Presented at the 5th International Linear Collider Workshop (LCWS 2000), Oct 24-28, 2000, Batavia, IL SLAC-PUB-10794

Java Based LCD Reconstruction and Analysis Tools

Gary Bower, Ron Cassell, Norman Graf, Tony Johnson, Mike Ronan

Stanford Linear Accelerator Center and Lawrence Berkeley National Laboratory

Abstract. We summarize the current status and future developments of the North American Group's Java-based system for studying physics and detector design issues at a linear collider. The system is built around Java Analysis Studio (JAS) an experiment-independent Java-based utility for data analysis. Although the system is an integrated package running in JAS, many parts of it are also standalone Java utilities.

JAVA ANALYSIS STUDIO AND LCD

Java Analysis Studio (JAS) is a desktop data analysis application aimed primarily at offline analysis of high-energy physics data. The application is independent of any particular data format, so that it can be used to analyze data from any experiment. The application features a rich graphical user interface (GUI) designed to make the program easy to learn and use. By writing analysis modules in Java using the built-in editor and compiler, users can perform arbitrarily complex data analysis tasks. The application can be used either as a standalone application, or as a client for a remote data server – a feature particularly targeted at allowing remote users to access large data samples stored at a central data center in a natural and efficient way. Users can write "plugins" to extend the functionality of JAS in experiment or analysis specific ways.

More information about JAS can be found in the proceedings of CHEP98[1] and CHEP2000[2]. Since these papers were written many enhancements have been added, the details of which can be found on the JAS website[3]. Recently-added features of particular relevance to the linear collider effort include much greater ease in comparing plots from different datasets and algorithms, greater ease to download and install plugins and greatly improved support for printing plots.

The overall goal of the LCD project is to develop a complete simulation and analysis package for studying the physics and detector issues of linear colliders. We have developed an extensive suite of programs written in Java that can be used inside JAS, or as standalone utilities[4].

Work supported by the Department of Energy contract DE-AC02-76SF00515.

Stanford Linear Accelerator Center, Stanford University, Stanford, California, 94309

Using either approach the user can select either of two paths to study physics and detector issues. One option entirely within the package is to generate events, run them through a Fast Monte Carlo, reconstruct them and analyze the results. In the other option, the package can take input files from an external source containing events that fully simulate the response of a detector. Transfer of data between different packages is achieved using SIO, a platform independent serial IO format developed for the LCD work. (SIO replaces our earlier use of ASCII files, and results in improvements in IO speed and size of files, as well as eliminating problems with loss of precision present with the ASCII files). A second data format, called LCD format, allows random access to parts of events and is optimized for data analysis tasks.

LCD RECONSTRUCTION TOOLS

The present North American Group effort is focused on two detector designs described elsewhere in these proceedings.[5] The reconstruction has been designed to work with both of these detector designs, and to be easily modifiable for other detector designs of interest in future. The reconstruction has also been built to support a variety of different analysis algorithms, so that it can be used to experiment with different reconstruction methodologies as well as different detector geometries and technologies.

The track reconstruction runs in two stages, first track finding using pattern recognition tuned for the two detector types, followed by track fitting based on the SLD weight matrix fitter. The fit can be based on a single tracking detector or combined with a vertex detector. Hit smearing for realistic efficiencies is included. Coming developments include more algorithms for track finding (cheater, projective geometry) and fitting (Kalman Filter), end cap tracking and hit merging, and overlays of both random backgrounds and beamstrahlung background. The NLC design involves bunch trains interacting in a series of beam crossings too close in time to readout separately from a tracking chamber. Each crossing will produce background that will be adjusted in time for a TPC and combined in calorimeter cells.

There are currently three algorithms available for calorimeter cluster finding: a cheater which uses MC truth to make a single cluster for each particle, a simple cluster builder which creates a cluster out of touching cells and a radial cluster builder. Future plans are to refine the cluster packages to combine EM and HAD clusters and endcap and barrel overlap clusters. Track-Cluster association is still needed. The fine segmentation and readout possible for new calorimeters will be exploited with developments of tracking-like cluster descriptions including directionality, entry point, etc. Finally, methods are being developed to allow a full simulation with very fine segmentation to undergo cell merging at reconstruction time to study effects of segmentation size without rerunning the full simulation.

A package is under development that simulates and reconstructs (both at reconstruction time) output from various particle ID technologies such as CRID and dE/dx. There is a package of quality control diagnostic plots for checking the validity of generator, simulator and reconstruction output. The package also contains a Fast Monte Carlo and vertex reconstruction capabilities based on the SLD ZVTOP utility. For details on these latter two features see the article in these proceedings[6]. For a description of a similar approach to Fast Monte Carlo based on the C++ Root package see the article in these proceedings[7]. There is an extensive repository of full simulation data available for use with the JAS remote processing capability.

