

# Detector Simulation Visualisation

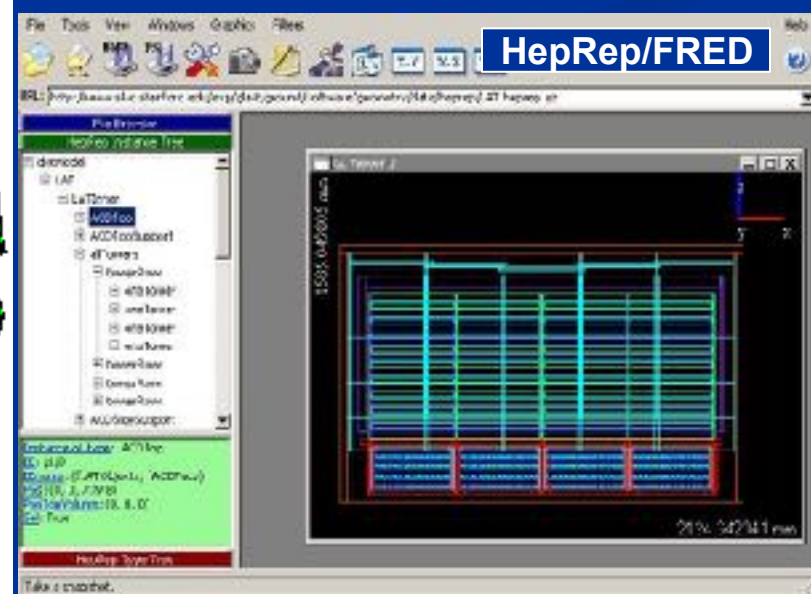
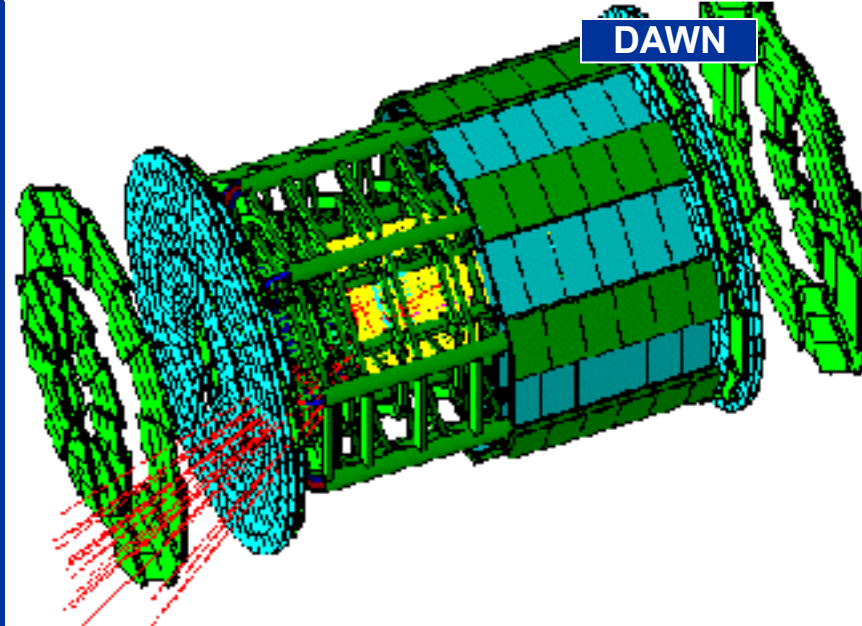
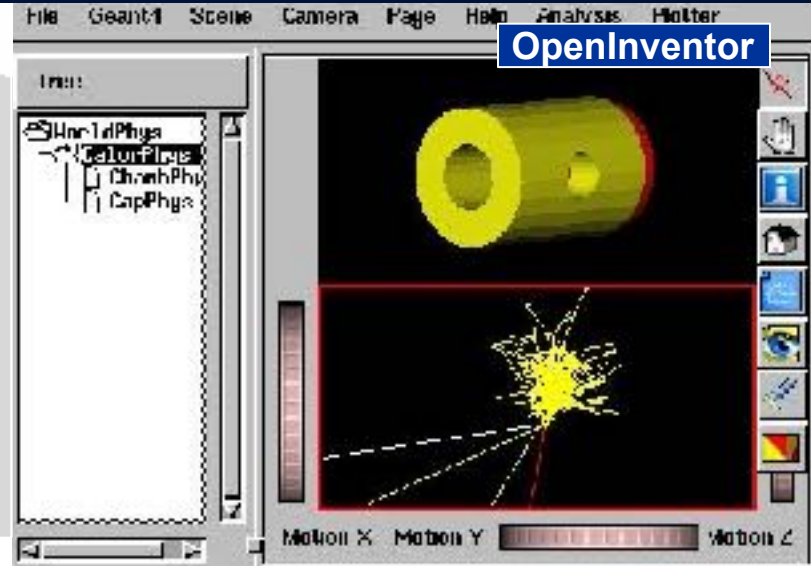
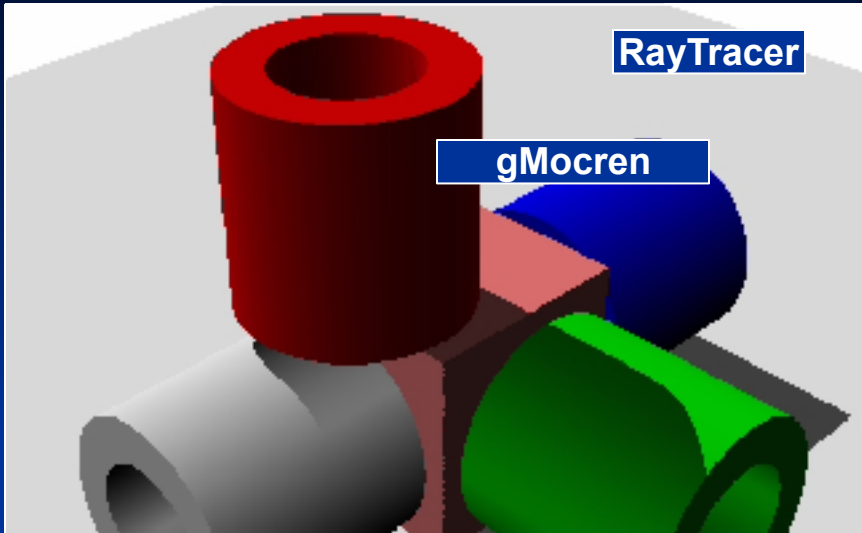
Witek Pokorski  
Alberto Ribon  
CERN

12-13.02.2018

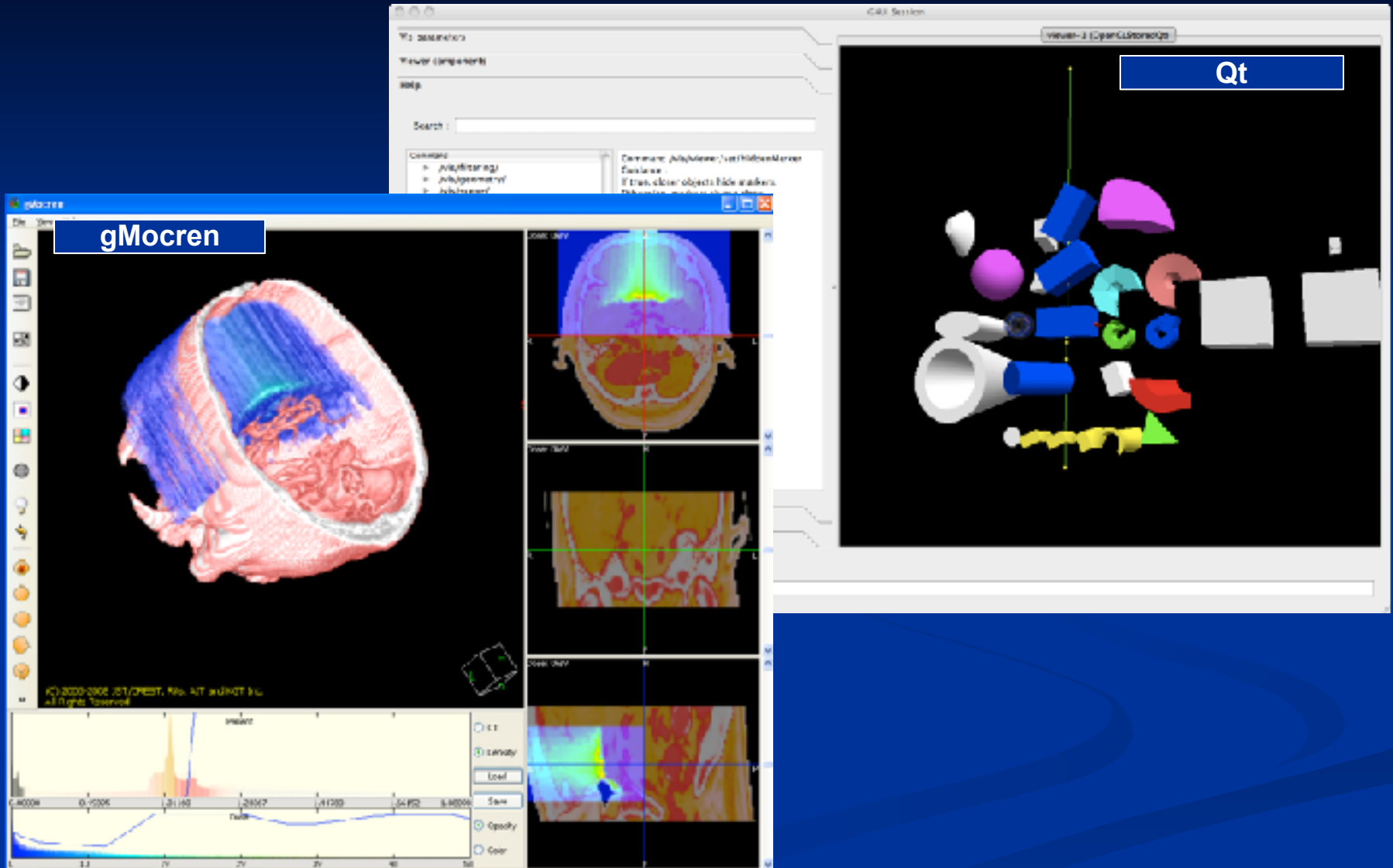




# Introduction to Geant4 Visualization



# Introduction to Geant4 Visualization



# How this Document Fits with Other Tutorial Materials

- This presentation can be used on its own, but gives the most comprehensive introduction to Geant4 visualization when used as part of the following full set of documents:
  - [Introduction to Geant4 Visualization](#)
  - [Geant4 Installation Guides](#)
  - [Geant4 Visualization Tutorial using the HepRApp HepRep Browser](#)
  - [Geant4 Visualization Tutorial using the DAWN Event Display](#)
  - Geant4 Visualization Tutorial using the OpenGL Event Display
  - Geant4 Visualization Commands
  - Geant4 Advanced Visualization
  - See the URLs at the end of this presentation
- This presentation discusses eight visualization drivers:
  - OpenGL
  - OpenInventor
  - HepRep
  - DAWN
  - VRML
  - RayTracer
  - gMocren
  - ASCII Tree





# What Can be Visualized

- Simulation data can be visualized:
  - Geometrical components
  - Particle trajectories and tracking steps
  - Hits of particles in the geometry
  - Scored energy, dose, etc.
- Other user defined objects can be visualized:
  - Polylines
    - such as coordinate axes
  - 3D Markers
    - such as eye guides
  - Text
    - descriptive character strings
    - comments or titles

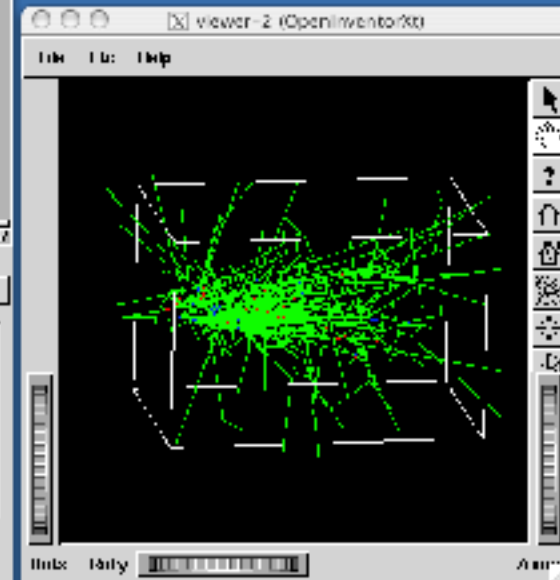
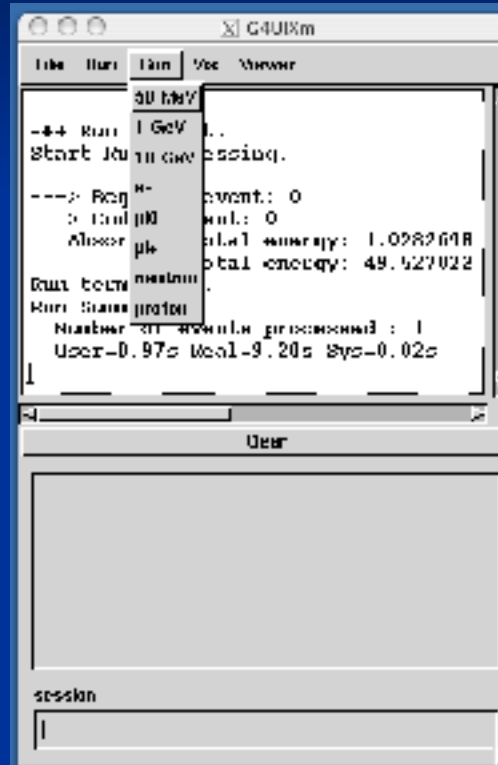
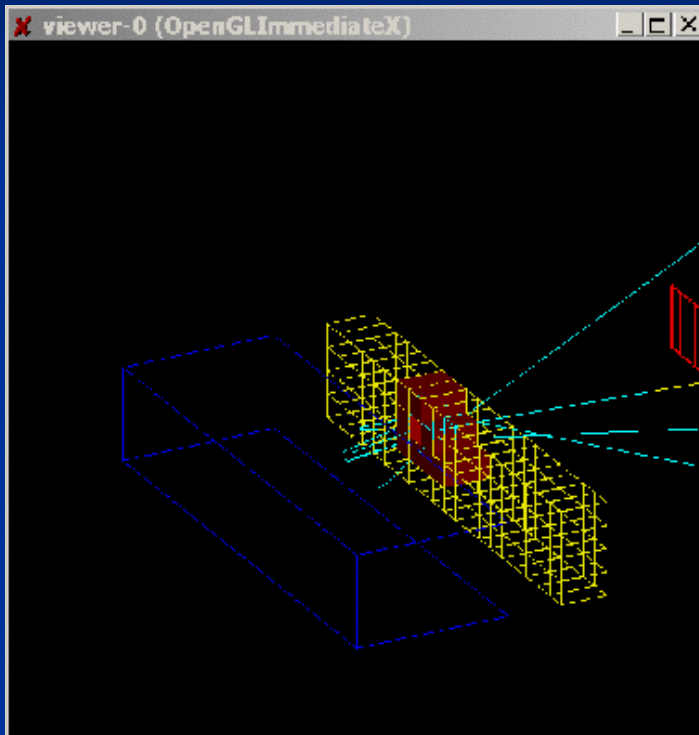
# Quick Overview of Visualization Driver Choices

