Aperture Display

Wanted : flexible tool to display beamline geometries with apertures

for overall optimization and as first step for detailed, integrated (machine+detector) simulations

Ingredients and steps

1. Lattice and aperture, Run **MAD-X** and make output **tables**, 2 dim array's (tfs files)

.1 **with apertures, and optionally tracks**; in local Courant Snyder coordinates

.2 **survey coordinates** for transformation to global Euclidian coordinate system

2. Read and combine the tables, and display the geometry in Euclidian co-ordinates

around some reference point like the IP, and range, like zmax = 320m from IP

allow for transverse scale factor (x100 in example shown here

Possible tools include:

commercial : CATIA, Mathematica (BeamLineGraphicsExamples, JJ), MathLab

free : Python, GEANT4, FLUKA, BDSIM or **ROOT**

ROOT Geometry and Event Display

Well documented, easily available including source code (git co, cmake, make), libraries, executables..

Supported and updated; (CINT --> CLANG), good OS-X integration (no need for X11)

http://root.cern.ch ,

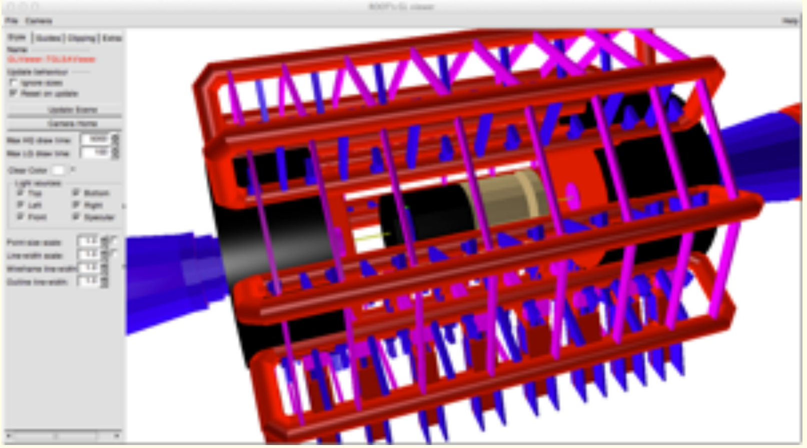
http://root.cern.ch/drupal/content/users-guide with Chapter 18 on Geometry

has standalone tracking and VMC interface to GEANT3,4 and to some extend FLUKA

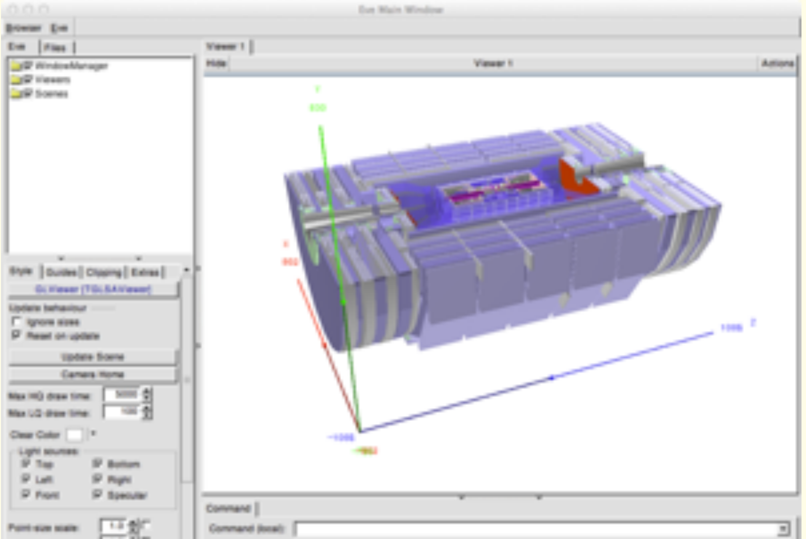
EVE, the Event Visualization Environment

and tutorials

Example sessions :



```
cd $ROOTSYS/tutorials ;  root
.x geom/geomAtlas.C
```



```
cd $ROOTSYS/tutorials/eve/ ; root
.x geom_cms_playback.C
```

ROOT geometry from MAD-X input

Define the ROOT top volume, large empty (material vacuum) cube

Read tfs optics+aperture and survey files to 2dim arrays; Combine information for elements

Loop over the elements, construct volumes and insert them in the top volume :

```
CurVol = gGeoManager->MakeTube(Name,medium,rmin,rmax,length);
TGeoTranslation trans(xpos,0,zpos);  // volume position
TGeoRotation rot; rot.RotateY(theta_survey*deg);  // CS --> Euclidian
CurPos= new TGeoHMatrix(TGeoCombiTrans(trans, rot));
top->AddNode(CurVol,1,CurPos);
gGeoManager->Export("LHC_IR_5.root");
```

