

## Future DØ Graphics

Sharon Hagopian, Saul Youssef

*Florida State University  
Tallahassee, Florida, USA*

Michael Shupe

*University of Arizona  
Tucson, Arizona, USA*

Norman Graf

*Brookhaven National Laboratory  
Upton, New York, USA*

Nobuaki Oshima

*Fermi National Laboratory  
Batavia Illinois, USA*

David L. Adams

*Rice University  
Houston, Texas, USA*

For the DØ Collaboration

The DØ Experiment at the Fermilab Tevatron Collider is preparing for major revisions of all of its software: data structures, databases, user interfaces, and graphics. We report here on the progress of the DØ Graphics Working Group, which has considered the requirements of DØ for interactive event displays and their role in the process of detector debugging and physics analysis. This report will include studies done by the group, and the evolving view of the future of DØ graphics.

Graphics is an important tool for a high energy experiment in the planning, testing, data taking, data analysis and presentation of results. For a large collider experiment such as the DØ experiment at Fermilab, the graphics software must be useful over a seven to ten year period. It must maintain stability during changes in computing platforms, operating systems, and graphics standards.

The DØ experiment is approaching a long pause in data taking between the first stage of the experiment and the upgraded experiment. A Future DØ Graphics Group was formed in Spring, 1995, and charged with several responsibilities: to review what we have learned from our graphics experiences during the first stage, to examine graphics techniques used by other experiments, to look at available and upcoming graphics standards, and to plan the graphics to be used in DØ into the

next millenium. This paper is an interim report from this group on the requirements and specifications for future  $D\emptyset$  graphics.

Event displays have the following uses for large collider experiments: prototyping in the design stage, display of testbeam data, setup and debugging of hardware, event monitoring during data acquisition, algorithm development for event reconstruction, studies of unusual events, displays of physics results, and presentations to general audiences.  $D\emptyset$  has used its current event displays in all these ways in the past and will continue to need various types of event displays in the future for all these purposes and more.

There are three event displays currently used by  $D\emptyset$  for data from the first run. (1) PIXIE is a general purpose  $D\emptyset$  event display, written by a team of  $D\emptyset$  collaborators<sup>1</sup>. It uses DI3000, which is a device-independent commercial package, available from Visual Numerics<sup>2</sup>. PIXIE has 2-dimensional displays of  $D\emptyset$  sub-detector data, which can be plotted individually or combined in overall displays, 2-D specialized plots, LEGO plots, and a primitive 3-dimensional wireframe display.

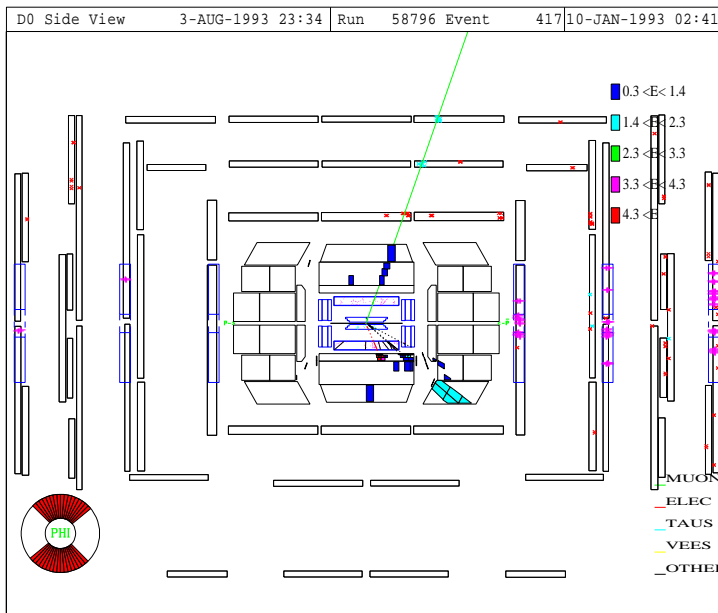


Figure 1: Side view of  $D\emptyset$  detector, showing e-mu top candidate event.