- First I'll explain why there are so many visualization driver choices
- Then I'll describe each visualization driver in more detail



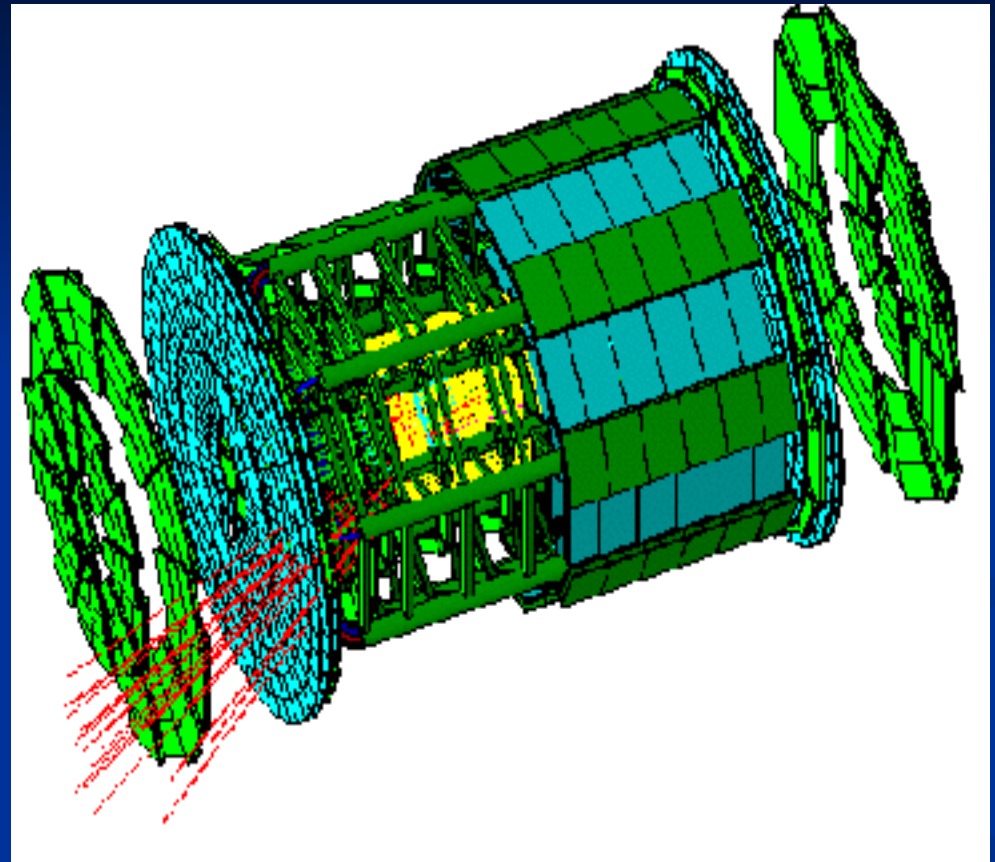
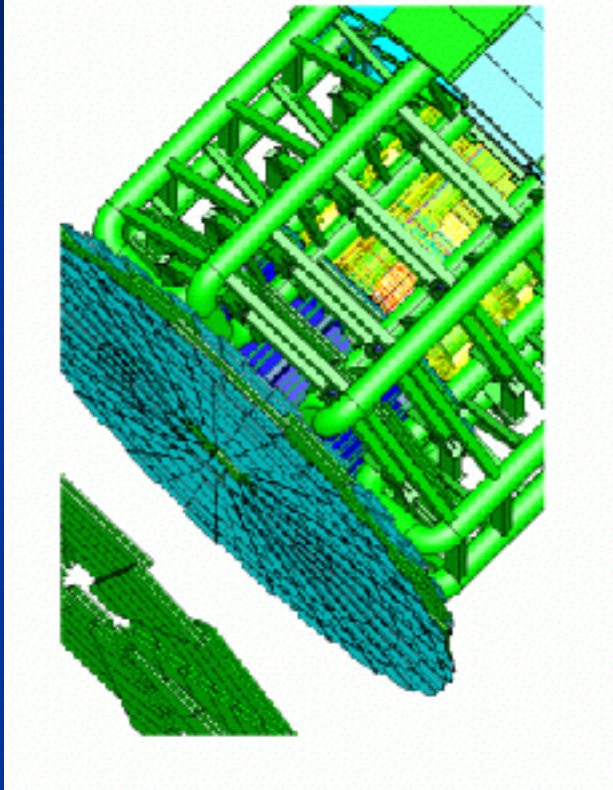
# Quick response with flexible camera control to study geometries, trajectories and hits

- OpenGL, Qt and OpenInventor



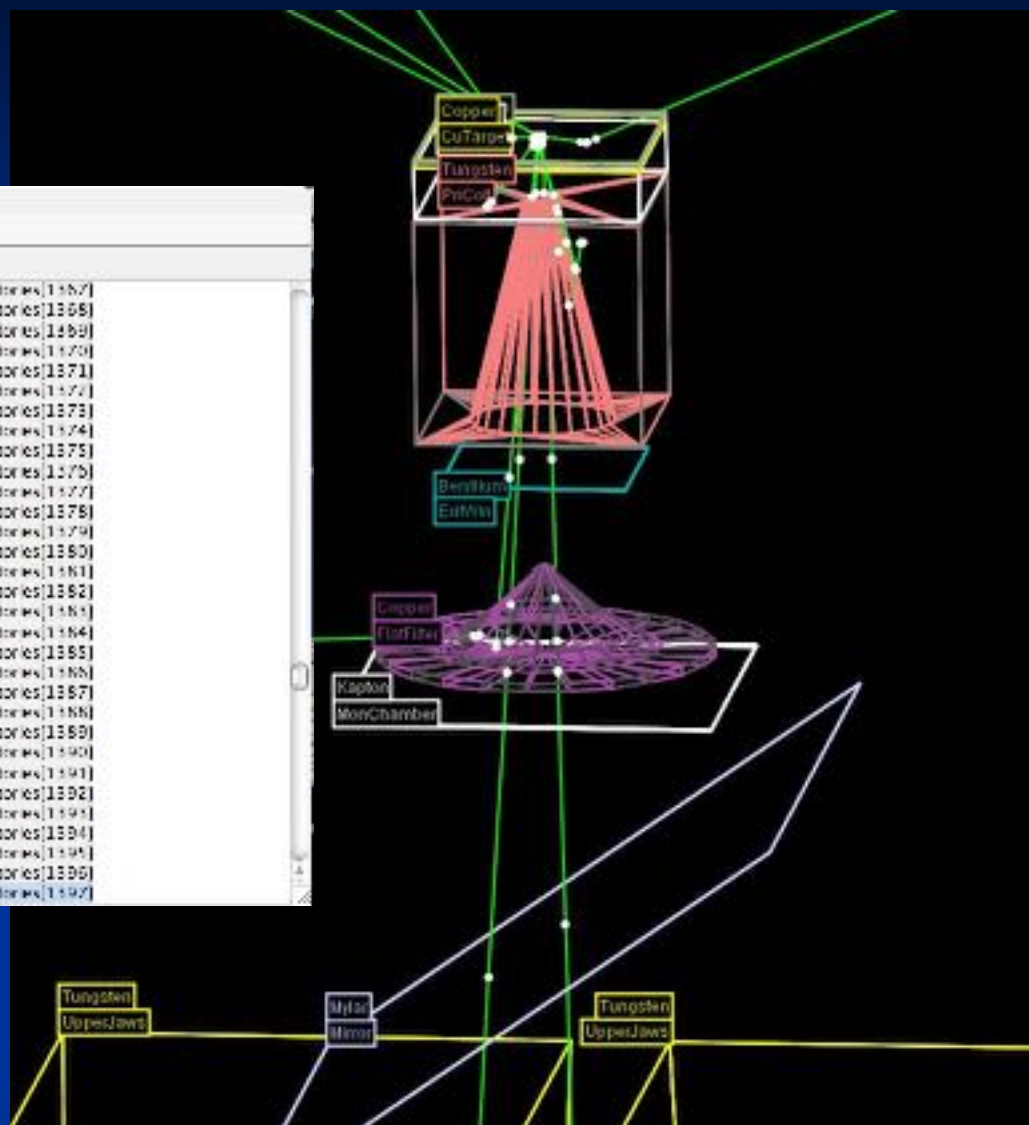
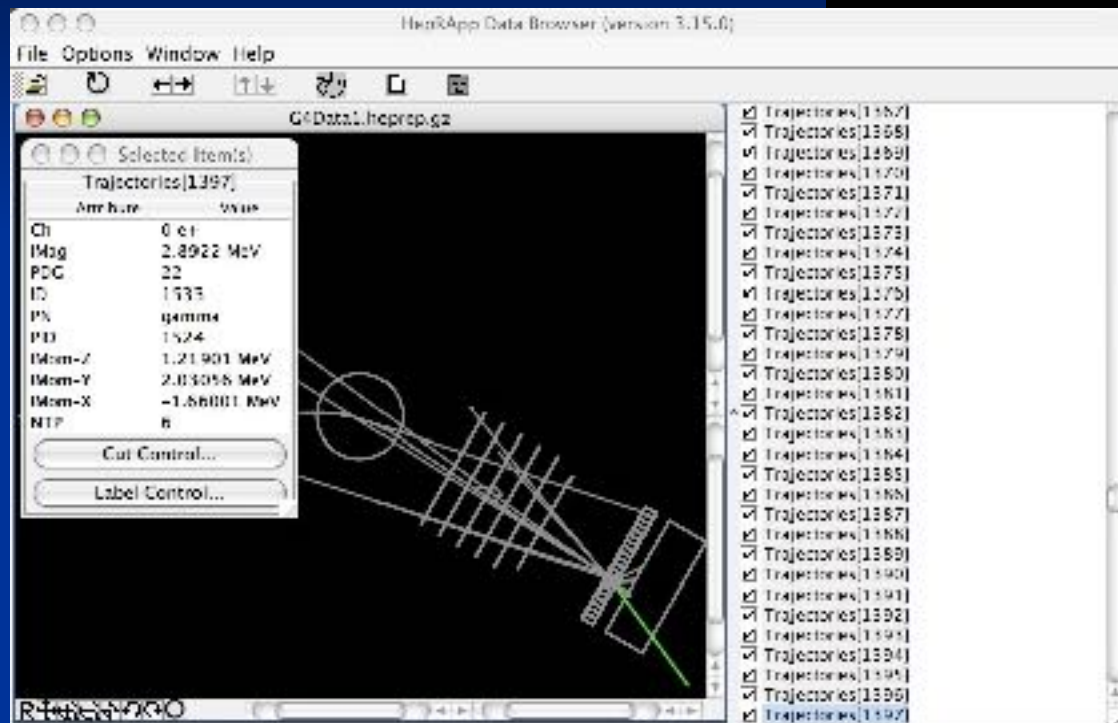
# High-quality Output for Publications

- DAWN



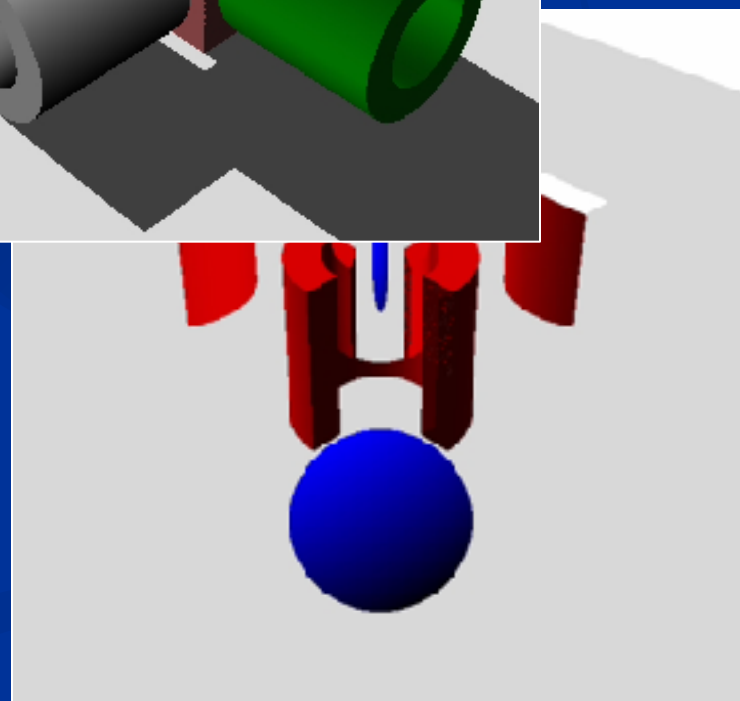
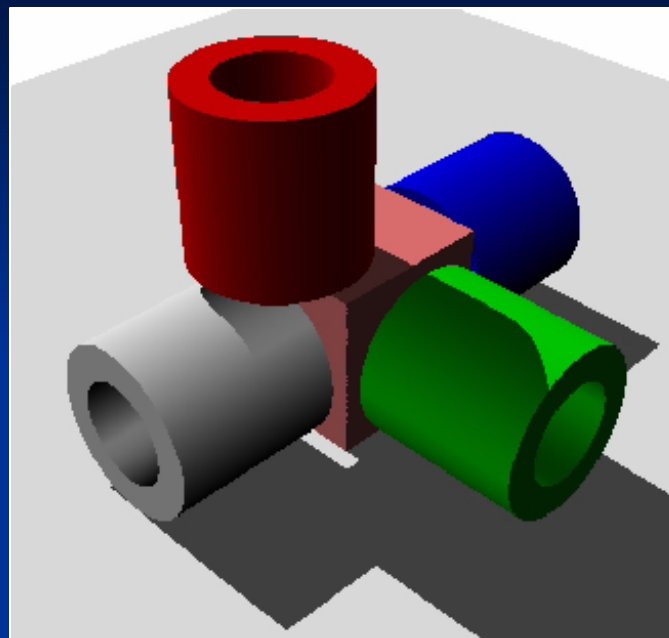
# Interactive Picking to Get More Information on Visualized Objects