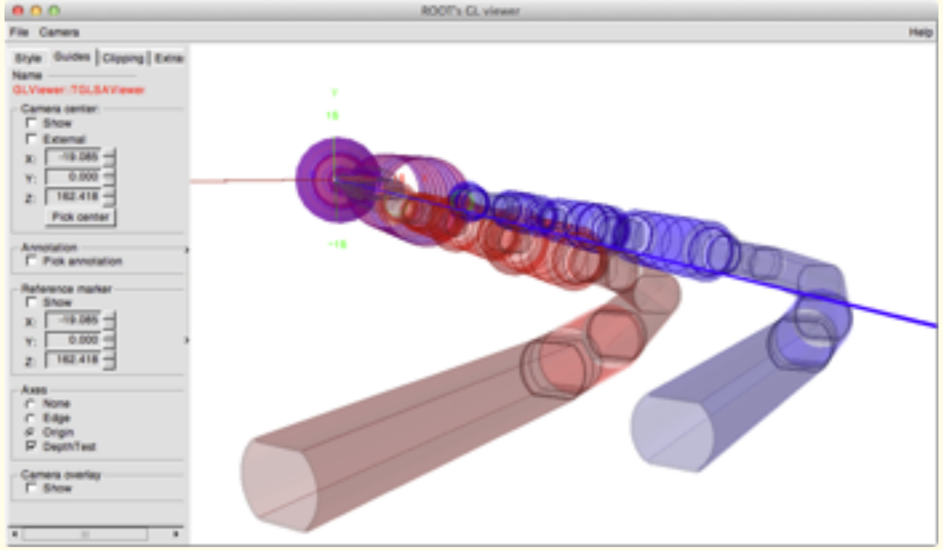
Optionally add tracks and display tracks

```
TEveManager::Create();  // create an event display, sets up global gEve
gGeoManager = gEve->GetGeometry(GeomFile);     // Load the geometry file
TEveLine* line = new TEveLine(“track_name”);   // create new track, give it a name
line->SetNextPoint(xt[i],yt[i],zt[i]);        // Add track points
line->SaveAs(“...”);      // Optionally save the track, for example in root format
gEve->AddElement(line);   // Add the line to the event manager
gEve->Redraw3D();         // draw geometry + track
```

LHC machine aperture display

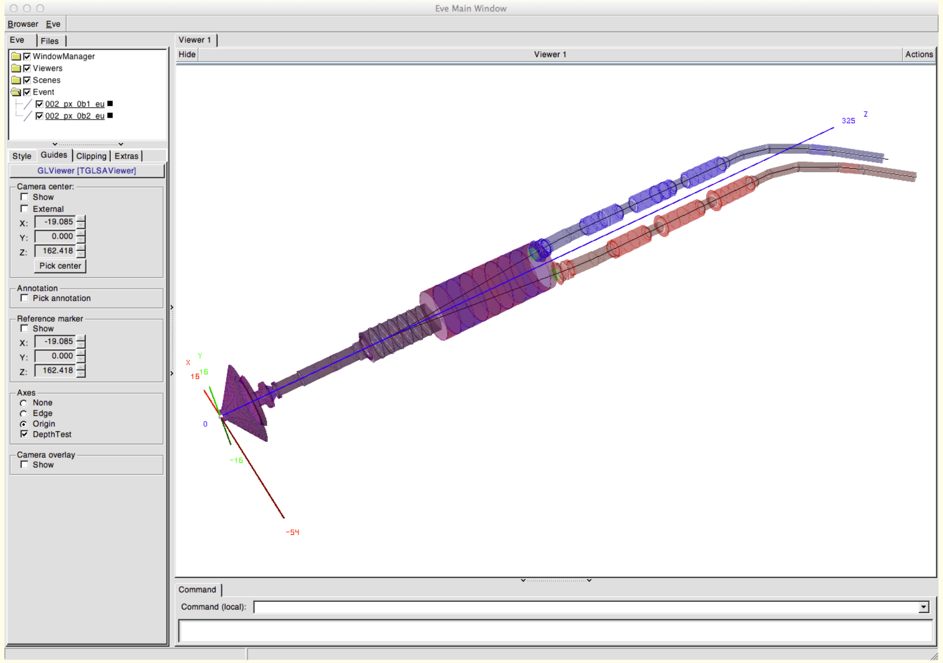
root

```
TGeoManager::Import("http://hbu.web.cern.ch/hbu/Geom/LHC_IR_5.root");
gGeoManager->GetTopVolume()->Draw("ogl");
```



Name
GLViewer::TGLSAViewer
Camera center
□ Show
□ External
X: -19.085
Y: 0.000
Z: 162.418
Pick center
Annotation
□ Pick annotation
Reference marker
□ Show
X: -19.085
Y: 0.000
Z: 162.418
Axes
○ None
○ Edge
● Origin
☑ DepthTest
Camera overlay
□ Show

With event (track) display



☑ WindowManager
☑ Viewers
☑ Scenes
☑ Event
☑ 002_px_0b1_eu
☑ 002_px_0b2_eu

GLViewer [TGLSAViewer]
Camera center:
□ Show
□ External
X: -19.085
Y: 0.000
Z: 162.418
Pick center
Annotation
□ Pick annotation
Reference marker
□ Show
X: -19.085
Y: 0.000
Z: 162.418
Axes
○ None
○ Edge
● Origin
☑ DepthTest
Camera overlay
□ Show

Command (local):

```
Run my demo
http://hbu.web.cern.ch/hbu/Geom/my_Geom_Eve_Display.C
root
.x my_Geom_Eve_Display.C
```