Argonne NATIONAL LABORATORY

... for a brighter future

U.S. Department of Energy

THE UNIVERSITY OF CHICAGO

Office of Science U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by The University of Chicago

Eclipse as an Application Workbench

Kenneth Evans, Jr.

Presented at the Software Collaboration Meeting

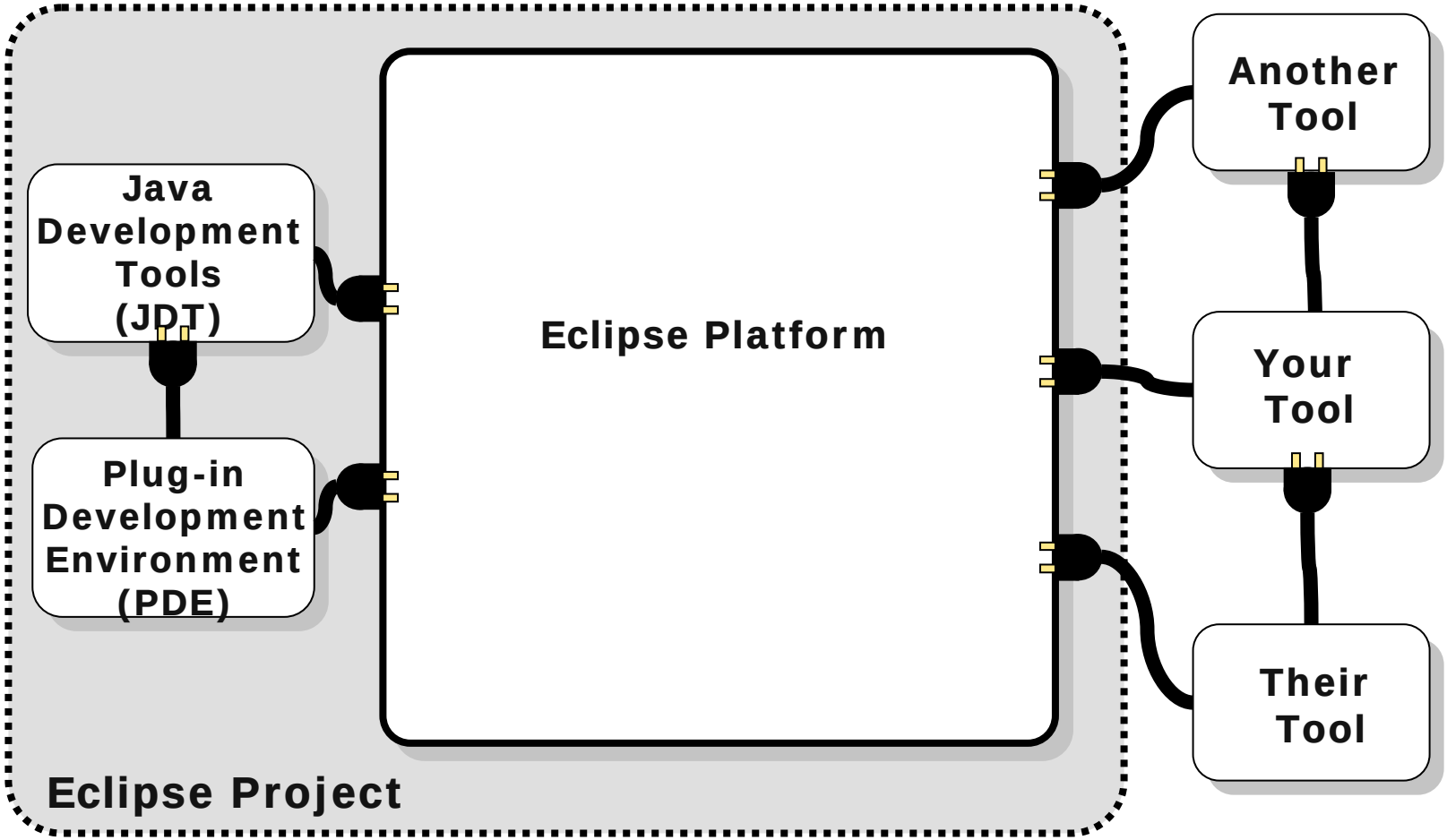
January 24, 2007

Oak Ridge National Laboratory, Oak Ridge, TN

Eclipse

- Eclipse is an Open Source community
- It was started in 2001 by IBM
  - IBM donated a lot of research
  - Controlled the early development, but later relinquished control
- Out of the box it looks like a Java IDE
- It is really a Plug-in manager
  - That happens to come with Java Development plug-ins.
  - You can take these out and put your own (and/or others) in

Eclipse is Very Extensible and Very Flexible



Modified From: Tony Lam, ICALEPCS Presentation, October 2005

Eclipse Foundation Membership

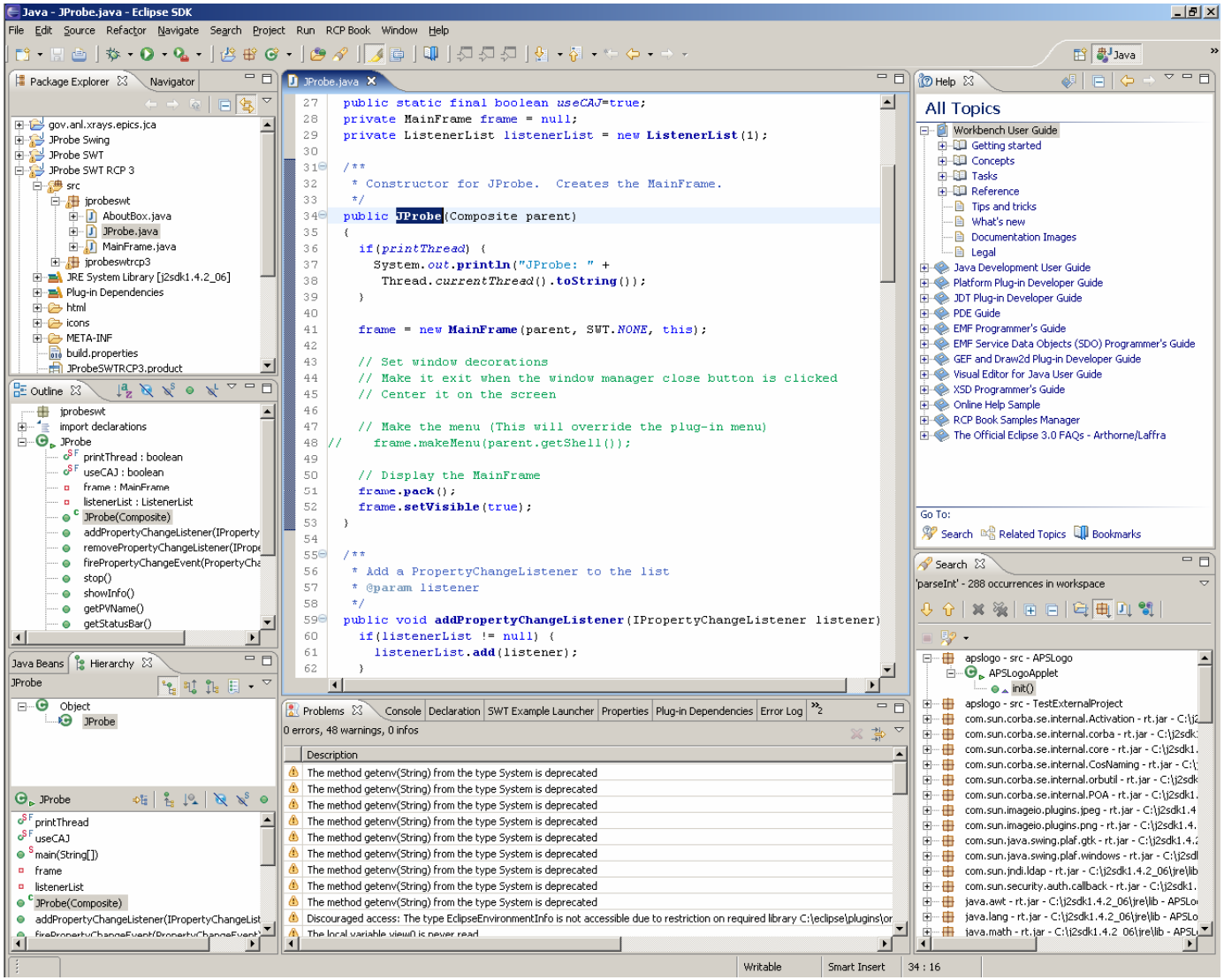
- Strategic Developers (13 as of Jan 2006)
  - At least 8 developers assigned full time to developing Eclipse
  - Contribution up to $250K
- Strategic Consumers (4)
  - Contribution up to $500K
  - Can reduce the dues by contributing 1-2 developers
- Three other tiers

- Bottom line
  - $$$ and Developers (currently > 150 full time)

Eclipse Consortium Strategic Members

* Strategic Consumer

Eclipse as a Java IDE

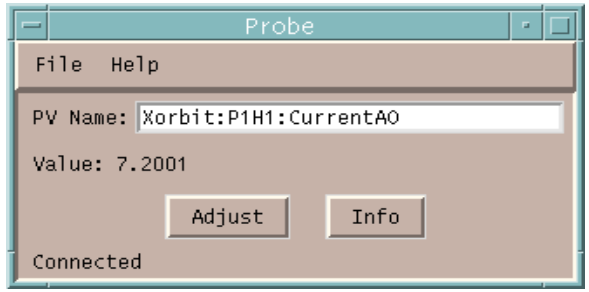


Rich Client Platform (RCP)

- "Rich Client" is a term from the early 1990's that distinguished applications built with Visual Basic and the like from "Console" or "Simple" applications
- Eclipse is particularly suited to Rich Client applications
- The possibility of using the Eclipse platform for applications was there from the beginning, but foreshadowed by its use as an IDE
  - In the early days it required hacking to make Rich Clients
- RCP is now (as of Eclipse 3.1) supported by the interface and encouraged
- You essentially use Eclipse as a framework for your application
  - You inherit all of its built-in features
  - As well as those from other community plug-ins
- You include only the plug-ins you need
- Is a very extensible development platform
  - You can use plug-ins developed by others as needed
  - Others can use yours and extend them

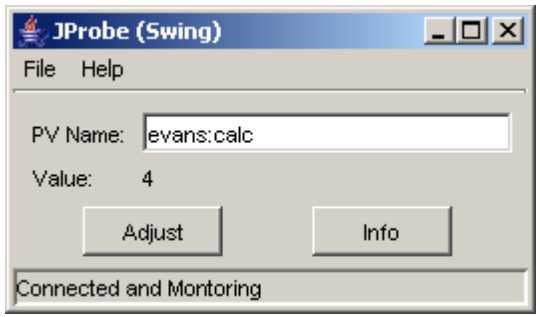
Eclipse as a Rich Client Platform

- Looks like an application, not an IDE
- Inherits a lot of functionality
  - Persistence (Properties and Preferences)
  - Help
  - Featured About dialog (like Eclipse's)
  - Splash screen
  - Dockable windows, and much more …

