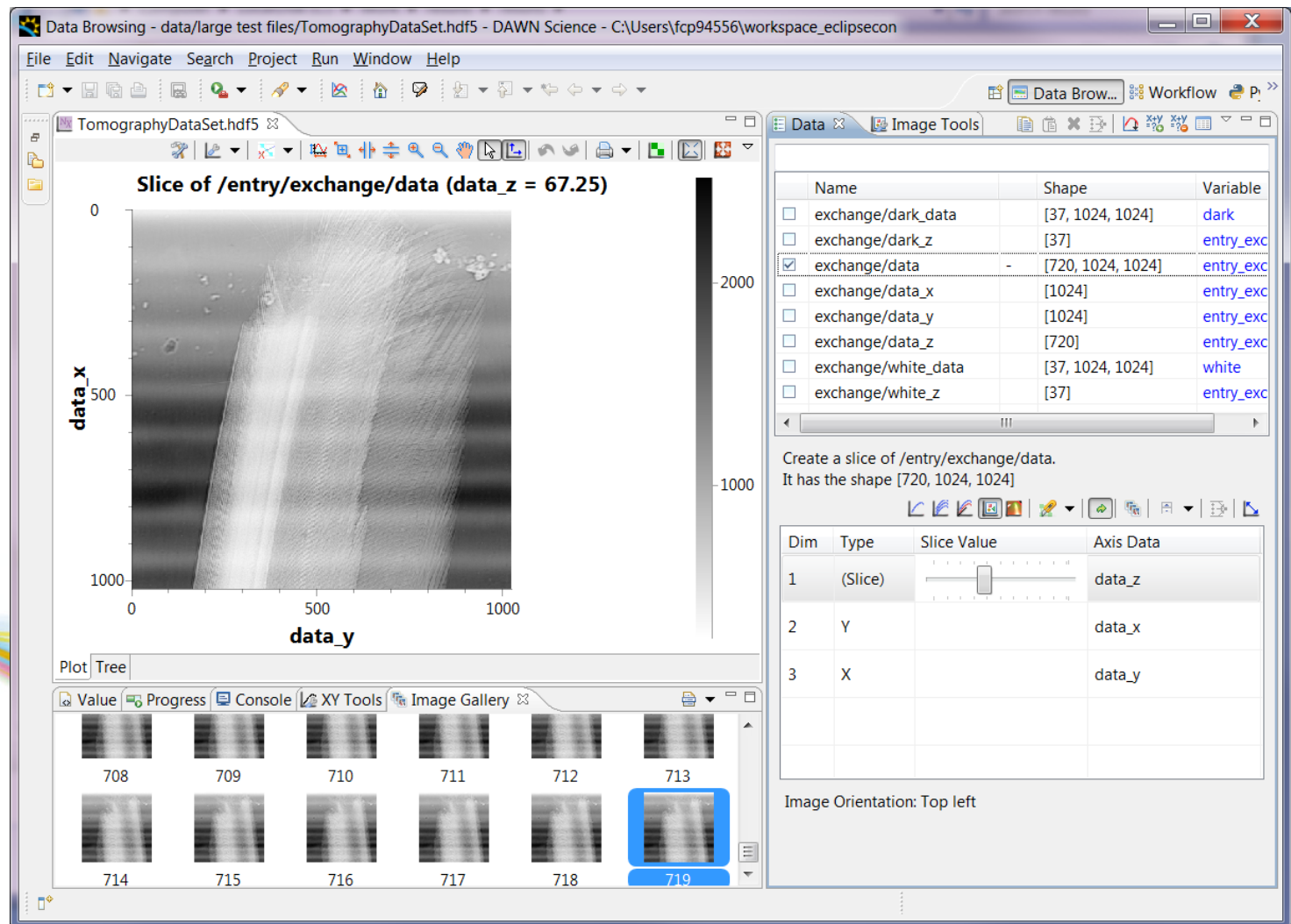
# Slicing data

- Cutting through N-dimensional data
  - With an XY plot
  - As an image
  - As a 3D iso-surface
  - Hyper 3D
- Important to run everything concurrently
  - Use of Jobs
  - Use of ordinary threads
  - Use of blocking queues



