Java Analysis Studio and the hep.lcd class library

Mike Ronan - LBNL

Joanne Bogart, Gary Bower, Tony Johnson - SLAC

Nick Sinev - Oregon

Don Benton - U Penn

Physics and Experiments with Future Linear e+e- Colliders

Thursday, April 29 , 1999

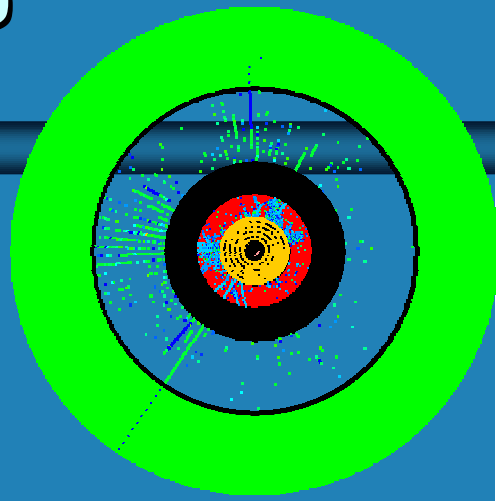
Contents

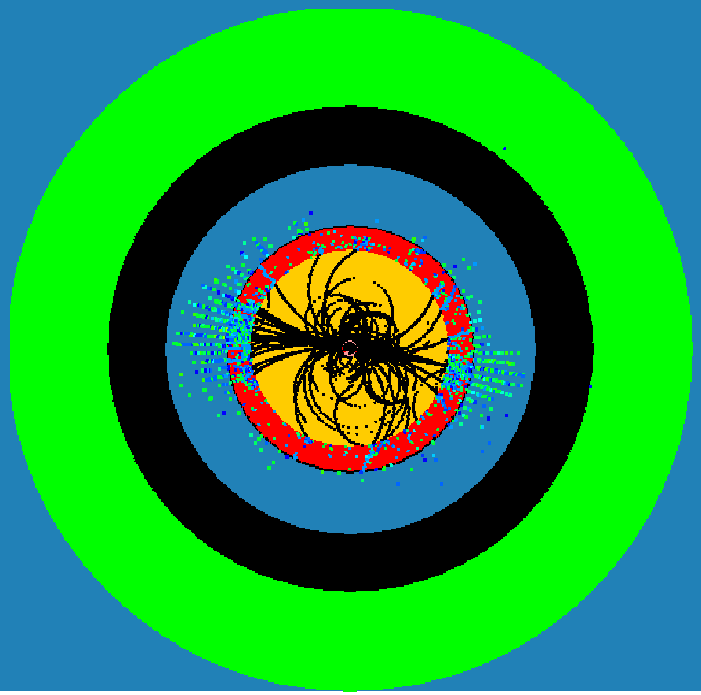
- The hep.lcd framework for LC physics studies
  - Overview
  - Fast MC
  - Tracking Reconstruction
  - Cluster Finding
- Java Analysis Studio
- Distributed Physics Analysis
- Performance Comparison
- Conclustions - How to try it out!

LC Detector studies in US

Goals:

- Detailed Study of physics processes in a variety of possible LC Detectors.
  - Reference Small and Large detectors
- Full simulation with GISMO
  - Switch to Geant4, when ready
- Analysis using
  - Java & Java Analysis Studio
  - C++ & Root
- Software Requirements
  - Flexibly handle different detector geometries and technologies
  - Rapid development of variety of reconstruction and analysis algorithms