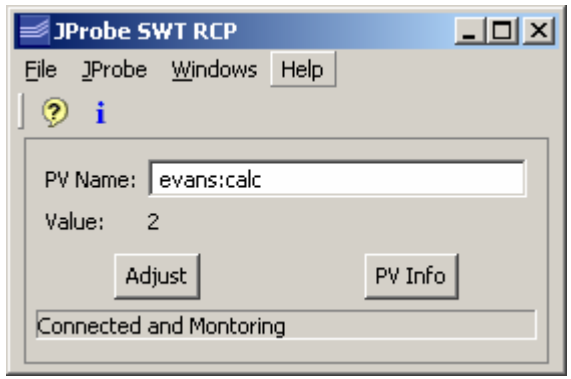


Pyre Application

probe.py --frame.pvName=Xorbit:P1H1:CurrentAO

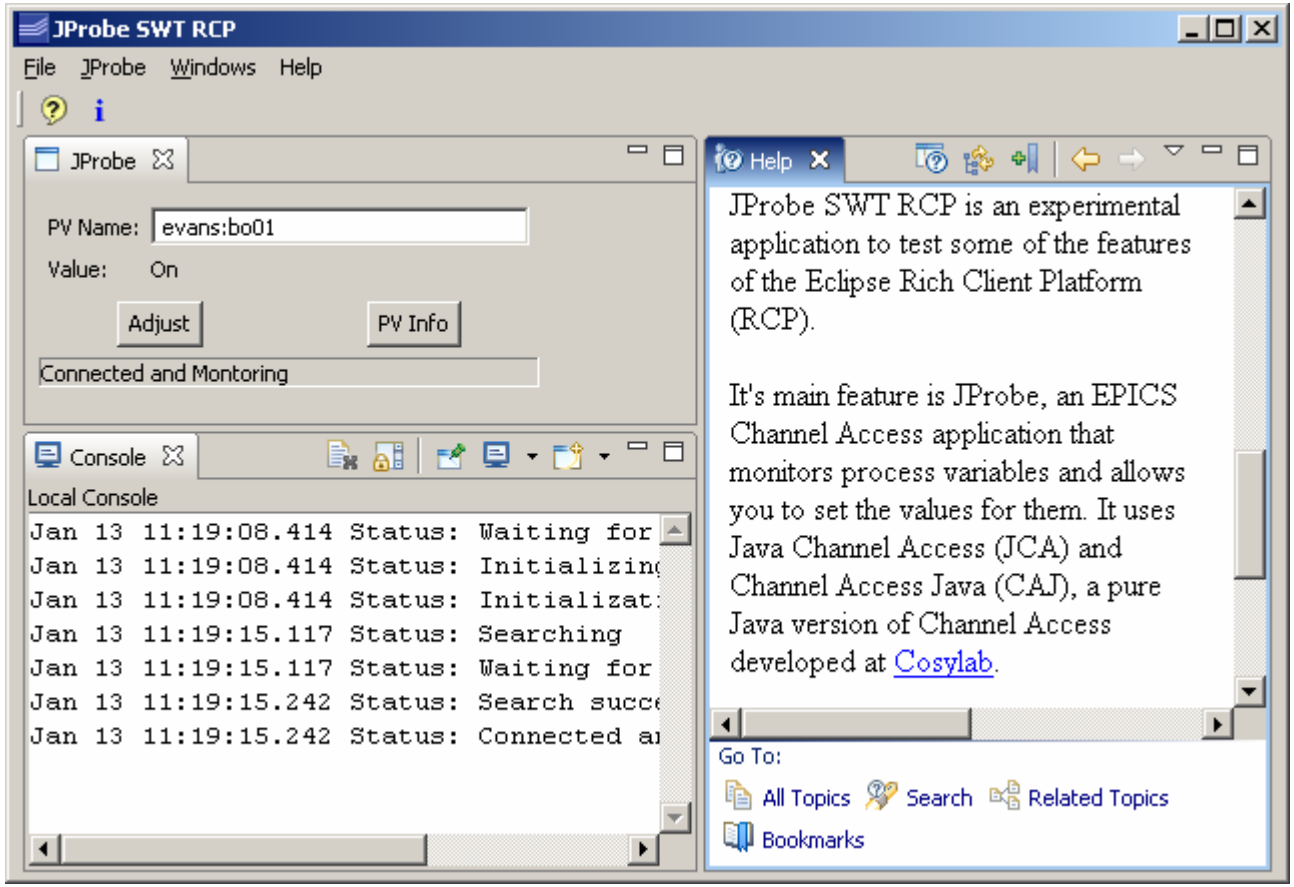


Java Application

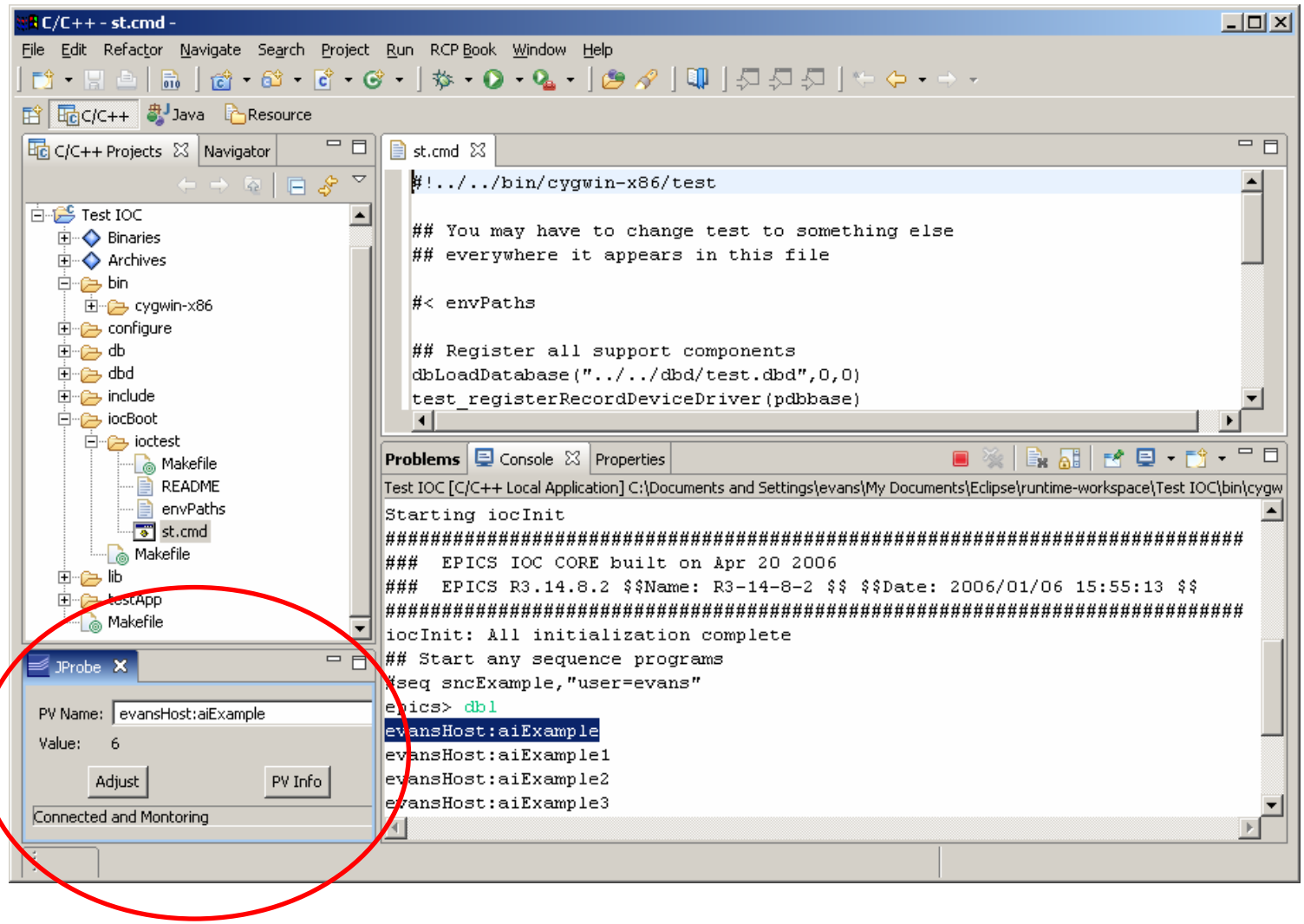


RCP Application

Probe on Steroids
Leveraging the Eclipse Framework



An RCP Application is Also a Plug-In



Bottom Line

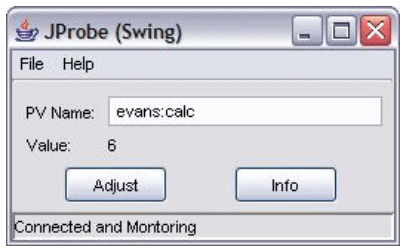
- Is a very powerful and extensible IDE and Framework
- Is also an IE - A way to organize your work
- Is Open Source
- Has a community
- Is supported by most of the industry
- Has a large number of developers (>150)
- Has significant financial backing
- Are many 3rd-party Plug-ins, both free and commercial
- Are more than 60 open-source projects
  - From Web Tools to Code Profilers
- Is continuing to expand and improve rapidly
- Is free
- Downsides
  - Is a continually changing, moving target

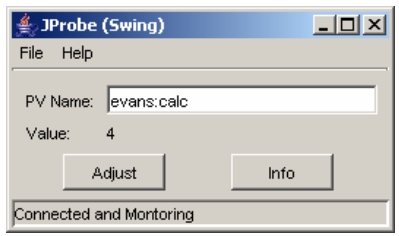
AWT vs. SWT - You Have to Decide

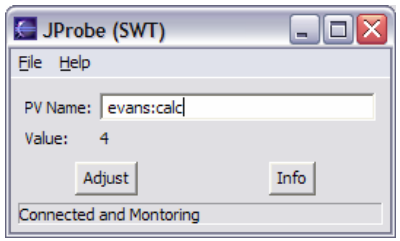
- AWT / Swing (Abstract Windowing Toolkit)
  - Write once, run anywhere
  - Formerly ugly, with bad performance
  - Now look and work well
  - Use garbage collection
  - Come with the JDK and JRE
- SWT / JFace (Standard Window Toolkit)
  - The important fact is that Eclipse uses SWT, not AWT
  - Supposed to look better, run faster
  - A thin wrapper around native widgets
  - SWT components must be disposed (vs. garbage collected)
    - *Owing to need to free native resources*
  - Need JNI libraries for each platform
  - Distribution is through the Eclipse Foundation, not Sun