# Demonstration — Slicing and dicing

Example opening a tomography file and slicing it





# Passerelle Origins

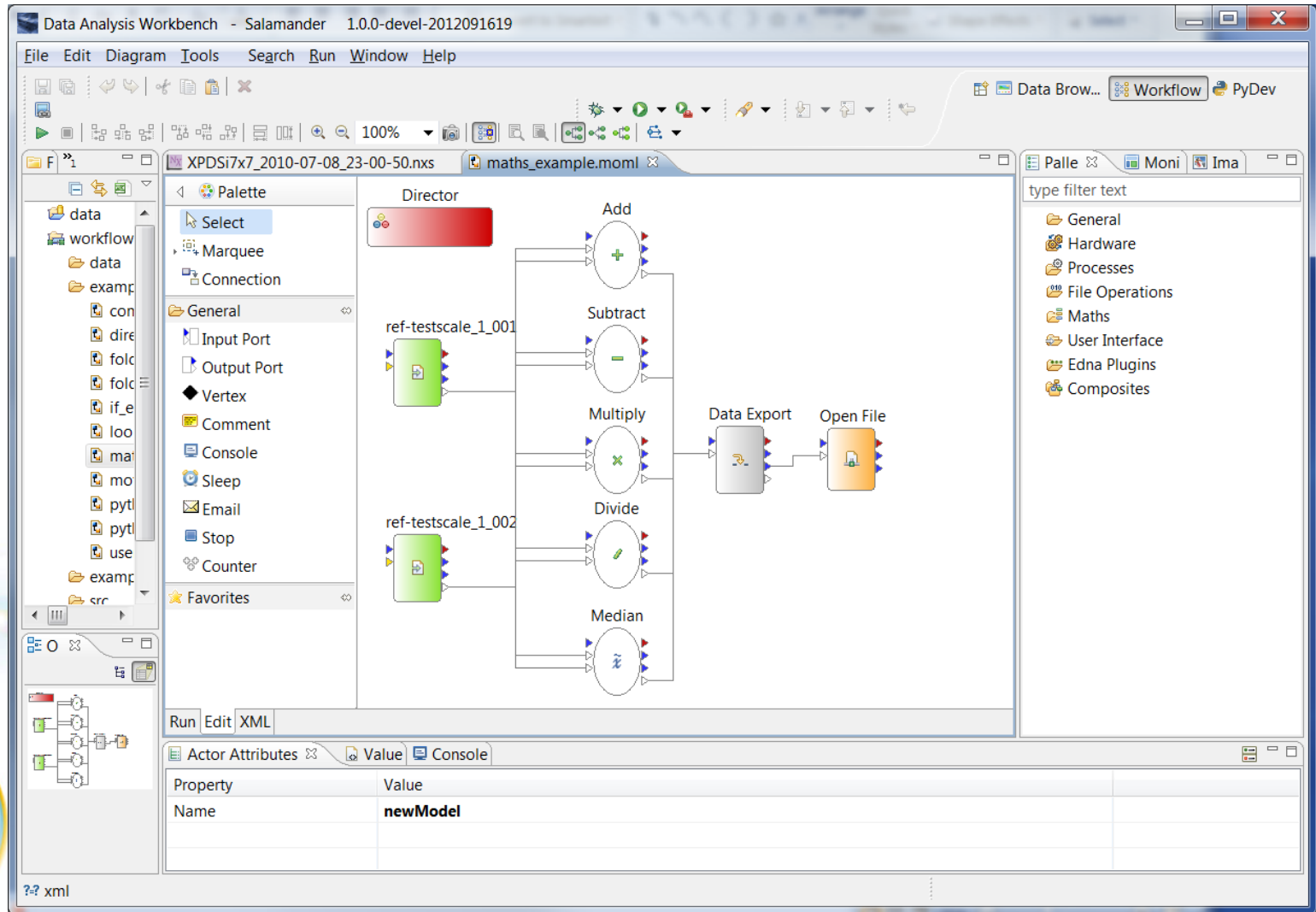
- Passerelle is a Ptolemy 2 based framework produced open source by Isencia Belgium.
  - Passerelle has **Swing**, **HTML5** and **SWT/RCP** versions today
1. Passerelle using Ptolemy 2 by extension / customization for projects in telecommunications
  2. Passerelle first used at the Soleil synchrotron - in its **Swing** incarnation
  3. A project completed with the ESRF to convert Passerelle UI to **SWT** in the RCP/Eclipse platform
  4. The **DAWN** project incorporates ESRF work and creates a new custom message to pass around actors.

# Common Message

- Messages passed between actors are complex
- Passerelle define a message with a header
- DAWN send multiple scalars and list values between actors in one message
- This enables graphs to be simplified at the expense of flexibility

## ***Demonstration*** – Simple Matrix Maths

Add, subtract – etc some images produced by an experiment...





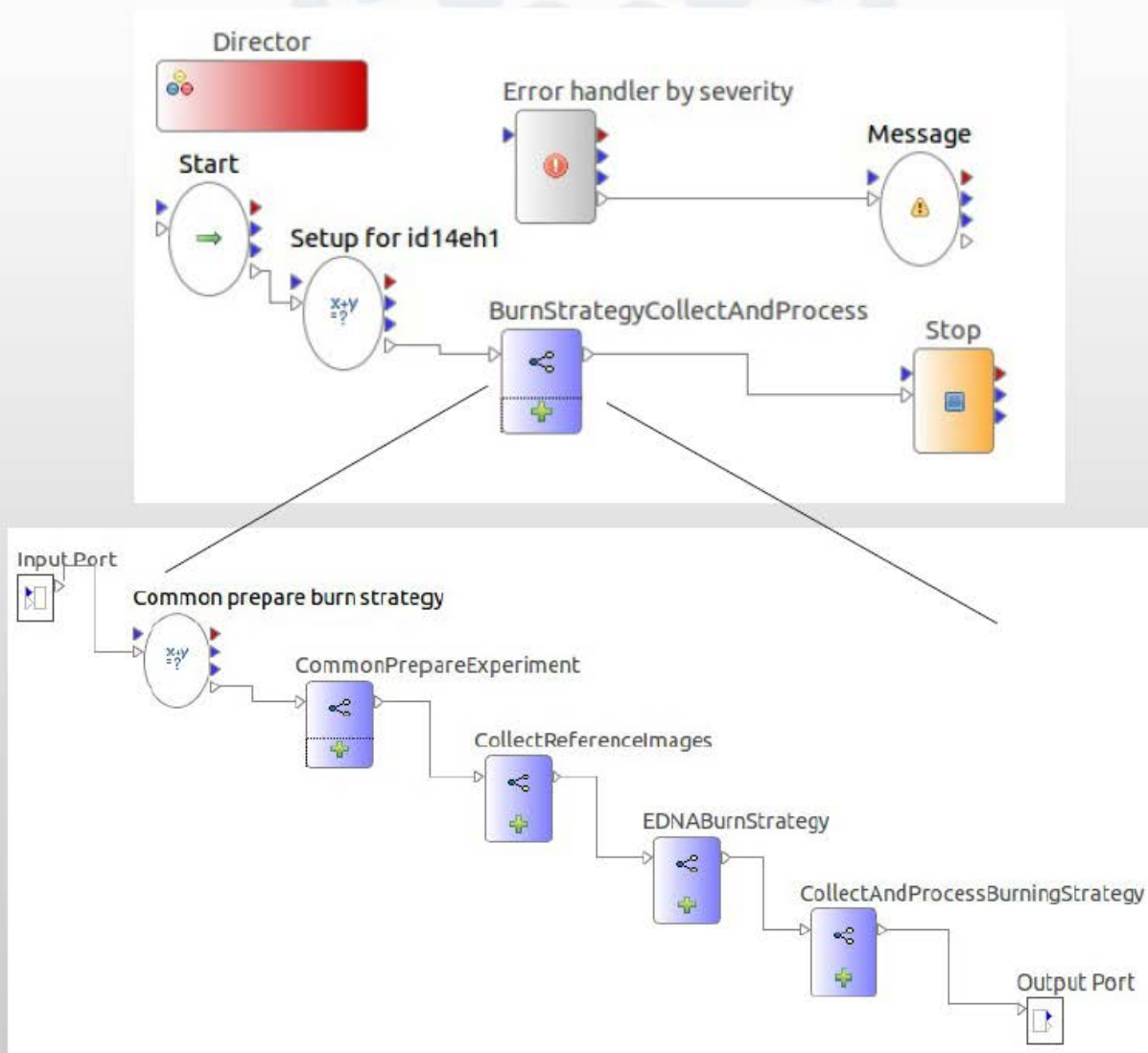
# Why use a workflow tool?

