The “o” packages and GEANT4

Guy Barrand

LAL, Batiment 200, 91405 Orsay France ; barrand@lal.in2p3.fr

The “o” packages provide a toolkit for building interactive, graphics oriented applications. Developped at LAL Orsay, they are based on an OO design and implemented using ANSI C and X11. They contain an interactive interface builder (the Wo package), a 3D library based on OpenGL (the Go package) and a set of Xt widgets (the Xo package). They propose also a generic access to objects of the application. Being used in NA48, NEMO, KLOE event display, they are used also as drivers of the GEANT4 and BABAR visualization system. The “o” packages will be presented along their use in the GEANT4 environment.

Key words: interface; graphic; GEANT.

1 The “o” packages and HEP event display

An HEP event display is a program that visualizes results of high energy collisions in a detector and helps writing the reconstruction algorithms of the various subdetectors. An HEP event display can be logically split in five software domains :

– get event : to fetch “events” from disk or the net.
– selection : to filter data to be visualized.
– representation : to build a geometric modeling of filtered data.
– graphic : to bring representations to the screen (or on paper).
– interface : to handle interactors (including graphic viewers).

See the presentations done at HEPVis 95 by the same author for more details.

The “o” packages has been developped to help implemening some of the above services. In an “o” implementation of an HEP event display, one assembles these components as follows : the interface service is realized with the “Wo” Xt widget manager. The graphic service is done with the “Go” multi-viewing graphic database. Go relies on the OpenGL complient “GLo” package to do the graphic rendering. It has been checked that Go is compatible with other OpenGL packages such as the free "Mesa" and proprietary ones such as DEC and SGI OpenGL. The XoCamera widget, found in the "Xo" package, is the Xt interface to Go, and is used in conjunction with Wo to place views in the interface of the application.

The data selection service could be realized with the "type manager" facility and the, Bourne shell like, "osh" interpreter found in the "Co" package.

Other "o" packages exists like "Ho" to handle histograming and "Xz" that contains an HPLOT Xt widget. The XoPlotter widget (found in Xo) is used in conjunction with Ho and Wo to visualize histograms and functions. This data analysis environment will be used for the NA48 monitoring. The HTML widget and WWW layer of Mosaic-2.6 has been put "as it" in the "HTMLo" and "W3o" packages. The XoMosaic widget (found in Xo) uses these packages to offer an HTML browser. It is used mainly in event display as basic support for the help of the application.

The portability is ensured by sticking to ANSI standards like ANSI C, and defacto standards like X11, PostScript, OpenGL. The only feature intensivly used and not normalised is the Go graphic database. This is due to the fact that among possible candidates (Inventor, QuickDraw3D,...) no one seems to emerge as a standard in this area. The "o" packages have been ported on most UNIX systems, a Window/NT platform having the X11 eXcursion software, and VMS. The "o" packages can be reached from the web at : http://www.lal.in2p3.fr/opacs.

2 The "o" packages and GEANT3

We are interested since a long time to GEANT. Most of the HEP experiment has a GEANT3 simulation, and then a GEANT3 description of their detector that can be stored on disk, and used as a geometry database for the visualization.

In GEANT3, the visualization package (GEANTG) was not open to external software. In particular it was not possible to reuse the volume "representations" of GEANTG in other graphic packages. For event display, we have been compelled to reimplement from scratch these representation. This job has been put in a consistent way in the "G3o" package. For example, it is used today in the event display of the NEMO collaboration.

3 The “o” packages and GEANT4

GEANT4 collaboration aims at producing a new generation of HEP experiment simulators written in C++. A subgroup, directed by John Allison at Manchester, is charged of the visualization. Roughly speaking this group tries to write an event display for GEANT4. The strategy is to implement the interface and graphic services as generic C++ classes where “drivers” could be plugged in.

At LAL two threads of work are followed. One is to provide “o” drivers to the interface and graphic services of the GEANT4 event display. In the first release of GEANT4, the interface service (G4UI classes) will have a “dumb terminal” and a Wo driver. The graphic service (G4Vis classes) will have an OpenGL (Manchester), an Inventor (Chicago), a DAWN (Fukui), an Xo (Orsay) and a Wo (Orsay) driver.

The other thread of work is to guarantee the accessibility of the GEANT4 software toward other “external” graphical packages like IRIS Explorer or the Wo generic event display.

In the first design of the visualization and interface parts, it appears that the representation service was not distinct of the graphic service. For example, it was not possible for an external software to reuse the “modeling” methods of GEANT4 volumes. A joint effort with John Allison has permitted to decouple these two services. Now the representations are handled in the “graphics_reps” classes that are only related to GEANT4 “kernel” (that handle geometry, tracking, ...). These classes are used by the “visualization” classes that implement the graphic service and could also be used by other graphic packages that have their own visualization strategy.

4 Conclusions

We have shortly presented the “o” packages. They offer in a unique way a coherent environment for handling interface and graphic, especially for HEP event display applications. They are used now in many experiments. Whilst the architecture of this pieces of software is now stable, the graphic part could evolve in the future to adapt to new coming standards for graphic database.

Being used as “drivers” in the GEANT4 visualization framework, the “o” packages are used also at LAL as an environment to evaluate the open architecture of the GEANT4 visualization domain.