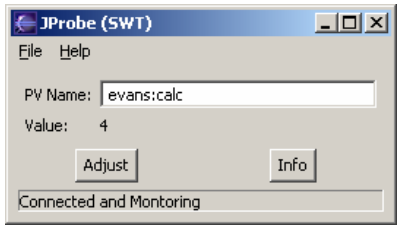
AWT vs. SWT - More Considerations

- It is not easy to convert between them
- The SWT look is not obviously better
- The performance difference may not be there either, today
- Eclipse uses SWT
  - They are supposed to mix and match, but ???
- Sun is unlikely to include SWT support in the JDK and JRE soon



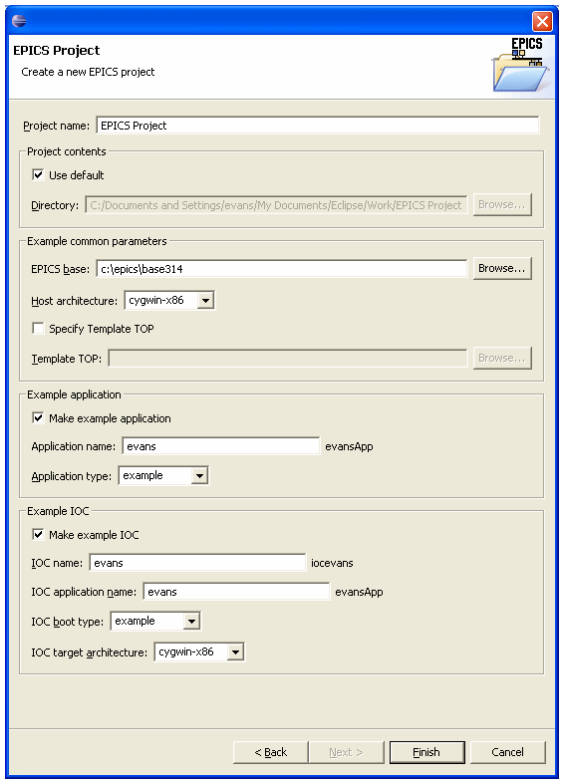


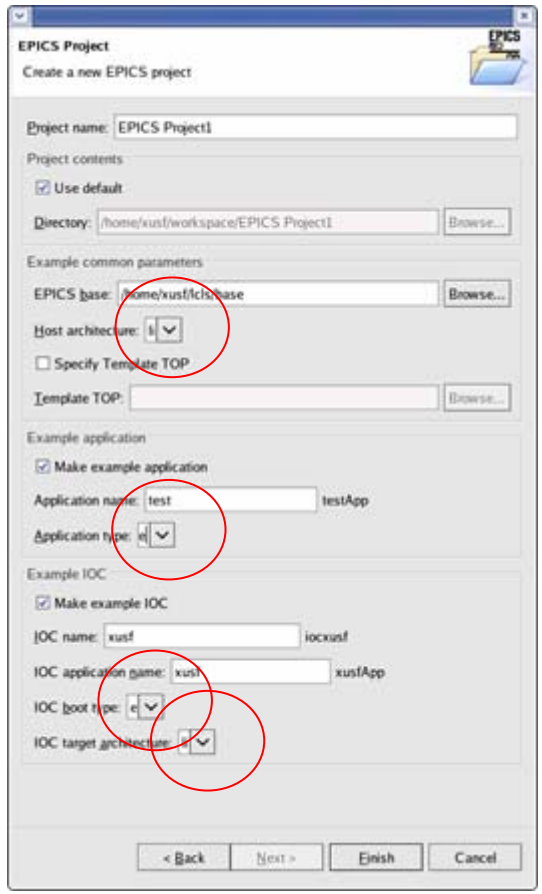




SWT Platform Dependence

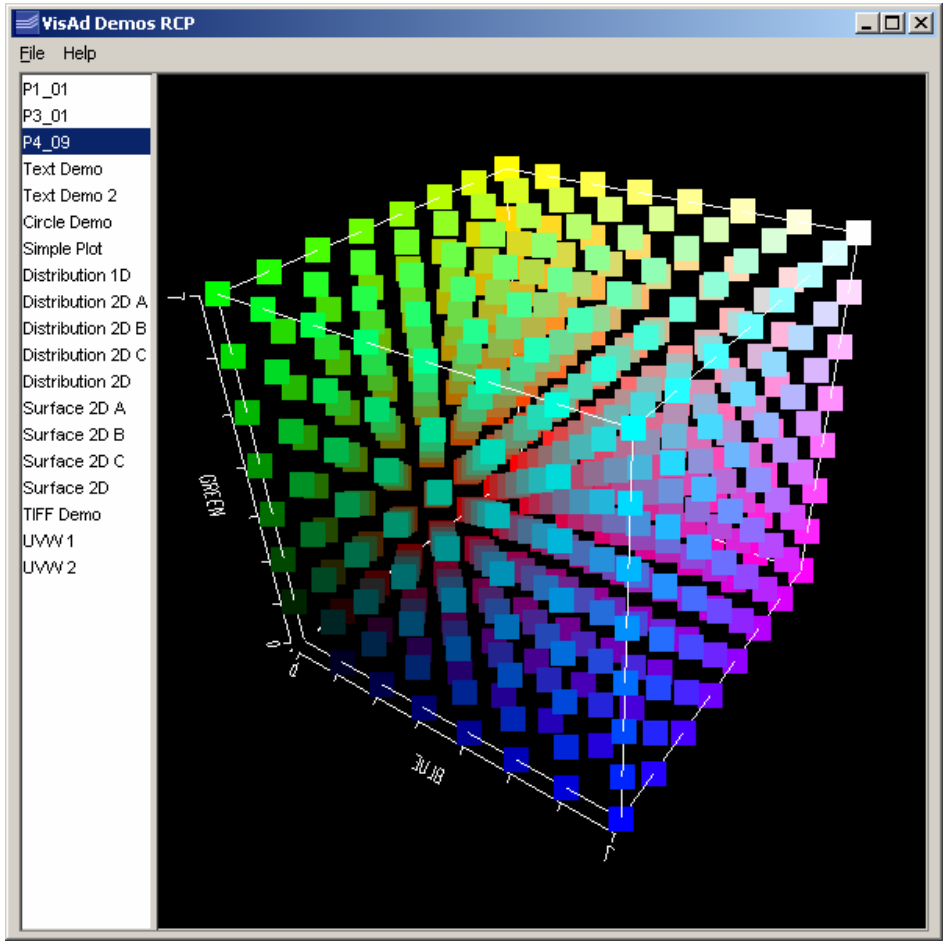
- Example: Working Windows dialog doesn't work right on Linux





Combining Swing and SWT - SWT_AWT Bridge

- ContentPane of JFrame is embedded in an SWT Composite
- Menu Initialization is separate from other UI initialization
  - Standalone Swing version uses Swing menus
  - RCP versions uses RCP workbench menus
  - Both can call same instance methods (or not)
- This application also uses JAI and J3D
  - Both are Java extensions
  - Don't play well with Eclipse

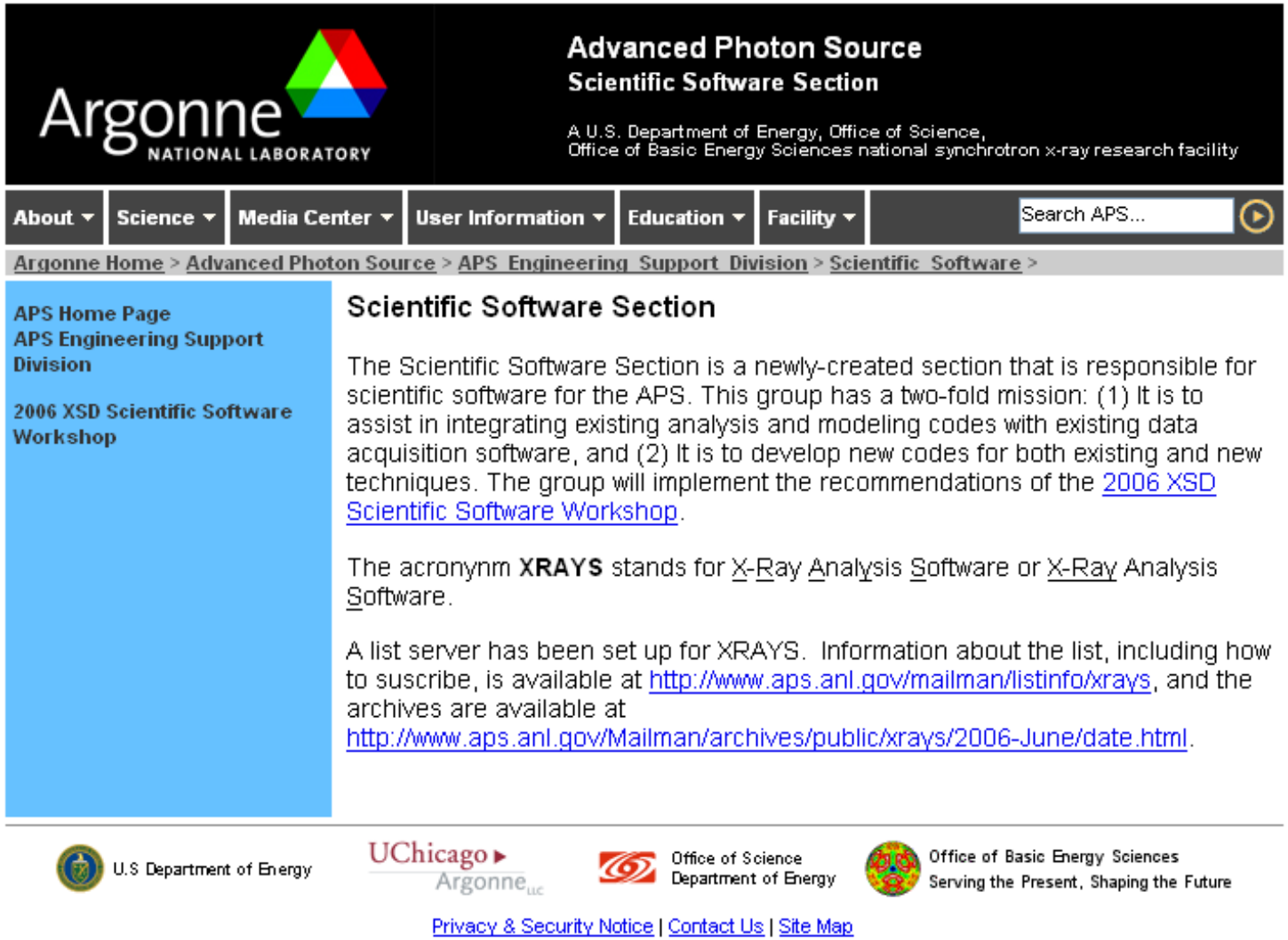


X-Ray Software Development at the APS

- Best described as “Uncoordinated”
- Wide variety of languages
  - FORTRAN, C, C++, Perl, Tcl/Tk, Python, Java, …
- Visualization relies on (different) commercial products
  - IDL, IGOR, Matlab, …
- Each beamline tends to do its own thing
- Modeling and Analysis is not well integrated with Data Acquisition
- Lack of real-time data reduction
- Little high-performance computing
- Little remote access
- No common data format

- The APS has been running for ten years
  - People are getting their work done
  - They are fixed in their ways

Scientific Software Web Page



Argonne NATIONAL LABORATORY

Advanced Photon Source
Scientific Software Section

A U.S. Department of Energy, Office of Science,
Office of Basic Energy Sciences national synchrotron x-ray research facility

About | Science | Media Center | User Information | Education | Facility | Search APS...