- They offer a visual, higher level programming language than traditional programming languages like C, Python, Fortran etc
- **The goal is not to replace these languages but to complement them**
- Workflows facilitates development of the high level analysis:
  - Visual programming → (beamline) scientists can participate in the design and make modifications
  - Easy to implement parallelism, error handling, LIMS connection etc.
  - Documentation by design!
- Workflow tools for data analysis :
  - Widely used in many scientific fields e.g. biology
  - New for synchrotron radiation facilities

# Workflows currently available at ESRF MX beamlines (in expert mode)

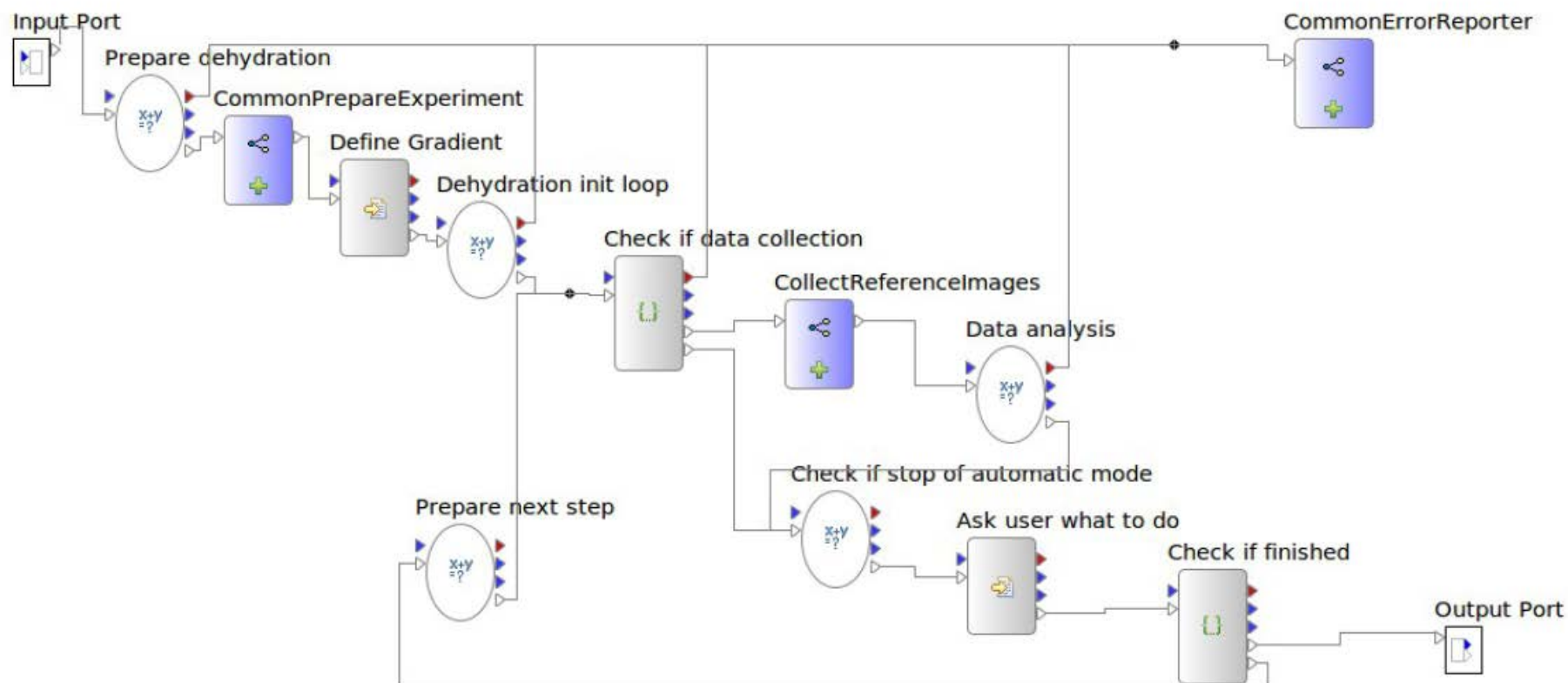
- Enhanced EDNA characterisation
- Accurate estimation of crystal radiation damage susceptibility
- Kappa goniostat re-orientation
- Automatic control of a Humidity Controller (HC)
- Various types of scans for diffraction intensity:
  - Line scan + move to strongest position
  - Mesh (2D) scan + move to strongest position
  - Automatic X-ray centring: a mesh scan, rotation of sample 90 degrees and a vertical line scan
  - On ID29 : using fast 4dscan