- HepRep



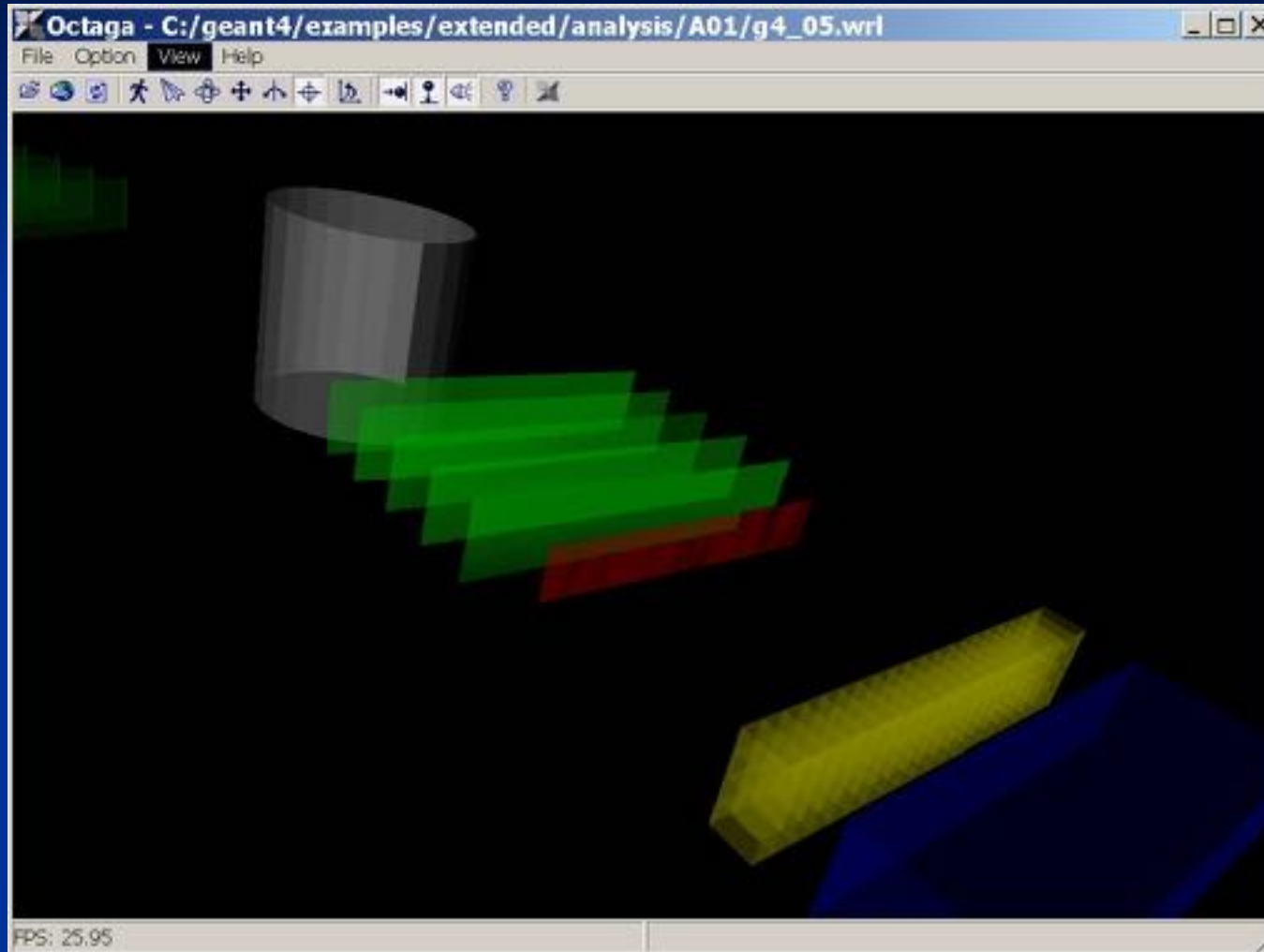
# Understand Complex Boolean Solids and Transparent or Reflective Surfaces

- RayTracer



# 3D Format Suitable for Web Distribution

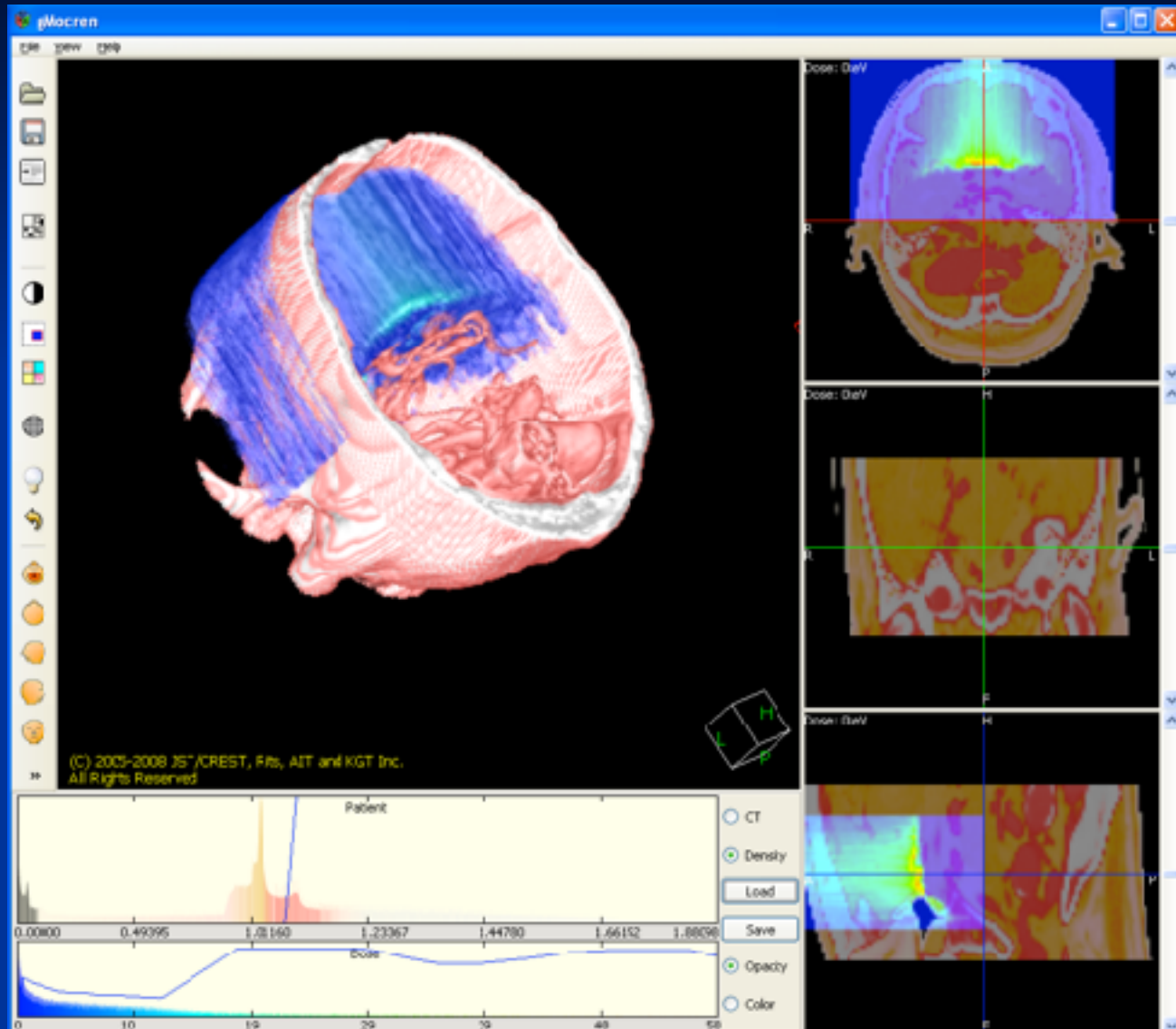
- VRML





# Visualize Volume Data, such as Dose

- gMocren



# Understand Geometry Hierarchies

- ASCIITree, HepRep

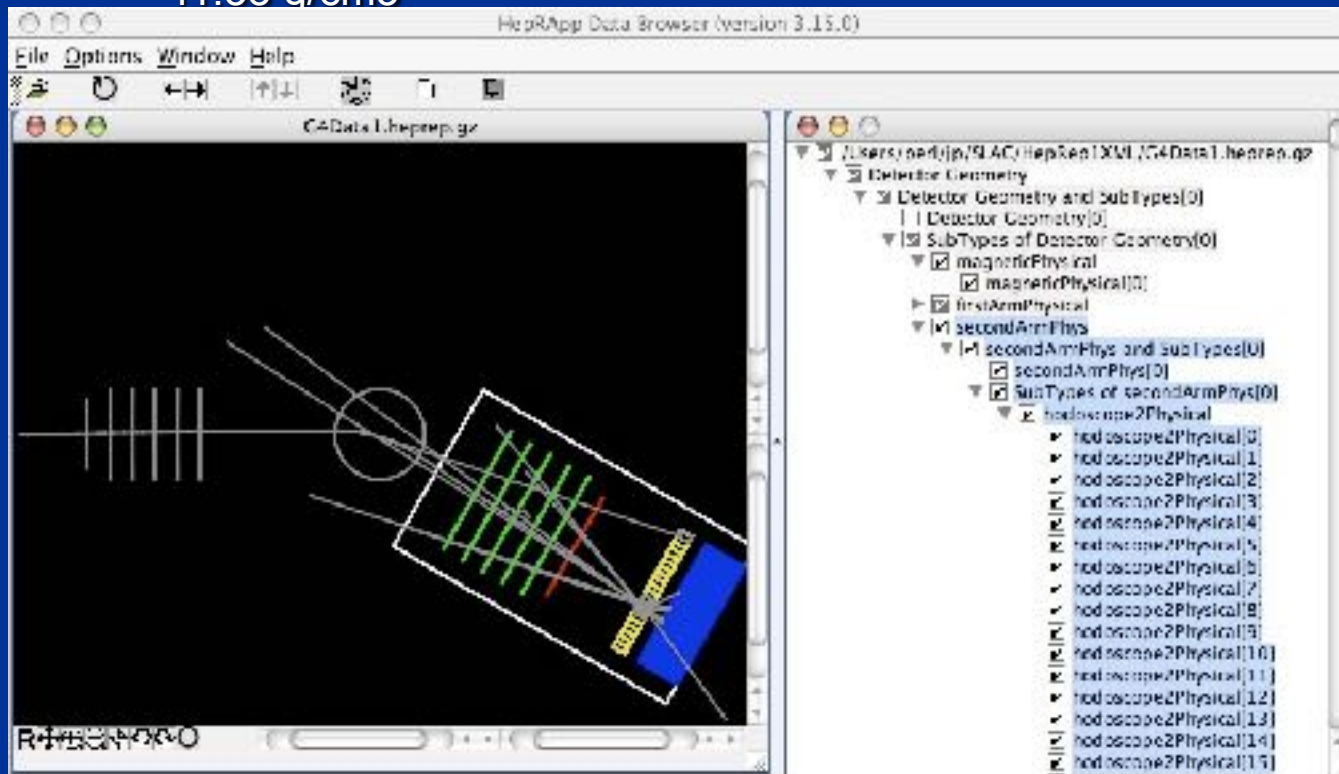
- /vis/ASCIITree/Verbose 4

- /vis/viewer/flush

- "HadCalorimeterPhysical":0 / "HadCalorimeterLogical" / "HadCalorimeterBox"(G4Box), 1.8 m<sup>3</sup>, 11.35 g/cm<sup>3</sup>

- "HadCalColumnPhysical":-1 (10 replicas) / "HadCalColumnLogical" / "HadCalColumnBox"(G4Box), 180000 cm<sup>3</sup>, 11.35 g/cm<sup>3</sup>

- "HadCalCellPhysical":-1 (2 replicas) / "HadCalCellLogical" / "HadCalCellBox"(G4Box), 11.35 g/cm<sup>3</sup>



# Design by Interfaces

- You may ask why Geant4 has so many different visualization systems.
- This is a natural result of Geant4 being a toolkit and not a single application.
- To support user communities who incorporate Geant4 into their own pre-existing software frameworks, Geant4 visualization is built around a set of well defined interfaces.
  - These interfaces make it straightforward to connect Geant4's core visualization tools to any visualization system
    - able to drive advanced systems that can natively display complex solids such as Geant4's cut cylinders
    - able to drive more basic systems that do not understand such solids (system can ask Geant4 visualization to deconstruct complex solids into simpler polygons)
  - For those users who want a ready-made visualization solution from Geant4, these same interfaces have made it straightforward for us to provide a variety of solutions, each with particular areas of strength.
- Interfaces discussed in detail in: [The Geant4 Visualisation System](#)  
J Allison, M Asai, G Barrand, M Donszelmann, K Minamimoto, J Perl, S Tanaka, E Tcherniaev, J Tinslay, Computer Physics Communications, Volume 178, Issue 5, 331-365, 1 March 2008

# Eight Visualization Drivers

- No Single Visualization Solution Can Meet all of Our Demands
  - Quick response with flexible camera control
  - High-quality Output for Publications
  - Interactive Picking to Get More Information
  - Complex Boolean Solids and Transparent or Reflective Surfaces
  - 3D Format Suitable for Web Distribution
  - Visualize Volume Data
  - Understand Geometry Hierarchies
- By exploiting the same interface design that we need anyway to support visualization systems of existing frameworks
  - we are able to take advantage of the best features of several different visualization drivers
  - with a common set of user commands
  - and minimal maintenance for many of the drivers
- We take advantage of the best features of many pre-existing visualization systems without having to reinvent those systems.

# Controlling Visualization

- Your Geant4 code stays basically the same no matter which driver you use
- Visualization is performed either with commands or from C++ code
  - For the present tutorial, we confine ourselves to command-driven visualization.
- Some visualization drivers work directly from Geant4
  - OpenGL
  - OpenInventor
  - RayTracer
  - ASCII Tree
- For other visualization drivers, you first have Geant4 produce a file, and then you have that file rendered by another application (which may have GUI control)
  - HepRepFile
  - DAWNFILE
  - VRML2FILE
  - gMocrenFile
- You can open more than one driver at a time.
  - For example, do a quick check in OpenGL, then save the same event for a beautiful DAWN plot



# Controlling Which Drivers are Available

- Six of the visualization drivers are always included by default (since they require no external libraries):
  - HepRepFile
  - DAWNFILE
  - VRMLFILE
  - RayTracer
  - gMocrenFile
  - ASCIITree
- Other visualization drivers are included only if appropriate environment variables have been set:
  - The standard Geant4 configure procedure takes care of these environment variables for you. It asks you which drivers you want and then produces a script, “env.sh” or “env.csh”, that you execute to set these variables.
- You can also add your own visualization driver.
  - Geant4’s visualization system is modular. By creating just three new classes, you can direct Geant4 information to your own visualization system.