Argonne Home > Advanced Photon Source > APS Engineering Support Division > Scientific Software >

APS Home Page
APS Engineering Support Division

2006 XSD Scientific Software Workshop

Scientific Software Section

The Scientific Software Section is a newly-created section that is responsible for scientific software for the APS. This group has a two-fold mission: (1) It is to assist in integrating existing analysis and modeling codes with existing data acquisition software, and (2) It is to develop new codes for both existing and new techniques. The group will implement the recommendations of the 2006 XSD Scientific Software Workshop.

The acronynm **XRAYS** stands for X-Ray Analysis Software or X-Ray Analysis Software.

A list server has been set up for XRAYS. Information about the list, including how to suscribe, is available at http://www.aps.anl.gov/mailman/listinfo/xrays, and the archives are available at http://www.aps.anl.gov/Mailman/archives/public/xrays/2006-June/date.html.

U.S Department of Energy | UChicago Argonne LLC | Office of Science Department of Energy | Office of Basic Energy Sciences Serving the Present, Shaping the Future

Privacy & Security Notice | Contact Us | Site Map

Scientific Software Section

- Specific goals:
  - Combine existing analysis and visualization codes with beamline data acquisition software and transform these codes into easy-to-use software
  - Provide a scientific workbench program that is easy to use and learn and from which users can access all the software that is necessary to manage the entire scientific work flow
  - Create new analysis and visualization applications that can be used on all beamlines and that are easily integrated into the standard workbench
  - Develop a software framework, perhaps more than one, that provides tested and debugged scientific routines, such as fitting and visualization, which can be used by developers to create applications
  - Create an interface to the facilities necessary to provide high-performance computing
  - Provide documentation, distribution, maintenance, and support

Rationalization for Eclipse

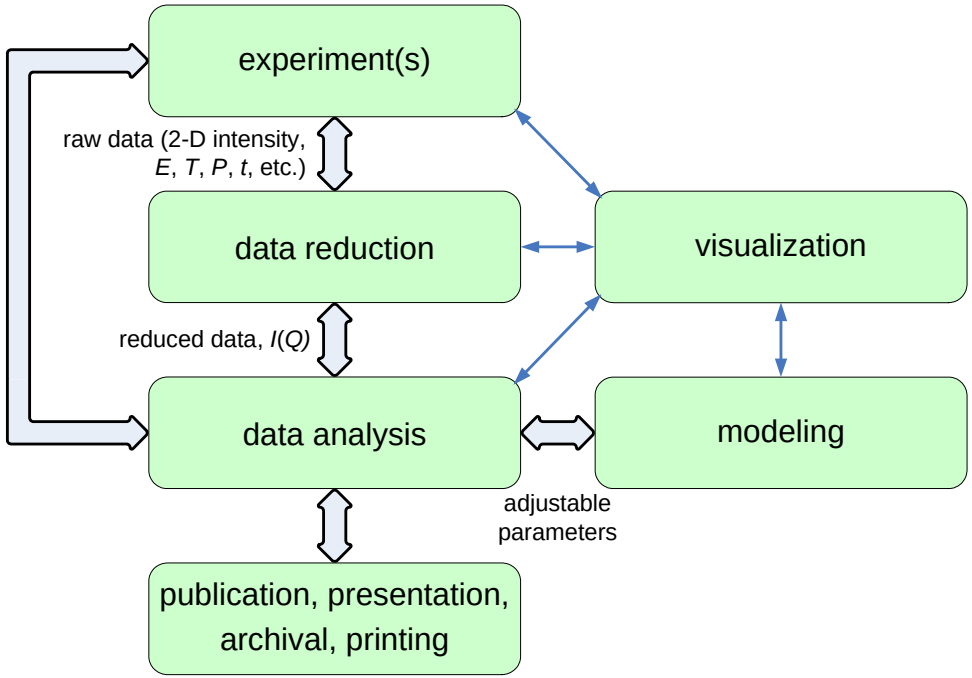
- Providing coordination is a primary goal
- Resources are limited
- Have to choose something
  - Eclipse seems like the best choice
  - Powerful, flexible, extensible
  - Open-source
  - Huge community with many projects

- Java development environment leads to high productivity
- Deployment via plug-ins appears to solve many problems

- Downsides
  - Most X-Ray beamline staff and users are not using Eclipse now

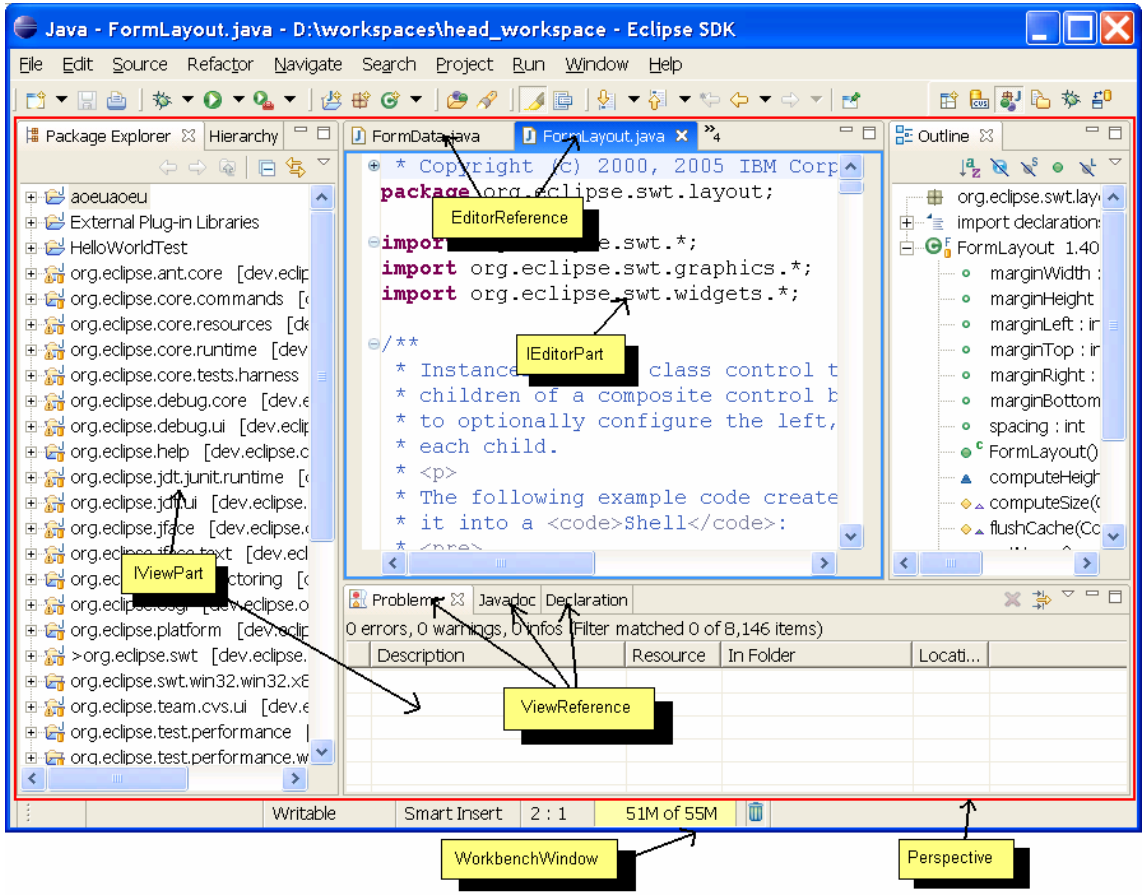
Want to Manage the Entire Experimental Data Flow

- Eclipse should provide the workbench to do this
  - An IDE without the “D”
  - A way to organize your work
  - A tool for anything in general and nothing in particular