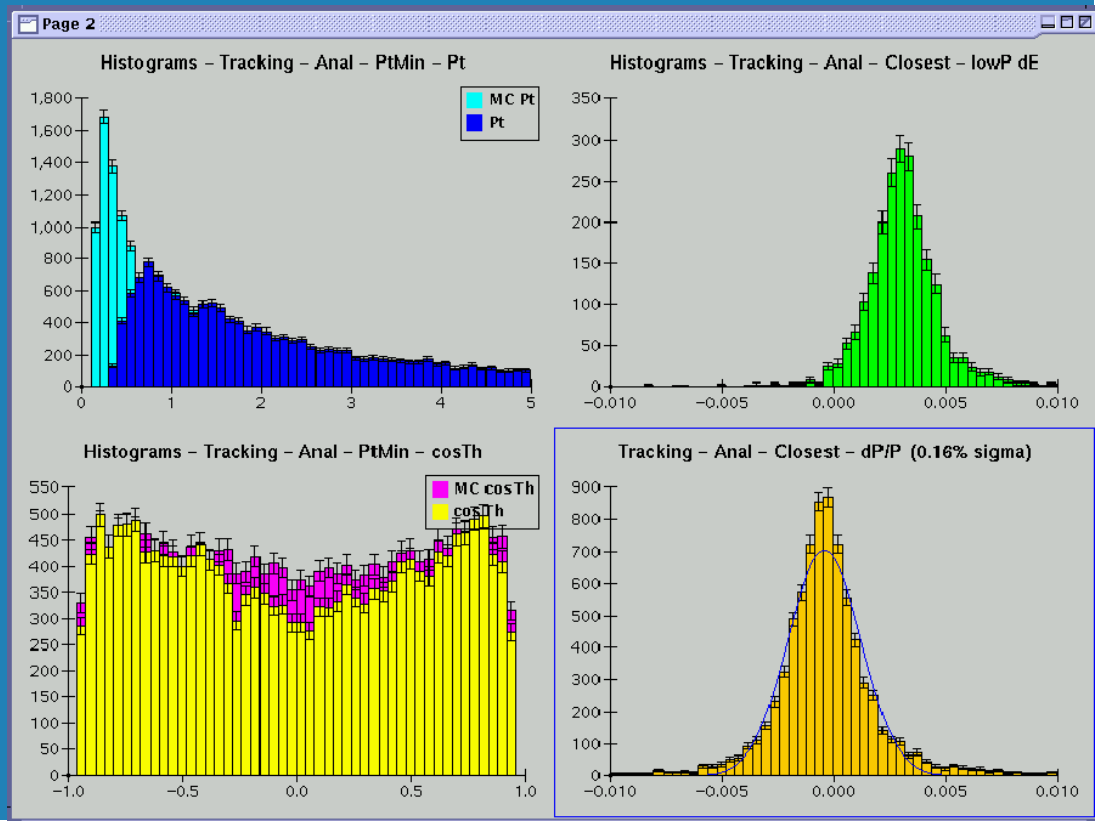


Java package hep.lcd

- **Framework**
  - Driver framework
    - interactively control
      - calling of processors
      - debugging/histograming
  - Parameter (Constant) access
    - driven by detector geometry
  - MC event input (StdHEP format)
  - IO system based on Java IO
    - random access files allows efficient access to subset of data
  - Can be run inside JAS or standalone
- **Reconstruction Processors**
  - Track finder + track fitter
  - Several clustering algorithms
- **Parameterized MC Processors**
  - Can read generator output (StdHEP) or Gismo output
  - Track and Cluster smearing
- **Analysis Utilities**
  - Event Shape + Thrust utilities
  - Jet finders (Jade, Durham)
  - Histograming
- **Event Display**
  - Simple 2D Event display currently

Track finding and fitting package

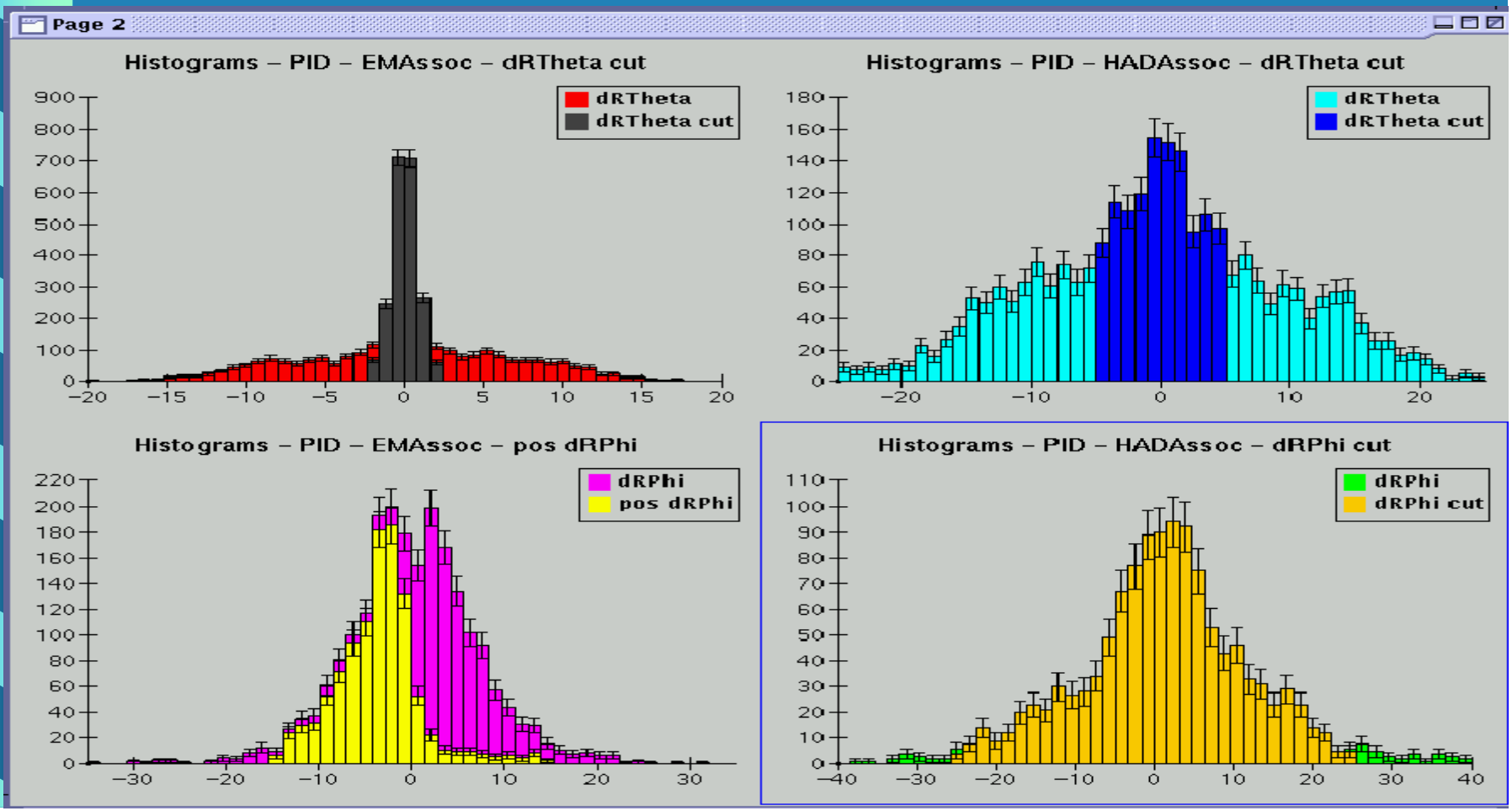
- Track finding package derived from TPC, Babar
- Track fitting based on SLD Weight Matrix fit algorithm
  - Tracking chamber or tracking+vertex detector fit possible



Clustering + FastMC

- **Three Clustering Algorithms** implemented:
  - Simple Cluster Finder (contiguous energy)
  - Cluster Cheater (perfect clustering using MC info)
  - JRBCluster - configurable cluster finder
- **FastMC**
  - Simple parameterized MC
    - Allows analysis directly from generator output without using full Gismo simulation
  - Produces same event format as Gismo
    - same analysis can be run with FastMC or Gismo.

