The Hepvis Class Library for Event Visualization

George Alverson [b], Amber Boehnlein [c], Joseph Boudreau [a], Xiaoling Fan [d], Lucas Taylor [b], Jeff Kallenbach [c]

[a] *University of Pittsburgh, Pittsburgh, PA*

[b] *Northeastern University, Boston, MA*

[c] *Fermi National Accelerator Laboratory, Batavia Il*

[d] *Northwestern University, Evanston Il*

The Hepvis class library is a set of HEP-specific classes derived from the Open Inventor toolkit for the purpose of building high-quality, economical, interactive event displays. Inventor classes encapsulate not only the geometry of the objects comprising the scene, but also much of the interaction between the user and the scene. The Hepvis library extends Open Inventor with geometrical objects representing parts of a particle physics detector and physics objects found therein, such as tracks and hits. It constitutes a palette of common shapes, and aims to be extensible rather than exhaustive. It is being ported to a variety of Unix platforms as well as Windows NT, and developed for all levels of graphics acceleration, from PC to VR.

Key words: Visualization; C++; Open Inventor; Virtual Reality; Event Display.

1 Requirements of an Event Display Program

The problem of event display, which arises in every major collider experiment, has one aspect which appears unique to particle physics. The complexity a physics event is so great, that that the display program can only show a small subsample of the event data at a time. Otherwise, it would overwhelm not only the graphics system, but also the user's ability to comprehend the event[1]. Instead, the event display serves as an index into the event. The user interacts with the physics objects through the graphical objects. Usually, the purpose of his interaction is information retrieval, but it can also be to modify the physics in some way; e.g., to actively realign detector elements, to combine objects like tracks into composite objects, like vertices.

We distinguish two kinds of interactivity: local and global[2]. Local interactivity means simply a change of viewpoint or camera angle, or toggling the visibility of some components (e.g. muon chambers, which typically occlude the rest of the detector). Global interactivity means interacting in some way with the physics objects, which requires going off and seeking additional physics information that is not in the display. Global interactivity, by far, accounts for most of the work of constructing an event display program.

Reasonable requirements for an event display program are that it be informative, multi-platform, that it have powerful, professional, user-friendly local and global interactivity, and that it be extensible to evolve with the changing needs of an experiment over its lifetime. In the past, experiments have either used many man-years to achieve these goals, or they have fallen short of them. It is not unreasonable to estimate the cost of a fully functional event display program in excess of a million dollars.

Now that the next generation of collider experiments have adopted C++ as the programming language, an opportunity for reducing the price tag has appeared. This opportunity is to use a commercially available class library, Open Inventor[3], as the basis for event display [4]. The authors of Open Inventor foresaw extensions of the toolkit through subclassing[5]. The mission of the Hepvis project is to create a special set of extensions for high energy physics. These extensions are being ported to a variety of Unix platforms as well as Windows NT; they target all levels of graphics acceleration, from personal computers to sophisticated virtual reality installations.

2 Anatomy of a Hepvis/Open Inventor Program

Major components of an Open Inventor application include a scene, some scene objects, and a viewer. First, the scene is instantiated; then, children are added to the scene. A complete hierarchy of shapes, transformations, material properties and other attributes can be built. This hierarchy is called a "scene graph". A *viewer* contains both a render area and a user interface. The user interface gives interactive control over zoom, camera position, drawing style, and much more. Several polymorphic viewers supply various flavours of local interactivity, and all can be used to display a scene graph.

The Hepvis library provides a palette of shapes that physicists can use to construct scenes. For example, one useful shape represents a helical track. One instantiates the track, then sets its data fields to the proper value, and adds it to the scene. The data fields are familiar to physicists: they are "track parameters" in an established, five dimensional representation. If the value of a data field changes during program execution, the shape of the object in the