# Example Visualization Command Sequences

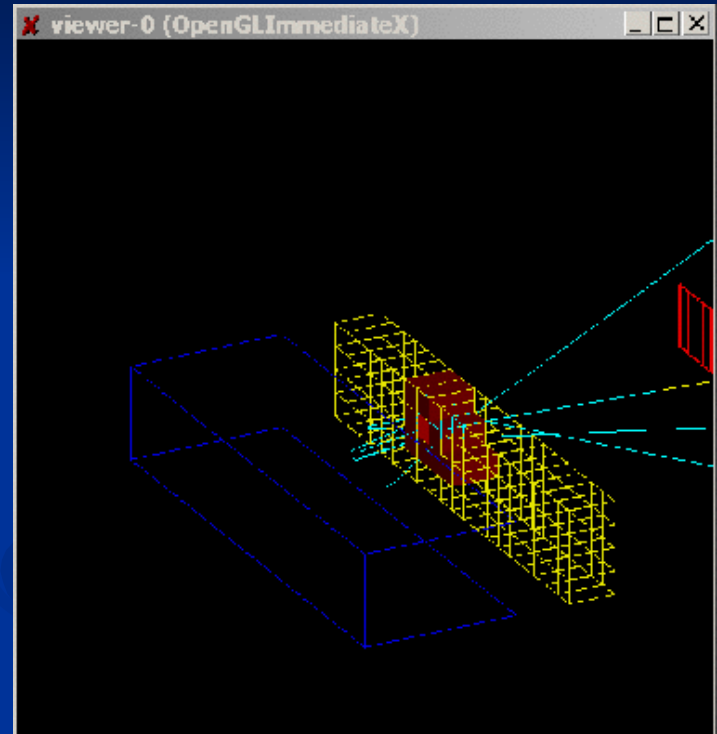
- Visualize a detector using the OpenGL visualization driver:
  - `/vis/open OGL`
  - `/vis/drawVolume`
- Visualize trajectories and hits for 10 events using the HepRep visualization driver:
  - `/vis/open HepRepFile`
  - `/vis/drawVolume`
  - `/vis/scene/add/trajectories`
  - `/vis/scene/add/hits`
  - `/run/beamOn 10`
- The step-by-step Geant4 Installation Tutorials tell you enough commands to get you started:
  - [geant4.slac.stanford.edu/installation](http://geant4.slac.stanford.edu/installation)
- A more detailed discussion of commands can be found in:
  - [geant4.slac.stanford.edu/Presentations/vis/G4VisCommands.pdf](http://geant4.slac.stanford.edu/Presentations/vis/G4VisCommands.pdf)
  - [geant4.slac.stanford.edu/Presentations/vis/G4VisAdvanced.pdf](http://geant4.slac.stanford.edu/Presentations/vis/G4VisAdvanced.pdf)

# Details on the Eight Visualization Drivers

- OpenGL
- OpenInventor
- HepRep
- DAWN
- VRML
- RayTracer
- gMocren
- ASCII Tree

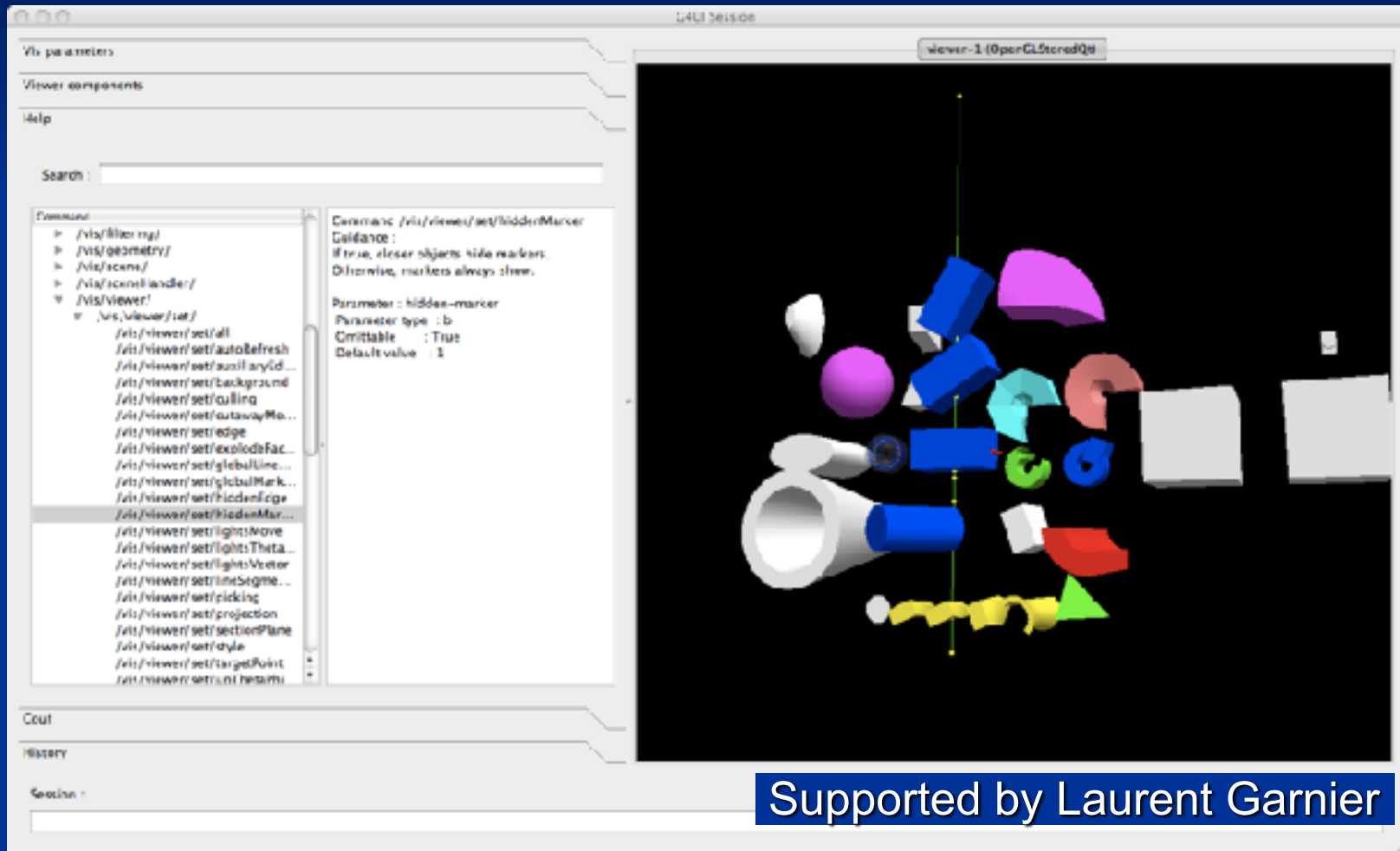
# OpenGL

- /vis/open OGL
- Features
  - Control directly from Geant4
  - Uses GL libraries that are already included on most Linux and Windows systems
  - Rendered, photorealistic image with some interactive features
    - zoom, rotate, translate
  - Fast response (can usually exploit full potential of graphics hardware)
  - Save as pixel graphics or vector EPS
  - Live movies
- Supported by John Allison, Laurent Garnier and Guy Barrand



# OpenGL with Qt Control

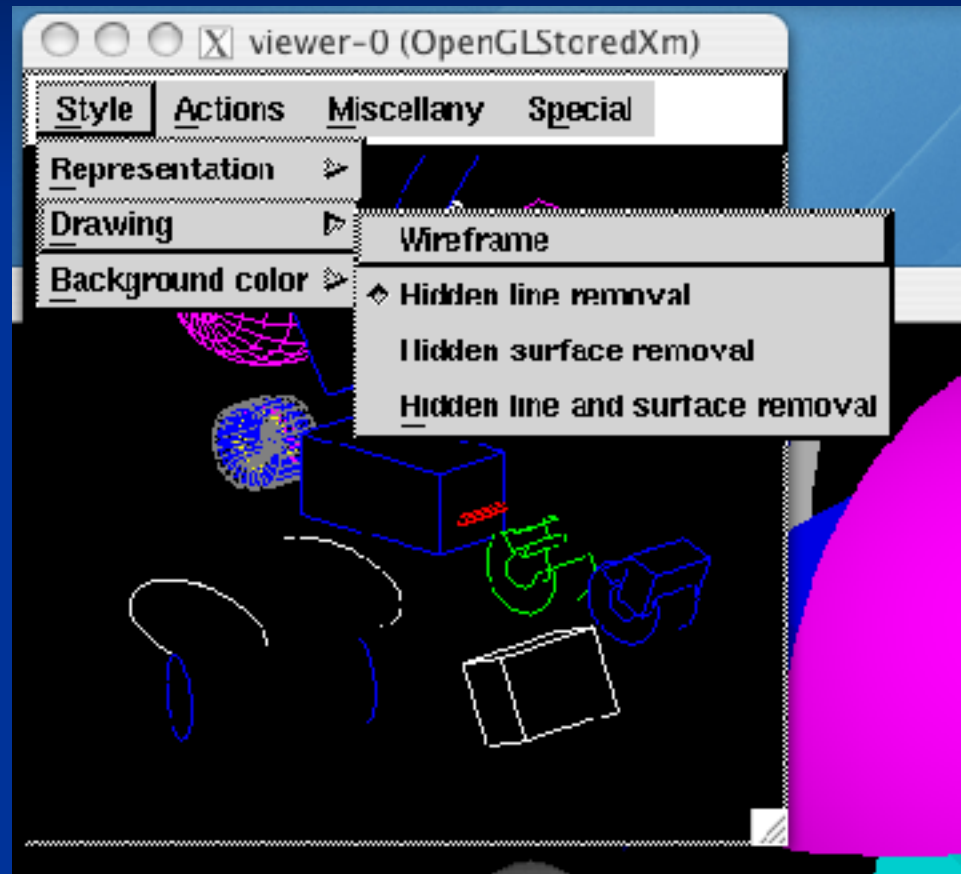
- New visualization driver and user interface
  - requires that you install Qt and link against this in your Geant4 ./Configure
  - you can customize the interface to control any Geant4 command





# OpenGL with Motif Control

- Somewhat obsolete now that Qt can take over this functionality
  - but still supported
  - requires that you have Motif and link against this in your Geant4 ./Configure



# OpenGL Additional Modes

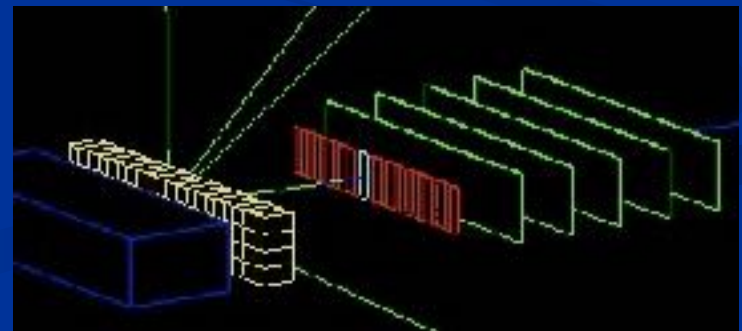
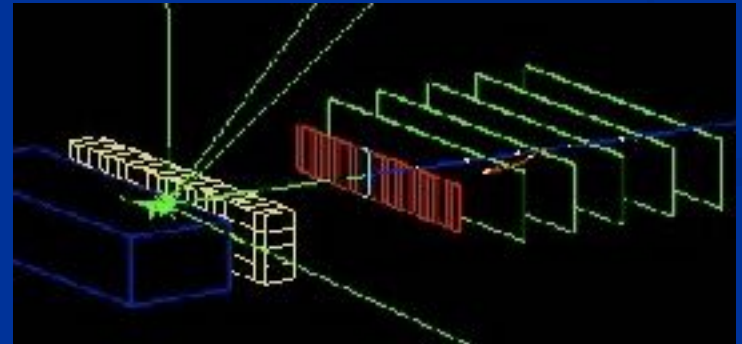
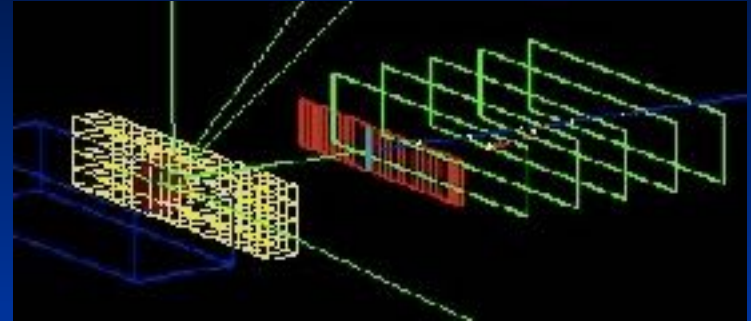
- There are actually two OpenGL modes, OGLS and OGLI
  - Stored mode: creates graphical database (display lists). Redrawing is faster because Geant4 only needs to resend parts that have changed. Nothing is lost on simple operations like change of viewing angle.
  - Immediate mode: draws only to screen, no “memory”; detector can be redrawn after view changes but event data is lost.
- And if you have Qt and Motif built and configured, even more options:
  - OGLSX, OGLIX, OGLSWin32 and OGLIWin32 are basic OpenGL (stored, immediate, stored for microsoft windows, immediate for microsoft windows)
  - OGLSQt and OGLIQt are OpenGL with Qt
  - OGLSXm and OGLIXm are OpenGL with Motif
- When you just use `/vis/open OGL`,
  - We provide you the most advanced OpenGL that you currently have configured - Qt if you have it, otherwise Motif, otherwise basic OpenGL
  - We give you Stored mode unless starts to use too much memory, in which case we switch to Immediate mode
  - We worry for you about whether you're on Windows or not
- To explicitly specify stored or immediate, but leave other decisions to us:
  - `/vis/open OGLS`
  - `/vis/open OGLI`

# OpenGL Additional Notes

- OpenGL supports Smooth shading and Transparency
- OpenGL supports non-symmetric scaling:
  - `/vis/viewer/scale <3 vector of scale factors>`
  - or
  - `/vis/viewer/zoom <3 vector of scale factors>`

# Hidden Line Removal

- OpenGL supports hidden line removal.
- You can control whether this removal is done and whether trajectories and hits are affected by this feature.
- By default, hidden line removal is disabled
- To turn on hidden line removal
  - `/vis/viewer/set/hiddenEdge 1`
- This hides edges of geometry, but lets trajectories through.
- To hide trajectories and hits as well
  - `/vis/viewer/set/hiddenMarker 1`



# Postscript Printing

- OpenGL can save the current view to high quality Postscript files
  - `/vis/ogl/printEPS`
- Default output is Vectored, but can also produce Pixmap output
  - `/vis/ogl/set/printMode` vectored or pixmap
- Additional options:
  - `/vis/ogl/set/transparency` True or False



# Movies: Time Development of the Event

- New features since release 8.2 allow you to do a new kind of Geant4 movie. You can make movies that show Time Development of an event
  - I.e., a shower in slow motion
- Based on technique of “time-slicing”, breaking trajectories into individual slices, each with a time attribute.
  - requires newer visualization features, rich trajectory and some extensions to the OpenGL driver
  - you can run these animations Directly from Geant4, does NOT involve stitching together a movie by hand
- A collection of example movies has been prepared by John Allison:  
<http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/>
- How-To Presentation:  
<http://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.ppt> <http://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.pdf>

50 MeV electron into  
lead-liqAr calorimeter

e<sup>-</sup> red  
e<sup>+</sup> blue  
 $\gamma$  green

Yellow circles are  
step points --  
geometrical  
boundaries or  
physical processes