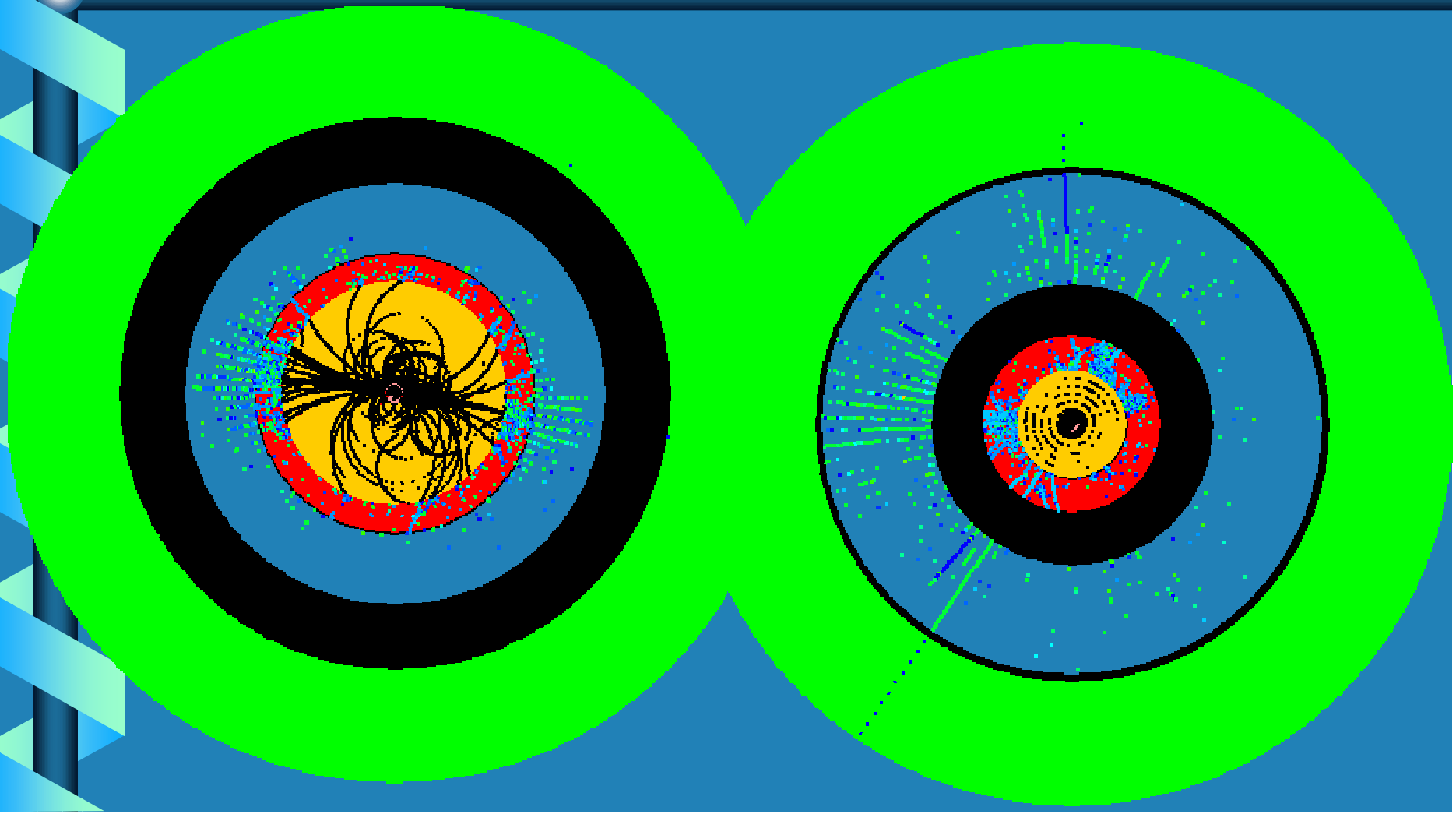
Track-Cluster Association being Developed



