A Novel Approach to Designing an Event Display

C.D. Jones [a], P.R. Avery [a]

[a] *Department of Physics, University of Florida, Gainesville, FL 32611 USA*

The goal of the **CLEO3D** event display project is to create an interactive *information display* which is easy to use, displays the full range of CLEO data, and can be easily extended and maintained. Contrary to the approach taken by other experiments of putting in high level graphics early, we have concentrated on the system design to provide important additional features, including non-graphical hierarchically organized lists. To ensure that the system really is useful to physicists, we have adopted a unique "user centric" approach. To test the design, we have created a functioning prototype and observed physicists as they attempt to use it.

Key words: CHEP97; Event display; graphics; CLEO; CLEO3D

1 Introduction

In designing the event display for the CLEO III detector, we concentrated on how the user interacts with the data rather than on how to graphically display the event on high end graphics workstations. In addition to this "user centric" design methodology we also required that the system be easy to extend in order to accommodate the inevitable user requested additions. Once we created an initial design, we built a prototype, called **CLEO3D** [1], which we test by having physicists use the program. The information learned by watching and questioning physicists is used to modify the design.

The idea of user centric design is wonderfully explained in *The Design of Everyday Things* [2]. The element of design we are most concerned with in this paper is the *conceptual model*. Explanations of the other elements of user centric design can be found in [3]. A *conceptual model* is the user's mental model of the system. A good conceptual model builds on concepts the user already understands and matches the actual internal model of the system.

2 System Design

The system design was driven by the two major goals of ease of use and extensibility. We began by determining the conceptual model that a physicist uses when he thinks about an event. We want our system design to reflect the physicist's mental model. To do that we first needed to identify objects in an event that are used by a physicist and then determined how she interacts with them. Once we understood what a physicist expects from an event display, we concentrated on how we could make the system extensible.

Instead of looking for specific objects that are used within an event, we started by finding groups of related objects. We grouped objects into categories based on similarity of purpose and life expectancy within the program. These categories are *detector components*, *response*, *reconstruction*, *analysis*, *Monte Carlo*, and *user defined*. Detector components are the physical and logical detector parts, for example the drift chamber end-plate. The response category contains all the objects associated with the response of the detector to the presence of a particle, e.g. hits in the drift chamber. Reconstruction objects are the results of pattern recognition applied to the response items e.g., tracks. The analysis objects are the objects of primary interest to a physicist, e.g. a pion. The Monte Carlo category contains objects which are generated by the simulation software. The final category is user defined objects such as a sphere placed at a spatial point of interest. These categories are not independent of one another. A pion found in an analysis is inferred by the presence of a track. That track is created by fitting a group of hits. The position of those hits is determined by the placement of the wires in the drift chamber. A user object may be associated with any of the previous objects.

Physicists will want to interact with these objects in many different ways: decide which objects are visible or invisible; choose the color of an object; examine the data associated with an object, e.g. energy of a kaon; create new objects, e.g. make a kaon from a track; determine the relationship between objects, e.g. which hits go with which track; and display an object in a variety of representations, e.g. display a track as a helix or as a vector. To interact with objects, the physicist picks the objects of interest and then applies the action. For instance a physicist could pick all showers with energy less than 200 MeV and then make them invisible.

To accommodate a physicist's wish to view an object in many different ways (e.g. show a track as a helix or as a list of data) our system model is designed around three concepts: Entity, Model and View. An Entity is an HEP object, such as a track. A Model is a representation of an Entity, such as a vector. A View is an image of the Model, including more abstract displays such as text spreadsheets and hierarchical lists of components. Every View must be able to support user selection of Entities and must highlight selected Entities. The core of the system, which takes care of the relationships between Entities, Models, and Views, and user interactions, is experiment independent. This core framework, called **Spectator**, interacts with each Model through an abstract interface which allows us to easily add new Models.