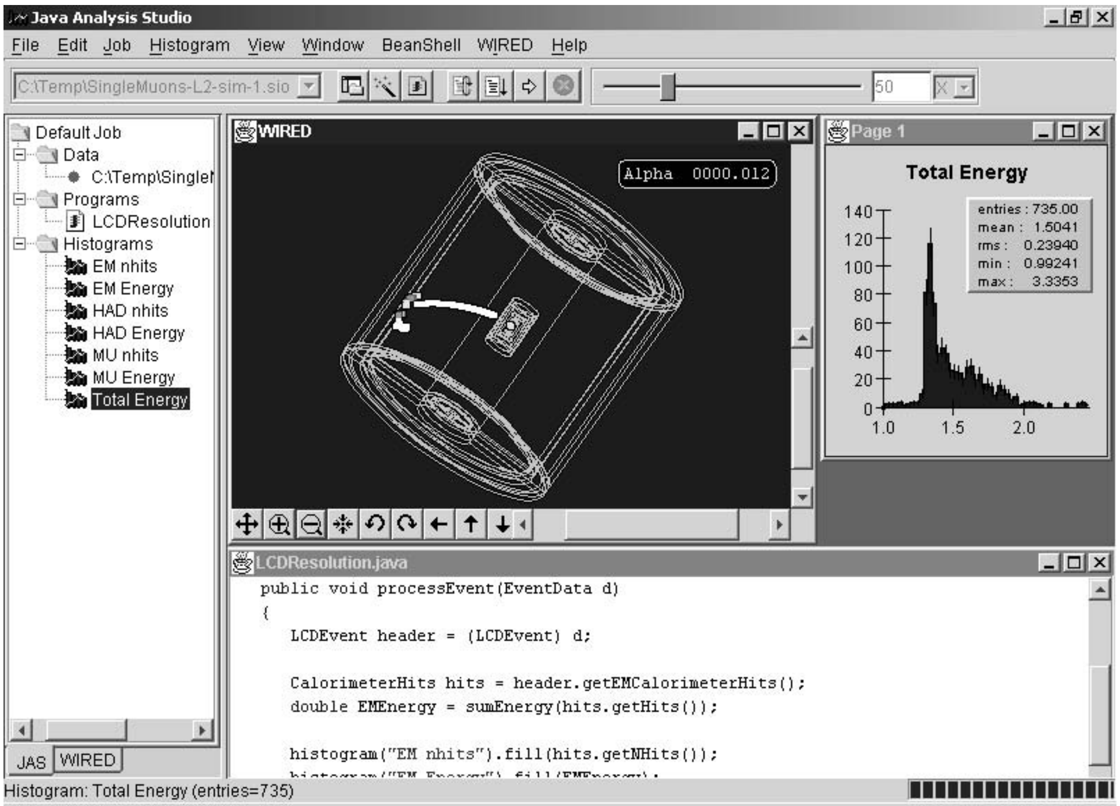

Figure 1 The JAS application showing histogram browsing, plot display, code editor and WIRED event display.

ANALYSIS TOOLS

In the area of physics utilities there have also been advances. There is a 2-D display suitable for debugging reconstruction and analysis routines and the CERN WIRED[8] 3-D event display has now been integrated into the LCD code (figure 1). It is possible to generate convenient MC particle tables and to display the particle hierarchy. The package of jet finders has been extended and additional 3- and 4-vector operations are included.

A diagnostic event generator has been added which can generate individual or sets of particles with a variety of user controllable parameters such as origin, 4-momentum, particle type and particle/anti-particle mixtures. Included is a feature for generating pairs of particles with user selectable separation for two-track resolution studies. The diagnostic generator can run either directly in JAS or standalone with stdhep output format. A standardized framework is being developed for including arbitrary event generators such as PYTHIA in JAS.

All code is in CVS for universal access. One may browse the repository from the Web or connect with any CVS client. There is a platform independent make (jmk) that builds identically on NT and Unix. There is full documentation (API) available for the entire suite and there is an extensive tutorial available for installing and learning to use the package[9]. All the sample code in the tutorials is automatically tested with each new release.

ACKNOWLEDGMENTS

The work was supported in part by the U.S. Department of Energy under Contract DE-AC03-76SF00515.

REFERENCES

[1] A.S. Johnson, "Java Analysis Studio", proceedings of the International Conference on Computing in High Energy Physics (CHEP 98), SLAC-PUB-7963. http://jas.freehep.org/documentation/Chep98/Chep98.htm

[2] A.S. Johnson, "Java Analysis Studio", proceedings of the International Conference on Computing in High Energy Physics (CHEP 2000), http://jas.freehep.org/Writeup/CHEP2000/JAS%20Chep%202000.htm

[3] The JAS Home page is at http://jas.freehep.org/

[4] Ronan, M.T., et. al., “Java Analysis Studio and the hep.lcd Class Library” in Proceedings of the Worldwide Study on Physics and Experiments with Future Linear e+e- Colliders, edited by E. Fernandez and A. Pacheco, Universitat Autonoma de Barcelona, Servei de Publicacions, 08193 Bellaterra (Barcelona). Spain, 1999, pp. 738-744.

[5] Cassell, R., “Full LCD Detector Simulation with GISMO” in these proceedings.

[6] Walkowiak, W., “Fast simulation and topological vertex finding in Java”, in these proceedings.

[7] Iwasaki, M., “LCD Root Simulation and Analysis Tools”, in these proceedings

[8] The WIRED home page is at http://wired.cern.ch/

[9] The LCD/Java home page is at http://www-sldnt.slac.stanford.edu/jas/documentation/lcd/