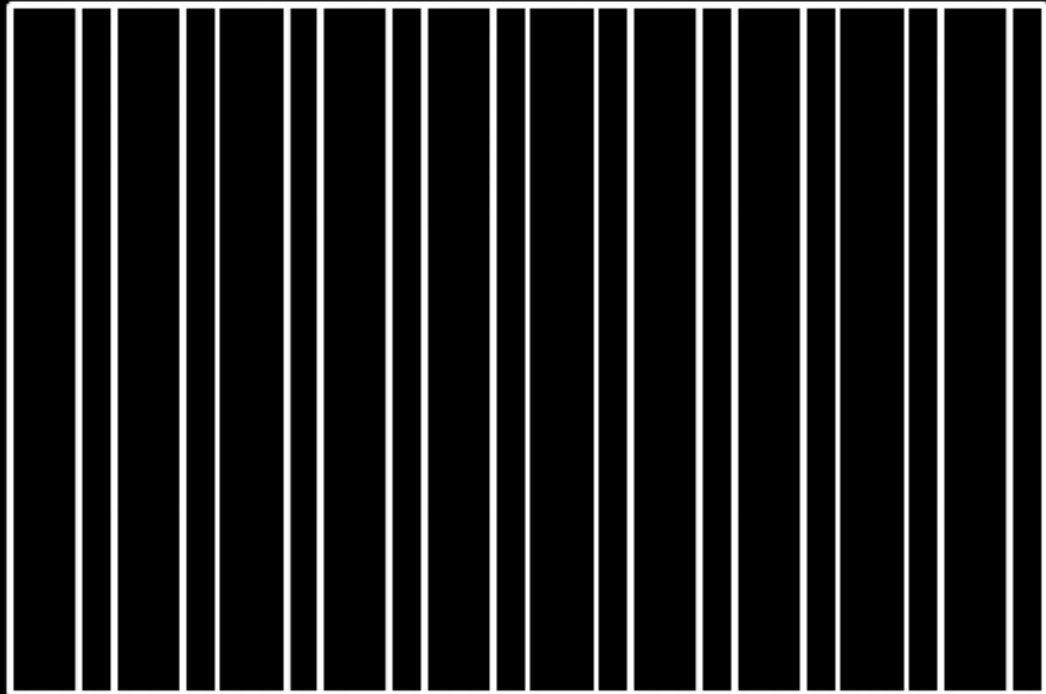
Another 10 GeV  $\pi^-$

3 fast  $\pi^-$

$\pi^+$  interaction

3 ns

Mpeg4 encoding with  
QuickTime Pro



# Movies: Live from Geant4 OpenGL



# What was in the Previous Movie

- <http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/pi-10Gevpi+neutronZoom.mp4>
- 10 GeV/c pi- on lead (in a lead-liquid-argon calorimeter, exampleN03 with QGSP physics)
  - A large number of slow pions, protons and neutrons
  - Three fast pi- and one fast pi+ that subsequently interacts again
  - Neutrons (yellow) hang around for several ns
- Green circle is the light front
  - Careful viewer will notice that near the end of the event, some particles appear to cross the light front. This was a bug in the way the light front sphere was rendered in perspective view, fixed in release 8.2

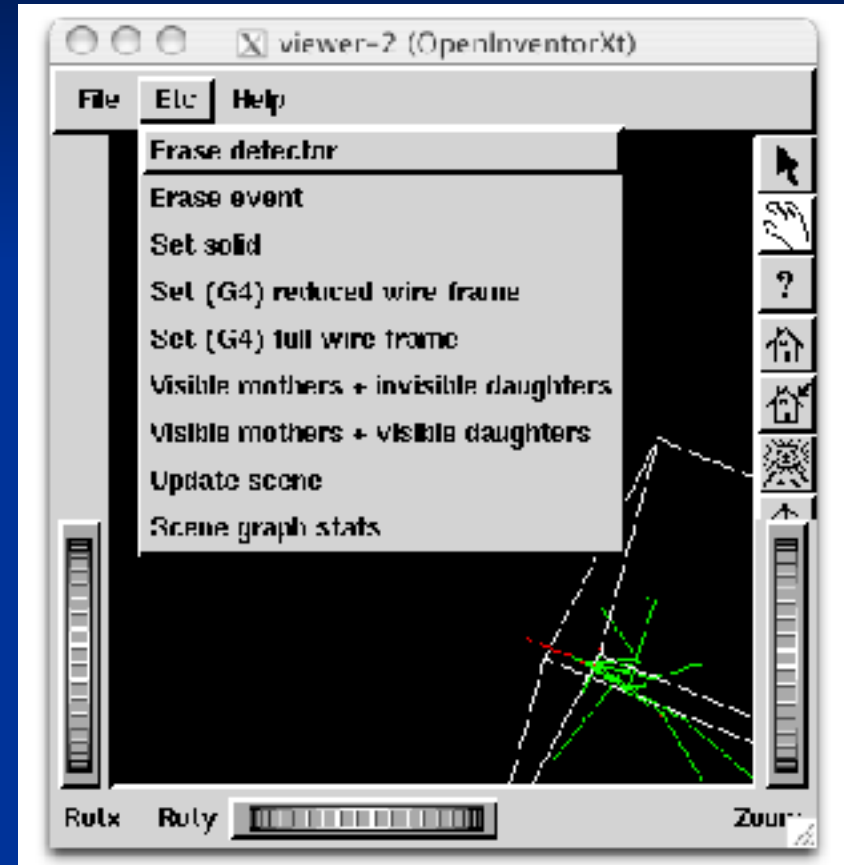
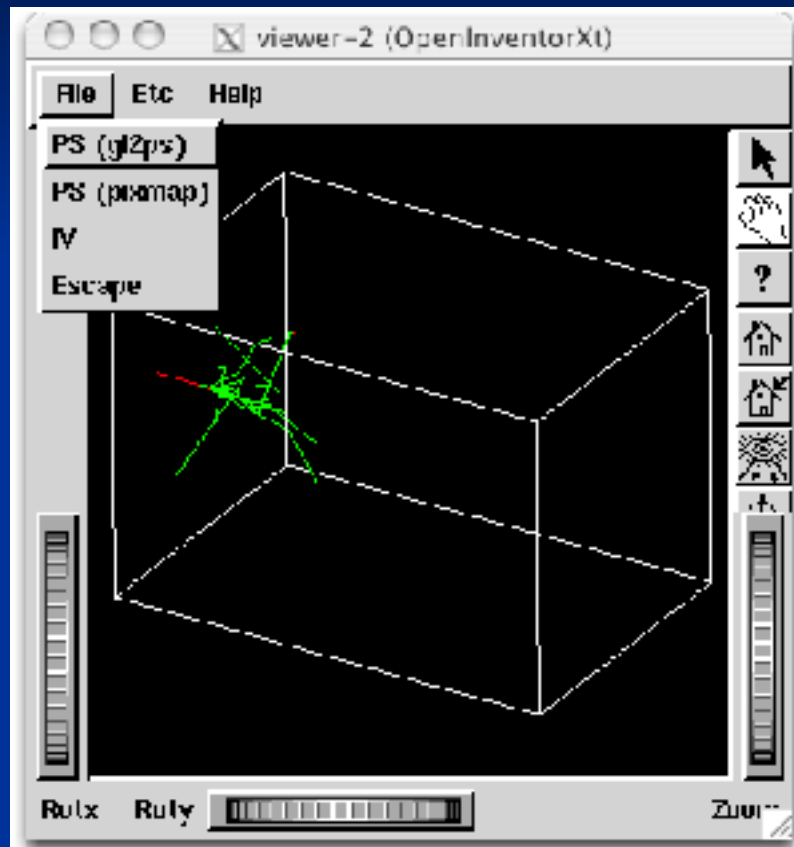
# OpenInventor

- `/vis/open OIX` or `/vis/open IOWin32`
- Features
  - Control from the OpenInventor GUI
  - Requires addition of OpenInventor libraries (freely available for most Linux systems and Windows).
  - Rendered, photorealistic image
  - Many interactive features
    - zoom, rotate, translate
    - click to “see inside” opaque volumes
    - click to show attributes (momentum, etc., dumps to standard output)
  - Fast response (can usually exploit full potential of graphics hardware)
  - Expanded printing ability (vector and pixel graphics)
- Supported by Guy Barrand



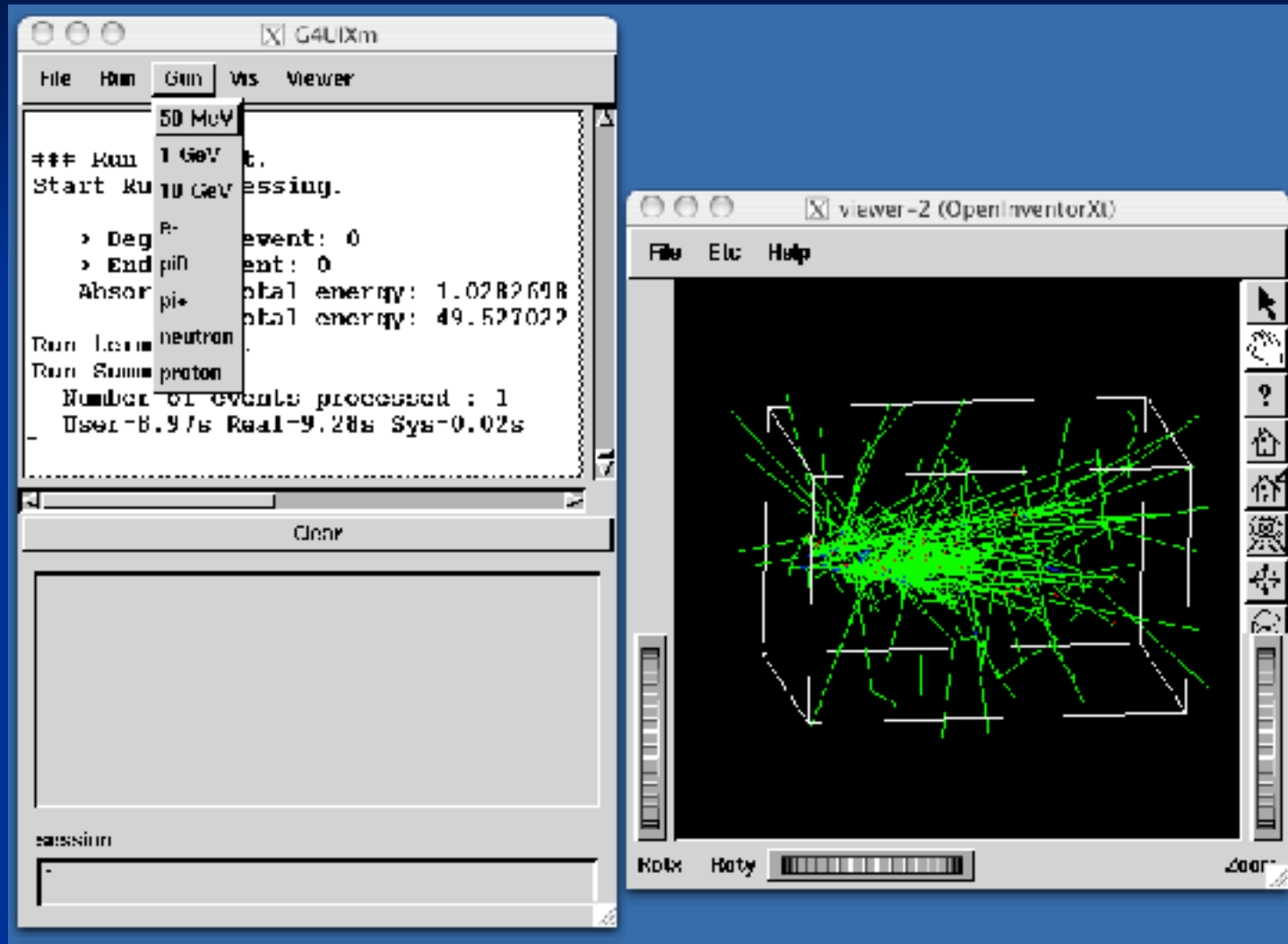
# OpenInventor: Start from Geant4

- With OpenInventor, start from Geant4, but then some control from OpenInventor GUI



# OpenInventor: More GUI Control

- You can also choose to control the Geant4 run from OpenInventor.



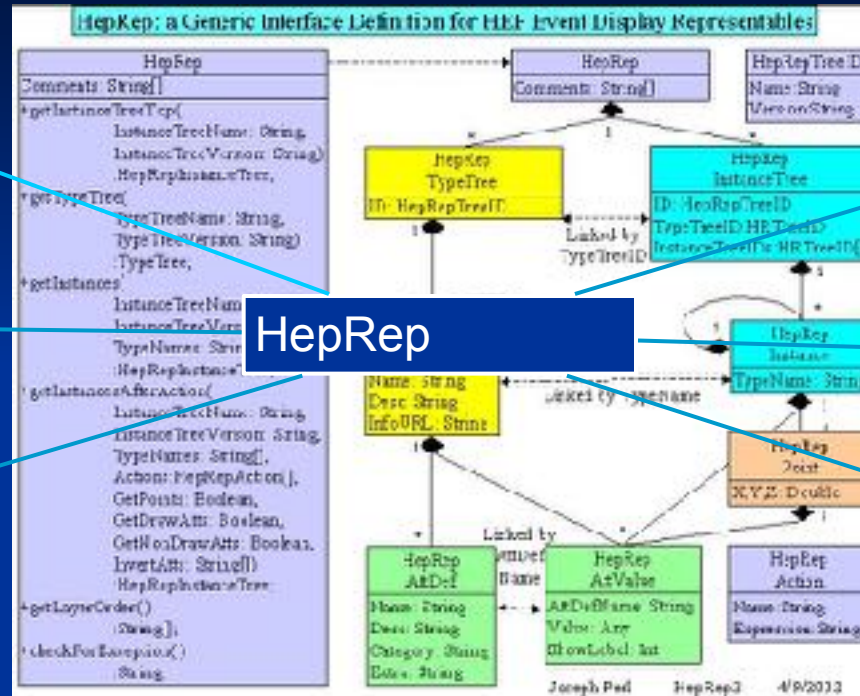
# OpenInventor Details

- Connected directly to the Geant4 kernel, using same language as that kernel (C++):
  - Can have direct access to Geant4 data (geometry, trajectories, etc.).
- Use of OpenGL for rendering:
  - Supports lighting and transparency
- Thumbwheel control to rotate and zoom
- Picking to ask about data
- “Control Clicking” on a volume turns on rendering of that volume’s daughters. “Shift Clicking” a daughter turns that rendering off:
  - If modeling opaque solid, effect is like opening a box to look inside
- OpenInventor is an open source project
- OpenInventor is described on the OpenScientist Home Page <http://openscientist.lal.in2p3.fr>
  - Follow the “Geant4 and Inventor” link at the left hand side of that page for details.

