Event Visualization

Harald Vogt

DESY-Zeuthen

Abstract. Event visualization is a basic tool for a detector study. An event display for a GEANT3 based detector study is presented. It reads a detector geometry definition file and a file with the detector response of generated events. The display is based on the OPACS visualization environment and provides sufficient functionality.

I INTRODUCTION

For the TESLA project two simulation tools have been developed, BRAHMS and SIMDET [1]. BRAHMS is a GEANT3 based full detector simulation, and SIMDET is a fast simulation using parametrized smearing and error distibutions (derived from BRAHMS simulations).

The results of both programs are written according to a common output structure [2]. It starts with a header record containing information about the structure of the following event records (standard or restricted). The standard event records have generator information and the detector response, the restricted records include only the detector response (Energy flow record).

The detector drawing is based on GEANT3 geometry structures which are read from a geometry file produced by GEANT3 after the TESLA detector was set up. Displaying the detector and the events generated is done using the OPACS visualization driver written in ANSI C. The detector is drawn using a wired modelling.

II THE OPACS VISUALIZATION ENVIRONMENT

OPACS [3] was developed at LAL Orsay using an object oriented design.

It is written in C and includes a widget manager (Motif/Athena). The hierarchies of widgets can be built and modified using ascii text files (.odb extension). OPACS has its own command language interpreter (osh - Bourne shell like). A simple and therefore fast 3D viewer is provided. There is however the possibility to use a more complex detector representation (rendering) because OPACS drawing primitives are based on the OpenGL library.

OPACS clear and flexible Fortran interface was decisive for our application. It is installed on a Linux PC using lesstif (Motif) [4] and Mesa (OpenGL) libraries [5].

III EVENT DISPLAY PROGRAM STRUCTURE

The following diagram shows the structure of the event display program:

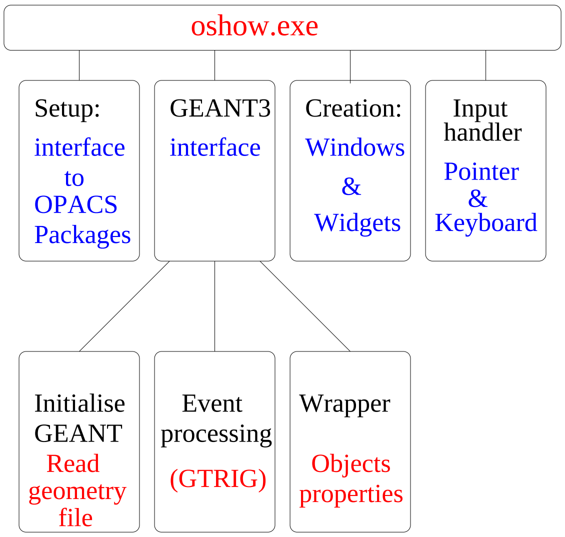

FIGURE 1. Program structure - all interfacing to Fortran is done via one C code unit G3oEXPO.c, all Fortran specific code is contained in the routines given by the second row.

IV DRAWING OF OBJECTS AND PICKING

The drawing of following objects can be performed:

- the detector geometry - wired volume shapes (VOLU)
- the generated particles tracks (MCT)
- the energy flow objects with 2 different presentations for
  - charged objects - fitted tracks (TKF)
  - neutral objects - energy towers (TOWE)

Following global information about the event is shown in the main widget: $N_{run} = ID_{(ISR)}, \quad N_{event}, \quad E_{CMS}[\mathrm{GeV}], \quad ID_{process}.$

Picking the objects above (moving the pointer to the object and push the left mouse button) the following object properties are displayed:

VOLU	properties	-	size[cm], position[cm]
MCT	properties	-	ID, p[GeV/c], ϕ[rad], θ[rad], dca[cm], z_0[cm], particle
TKF	properties	-	ID, p[GeV/c], ϕ[rad], θ[rad], dca[cm], z_0[cm], mass[GeV]
TOWE	properties	-	ID, x[cm], y[cm], z[cm], E_{tot}[GeV], E_{em}/E_{tot}

An example how this looks like is shown in figure 2.