# "Burning strategy" workflow

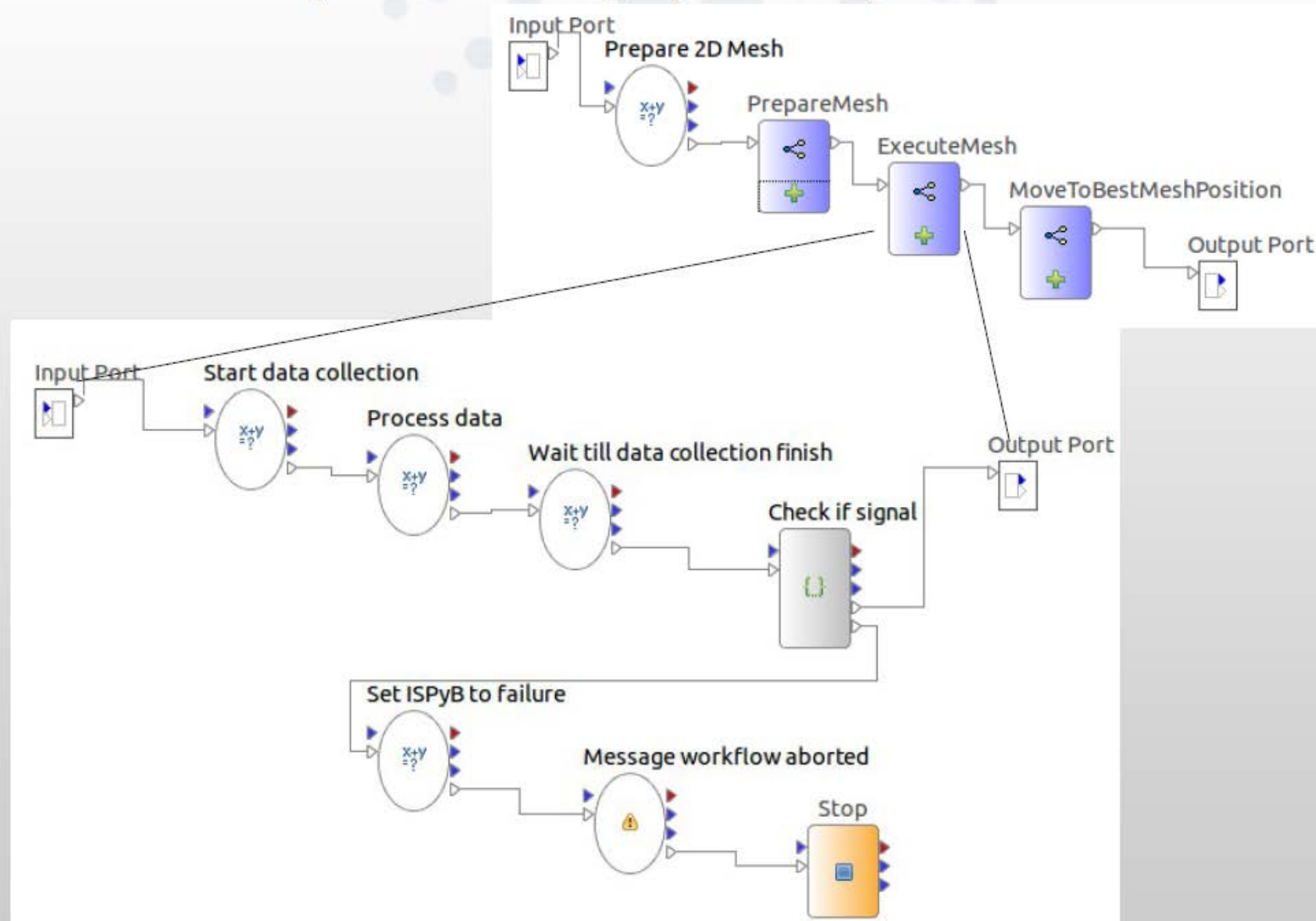




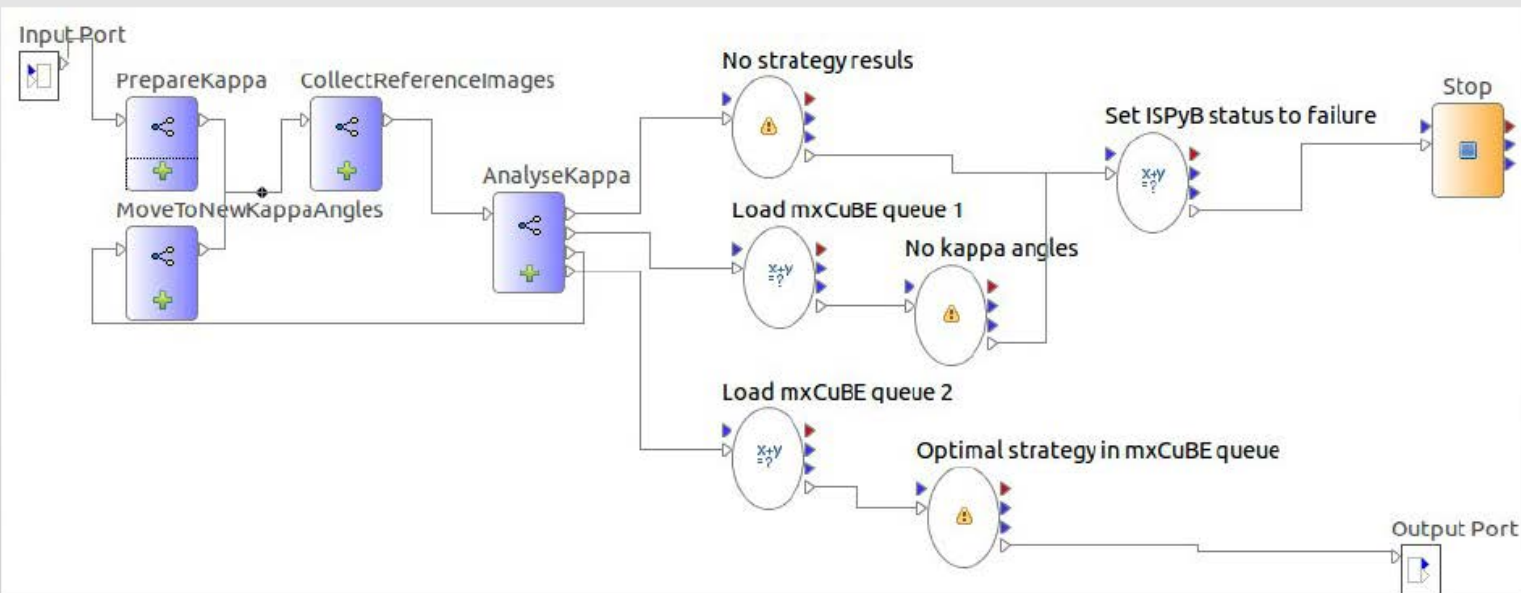
# Dehydration workflow



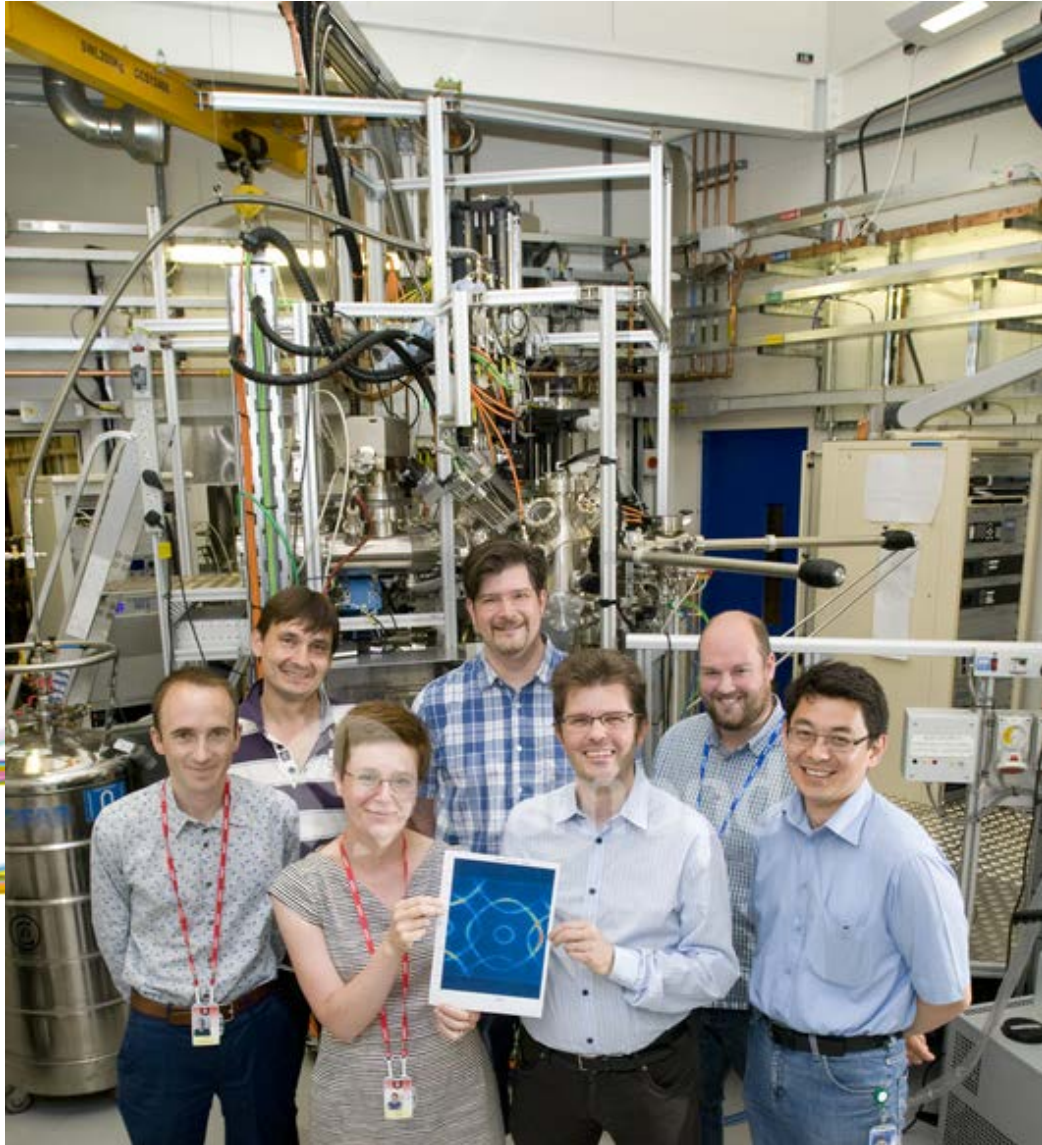