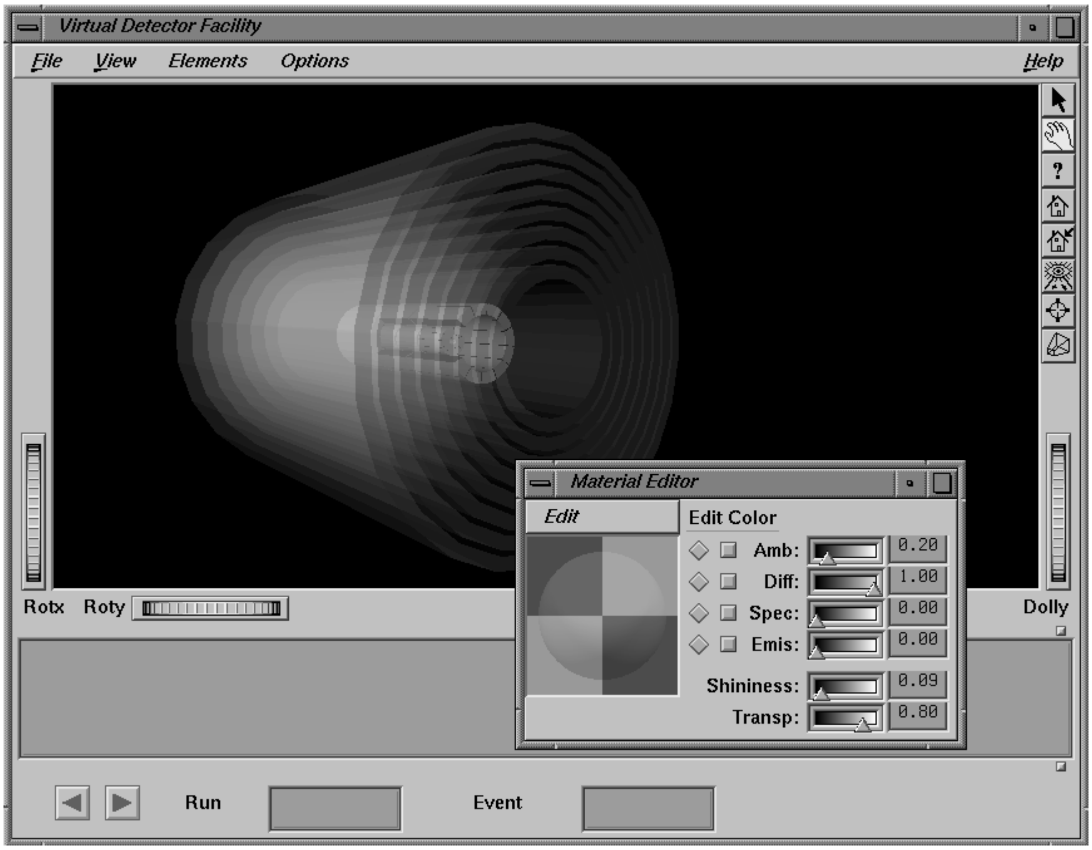

Fig. 1. An application making use of DetectorTreeKit, HepEventViewer, and several Hepvis shape classes. The user has expanded several levels of detail interactively, and is now changing the transparency of a central tracker.

render area changes as well. An effort has been made to keep physics behaviour out of the Hepvis objects, which are intended *only* for interactive display.

Smart nodes, derived from Inventor node kits, are also part of Hepvis. They can handle X-events. One such "smart node" represents a silicon strip detector and its digitizations. The pulse height profile is histogrammed on the surface of the wafer. Clicking on a strip displays the pulse height; Alt-MB1 allows interactive rescaling of the histogram using an Inventor manipulator; and Alt-MB2 allow interactive wafer realignment (if enabled by the application).

Another useful smart node, called a DetectorTreeKit, allows the application programmer to easily build up a hierarchy of shapes. Users can navigate this hierarchy by clicking on shapes; in response to this interaction each layer of the hierarchy is either displayed or hidden. Users can also interactively modify the material properties of each layer interactively, even making certain detector elements transparent, as in Figure 1. The DetectorTreeKit augments the local interactivity already provided by the Inventor viewers.

The Open Inventor toolkit, and its Hepvis extensions, effectively solve the problem of local interactivity. Global interactivity must be implemented by the application programmer, but we can at least suggest a mechanism: first, one uses the Open Inventor SoSelection node, inserted near the top of the scene graph, to manage a list of selected objects. A set of callback functions can be maintained by the SoSelection node; they are invoked whenever the node handles a selection event. In the callback, the application programmer has access to the address of the picked object. He can use a map, hash table, or dictionary to map the graphical object, flexibly, into some physics object. This is pure expression in C++ of the idea of an *index* into the physics.

Currently, the Hepvis library contains the following objects: **shapes:** ellipsoids, elliptical prisms, ellipsoid segments, silicon strips, reconstructed tracks, vertex constrained tracks, and helical tracks; also the GEANT geometry primitives Box, Tubs, Trap, Trd, and Cons; **smart nodes:** DetectorTreeKit, single-sided, double-sided, and double-sided low angle stereo silicon strip detectors; **viewers:** an event viewer containing event filtering, multiple views, viewing style selection, a text window, and commands for printing and 2D or 3D capturing. In the near future it is foreseen to add additional viewers, drift chamber classes, calorimetry classes, and classes for the visualization of vector fields. Hepvis is an open library, and contributions from any party are welcome. For general guidelines on how to contribute, please contact the authors.

3 Conclusion

The Hepvis extensions to the Open Inventor class library can be obtained from reference [6]. This library has reached maturity, and development continues. It is not too optimistic to think that by CHEP '98, a variety of Hepvis-based applications will demonstrate the benefits of this approach.

References

[1] H. Drevermann, "Is there a Future for Event Display?", Proceedings, 1992 CERN school of computing, L'Aquila, Italy, 30 Aug - 12 Sep 1992

[2] H. Stone, "Event Visualization in High Energy Physics", Proceedings, Hepvis '96 Workshop on HEP Visualization, CERN, Geneva, Switzerland, Sep 2-4 1996

[3] J. Wernecke and the Open Inventor Archtecture Group, *The Inventor Mentor*, Addison-Wesley 1993

[4] J. Boudreau, "Experiences with Open Inventor", Proceedings, Hepvis '96 Workshop on HEP Visualization, CERN, Geneva, Switzerland, Sep 2-4, 1996.

[5] J. Wernecke and the Open Inventor Archtecture Group, *The Inventor Toolmaker*, Addison-Wesley 1994

[6] http://cactus.phyast.pitt.edu/ joe/hepvis/hepvis.html, and links therein.