any WWW user to "fly through" the LHC Virtual Prototypes. The browser is called "i3D", it has been produced within a Joint Project between CERN and the Italian institute CRS4, directed by Prof. Carlo Rubbia, and it will shortly be available on the Public Domain. In its present version i3D is available only on Silicon Graphics stations, but it will soon be ported to all major UNIX platforms supporting OpenGL. It supports navigation through a 2D mouse or a Spaceball, and by the summer '95 it will support VRML (Virtual Reality Markup Language), stereo vision and shutter glasses for on-screen 3D imaging. I3D is tightly linked to ordinary WWW browsers. Clicking on a "3D" icon in Mosaic or Netscape will load a 3D world in i3D. Vice-versa, clicking on an hyperlink object during a flight in i3D will trigger the load of a new page in the textual browser. VENUS intends to provide all LHC Virtual Prototypes models "on the web", in i3D format. This will allow the HEP community to fly through them and obtain all kind of information related to the virtual objects, technical drawings, pictures, schedules at a mouse click.

The talk will focus on the usage of i3D as a graphic tool for LHC design integration and show a demo film.

more info: http://www.crs4.it/ 3diadm/i3d-help/i3d-help.html

ABS_71

Future D0 Graphics

S. HAGOPIAN (FLORIDA STATE)
M. Shupe (Arizona), N. Graf (Brookhaven), N. Oshima, R. Raja (Fermilab), S. Youssef (Florida State), D. L. Adams (Rice)

The D0 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 D0 Graphics Working Group, which has considered the requirements of D0 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 D0 graphics.

ABS_52

3D Graphics Module for Modeling, Visualization and Analysis

M. KAWAGUTI (FUKUI U.)
S. Tanaka (Fukui U.)

A general-purpose 3D graphics software (to be referred to tentatively as 'UniGr' in what follows) intended for use in various scientific applications is discussed. It represents the further extension of the work reported by the same authors at the previous CHEP94 conference.

The ultimate goal of UniGr development is hopefully to realize simple and straight-forward means of manipulating and rendering 3D data typically representing spatial configurations of rather complex experimental apparatus, as well as to visualize data handily in 3D way.

The language which describes 3D objects is made quite simple so that it can appeal to instinctive human recognition of the 3D object in reasonably natural manner without much training. This aspect is at variance with many of existing CAD systems.

The modeler of the UniGr parses the data, and converts it into internal description format of the 3D objects based on the boundary representation (B-rep) scheme.

The renderer of UniGr performs hidden line/surface removal based entirely on rigorous geometrical calculations. Therefore UniGr is completely system independent, free from any hardware restrictions such as pixel resolution on the display screen. The identical renderer serves various purposes: for displaying the 3D image on the CRT screen during the interactive session, as well as for generating fine quality figures drawn in preparation for publications.

UniGr is not just for modeling and rendering of 3D objects. For any arbitrarily selected location of a 3D point, UniGr can report whether it is inside of an object or outside of all the constituent objects. If the former is the case, it further tells within which object (or object tree if some objects are situated within another object in a recursive way) it is locateed. Likewise, for a given spatial directional line, arc or helix, UniGr can identify at what location on the surface of one of the objects it hits, if any, for the first time along the trajectory. UniGr performs these analyses relying on the same algorithms used for 3D rendering.

UniGr is written mostly in C++, so that the advantages of the object-oriented approach can be exploited. All the graphical objects are treated as C++ objects.

One of the motivations for developing UniGr came from our desire for a self-contained graphics software which can be run on any available workstation,free from any restrictions.

To be sure, PHIGS is recognized as a ISO-approved international graphics standard.