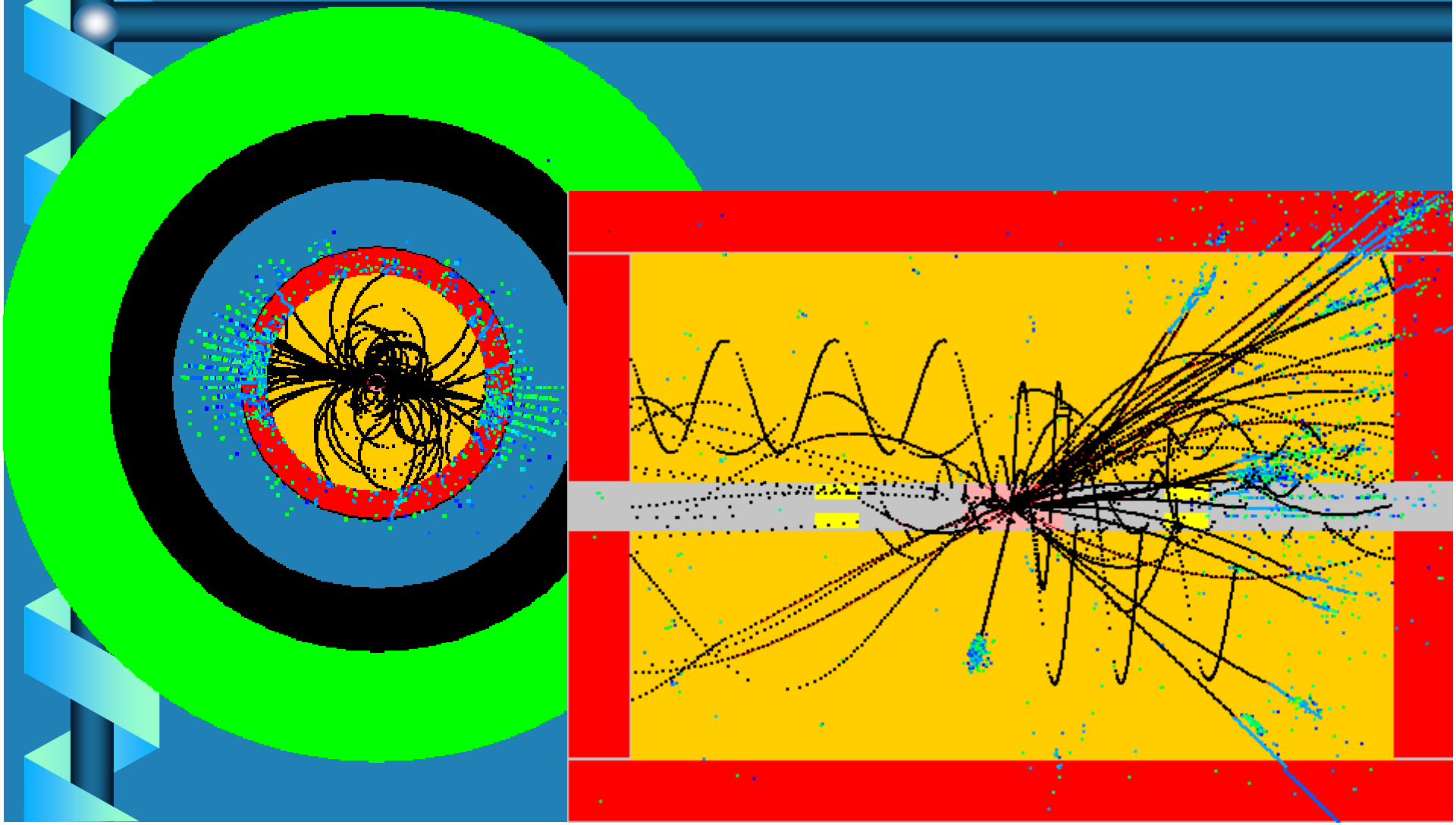
Physics Utilities

- **Physics Utilities**
  - 4-vector, 3-vector classes
  - Event shape/Thrust finder
  - Jet Finder
    - Jade and Durham algorithms implemented
    - Extensible to allow implementation of other algorithms
- **Histograming** (from Java Analysis Studio)
- **Event Display**
  - Suitable for debugging reconstruction and analysis
  - Plan to use Wired for full 3D support in future

