GEANT4 Visualization — A Report for CHEP97

John Allison [a] and Satoshi Tanaka [b]

[a] *University of Manchester, UK*

[b] *Fukui University, Japan*

GEANT4, the C++ re-write of CERN's particle physics detector simulation program, is soon to be released. This is an interim report on the visualization component. Comments and feedback are encouraged.

Key words: graphics; visualization; renderer; GEANT4; simulations; detector; C++; object oriented; OpenGL; Open Inventor; PostScript;

1 Introduction

The GEANT4 project, RD44 at CERN, was an opportunity to re-think all aspects of detector simulation — tracking algorithms, physics processes, visualization and many other areas. In visualization in particular, technological advances have been far-reaching. On the one hand this makes our task simpler, for we no longer have to support Tektronix 4014s, for example, but on the other, people's expectations have increased immensely. Fortunately, software and libraries developed outside particle physics have become more powerful and more standard.

A decision was made at the very beginning (a) to minimise the code that we ourselves had to write and (b) to exploit to the maximum existing graphics packages and libraries. We found two packages within the academic domain — DAWN [1] from Fukui University and OPACS [2] from Orsay — and negotiated the possibility of distributing them freely with GEANT4. We also noted that OpenGL and Open Inventor were becoming *de facto* standards. We assumed that all users would have access to a windows system.

Drivers exist for the four above-mentioned graphics packages — DAWN, OPACS, OpenGL and Open Inventor. The last two can be purchased for most platforms; a free client-side-implementation of OpenGL is available [3].

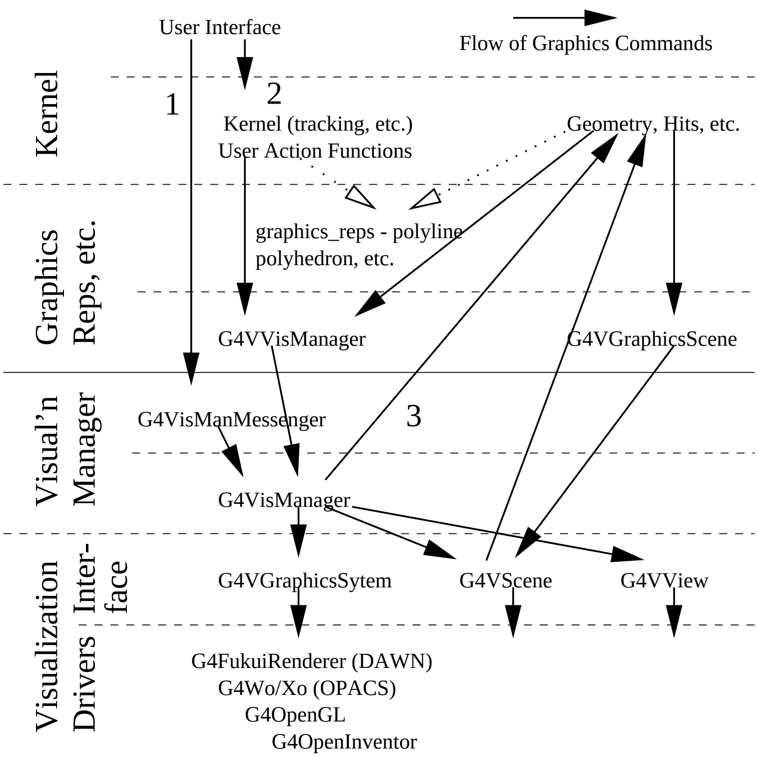

Fig. 1. The Structure of the Visualization Component of GEANT4 and the Flow of Graphics Commands. The numbers 1,2,3 refer to the items in the text.

2 Structure of the Visualization Component of GEANT4

The structure of the visualization component and the flow of graphics commands are illustrated in Figure 1. The graphics system drivers are defined by C++ classes derived from the Visualization Interface, which is a set a C++ abstract base classes. There are three ways the Interface receives drawing requests:

(i) The user issues visualization commands through the User Interface to `G4VisManMessenger`.
(ii) The user writes code which, during event simulation and particle tracking, invokes public functions of the class `G4VVisManager`, the abstract interface to the Visualization Manager. The user may make use of the GEANT4 graphics representations — polyline, polyhedron, etc. — if the drawing of the required object(s) is not supported directly.
(iii) `G4VisManager` and `G4VScene`, while interpreting (1) and (2), is free to make requests to the GEANT4 geometry, hits, etc., to draw themselves through `G4VGraphicsScene`. This route allows a sophisticated graphics system driver to render GEANT4 objects directly, and a less sophisticated one to obtain an appropriate GEANT4 graphics representation.