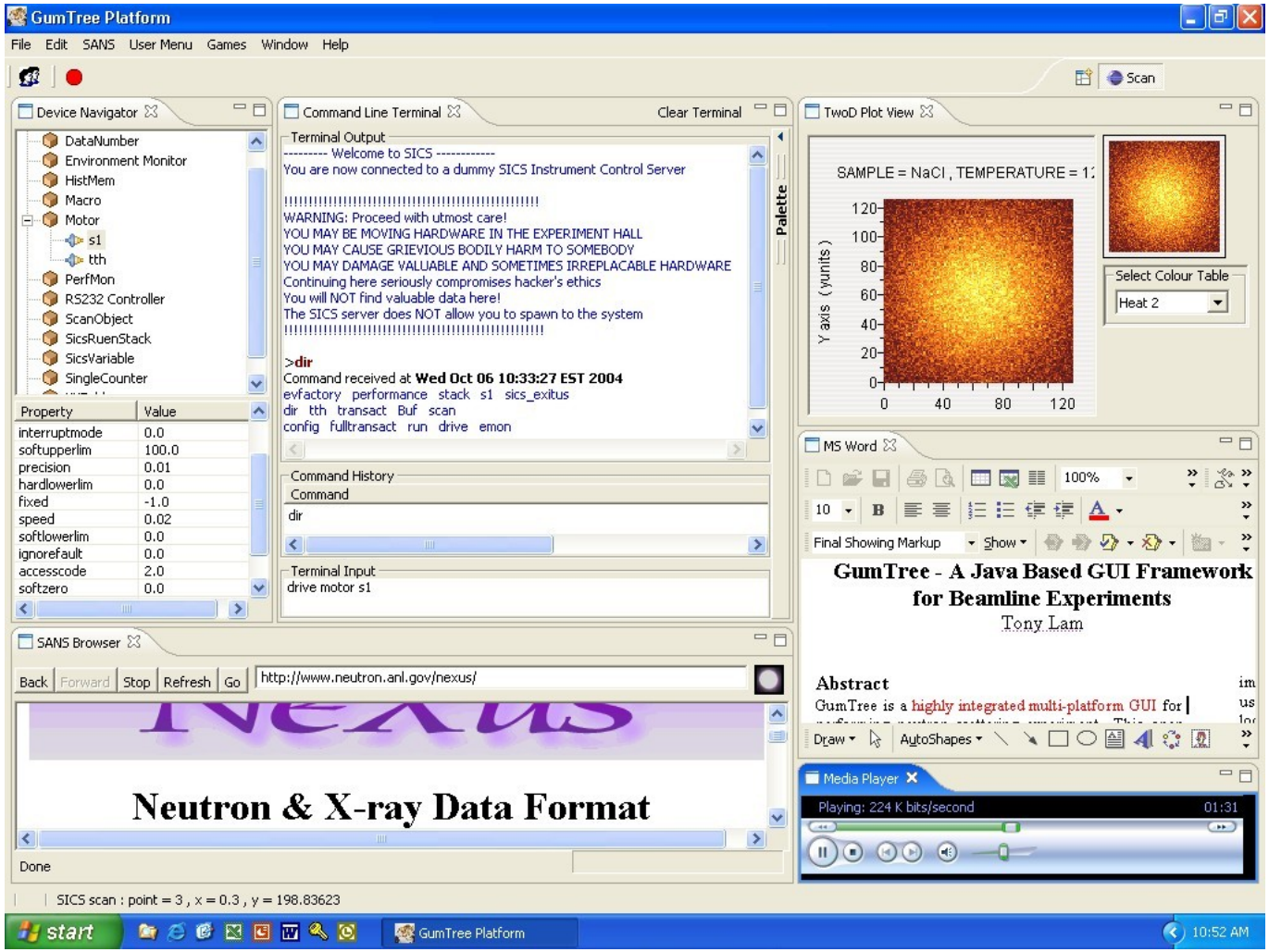
Eclipse Layout Fundamentals

- Perspective: A particular layout of a Workbench window
  - Has zero or one editor area and zero or more surrounding Views.



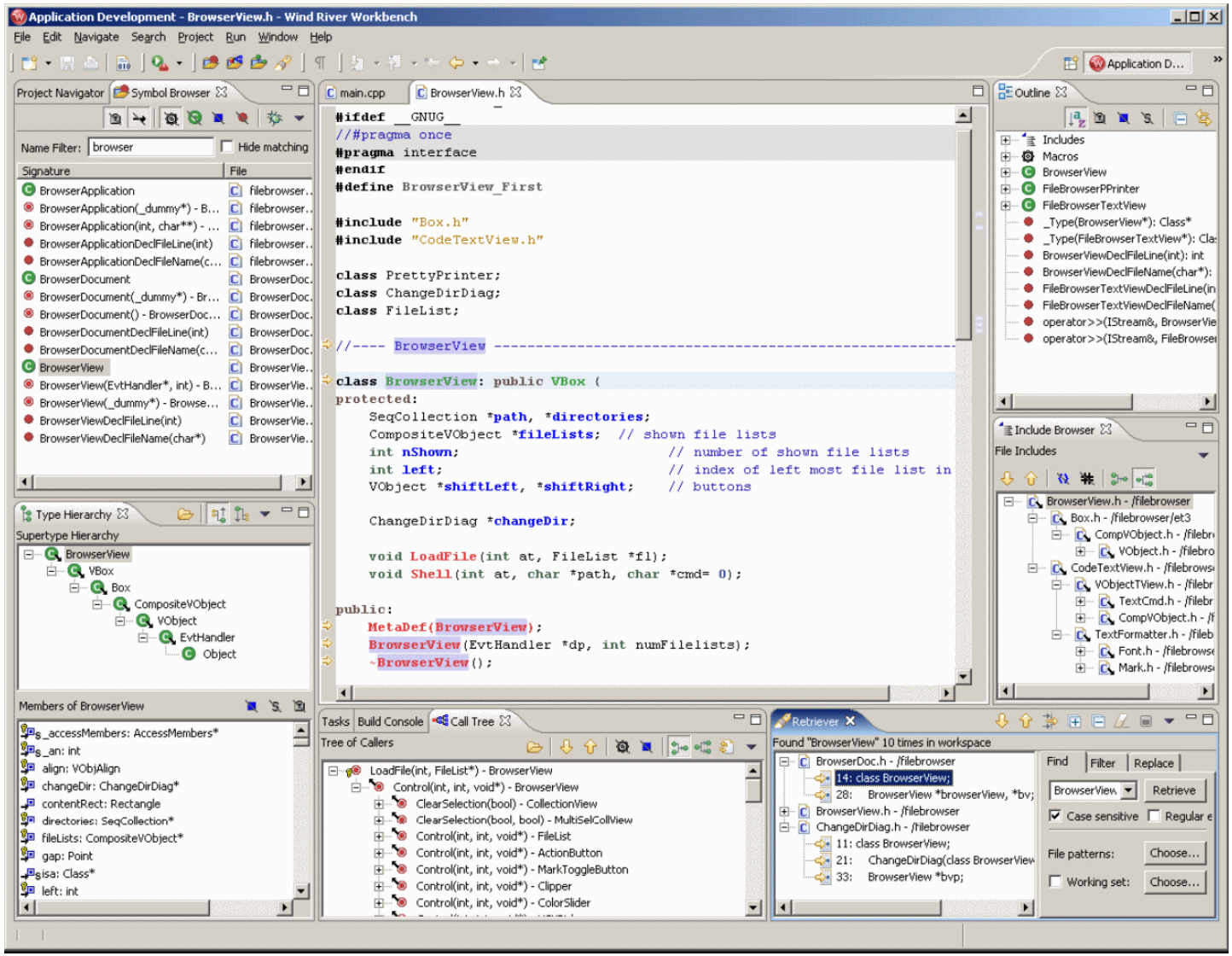
From: Stefan Xenos, Inside the Workbench - A guide to the Workbench Internals
http://www.eclipse.org/articles/Article-UI-Workbench/workbench.html

GumTree (ANSTO)