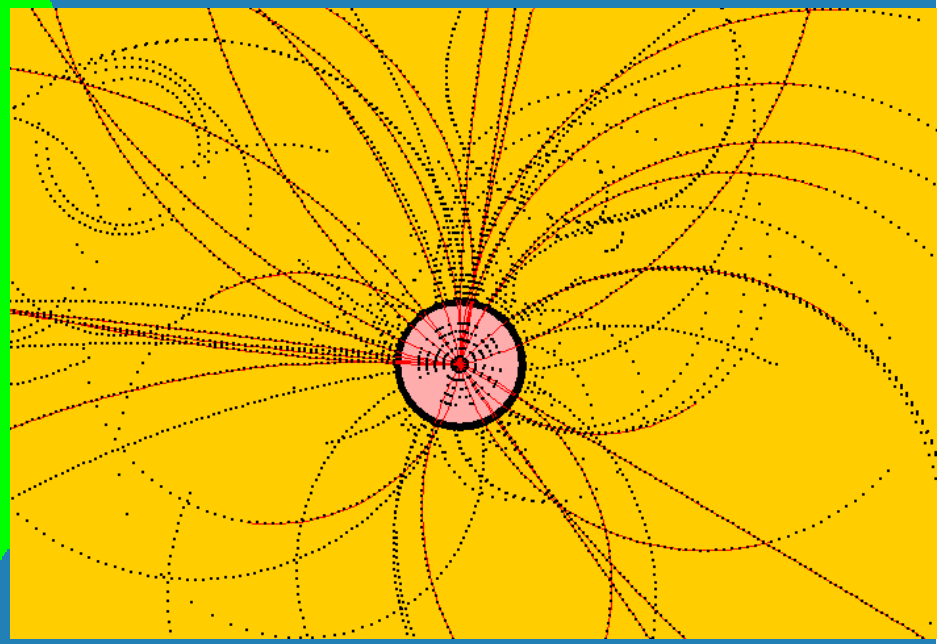
Event Display



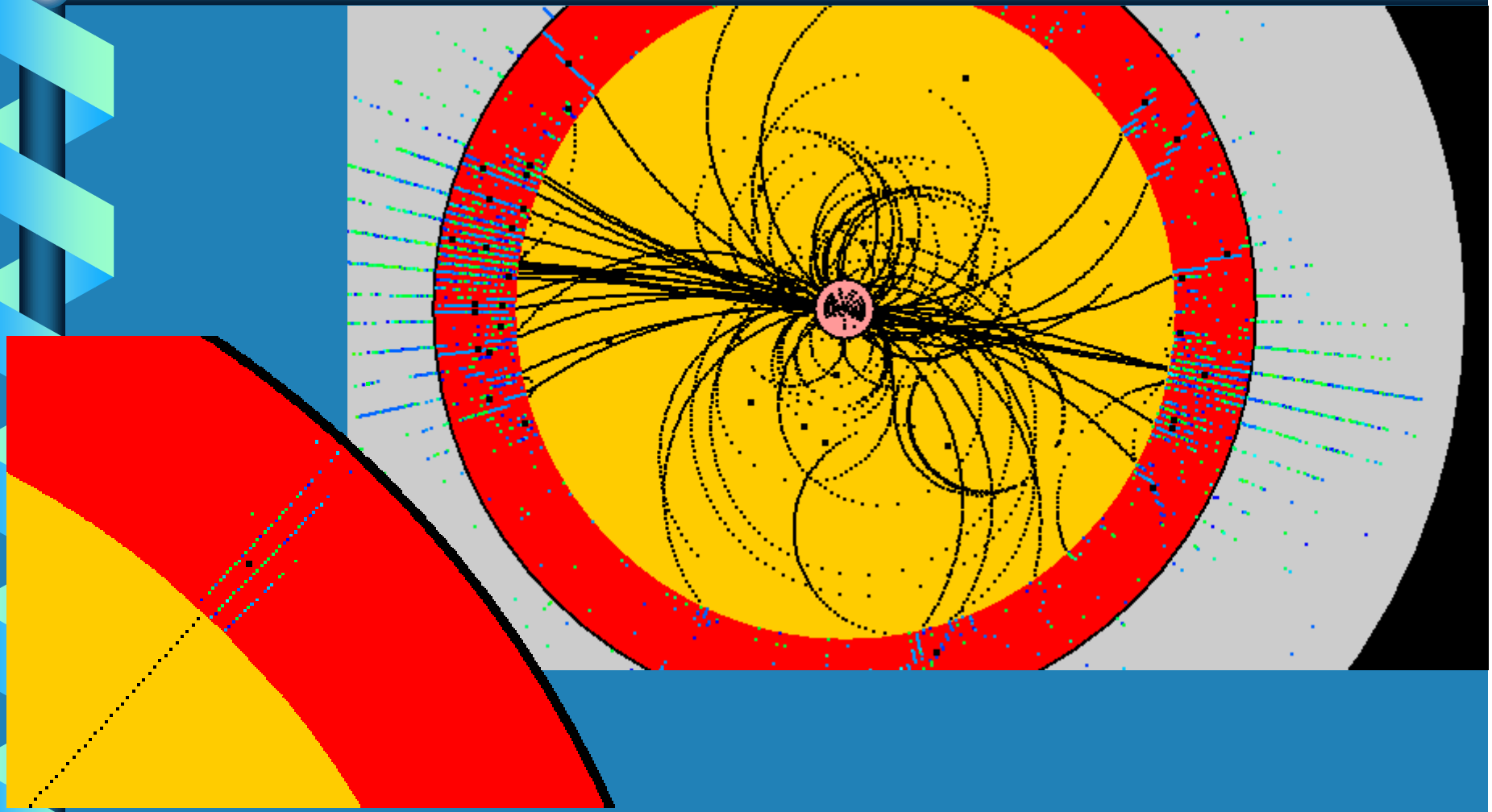
Event Display



Event Display



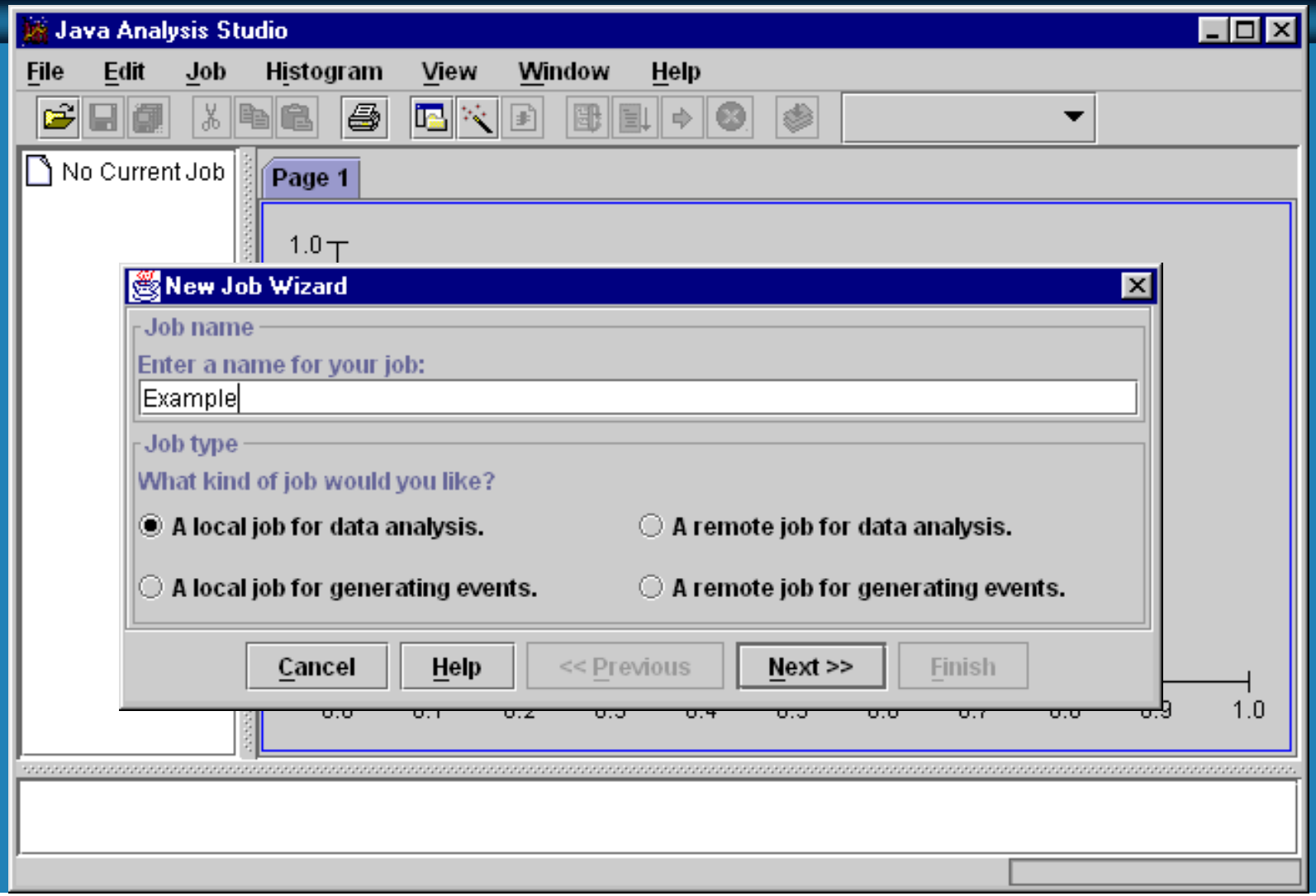
Event Display



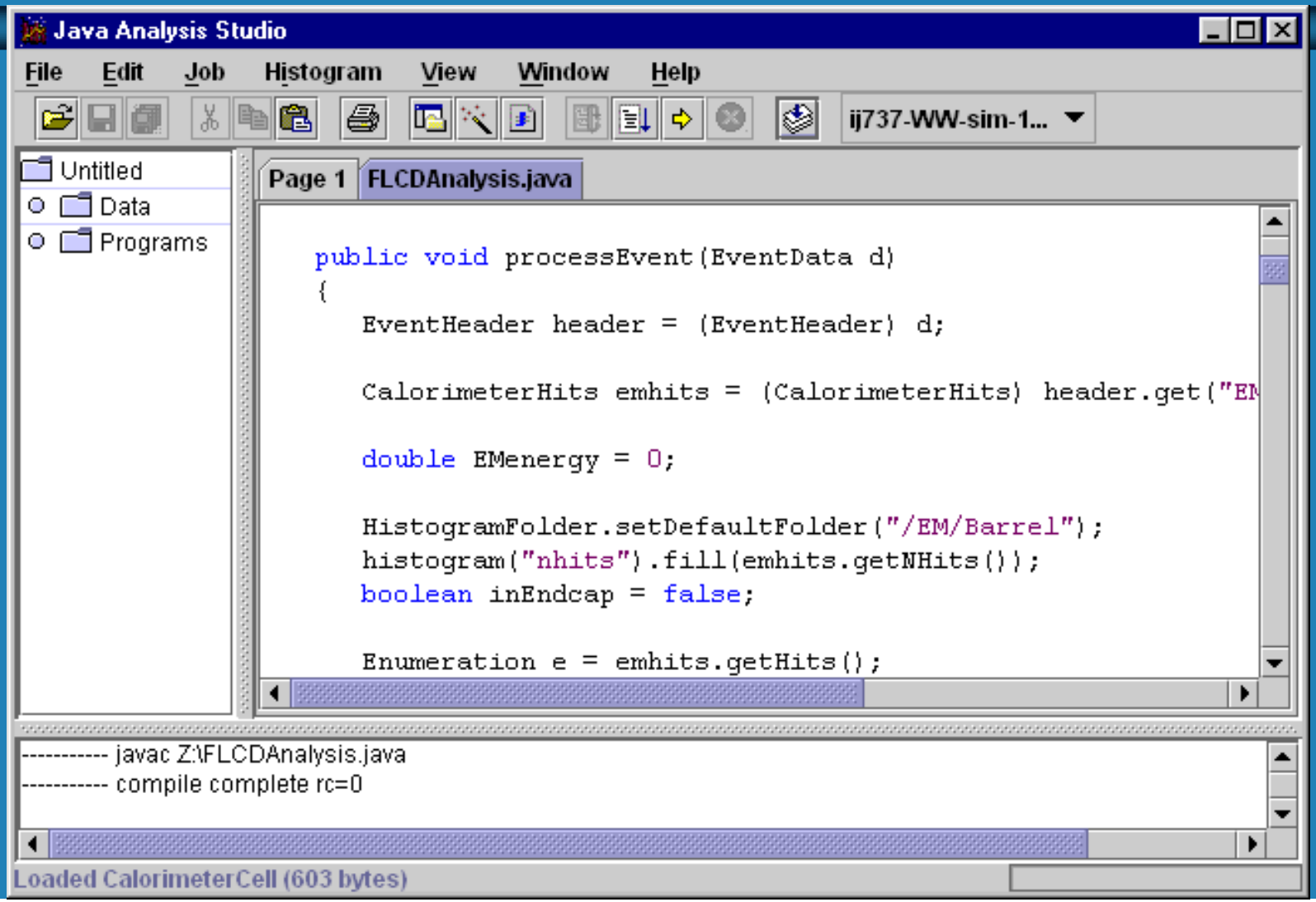
Java Analysis Studio

- Set of experiment independent analysis tools for event oriented (High Energy Physics) data
  - Data Access classes provide access to many common HEP data formats
  - Histogram/Scatterplot Accumulation + Manipulation Classes
