



Australian Government

**Ansto**

Nuclear-based science benefiting all Australians

# GumTree Decoded

Tony Lam, Norman Xiong, Paul Hathaway, Nick Hauser

**Australian Nuclear Science and Technology Organisation**

Sydney, Australia

# GumTree Decoded

1 single application

2 aspects: instrument control & data analysis

3 main languages: Java, Python, Adobe Flex

4 countries involved: Australia (ANSTO), South Africa (NESCA), Switzerland (PSI), France (ESRF)

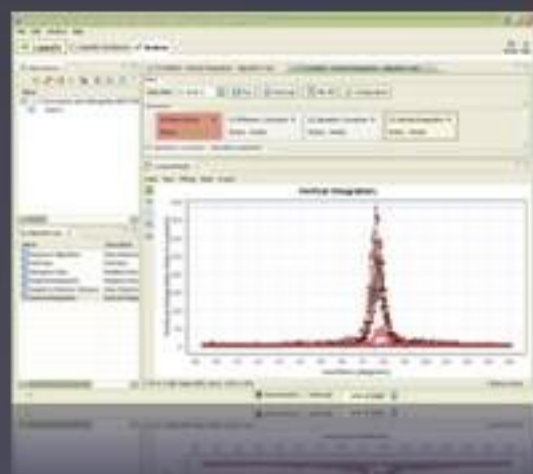
5 years of development

6 neutron scattering instruments supported

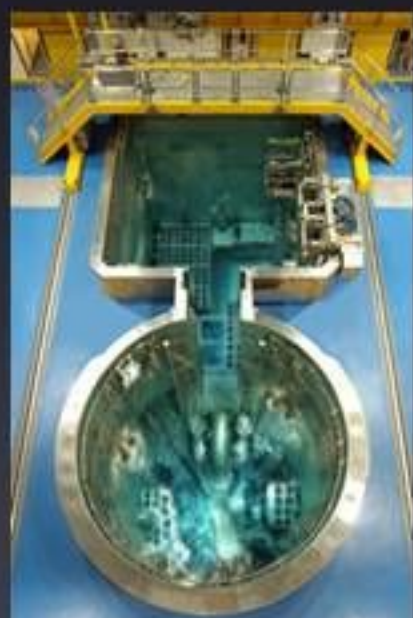