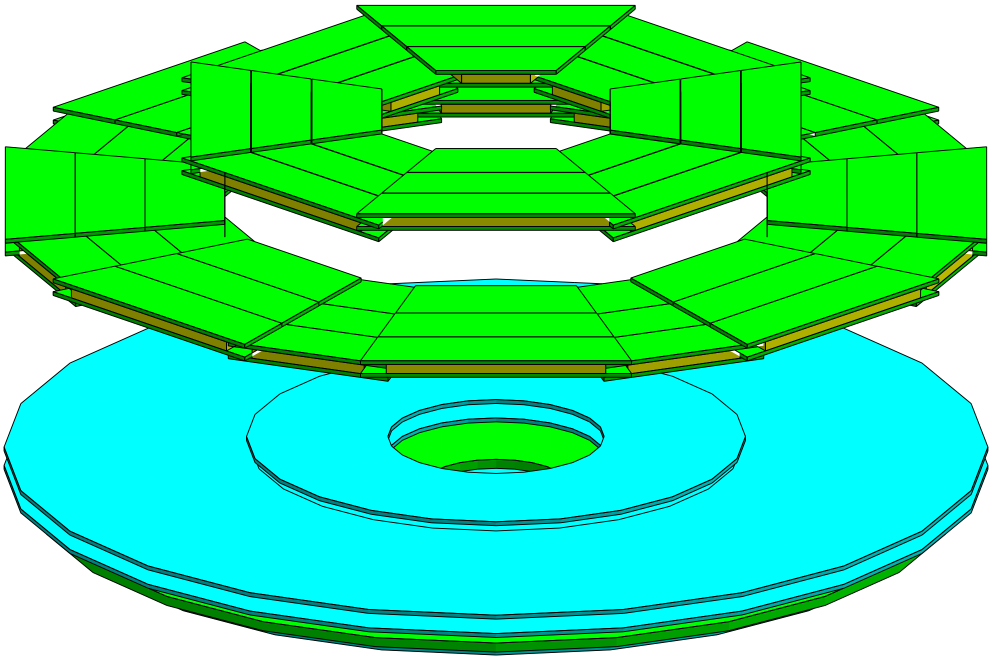

Fig. 2. Part of the ATLAS Detector modeled in GEANT4 and rendered by DAWN [1]

Thus there is a reasonable degree of flexibility and, through way (iii) above, an opportunity for maximum efficiency. Note that our graphics reps are data-holding objects only.

A User Guide [4] and a detailed Specification and Reference [5] are in preparation; drafts are available with the GEANT4 distribution.

3 The Available Graphics Drivers

The available drivers with their graphics systems complement each other — DAWN [1] produces high quality "technical" drawings and hard copy via PostScript, OPACS [2] interfaces well to X Windows and has a flexible user interface, OpenGL offers basic but fast drawing, and Open Inventor gives the user superb camera and lighting control and interactive scene editing. A rendering of part of the ATLAS detector with DAWN is shown in Figure 2.

4 Summary

A flexible structure for GEANT4 visualization is in place, which will hopefully allow a continuing development of visualization facilities.

5 Acknowledgements

We remember with fondness and acknowledge the great contribution of Minato Kawaguti, originator of the DAWN [1] rendering package from Fukui University, Japan, who was tragically killed at CHEP95 in Rio de Janeiro, and with whom we made the original negotiations and designs. One of us (ST) is honoured to be able to continue the work he started and provide the "Fukui Renderer" drivers. It is also appropriate to acknowledge the author of OPACS [2], Guy Barrand, LAL, Orsay, who has made his package freely available with the GEANT4 distribution, and has written drivers for it. Andrew Walkden, Manchester University, provided the OpenGL drivers and Jeff Kallenbach, FNAL, and Joe Boudreau, Pittsburg University, provided the Open Inventor drivers. Joe is also responsible for the "hepvis" [6] extensions to Open Inventor, also available with the GEANT4 distribution.

References

[1] DAWN (Drawing for Academic WritiNgs) or Fukui Renderer. See CHEP95. Available with the GEANT4 distribution or by anonymous ftp via `ftp://i1nws2.fuis.fukui-u.ac.jp/pub/graphics/fukui_graphics_system/`, M. Kawaguti, S. Tanaka, 3D Graphics Module for Modeling, Visualization and Analysis, Fukui University preprint, FU-IS/HEP-95-01, Fukui University, 9-1, Bunkyo 3-chome, Fukui, 910 Japan.

[2] OPACS ("O" Packages), Guy Barrand, LAL, Orsay. See HEPVIS95 and `http://www.lal.in2p3.fr/opacs/The_o_packages.html`.

[3] Mesa, a client-side-implementation of OpenGL, via anonymous ftp from `iris.ssec.wisc.edu` in `pub/Mesa`. Being client-side-implemented has the advantage that it runs over the X protocol without the need for a GLX server.

[4] Visualization in GEANT4 — An Overview and User Guide, in preparation, available with the GEANT4 releases.

[5] The GEANT4 Visualization Manager and Graphics System Interface Specification and Reference, in preparation, available with the GEANT4 releases.

[6] hepvis is a set of Open Inventor C++ classes derived from SoShape, extending the basic shapes to include some useful to High Energy Physics. It was started at HEPVIS97 (CERN, September 1996). It is being coordinated by Joe Boudreau, Pittsburg University — see Web page `http://cactus.phyast.pitt.edu/~joe/hepvis/hepvis.html`.