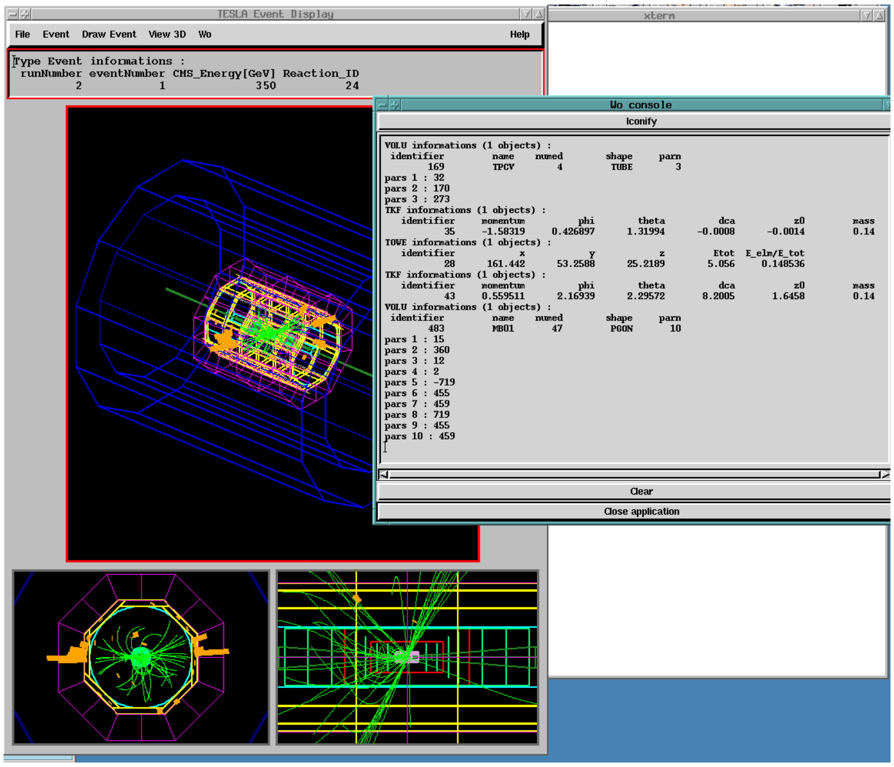

FIGURE 2. TESLA event display with properties of drawn objects (picking) in a widget called 'Wo console'.

V MENUS

At the top of the event display window there are several pull-down menus:

- the 'File' menu - with the 'GEANT' button to manipulate the VOLU objects,
- the 'Event' menu - with the 'Next Event' button to read and draw the next event,
- the 'Draw Event' menu - with buttons for drawing/erasing event objects,

- the 'View 3D' menu - with buttons to manipulate the drawings (e.g. rotate),
- the 'Wo' menu - with the 'osh' button to send a command to the application,
- the 'Help' menu - with buttons to provide specific information.

The 'Help' information is shown in a xterm window using the lynx browser to present the corresponding HTML files.

VI SUMMARY

- A TESLA event display has been developed for data written by BRAHMS or SIMDET [1].
- The detector geometry is drawn by a simple wired presentation.
- Three types of event objects have been defined that can be drawn (MCT, TKF, TOWE).
- A help facility is included.
- Widget hierarchies are defined by ascii files.

VII FUTURE DEVELOPMENTS

- OPACS has also a GEANT4 interface (C++) and is used already by the LHC experiment ALICE.
- Rendering of volumes can be done with the HEPVIS/OpenInventor/OpenGL graphics interface [6] [7].
- Using JAVA with OPACS is in progress [8].

REFERENCES

1. http://www.ifh.de/linear_collider/ (H. Vogt et al.)
2. http://www.ifh.de/linear_collider/users_guide_simdetv301.html
3. http://www.lal.in2p3.fr/OPACS/ (G. Barrand et al.)
4. http://www.lesstif.org/
5. http://www.mesa3d.org/
6. http://physics.web.cern.ch/Physics/Workshops/hepvis/welcome.html
7. http://oss.sgi.com/projects/inventor/index.html
8. G. Barrand, private communication, Nov. 2000