Rich Client Platform for Synchrotron Science

- An open source not for profit project

- 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 12 years with Java.
- An Eclipse/RCP fan
- I will attempt to explain a bit of the science for your enjoyment (hopefully not schadenfreude).
- The rest of the talk will focus on software.
- The eclipse features shown in this talk, such as the pipelining tools can be reused outside Science.

Matthew Gerring
Synchrotron

AKA – cool word, but what does it mean?

**syn·chro·tron/**ˈsiNGkrəˌträn/**

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 various scientific experiments and sometimes other things.”
Where I used to sit

Where I sit now
The Queen and Duke of Edinburgh at the official opening of DLS
Inside the ring (not star-trek conduit...)

Scientists with some of the hardware used in their research

Video of Diamond...
Responsibilities

AKA – what they want developers to do...

• **Software for controlling experiments**
  – Motors, detectors, data collection scripts, configuration.
  – A high quality but flexible GUI.

• **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 during experiments**
  – Workflow and pipelines. Hard coded and/or user configurable options.
  – Real time visualization of analysed results.
Integration Tools
AKA – how we are getting it done

- EIDES – Eclipse IDE, Stupid - around 15 developers
- Eclipse RCP product built using **Buckminster** (previously PDE)
- Usage of **Jenkins** for continuous integration. Specific releases still produced and tested, appearing as separate items in Jenkins.
- Unit tests using **JUnit** and **JUnit plugin** tests.
- **Git** repositories – we use and like **eGit** and git command line.
- Almost no code walkthroughs and pair programming. Agile practices being used where otherwise possible.
- We like **Cheat Sheets** and use them for tutorial and testing guides.
‘Shoulders of Giants’
AKA – we didn’t think of everything

• 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 6x, and many more...
A quick look at lazy

What it does

• Viewer reads contents as incrementally (normal, however...)

• Items off screen *not* loaded
  – Not icon read, no label
  – Resources for the item unassigned

• Allows large viewer contents to exist but not take up resources until used

Things to do in the source code

```java
tree = new TreeViewer(parent, SWT.MULTI
  | SWT.H_SCROLL | SWT.V_SCROLL
  | SWT.FULL_SELECTION | SWT.BORDER
  | SWT.VIRTUAL);

tree.setUseHashlookup(true);

public class FileContentProvider implements ILazyTreeContentProvider {
...
}

tree.setContentProvider(new FileContentProvider());

tree.getTree().setItemCount(...);
```
Demonstration - Lazy Trees

Navigate the file system and open a file containing a large tree
Pipelines / workflows

- **Eclipse feature, EPL /Apache dependencies**
- Easy to create **multi-threaded** algorithms using branches
- **Balance** over multi-CPU systems using JDK threading
- Runs in **separate process** using Eclipse debug plugins
- Easy to integrate new actors (**extension points**)
Demonstration — Simple Matrix Maths

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

• Concept of source actors, for example:
  – Monitor a file or directory
  – Monitor a web service
  – Iterate a large folder of data files

• Ability to deploy as a web or rcp service
  – Run as external process
  – Communicate to and from controlling process during model run

• Add actors using extension points
Demonstration — Efficient Pipelines

Pack a large number of files into a high density file (HDF5)
Fridge Control Example

- Controlling a fridge using Ptolemy II
- Adding and running your own actors
- The actors sleep instead of doing things

*Taken from examples on our github repo ‘DawnScience’*
Demonstration — Acme fridge control

Example using actors which control a fridge, for instance
Slicing data

• Cutting through N-dimensional data
  – With an XY plot
  – As an image
  – As a 3D iso-surface

• Important to run everything concurrently
  – Use of Jobs
  – Use of ordinary threads
  – Use of blocking queues of various sizes
  – Third party native APIs using Bundle-NativeCode
Demonstration — Slicing and dicing

Example opening a tomography file and slicing it
Conclusion

• **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

• We will hopefully be an eclipse project one day...

• **Thanks** for listening and please approach me at the conference if you want to talk about things

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