150+ plug-ins developed

200+ third party plug-ins used

2,000,000+ LOC



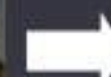
# Neutron Scattering



Neutron Source

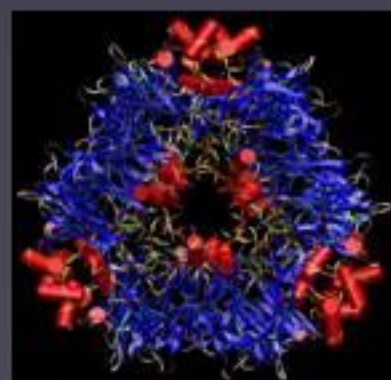


Instrument Optics



Sample

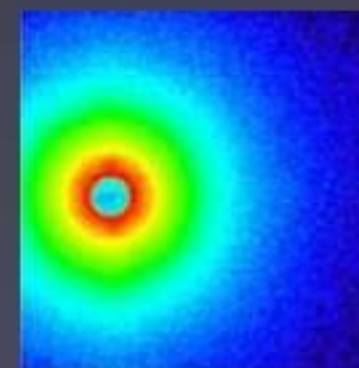
Neutrons



Science



*analysis*



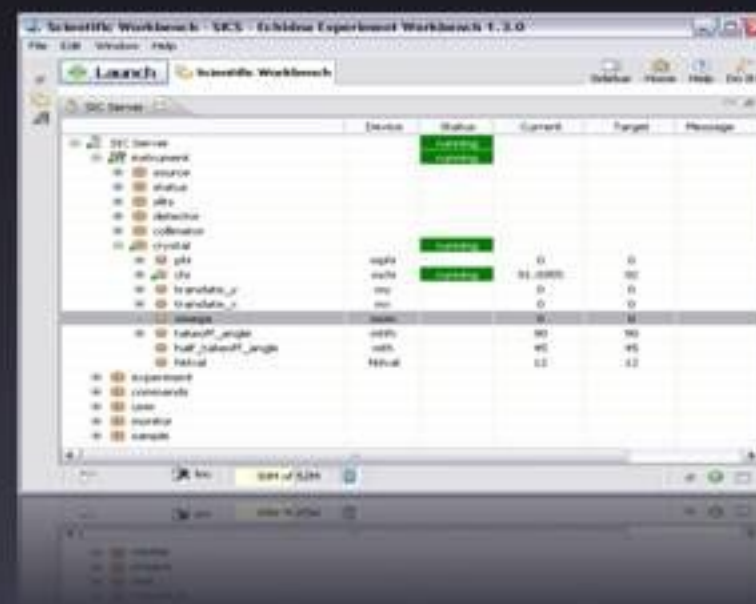
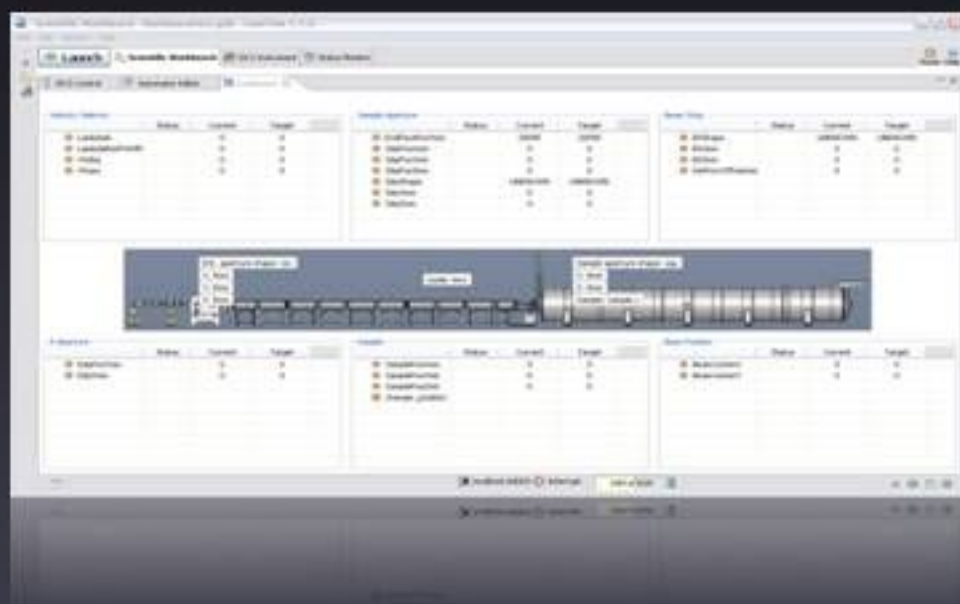
Result



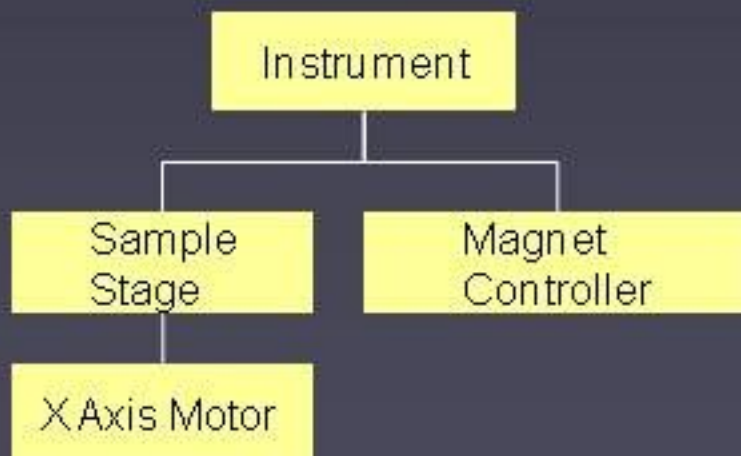
*acquisition*



# Instrument Control Client



## Instrument Model:



transported and serialised using Service Data Object (**SDO**)