  - Plot Display classes
  - Lightweight framework for users to create physics analysis applications in Java.
- Tools work alone, in combination, or within
  - Java Analysis Studio GUI which gives:
    - Integrated editor and compiler
    - Efficient access to local and remote data
    - Extensibility via Plug-ins, Fitters, Functions etc

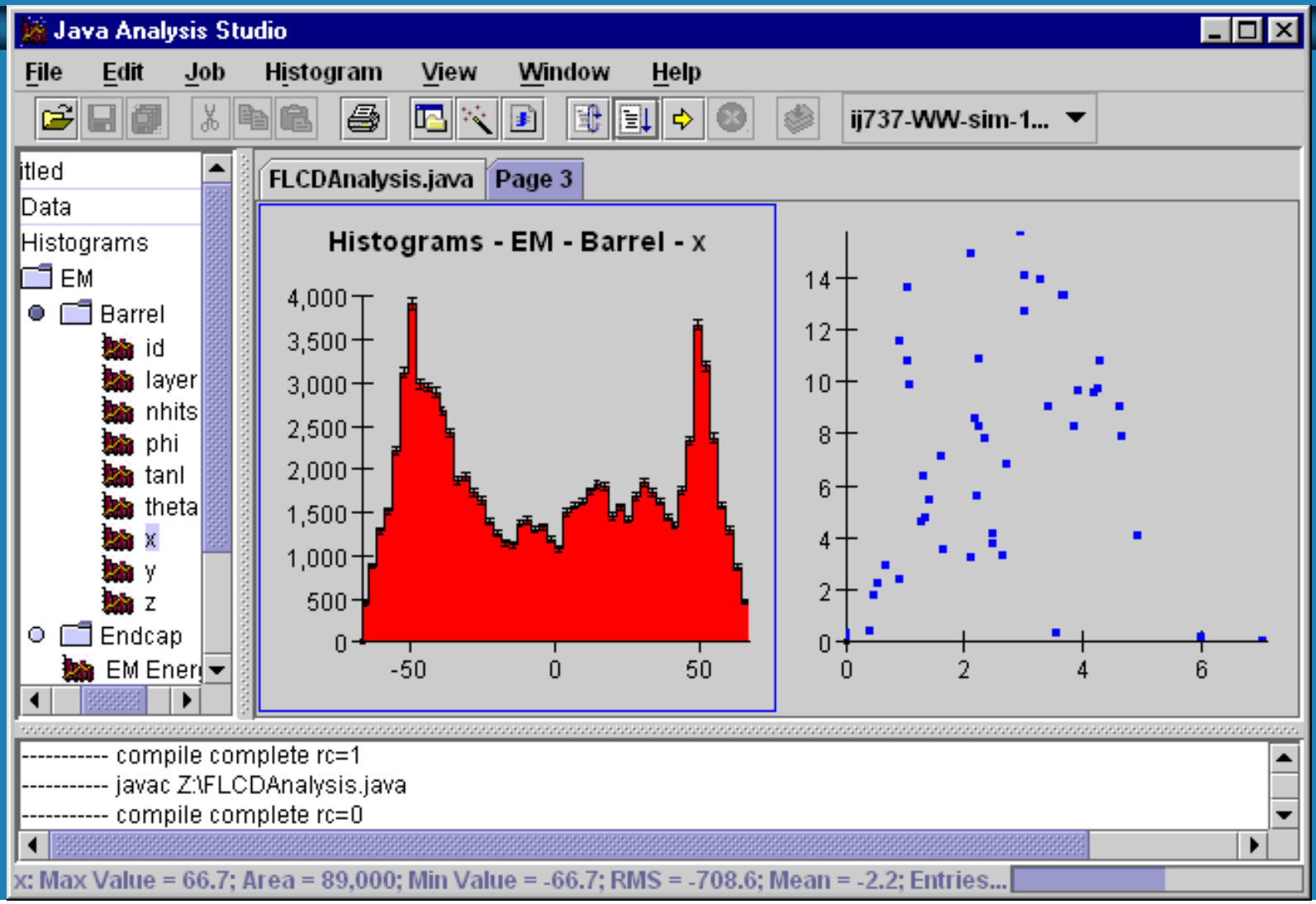
GUI makes getting started easy
"Wizards" guide beginners