# X-ray centring (mesh) workflow



# Kappa goniostat re-orientation



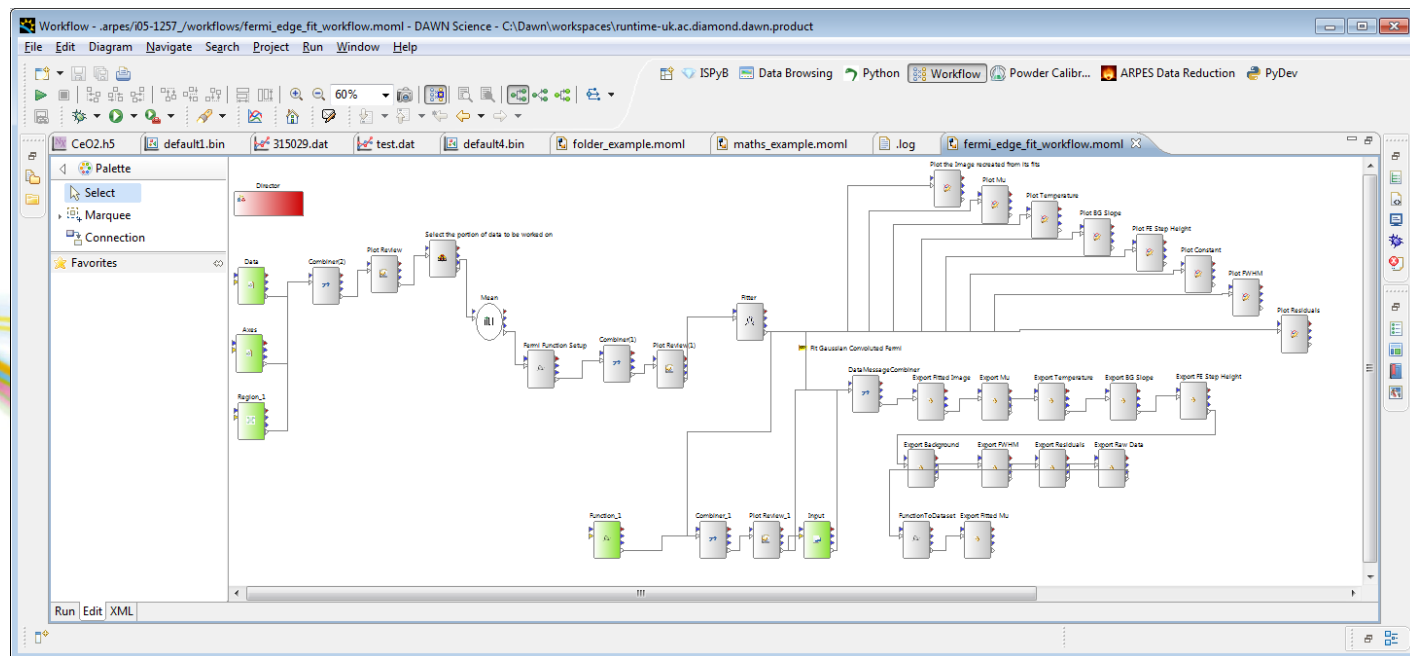




- Angle-Resolved PhotoEmission Spectroscopy
- Used to look at the Electron properties on surfaces.