TCP/IP

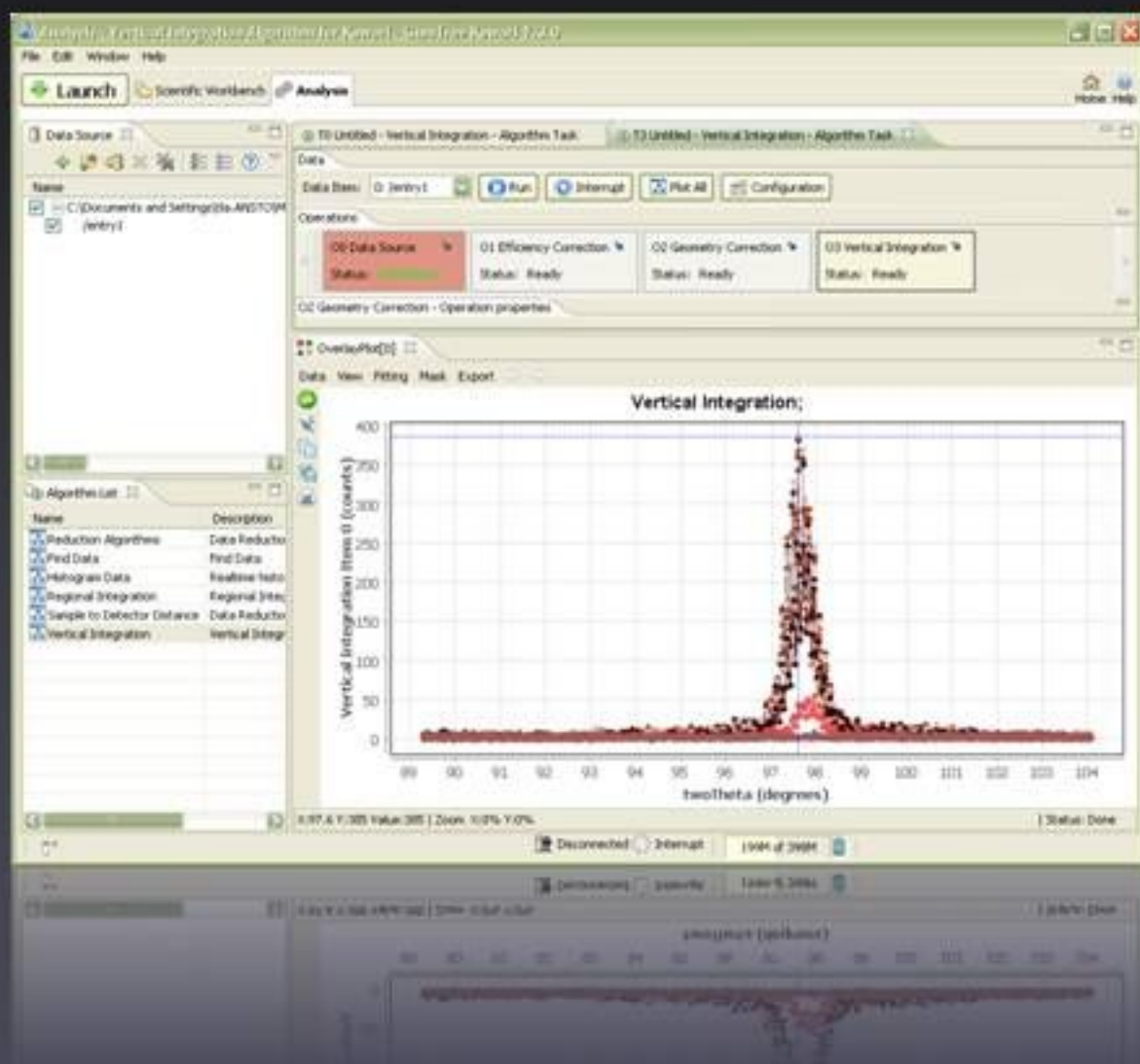


**SICS Instrument Control System**

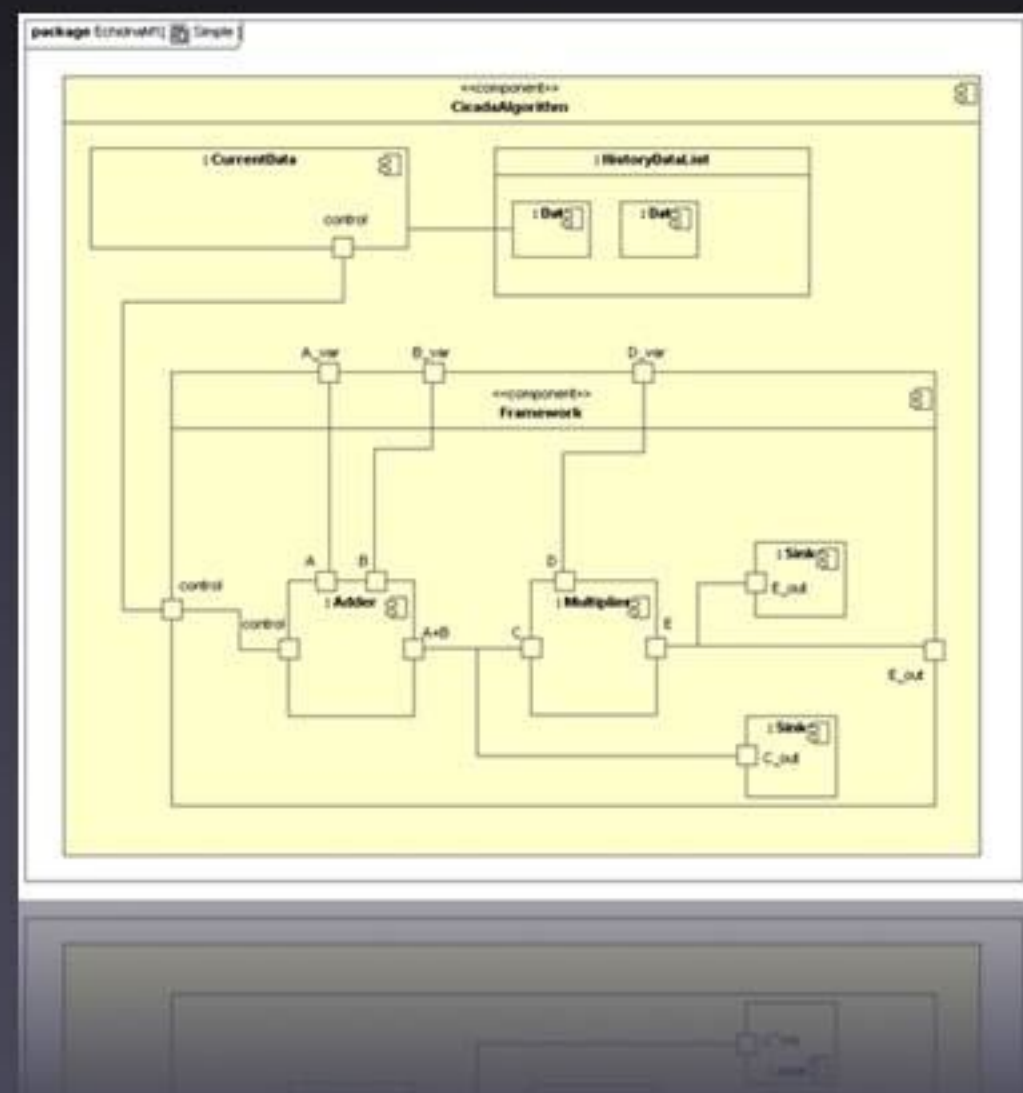
## JSON Protocol:

```
{  "transaction":101,  "object":"motor1",  "data":{    "position":10.4,    "unit":"mm"  }}
```