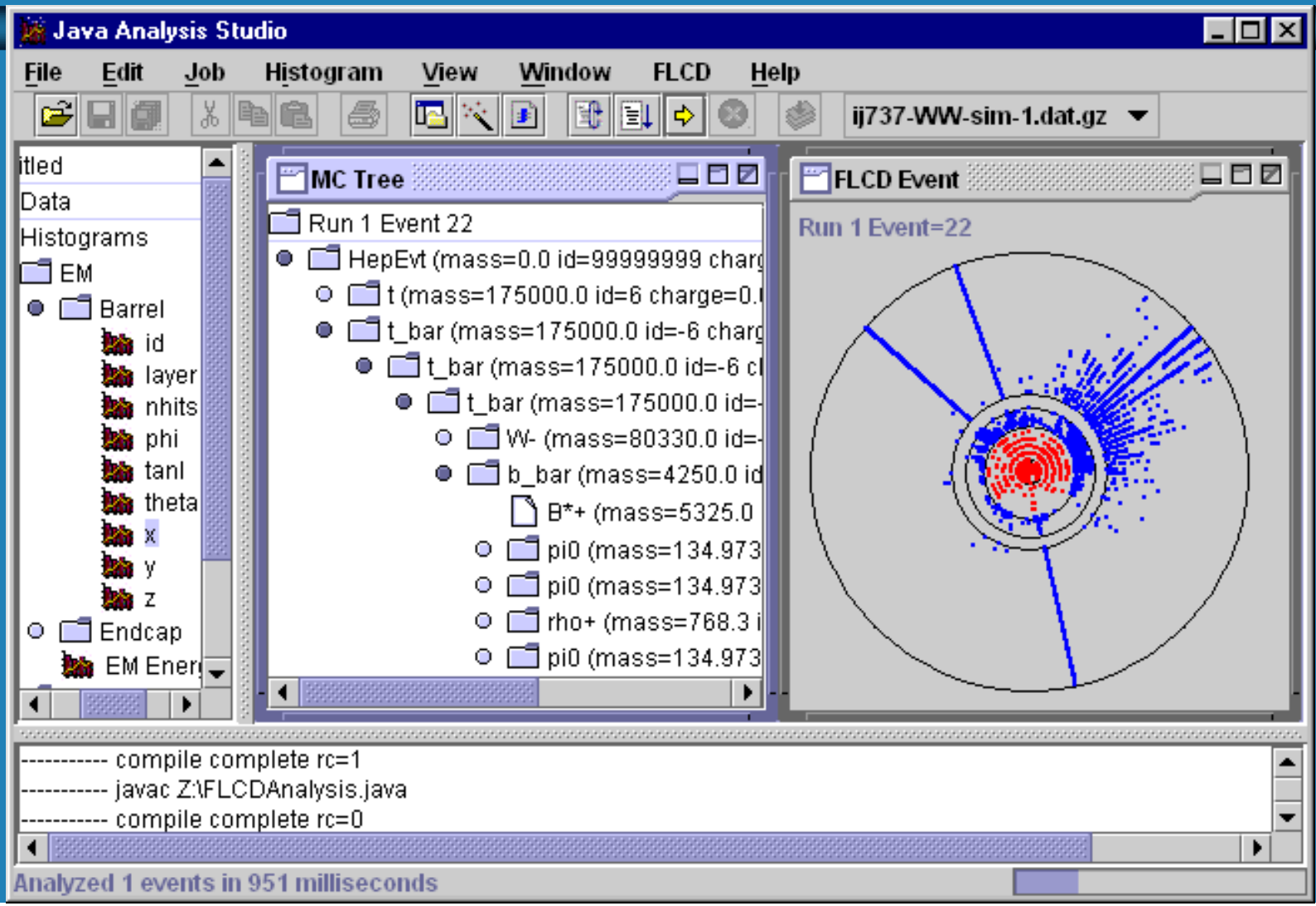
Built in Editor and Compiler for writing analysis code