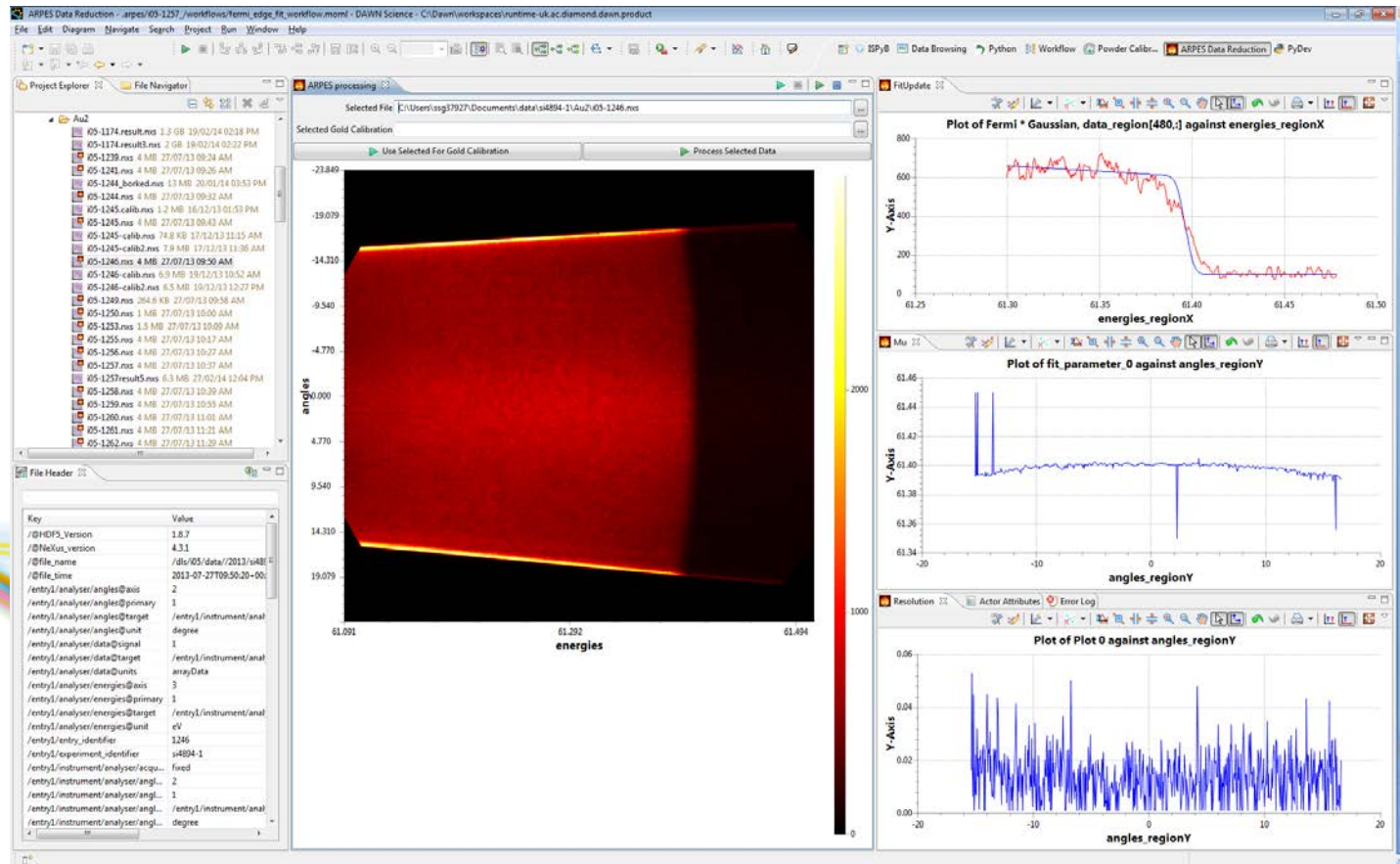
## I05 ARPES Why Workflows?

- Easier to work with beamline scientists, it seems less like black box data processing.
- Rich data message makes components reusable
- Individual plugins are more testable and stable.



# I05 ARPES User Interaction

- Users interact with this through a front end, and never see the workflow behind.





# Cluster Project

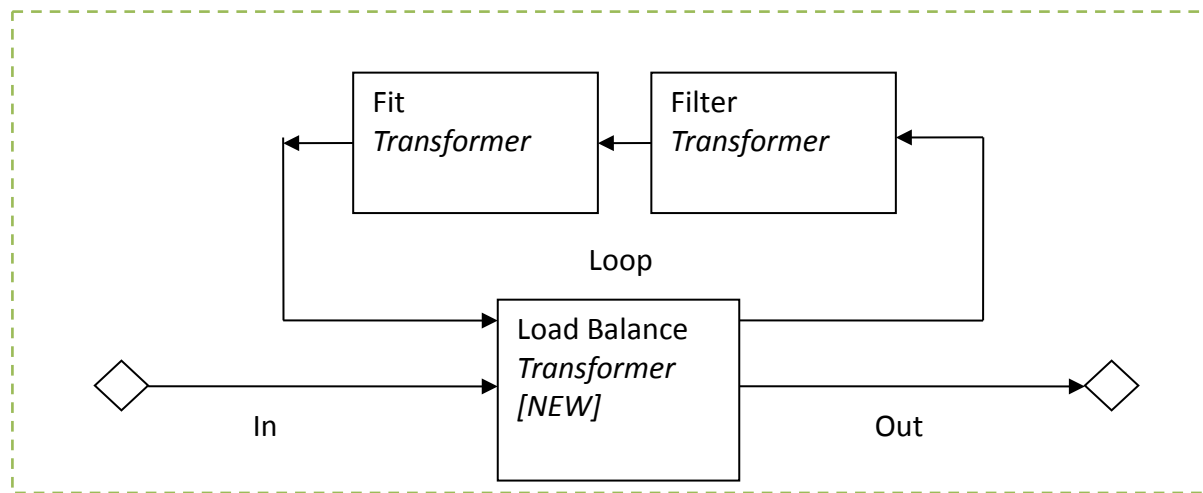
**Non-crystalline diffraction beamlines have an existing algorithm in Ptolemy 2 / Passerelle:**