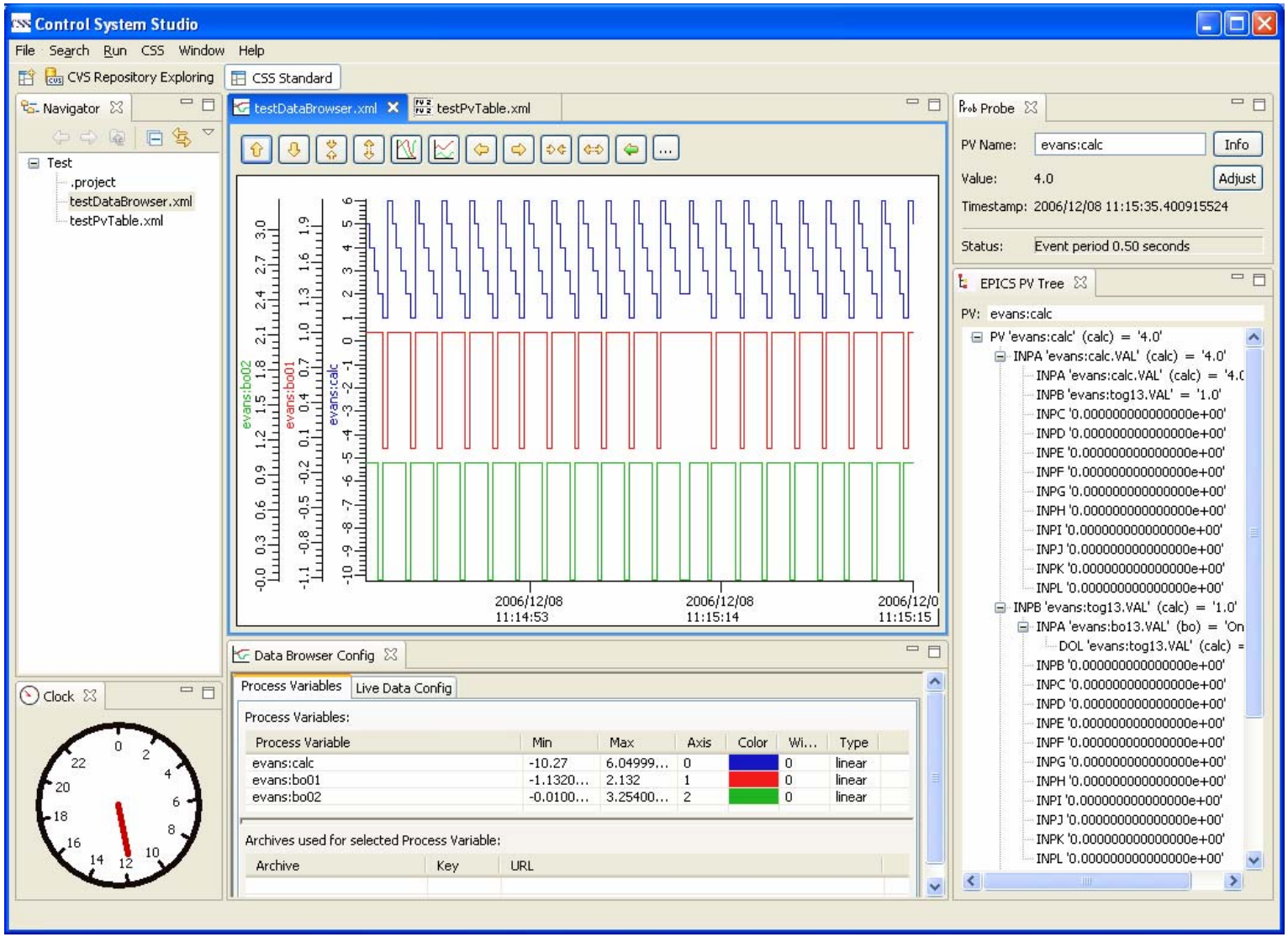


From: Tony Lam, ICALEPCS Presentation, October 2005

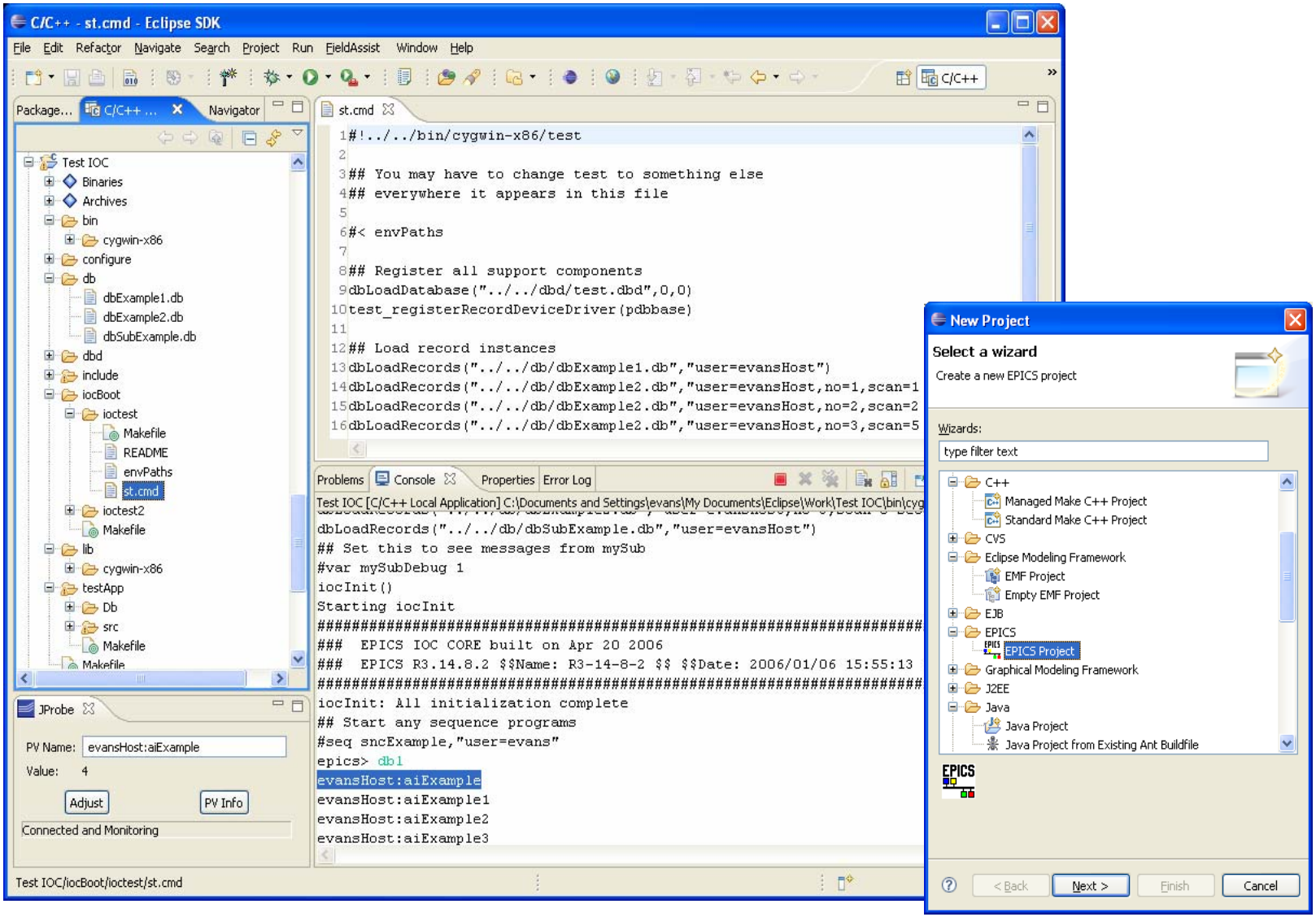
Wind River Workbench



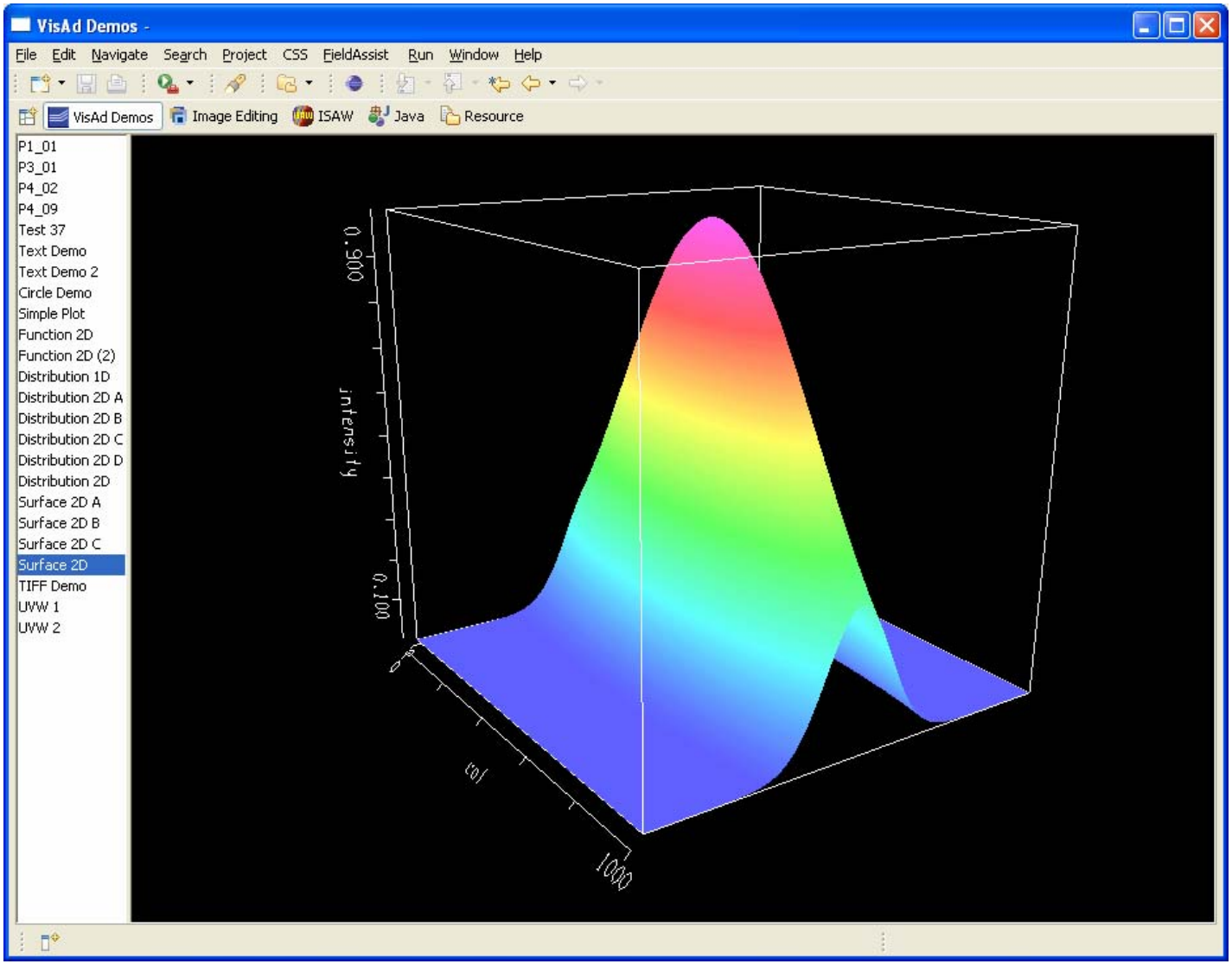
EPICS Control System Studio



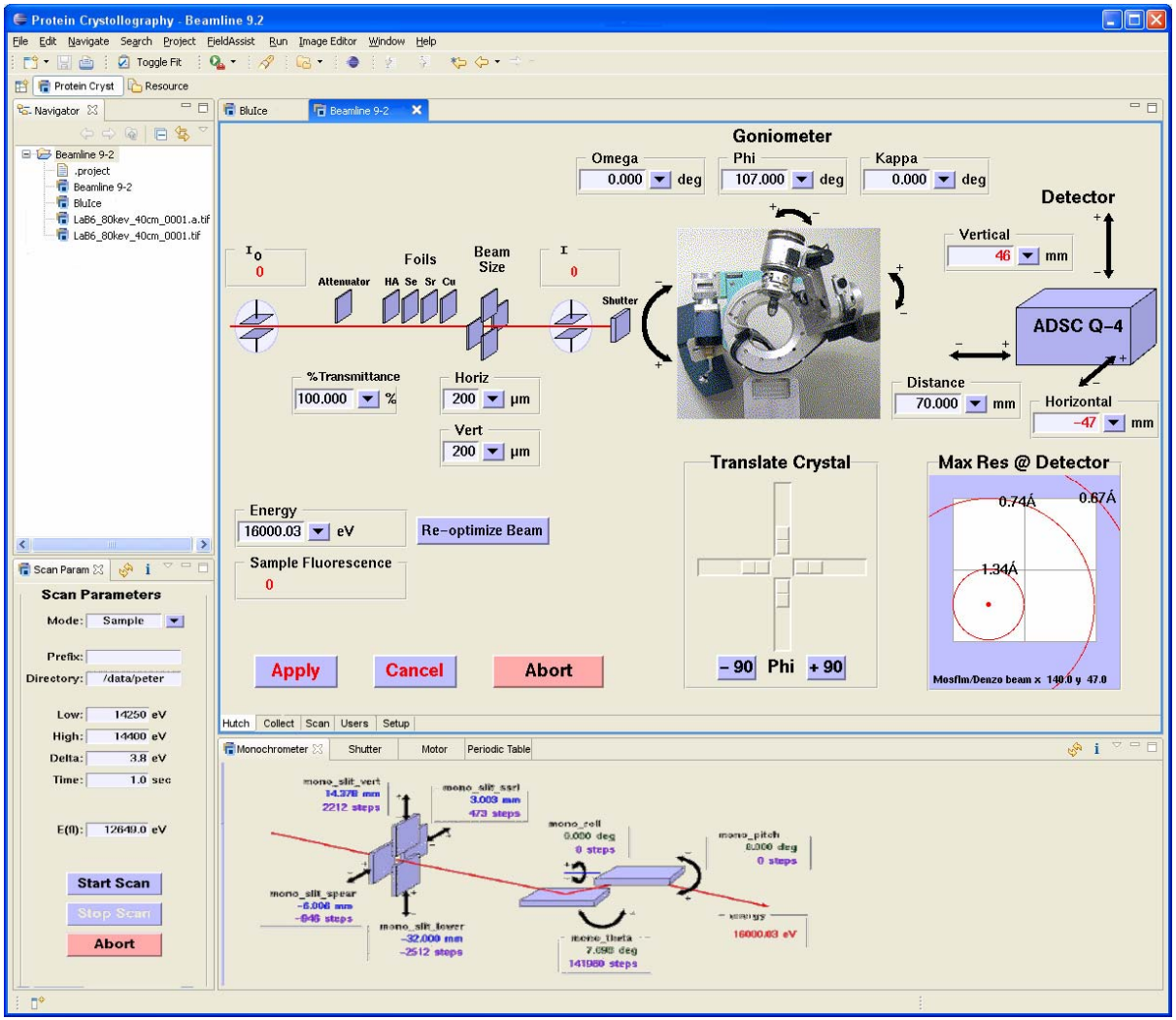
EPICS IDE : IOC Development



VisAD: A Perspective Can be a Single Application