(2)  $D\emptyset$  DISP, written by Mike Shupe, University of Arizona, is a 3-D wireframe  $D\emptyset$  display which uses z-buffering and intensity and line thickness to encode the importance of objects. It is written in GL, and runs on Silicon Graphics workstations. It was used to scan top candidate events. (3)  $D\emptyset$  also uses "Ctool", a detector independent display package written by Saul Youssef of the Supercomputer Com-

putations Research Institute (SCRI)<sup>3</sup>. The CTool display combines polygonal and wireframe graphics objects constructed from items in Zebra banks with calorimetry information constructed from simple input files and items from GEANT geometries into an integrated display using the GL based SciAn general purpose visualization package, also written at SCRI<sup>4</sup>. Ctool is particularly good at displaying calorimetry and its integration with other analysis operations makes it a useful tool for detailed investigation of rare events.

Future DØ event displays should meet certain requirements. The displays should work on all the platforms used by DØ data acquisition and event reconstruction programs. The displays must be able to work under different operating systems, even on the same platforms, since the operating systems may change in time. To be of use to many different people for many different purposes, it should be possible to make displays in various frameworks and to incorporate additional user displays as needed. In general, these displays must be written in such a way that they can be easily combined and ported and will work effectively over the lifetime of the experiment. The best way to accomplish this is to have a generic graphics layer between the application code and the current graphics language. This layer can be expanded as new graphical features appear and it can be rewritten as new graphical languages become the standard. A mixture of general purpose event displays, special purpose displays, and displays on high-end graphics workstations should be developed to meet the various needs of the experiment.

Displays in 2-D should be supported and maintained in future DØ event displays. These displays are usually easier to understand and interpret than 3-D displays. But care should be given to incorporating useful features in these displays. The ALEPH display has many examples of such useful features<sup>5</sup>. These include transformations such as fisheye or “V” plots, which give easier pattern recognition or more physics insight. ALEPH displays also use rubberbanding to select viewing areas for multiple windows or for spatial measurements. Stylistic features which are also desirable in 2-D displays include using solid filled areas, using specialized summary display plots together with more detailed plots of specific detectors, and LEGO plots with special features.

Future displays should have modern 3-D graphics capabilities including fast 3-D manipulation, hidden line and surface removal, solid body rendering, shading and light effects, and translucency and transparency effects. There are also stylistic features which are desirable in 3-D displays. In order not to overwhelm the viewer with too much information, some selection criteria should be used in the choice of information displayed. This can be done by using cut planes to eliminate information outside a rectangle, by choosing the viewpoint under user control, or by increasing the amount of information as one zooms in for higher resolution. Intensity or line thickness can also be used to indicate the importance of objects. Since event displays may be used on a variety of workstations with different capabilities, being able to interactively choose the type of modeling is desirable, wire-frame pictures on slower workstations, and solids with variable numbers of facets on faster stations. Double frame buffering should also be used to keep the user from having to wait for the display to be drawn.

Displays should also have highly interactive user control. It should be possible

to start a viewing session with predefined choices of displays, interactively turn on/off subdetectors and types of information displayed, interactively rotate and zoom where appropriate, construct combined displays from individual subdetector displays or predefined displays, have different displays in different zones of the screen, and to save display choices and viewing parameters for use in future sessions.

Picking with a cursor is also an important modern graphics feature. The user should be able to pick objects for additional text information and to pick objects for additional graphical information. Picking can also be used for viewpoint manipulation, for marking boundaries of zoom, and to mark the place for text annotation of displays.

In order to satisfy these general requirements, the event display software should meet certain design specifications. The routines should be modular as to function. This includes well defined geometry access and well defined event data access. Separate application routines for each type of object to be displayed will also make it easier to control the visibility of objects interactively. In order to make changes in graphics standards possible over the long timescale of the experiment, graphics routines that draw and set attributes should be isolated, and the rendering and view manipulation routines should be separate from the drawing routines. The program should be able to produce high resolution display output files in either black/white or color and in various formats such as postscript and GIF for World Wide Web.

Future DØ event display programs must also satisfy certain interface constraints. They must be able to get information from DØ geometry databases and DØ run information databases. They must be able to read in and get information from DØ event data structures. They should also use the DØ menu system, where appropriate.