# Data Analysis Tool



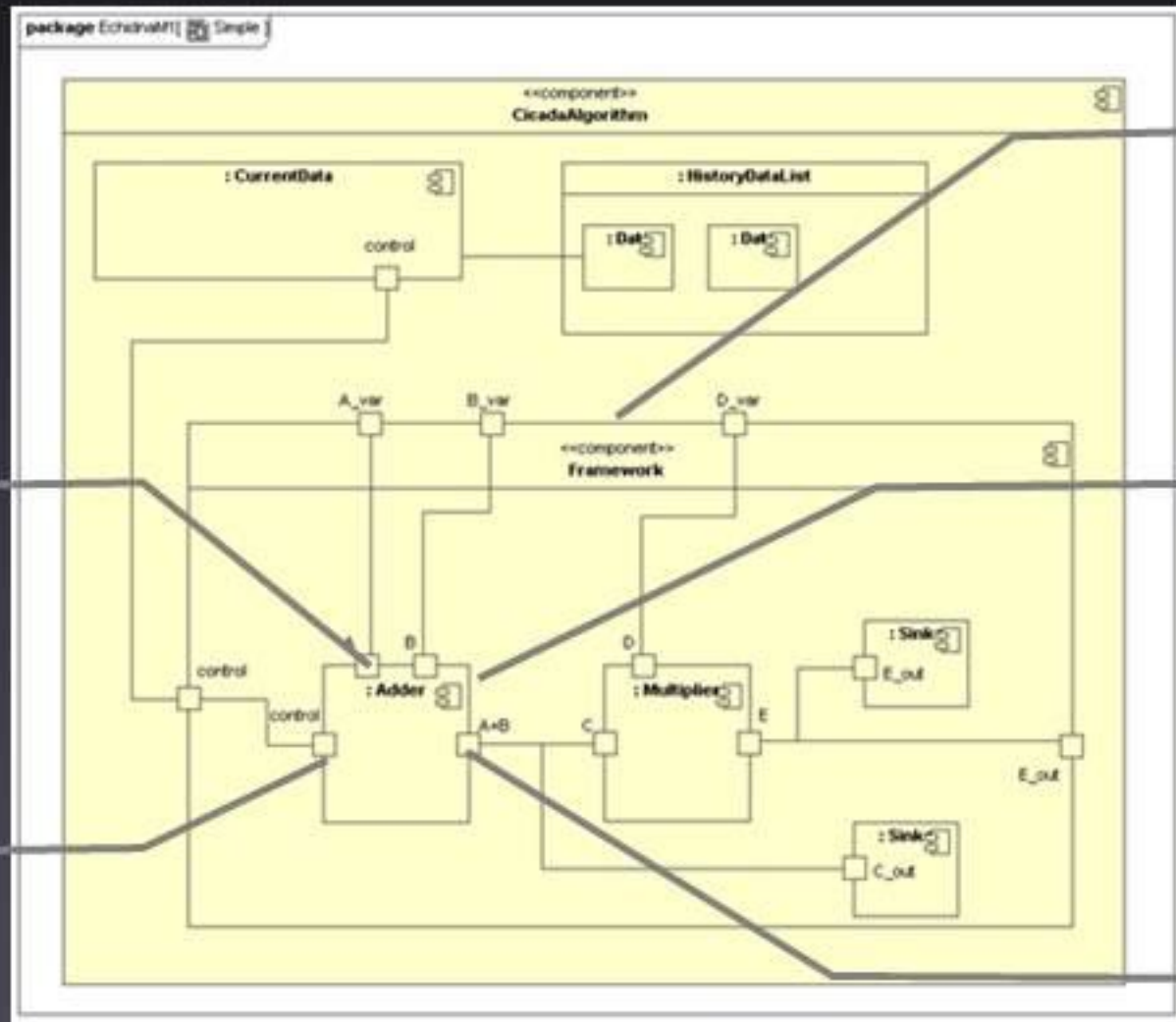
**Kakadu  
Data Analysis Suite**



**Cicada  
Data Processor Framework\***

\* **THP076**: Multi-platform Processor Framework for Data Analysis, Data Acquisition and Simulation

# Data Analysis Tool



Tuner Port

Input Port

Processors are composed and linked by XML

Pluggable Algorithm

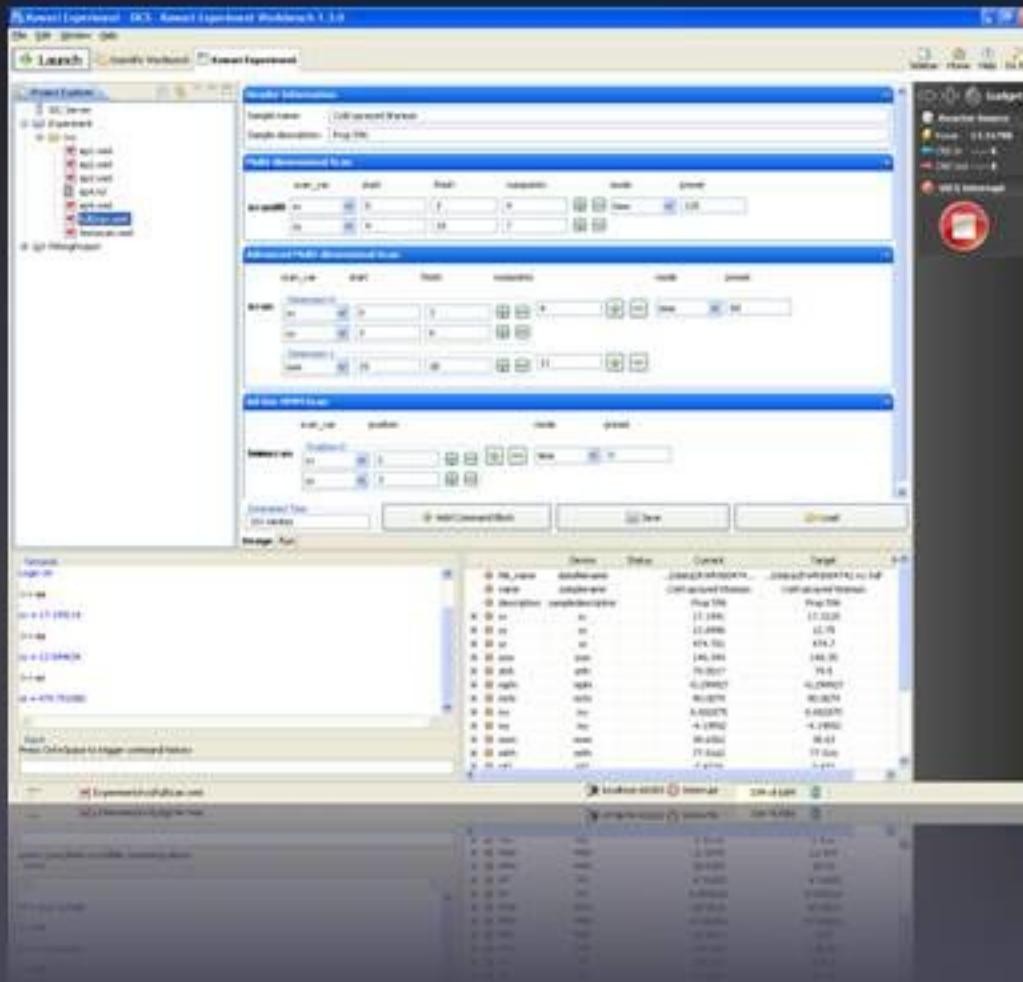
Output Port

**Cicada**

**Data Processor Framework\***

\* THP076: Multi-platform Processor Framework for Data Analysis, Data Acquisition and Simulation