- On i7 ~120 images take 4 minutes to process
- Image stack processed in parallel using load balancing (Fork/Join Java 7)

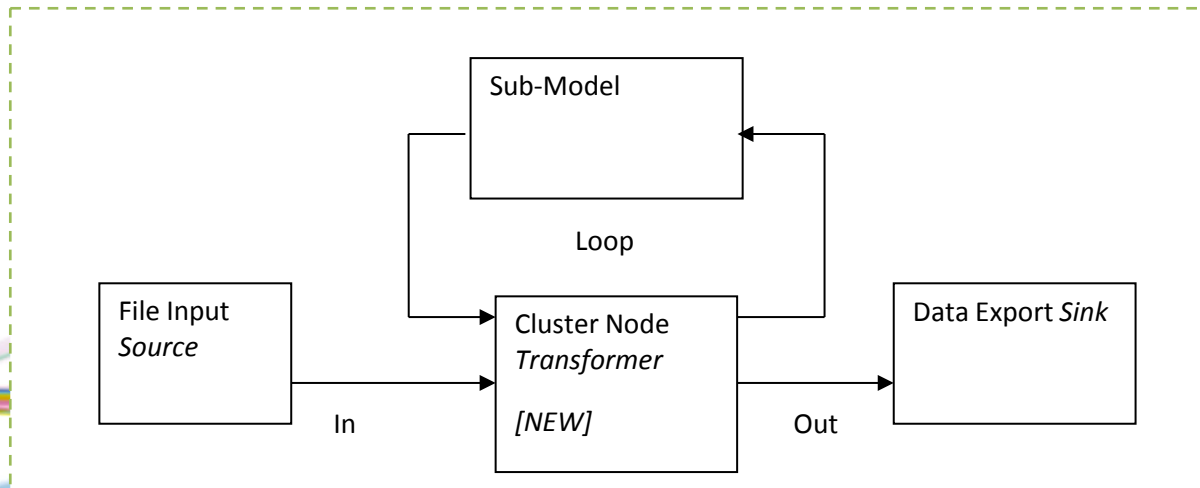
**We would like to run this *FAST***

- Split stack into chunks
- Process chunks on cluster nodes
- Cluster node actor to process chunks