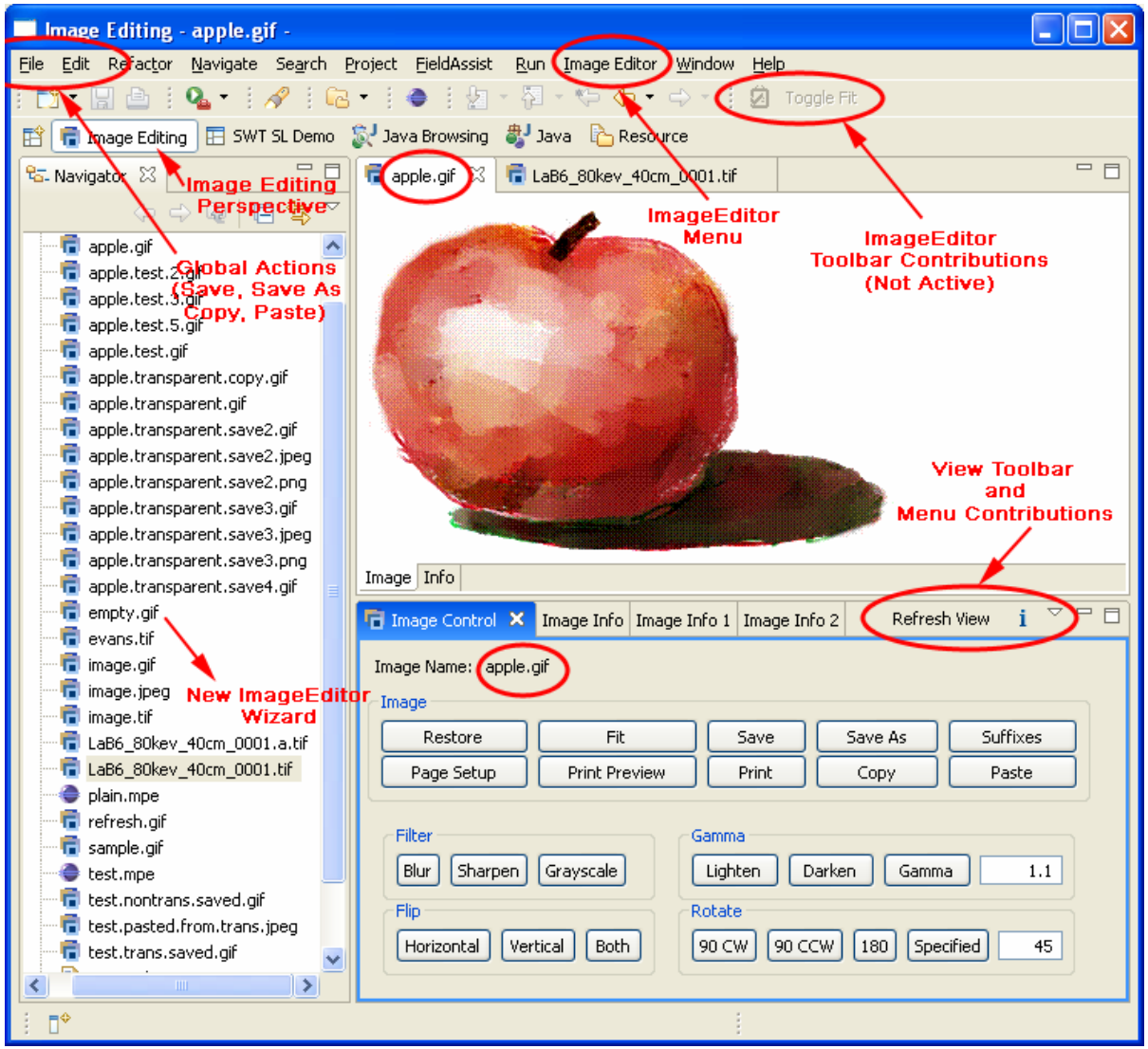


X-Ray Experiment



Images from: BLU-ICE and the Distributed Control System, NOBUGS III, January 2000

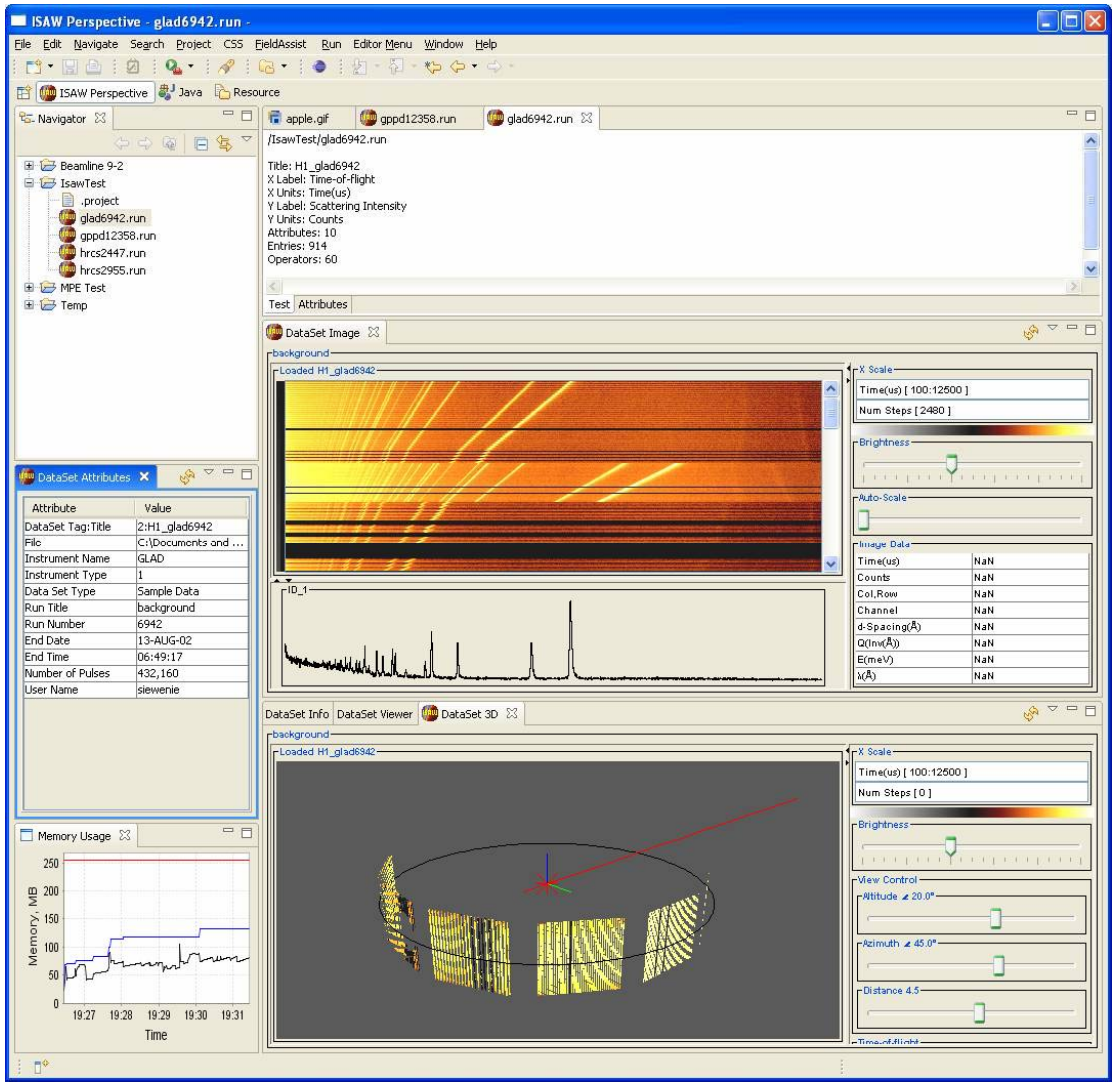
Image Editor as an Experiment Prototype



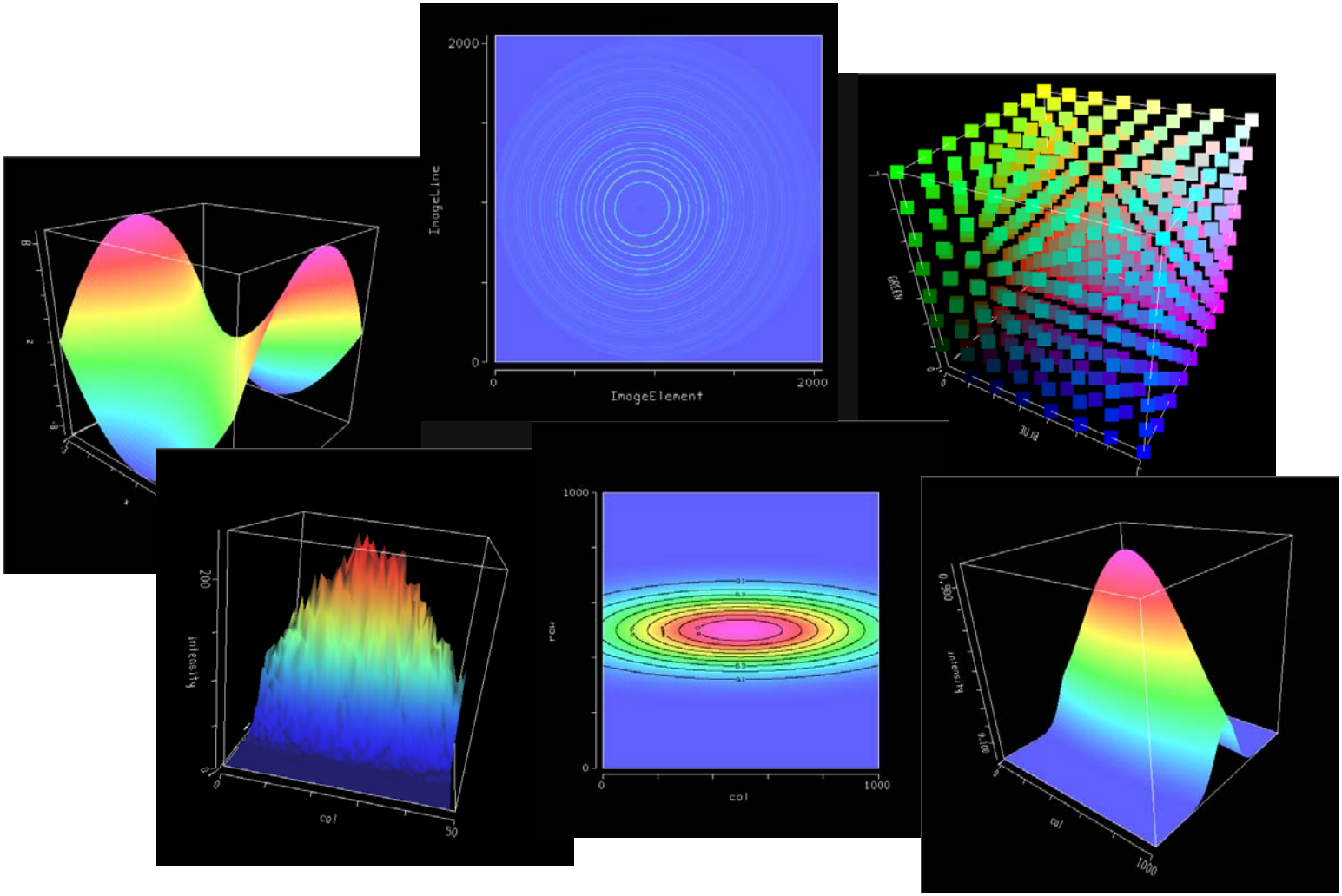
Prototype Implementation of ISAW in Eclipse