# Automated Experiment Control



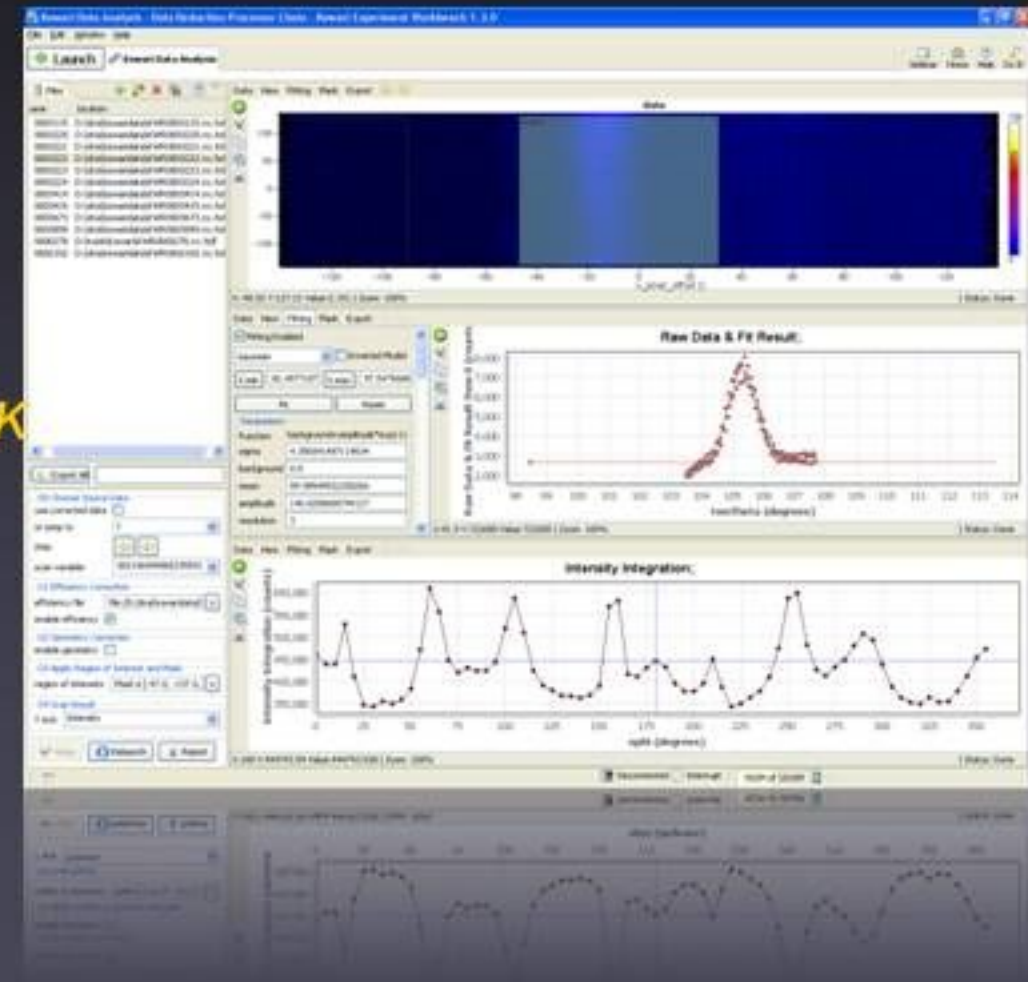
Visual Experiment Planning Window



1. CONTROL



SICS Instrument Control System



Live Data Reduction Window



2. DATA

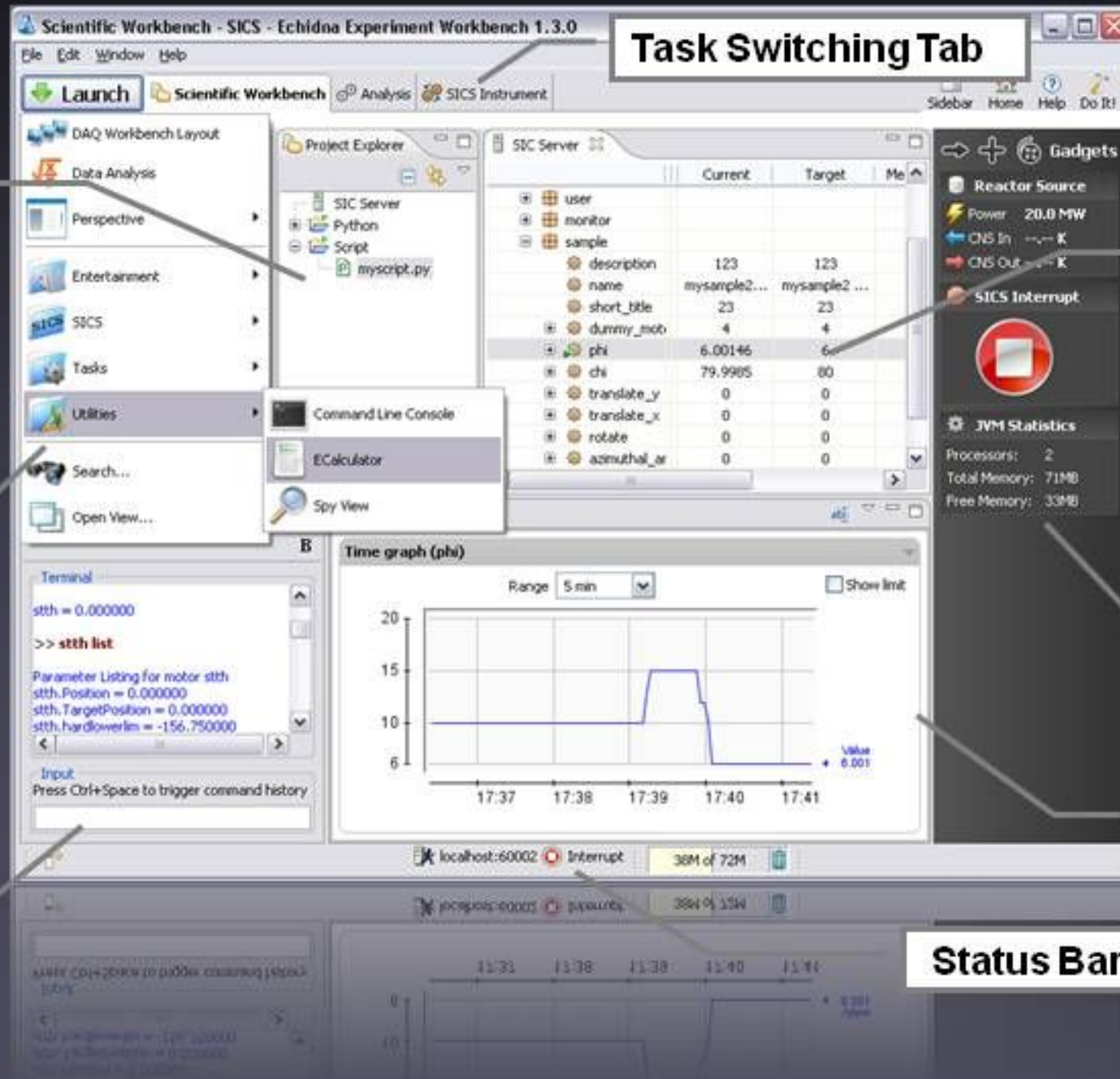
3. FEEDBACK





JFreeChart .netCDF  
Spring Restlet   
Eclipse OSGi xStream  
EMF JAS3 SLF4J HDF OpenGL  
Python db4o DOM4J Log4J  
Java

# Eclipse Rich Client Platform



# Eclipse Rich Client Platform

Workspace



JSR-223  
Scripting  
Engine

The screenshot shows the Scientific Workbench IDE interface. At the top, there's a menu bar with 'File', 'Edit', 'Window', and 'Help'. Below it is a toolbar with 'Launch', 'Scientific Workbench', 'Sidebar', 'Home', 'Help', and 'Do It!'. The main workspace is divided into several panes. On the left, a 'Project Explorer' shows a tree view of 'Scripts' containing files like 'gdm.py', 'info.py', 'integration.py', 'LoopTask.py', 'mayavi.py', 'Multiworkflow.py', 'overloading.py', 'plot.py', 'QuokkaResult.py', 'service.py', 'test.py', and 'traits.py'. The central pane is a code editor displaying Python code for 'QuokkaResult.py'. The code includes imports, class-like structures, and loops. A code completion popup is visible over the code, showing suggestions for 'DetectorHighVoltageController : PyJobobject' and 'SicsCore : PyJobobject'. At the bottom, a 'Terminal Console' shows a command prompt with 'ts\gumpy\quokka\ >> sys.version' and the output '2.5.4 (r254:679 [MSC v.1310 32 b...'. A 'Python History' pane on the right shows a list of commands and their execution times.