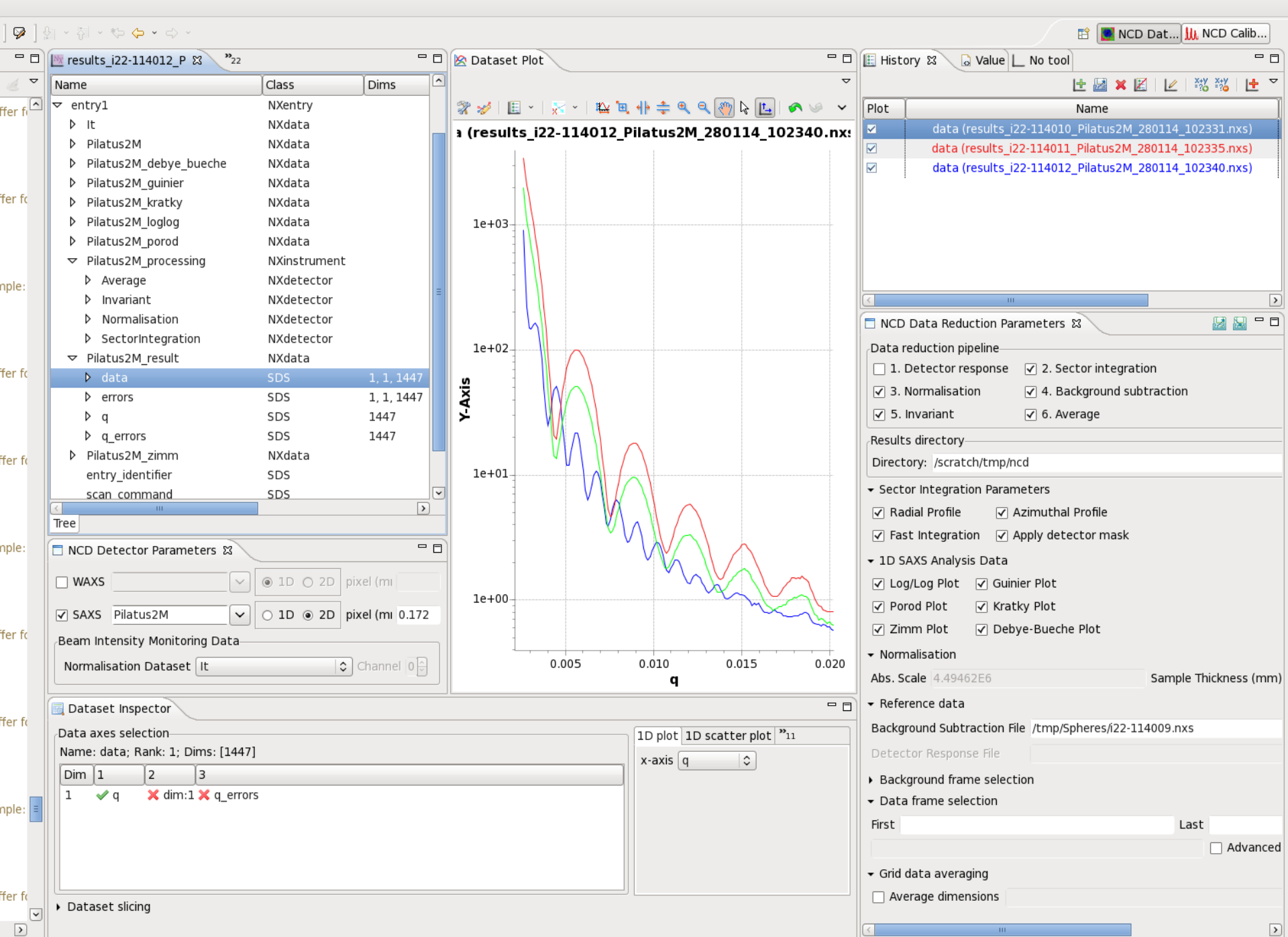
**Use of JMS and DRMAA planned**

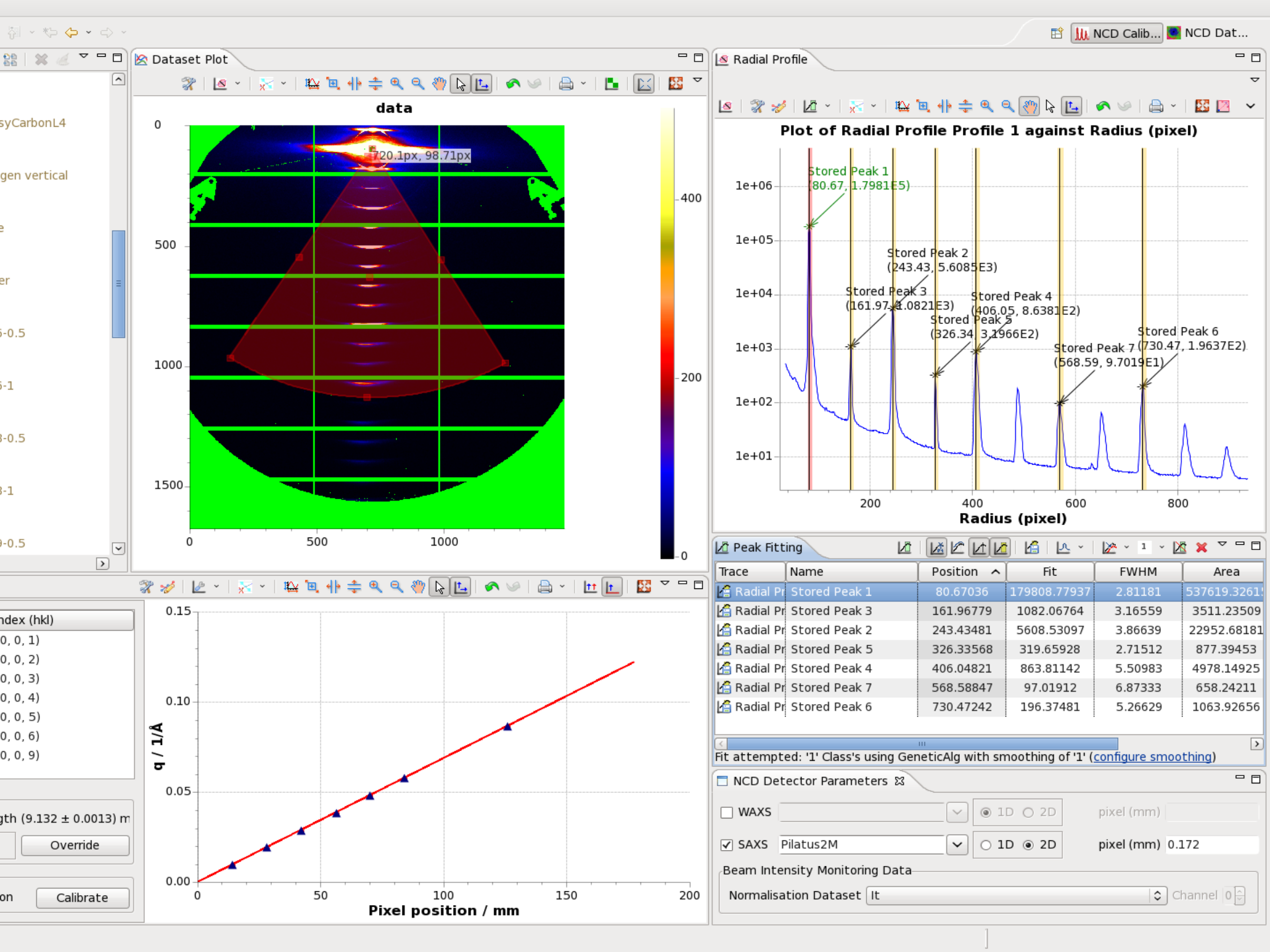


Load Balance



Cluster chunks







# Future of DAWN (wrt Ptolemy)

- New RCP workflow editor using Graphiti
  - New routing options
  - Improved graphical layer and tools
  - [eclipse.org/graphiti/](http://eclipse.org/graphiti/)
- Cluster connectivity
  - Load balancing actor
  - Cluster node actor based on DRMAA [drmaa.org](http://drmaa.org)
- Increased support for data regions and functions

# Ptolemy 2 (Questions about the) Future

*...Brainstorming*

- **Usage of the Fork/Join capability in Java 7?**
- **How to make best use of Lambda functions in Java 8?**
- **How is the Kepler RCP project going (is there one)?**
- **Can we collaborate in the future between Kepler and DAWN or Passerelle?**

# Conclusion

- **Thanks** to Ptolemy 2 and Passerelle for their API which has been useful for our workflows feature.
- **Thanks** to Eclipse for providing a great tool
  - RCP is fast and scalable too, using OSGI
  - SWT has ability to be configured for very large data
  - Ability to integrate native code in plugins if needed
  - Maybe we can support web application with RAP one day
- **Thanks** to the Java community for its APIs

Diamond Light Source Ltd. [www.diamond.ac.uk](http://www.diamond.ac.uk)  
ESRF [www.esrf.fr](http://www.esrf.fr)  
Data Analysis Workbench, [www.dawnsi.org](http://www.dawnsi.org)