Histogram and Scatterplot display Interactive Fitting and Rebinning

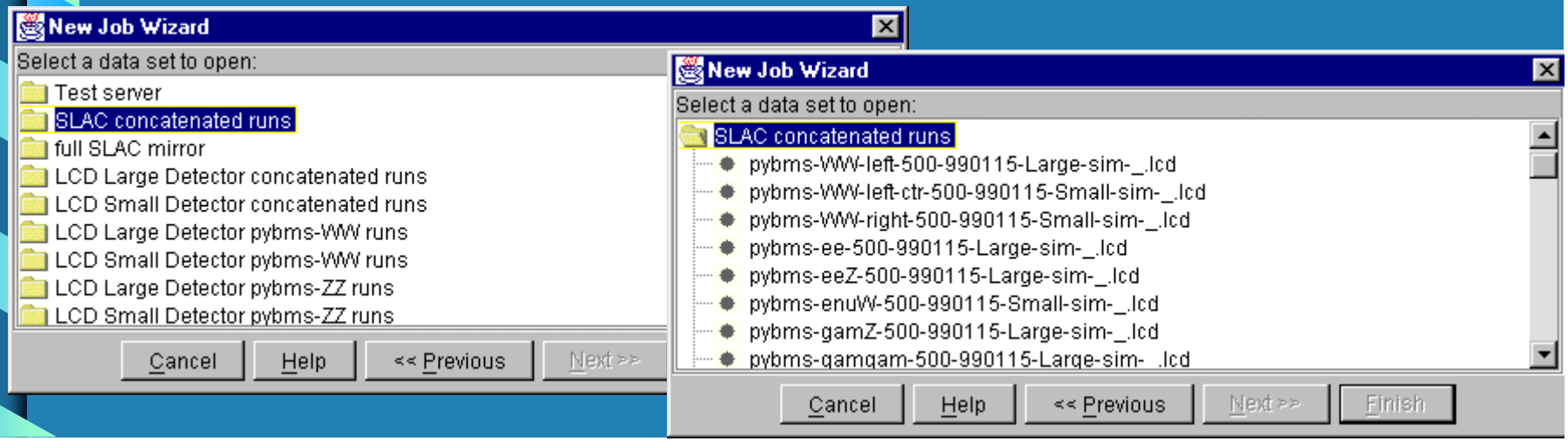


GUI can be extended to add experiment specific features

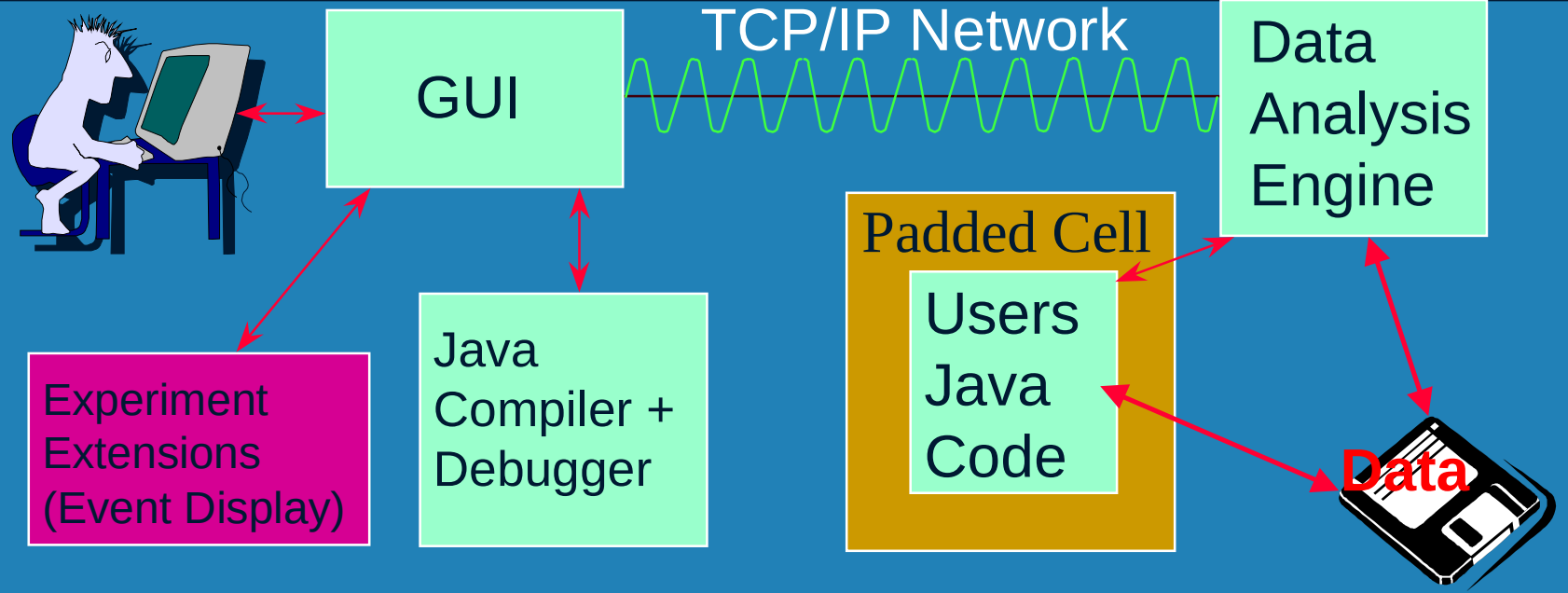


Distributed Data Analysis with JAS

- With many different simulated detectors and many physics processes, total MC data sample is large
- JAS has built in support for efficient distributed physics analysis
- LCD has set up central data repository at UPenn, accessible from anywhere



Distributed Data Analysis with JAS



- Java allows objects to move from client to server - even across different platforms
- Since analysis code is moved to data - analysis is fast
- Transparent to end user, who "feels" as if analysis is running locally

Is Java fast Enough for HEP offline?

- Current (266Mhz PII, JDK 1.1.7)
  - Clustering .6 secs/event
    - 13.5 Million Calorimeter Cells
  - Fast MC 6 ms/event
  - Track Finding + Fitting ~5secs/event
  - Very competitive with C++/Root implementation (where they exist)
- Will get even better!!!
  - JDK 1.2, HotSpot - Run-time optimization
- In real life may be faster than C++
  - Better, cheaper performance analysis tools
  - Java encourages lightweight, module interfaces which promote efficienct coding styles

Try it out!

- Works on Windows (95/98/NT) or Unix (Linux, Solaris,...)
- Online tutorial available
  - Suitable for complete beginners:
    - no knowledge of Java or JAS assumed
    - starts with instructions on downloading and installing
    - Shows simple sample analysis jobs
  - http://www-sldnt.slac.stanford.edu/jas/documentation/lcd/
- JAS Home Page
  - http://www-sldnt.slac.stanford.edu/jas

Conclusion

- **Use of Java + JAS looks very promising**
  - Have been able to develop complete framework + full reconstruction package in < 6 months
  - People have quickly learned and use it, and to contribute to the reconstruction package
  - Performance looks good
- **Future**
  - New version of JAS available this month
  - Standard Java interface to Geant4?
  - Continue development of reconstruction and FastMC
    - Direct speed comparison with C++ code