Editing  
Support

Terminal  
Console

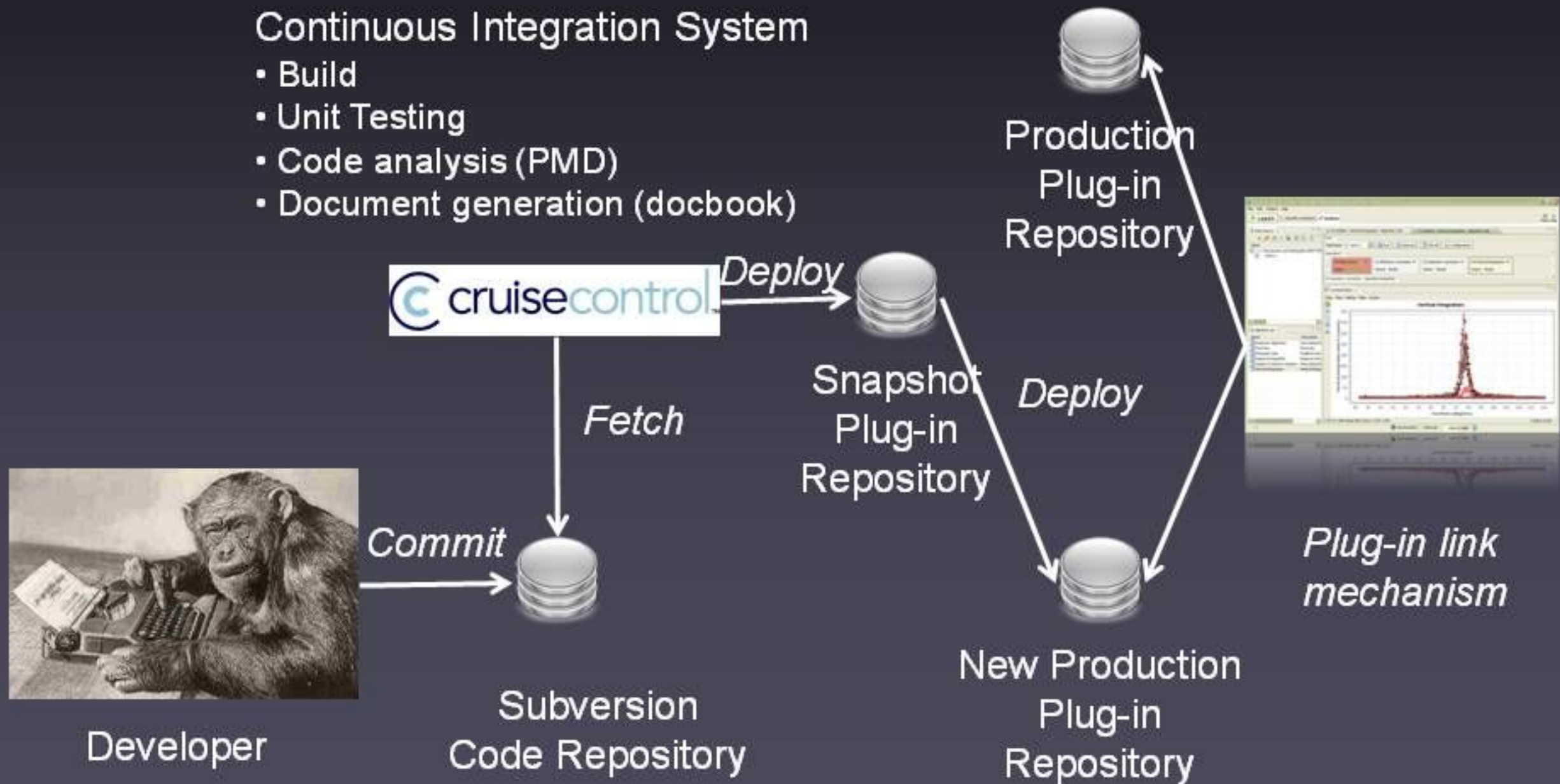
Code  
Completion



# Deployment Model

## Continuous Integration System

- Build
- Unit Testing
- Code analysis (PMD)
- Document generation (docbook)



<http://gumtree.codehaus.org>



Photo of the **QUOKKA** (Small-Angle Neutron Scattering) Instrument