Current graphical standards being considered include OpenGL, as a graphics language<sup>6</sup>. OpenGL is a relatively new industry standard which evolved from Silicon Graphics GL language. The specification of OpenGL is controlled by an industry consortium, the OpenGL Architectural Review Board. OpenGL has over 120 commands which specify objects and operations. It uses 3-D graphics and works on many platforms, including SGI, DEC ALPHAs, SUNs and many PCs. There are several associated libraries which augment the basic OpenGL features. OpenGL Utility Library (GLU) contains routines for fixing view orientations and for making projections. OpenGL X-windows extensions (GLX) helps make OpenGL more useful in an X-windows environment. It contains routines which associate OpenGL commands with specific X-windows, allows user inputs from X-windows, and supports the client-server display list model which makes OpenGL very efficient in networked applications. The OpenGL Utility Toolkit, (GLUT)<sup>7</sup>, provides for multiple X-windows, and allows for various inputs and fonts. It also supports bitmaps.

DØ is also considering EXPLORER as a framework for developing graphics applications<sup>8</sup>. EXPLORER was written for Silicon Graphics workstations but has been ported to several platforms, such as SUN and IBM, and is available commercially<sup>9</sup>. It is currently being used by the DØ Fiber Tracker Upgrade group. (See talk by Jeff Kallenbach).

Event displays that require interactive access to data banks usually must be

directly linked to these banks. But specialized event displays or pipelined displays may use intermediate graphics data files. The OpenInventor file format is a popular ASCII format for such files. This format was originally written for Silicon Graphics workstations, but is now public domain. It has true 3-D information stored in a hierarchical object-oriented data structure. This format is being considered for use in DØ graphics, since many commercial and public domain viewing packages accept OpenInventor input.

In summary, future DØ event displays must be portable to different platforms. They must be able to work under various operating systems. The displays should have modular application and graphics routines. They must be able to change as new graphics standards and features become useful. Event display packages and a generic graphics interface need to be written to meet these requirements. A mix of general purpose event displays, specialized application dependent displays and high-end graphics displays can meet the various future graphics needs for DØ, over a seven to ten year time span. Developing this software will not be an easy task, and the final result will necessarily involve some compromises. We have our work cut out for us.

## References

1. PIXIE graphics group: S. Hagopian, head, R. Avery, J. Bantly, S. Blessing, O. Callot, J. Glicenstein, P. Grudberg, G. Howell, T. Krammer, Q. Li-Demarteau, N. Oshima, H. Prosper, T. Trippe
2. Visual Numerics, 6230 Lookout Road, Boulder, Colorado, 803001, USA  
WWW URL: <http://www.vni.com>
3. Detector Independent Event Displays (mostly of DØ), S. Youssef, HEPVis95 Workshop, August 7-9, 1995.  
WWW URL: <http://cdibm.fnal.gov/workshop/talks>
4. Sci-An, E. Pepke, HEPVis95 Workshop, August 7-9, 1995. WWW URL: <http://www.scri.fsu.edu/mimi/scian.html>
5. "Is there a Future for Event Display?", H. Drevermann, D. Kuhn, B.S. Nilsson, Proceedings of the 1992 Cern School of Computing, L. Aquila, Italy, September, 1992; WWW URL: <http://cdibm.fnal.gov/workshop/talks>
6. The OpenGL Technical Library, Addison-Wesley, Reading, Massachusetts, 1992. WWW URL: <http://www.sgi.com/Technology/OpenGL>
7. "An OpenGL Toolkit", M. Kilgard, The X Journal, SIGS Publications, Nov/Dec 1994;  
WWW URL: <ftp:sgigate.sgi.com/pub/opengl/xjournal/GLUT>
8. "IRIS Explorer: A Framework for Investigation", D. Foulser, Computer Graphics May, 1995;  
WWW URL: [http://www.sgi.com/Products/appsdirectory.dir/Applications/Visualization\\_Tools/ApplicationNumber6296.html](http://www.sgi.com/Products/appsdirectory.dir/Applications/Visualization_Tools/ApplicationNumber6296.html)
9. Numerical Algorithms Group, Inc., 1400 Opus Place, Suite 200, Downers Grove, Il, 60515-2337; WWW URL: <http://www.nag.com/ICE>
10. Open Inventor C++ Reference Manual, Addison-Wesley, Reading, Massachusetts, 1993.