# HepRep

- /vis/open HepRepFile
- Features
  - Create a file to view in the
    - HepRApp HepRep Browser
    - WIRED4 JAS Plugin
    - or FRED Event Display
  - Requires one of the above browsers (freely available for all systems)
  - Wireframe or simple area fills (not photorealistic)
  - Many interactive features
    - zoom, rotate, translate
    - click to show attributes (momentum, etc.)
    - special projections (FishEye, etc.)
    - control visibility from hierarchical (tree) view of data
  - Hierarchical view of the geometry
  - HepRApp and WIRED4 can export to many vector graphic formats (PostScript, PDF, etc.)
- Supported by Joseph Perl

# HepRep is Not Just for Geant4 and Not Just for HepRApp



BaBar Server

GLAST Gaudi Service

Geant4 Server

HepRApp  
HepRep  
Browser (Java)

Wired4 Jas  
Plug-In (Java)

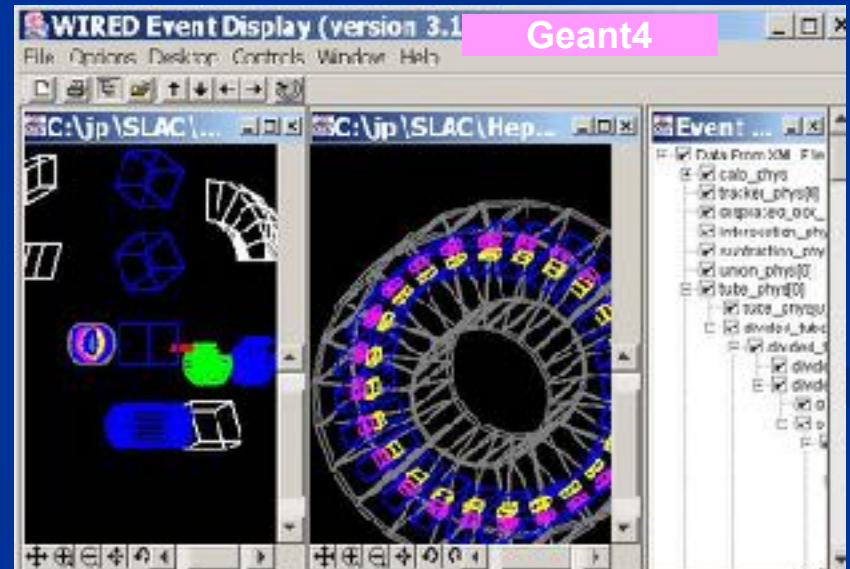
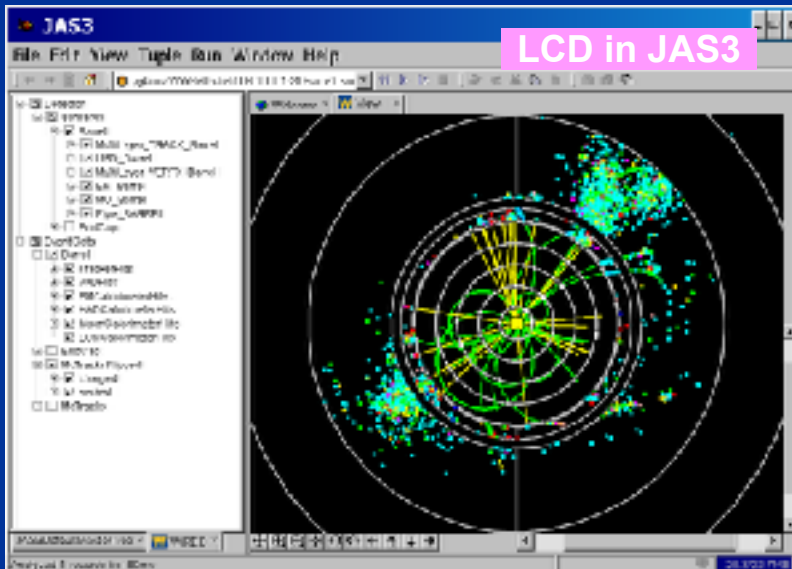
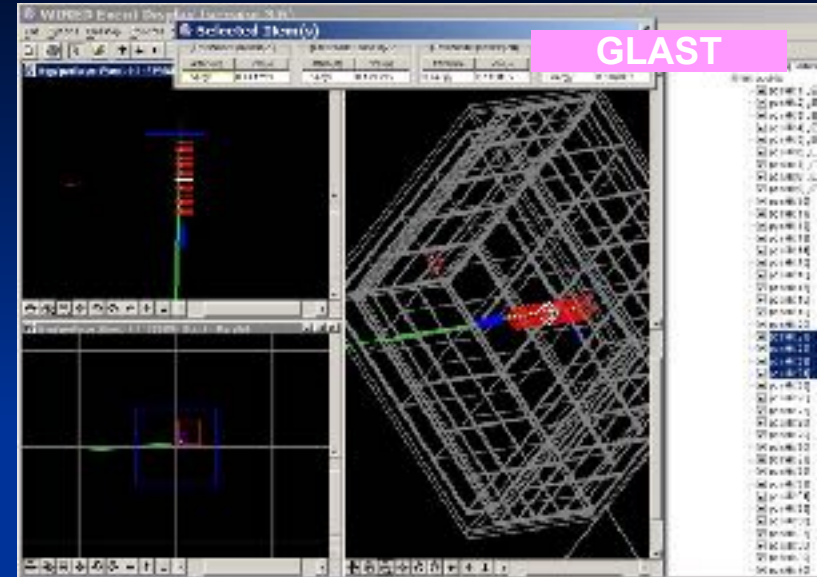
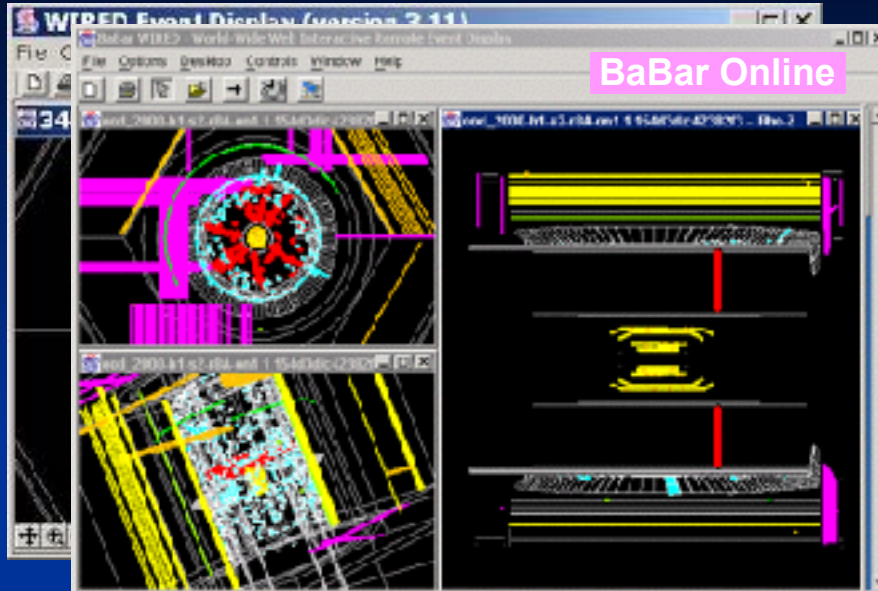
FRED Client  
(C++/Ruby)

The HepRep interface breaks the dependency between any particular experiment's event display server and any particular event display client.

The HepRep format is independent of any one particular language or protocol. It can be used from C++ or Java and can be shipped as Corba, RMI, XML, C++, Java or JNI for consumption by HepRApp, Wired4, FRED or any other HepRep-enabled event display client.



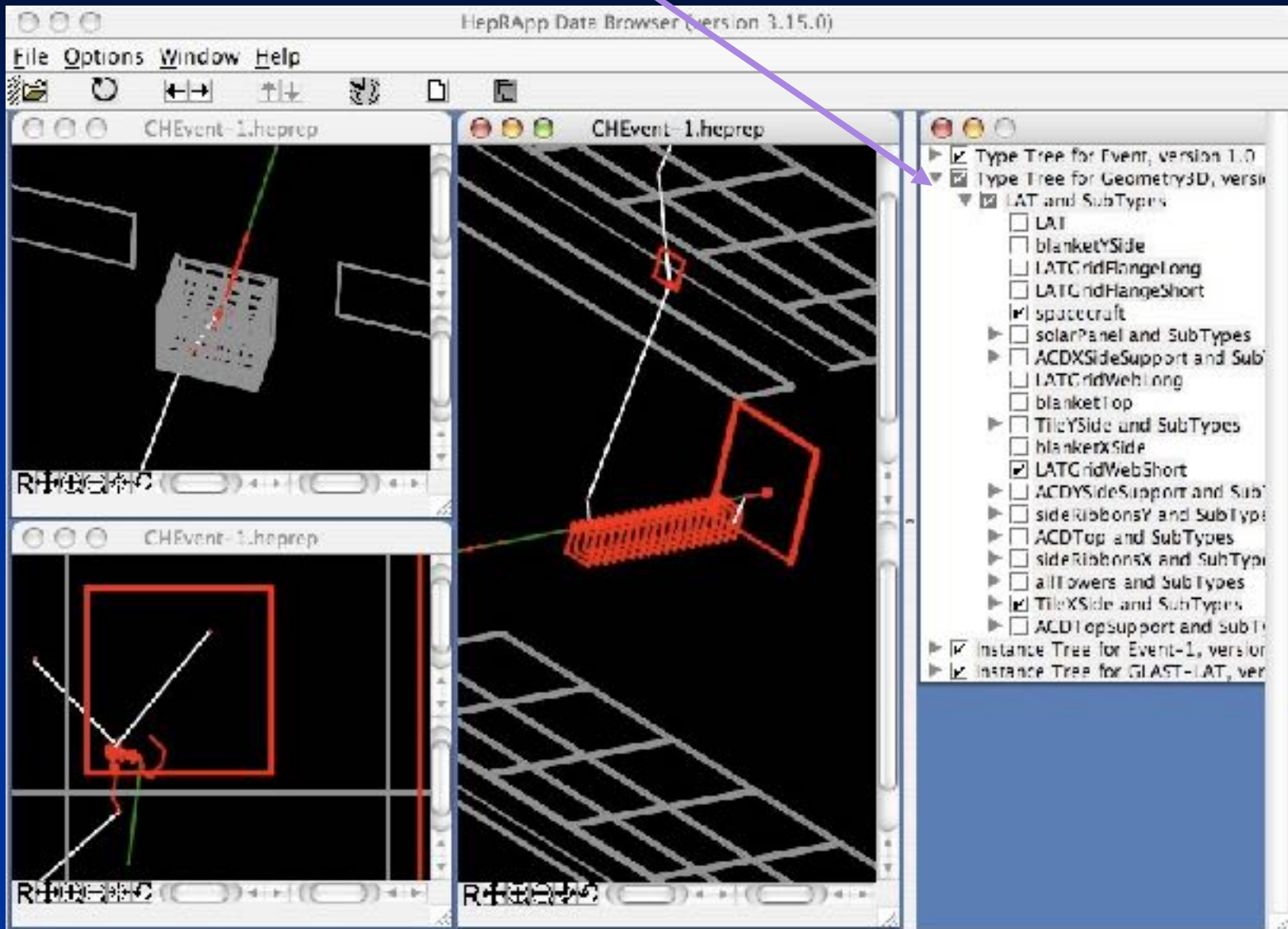
# Who's Using HepRep





# HepRApp: Shows Geometry Hierarchy

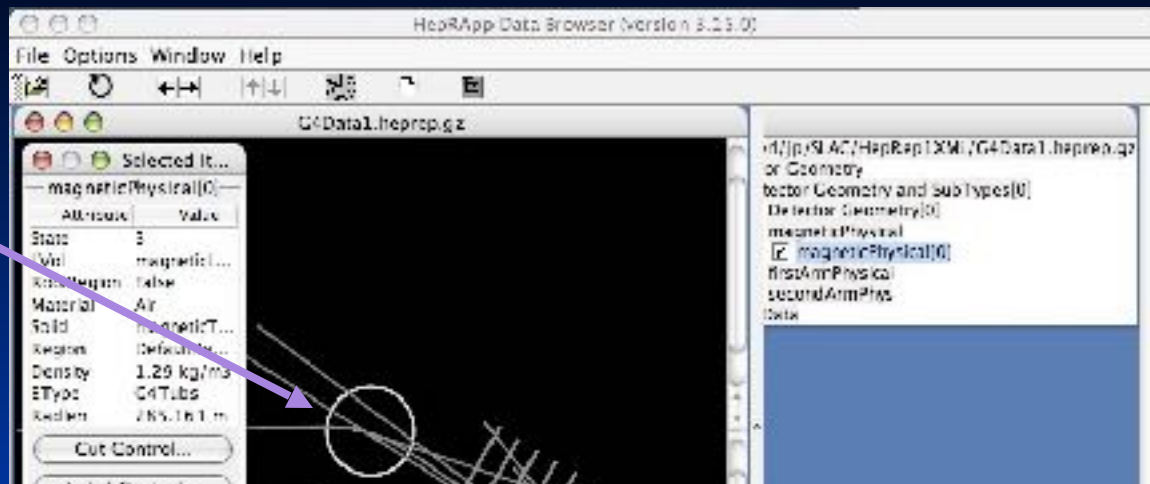
Turn visibility on and off from hierarchical control