3 User Testing

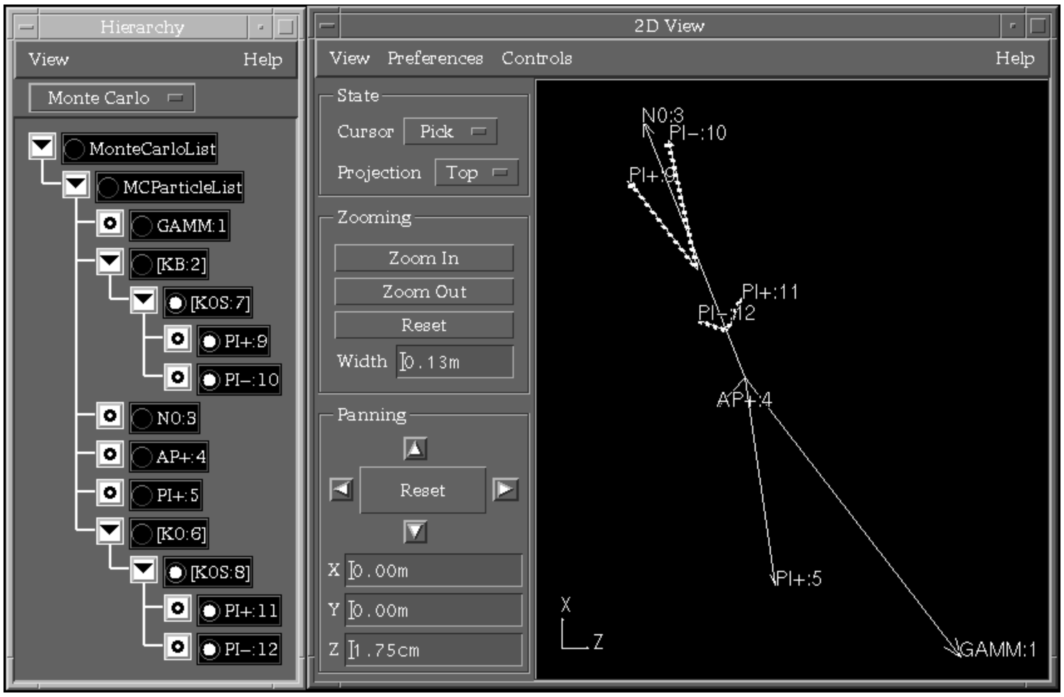

Fig. 1. Picture of the Hierarchy and 2D view both only showing Monte Carlo particles. The 2D view represents the Monte Carlo particles as vectors labeled with their names. To simplify the image, the kaons are invisible. The views show that the K_s^0's and their daughters have been selected.

We have written a design prototype to test the usability of our design. The prototype has a limited number of Entities (a few detector components, tracks, showers, and Monte Carlo particles), two user actions (set the color and visibility of an Entity), and two Models (a hierarchical list and a 2D graphical display shown in Figure 1). In addition, a track, shower or particle can be represented by a path, momentum vector, or point in the 2D display and they can be labeled with any data associated with that Entity (E.g. a track's reference number or x value of its momentum). Entities may be picked by either clicking on their name in the hierarchical view or clicking on their image in the 2D display. The prototype is written in C++ and runs under UNIX using Motif on DEC Alphas and IBM RS6000s. We use the OpenGL API to render our graphics and the Mesa library to translate OpenGL calls to X-windows.

We designed our motif graphical user interfaces using X-Designer [4].

We test the prototype by watching physicists try to use the prototype. While watching them, we ask them to perform simple task (make track 1 blue) and if an action does not give the desired result, we ask them what they expected would happen. So far we have determined that physicists are comfortable with our choice of objects but it is too early to tell if they will accept the Entity-Model-View concept. We were surprised at how useful the hierarchical view has become. The hierarchy nicely shows the parent-child relationship between Entities (e.g. the π^- is the child of the K_s^0) and allows the user to easily identify and select Entities. Our biggest difficulty has been that a first time user has a very difficult time deciding what actions he can perform. We hope that changes to our user interface will relieve this problem.

4 Conclusion

Future releases of **CLEO3D** will support 3-D views, visualize hits, allow flexible on-screen labeling, permit users to add their own information, incorporate a command line interface and, using a spreadsheet-like display, view any information associated with an object, e.g. track parameters, magnet dimensions, etc. After several iterations of design and user scrutiny, we have a prototype version of **CLEO3D** which has proved highly useful to CLEO.

This work is funded by the Department of Energy, grant DEFG0586ER40272.

References

[1] CLEO3D Home Page, http://www.phys.ufl.edu/~hepvis/cleo3d.html.

[2] D.A. Norman, *Design of Everyday Things*, Doubleday, New York (1988).

[3] C.D. Jones, *The Cleo3D Event Display*, HEPVis96 Conference Proceedings, http://www.phys.ufl.edu/~cdj

[4] OpenGL Home Page, http://www.sgi.com/Technology/openGL/
Mesa 3D Library Home Page, http://www.ssec.wisc.edu/~brianp/Mesa.html
X Designer Home Page, http://www.ist.co.uk/xd/index.html