- Perspective
- Editor for DataSets
  - .run, .isd
- Some Views

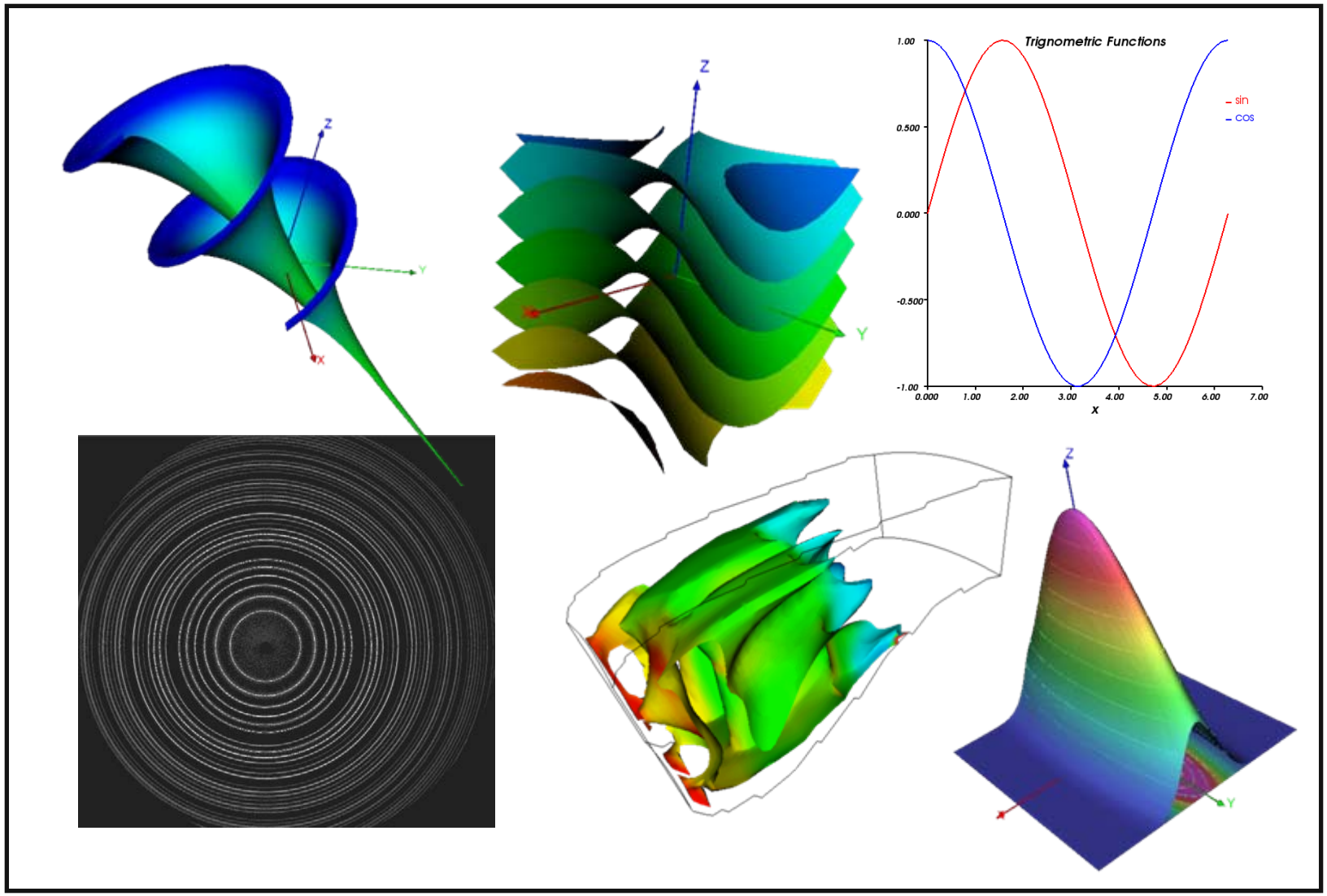
- All work together



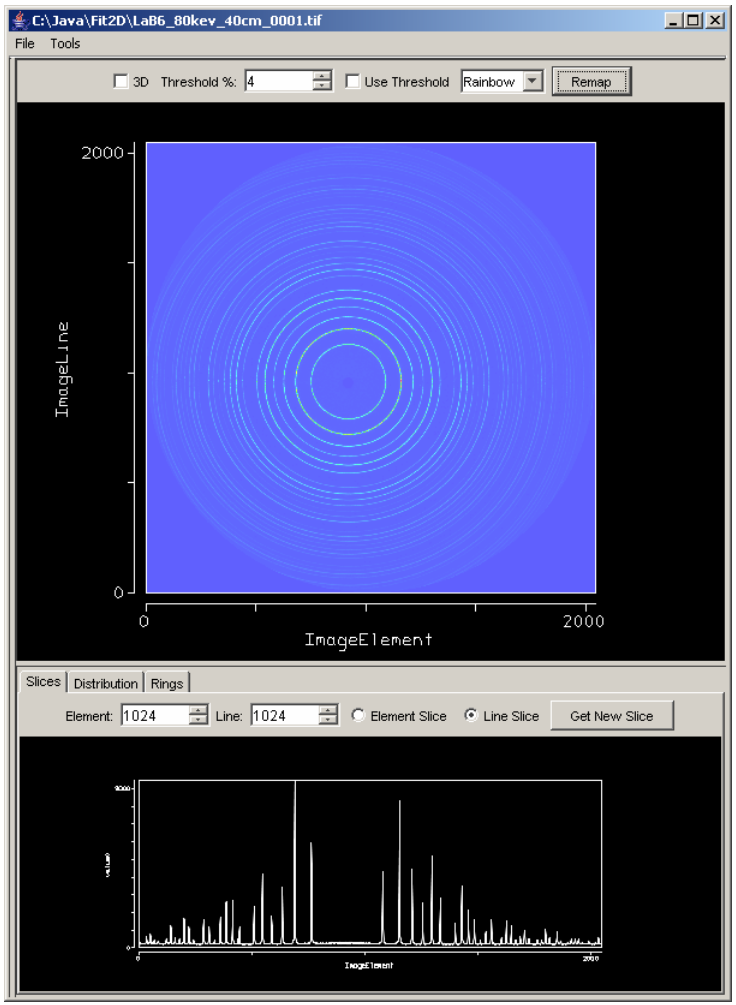
Visualization: VisAD

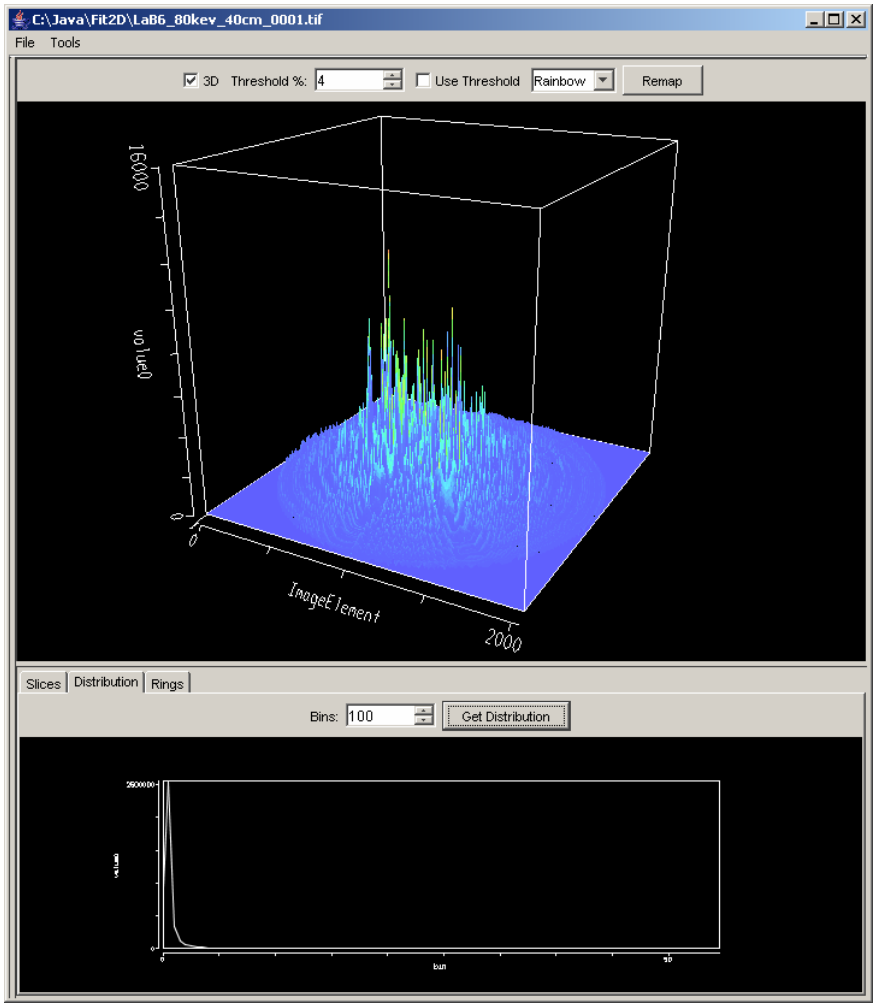


Visualization: VTK



Prototype Image Analysis Tool using VisAD Graphics





Thank You

This has been a

Scientific Software Presentation

Thank You

This has been a

Scientific Software Presentation