# HepRApp: Pick to Show Physics Attributes

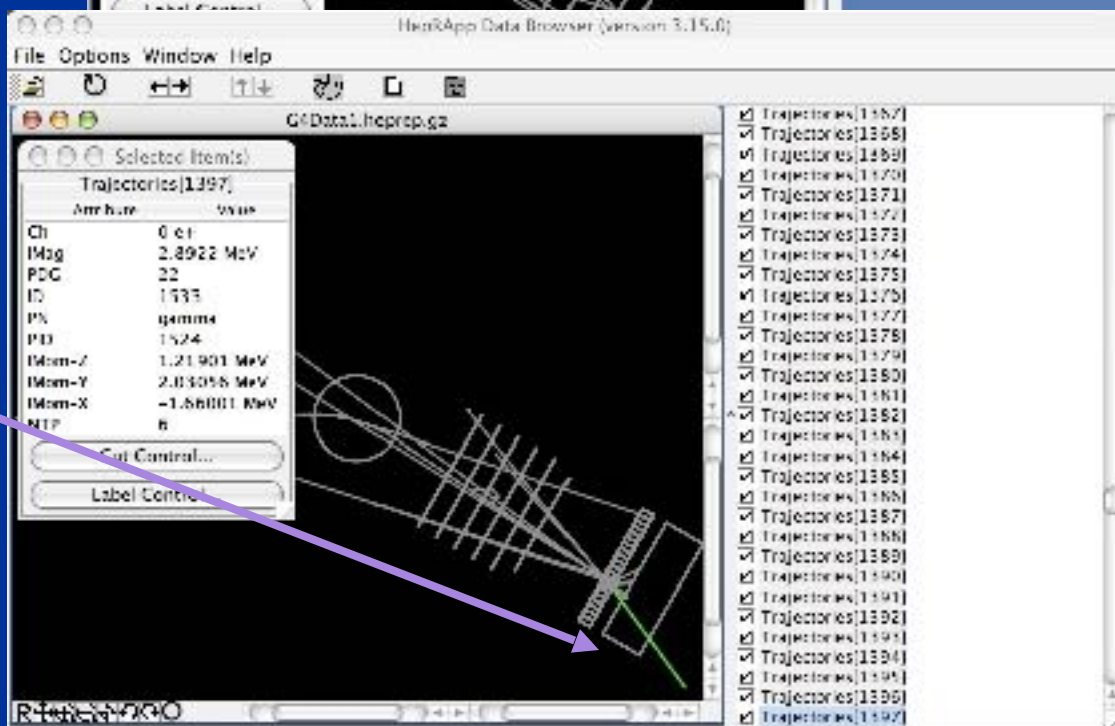
Picked on this volume to show

- Material
- Density
- RadLen
- etc

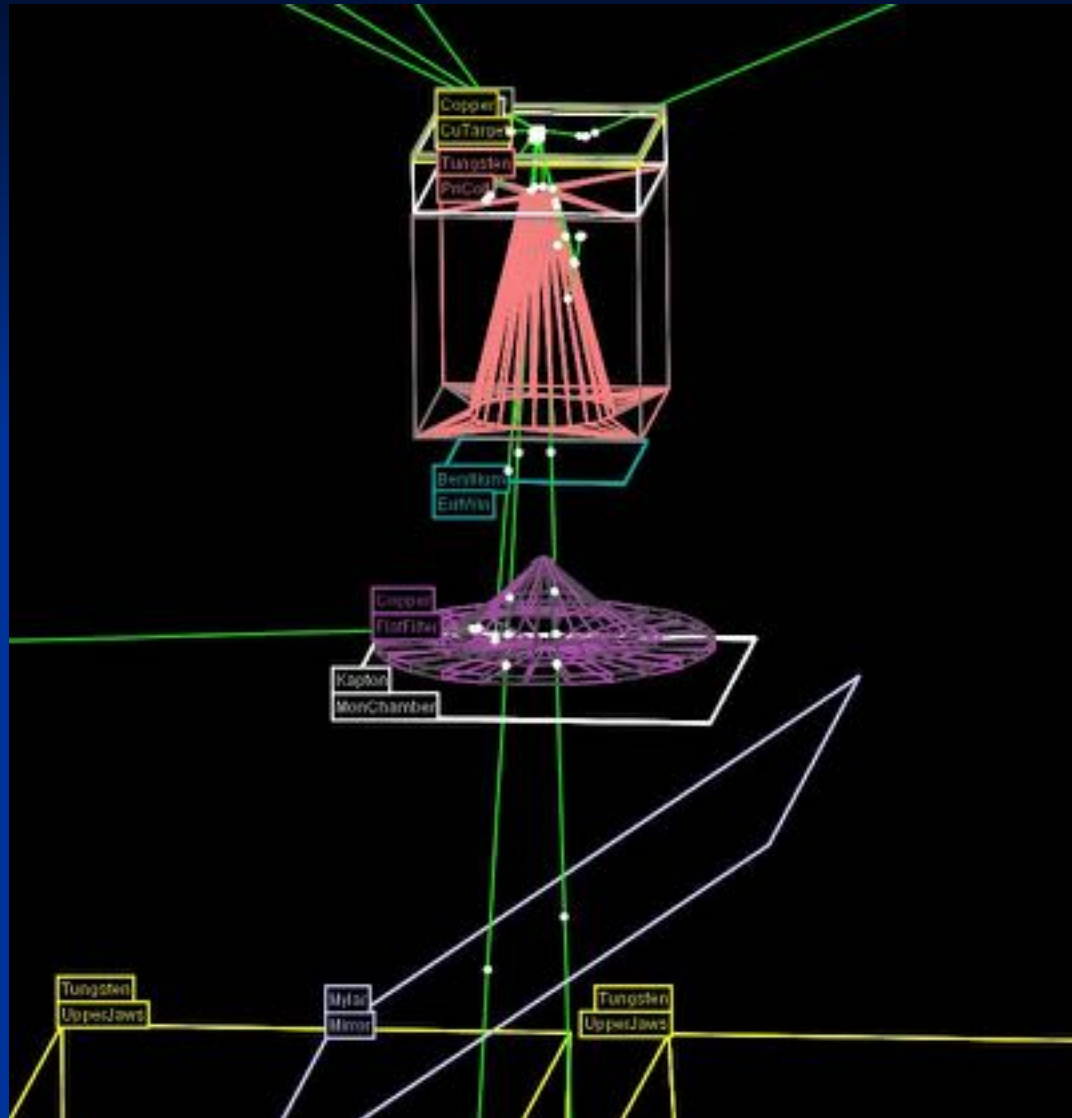


Picked on this trajectory to show

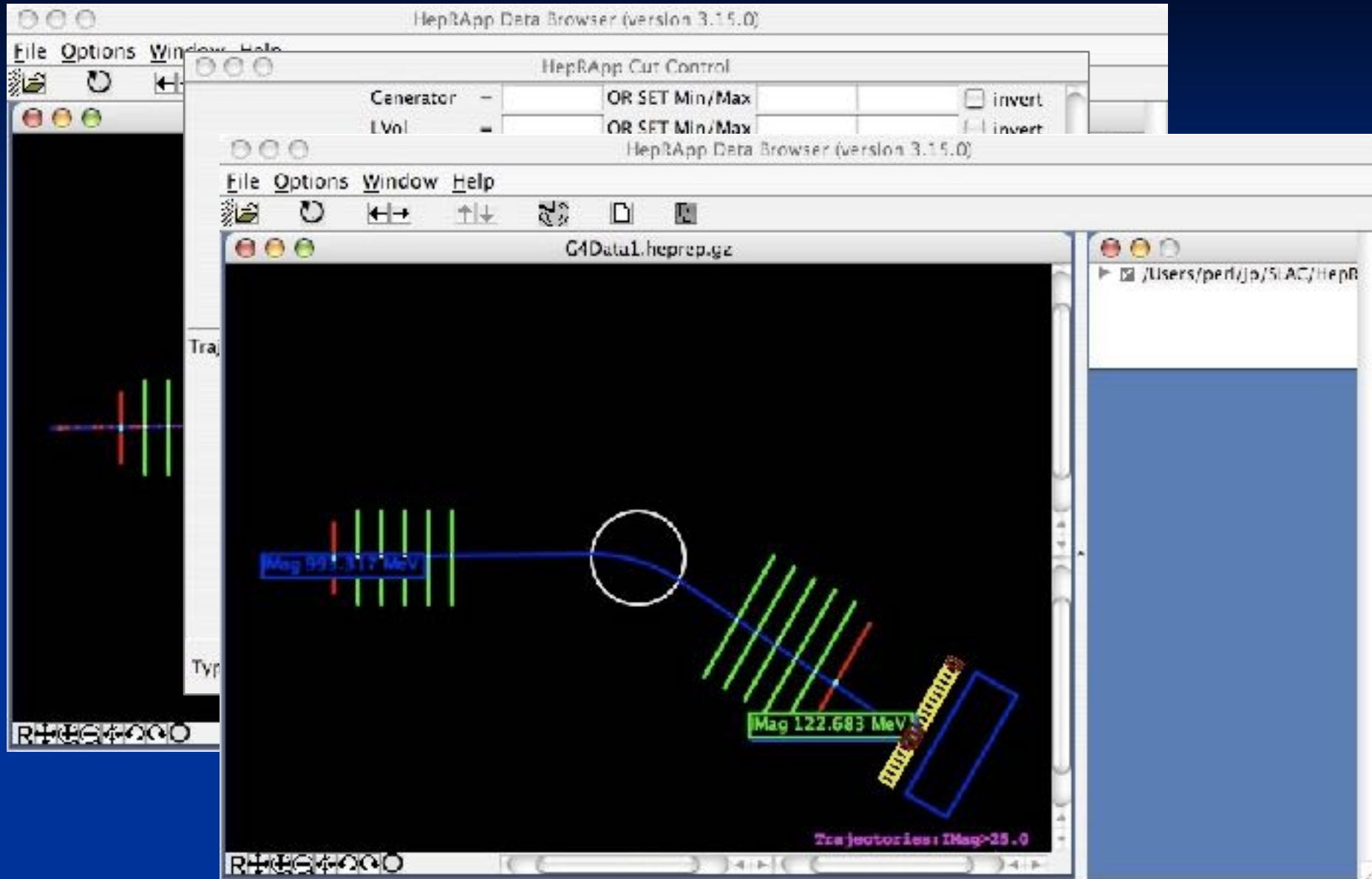
- Particle ID
- Charge
- Momentum
- etc.



# HepRApp: Labeling by Any Attribute

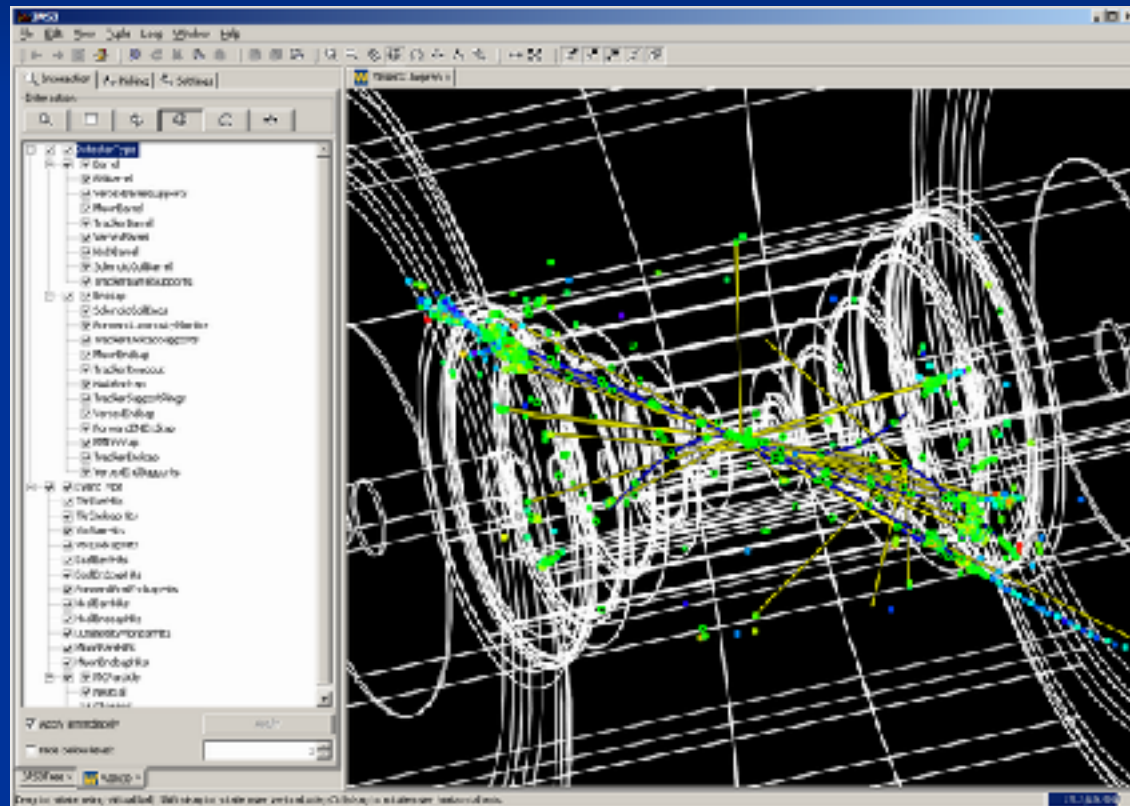


# HepRApp: Cut by Any Attribute



# WIRED4: Part of JAS

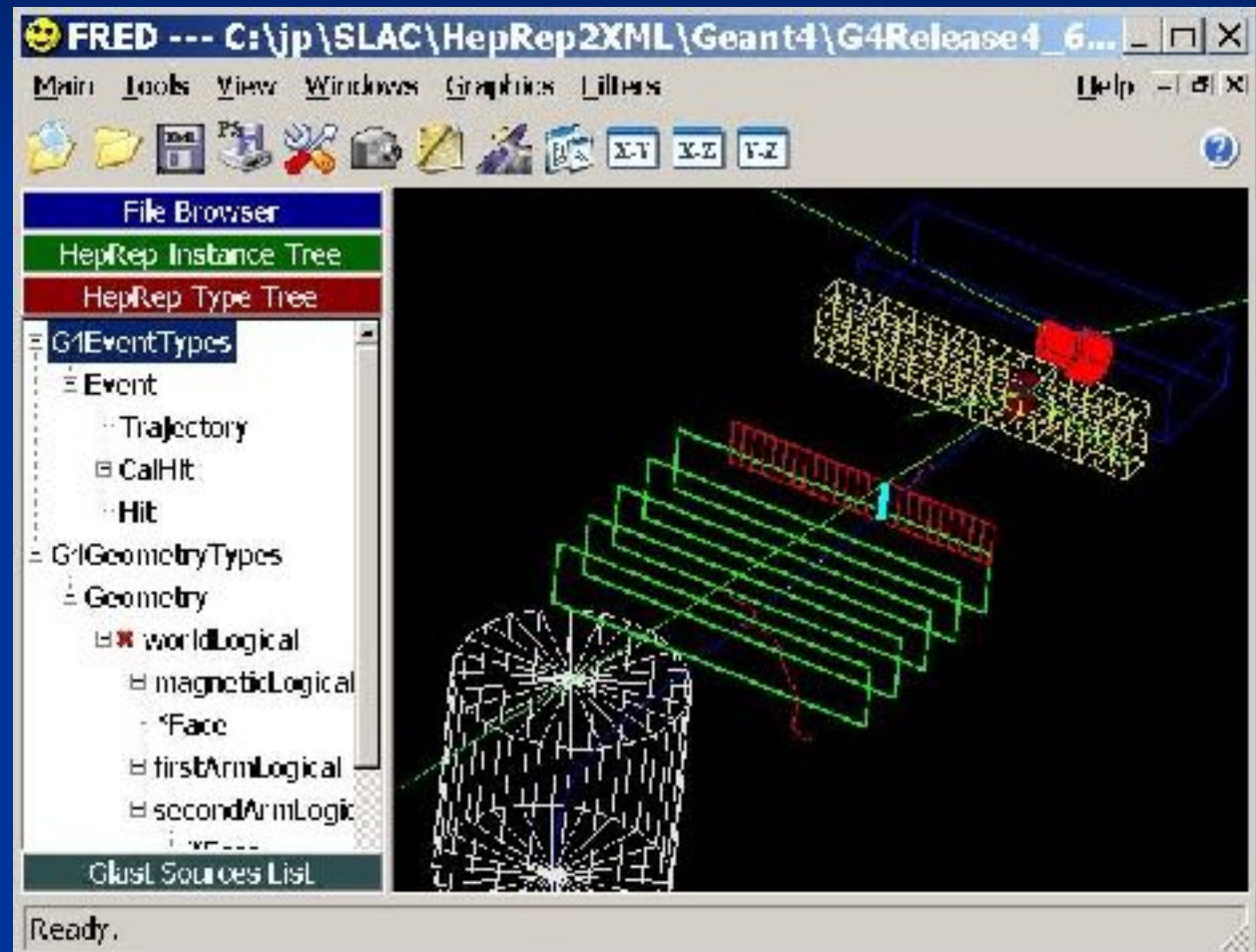
- Install JAS (Java Analysis Studio)
- Then download WIRED4 Plug-in
- <http://wired.freehep.org>





# FRED: Fox Ruby Event Display

- An additional HepRep-compatible browser developed by members of the GLAST space telescope collaboration.
- Includes the fast rotations and beautiful rendering of GL plus HepRep interactivity
- Allows scripting to change any attribute based on logic involving other attributes, hence things like "color by momentum" are scriptable.





# DAWN

- `/vis/open DAWNFILE`
- Features
  - Create a `.prim` file
  - Requires DAWN, available for all Linux and Windows systems
  - DAWN creates a rendered, photorealistic PostScript image
  - No interactive features once at PostScript stage
  - Highest quality technical rendering - vector PostScript
  - View or print from your favorite PostScript application
- Supported by Satoshi Tanaka

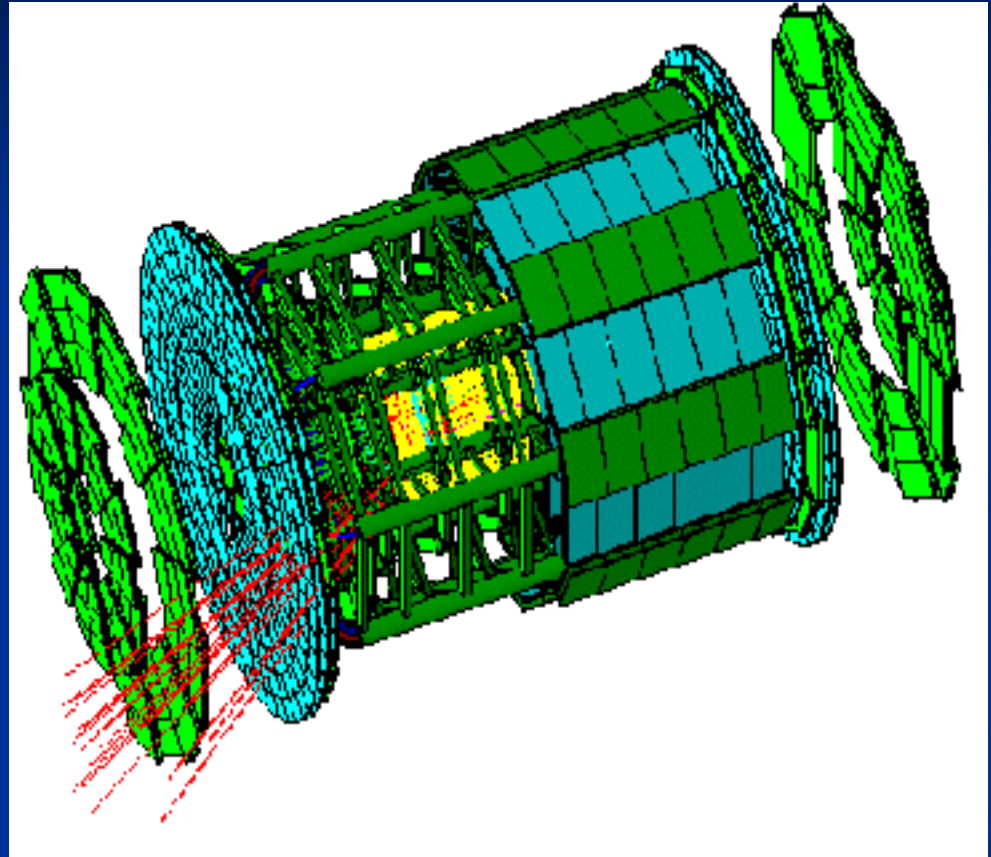
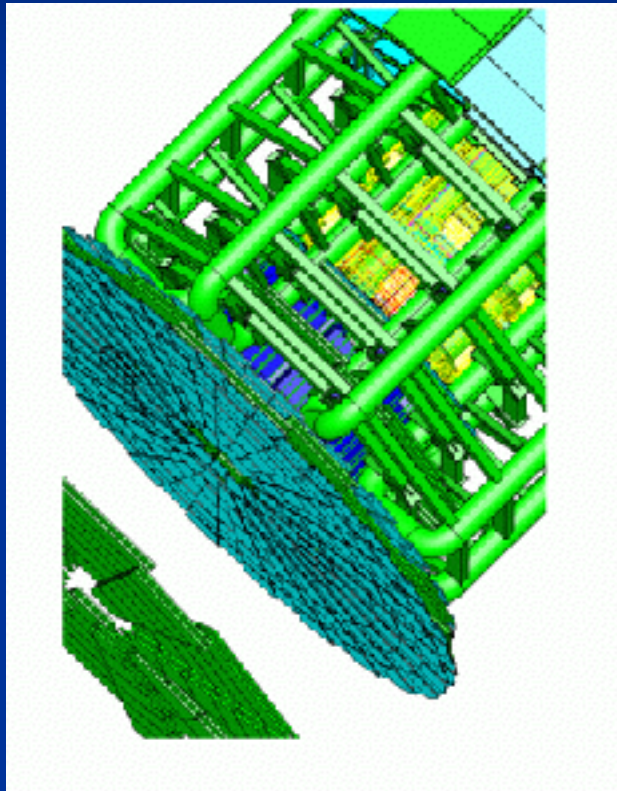
# Origins of DAWN

Fukui Renderer DAWN (Drawer for Academic WritINgs).

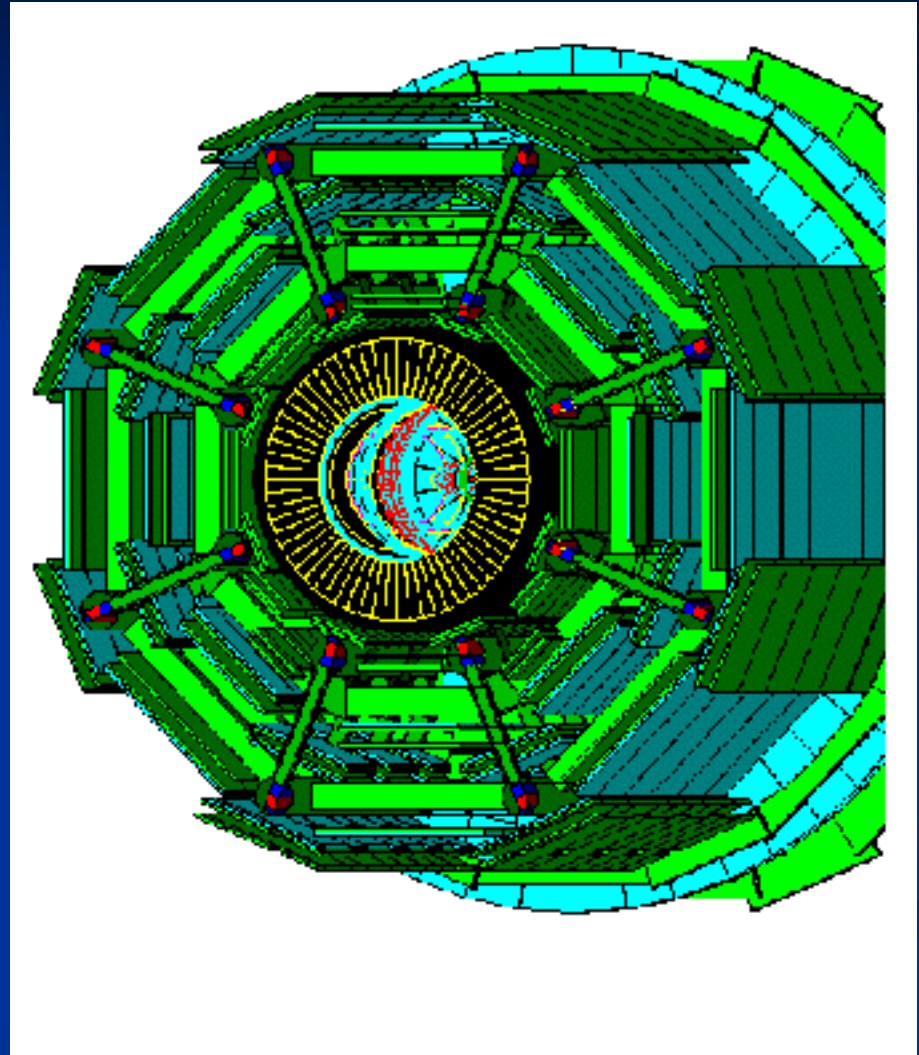
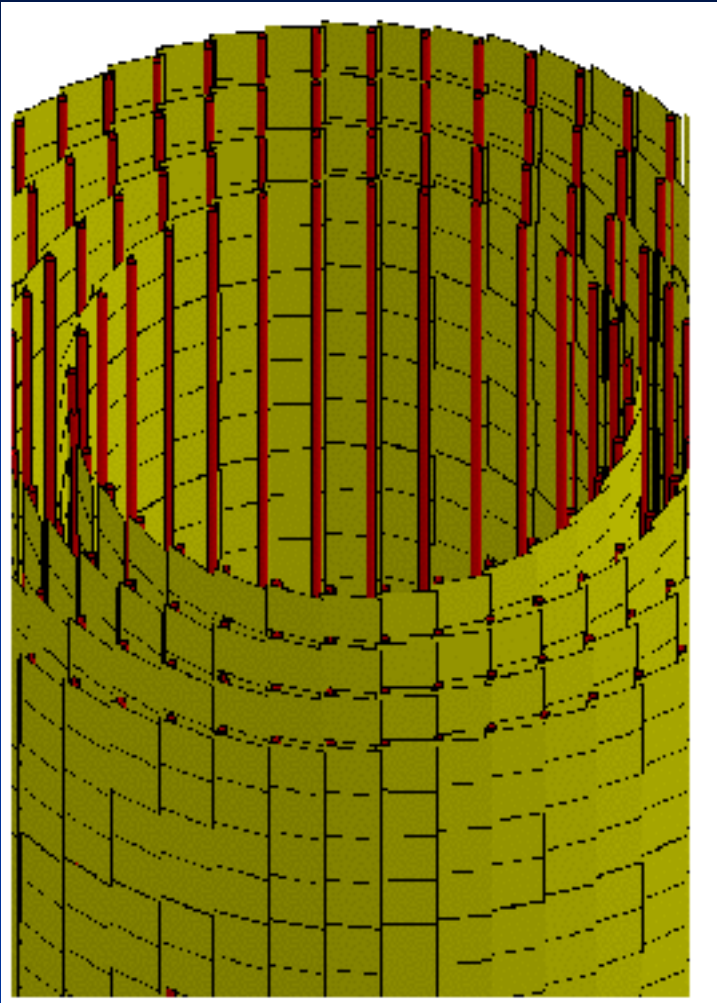
- A vectorized 3D PostScript processor with analytical hidden line/surface removal intended for precise technical drawing of complicated objects.
- Specifically designed for Geant4.
- Primitives set is same as Geant4 primitives set.
- Produces device-independent vectorized graphics for high quality technical applications.

# DAWN Examples

- From a repository of beautiful images at
  - [http://geant4.kek.jp/~tanaka/GEANT4/ATLAS\\_G4\\_GIFFIG/](http://geant4.kek.jp/~tanaka/GEANT4/ATLAS_G4_GIFFIG/)

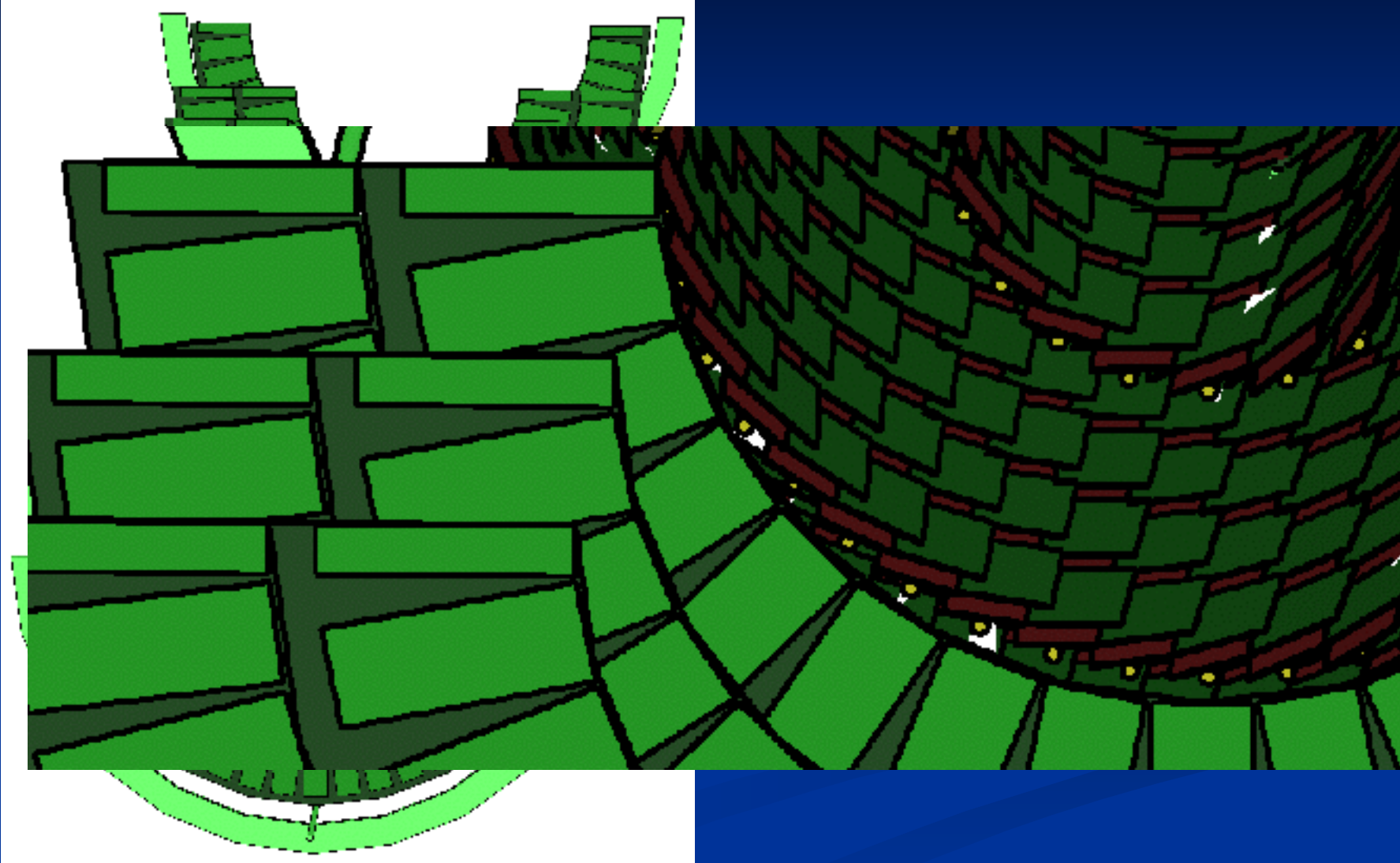


# DAWN Examples



# DAWN makes True Vector PostScript

- So when you zoom in with your PostScript browser, the images retain high resolution



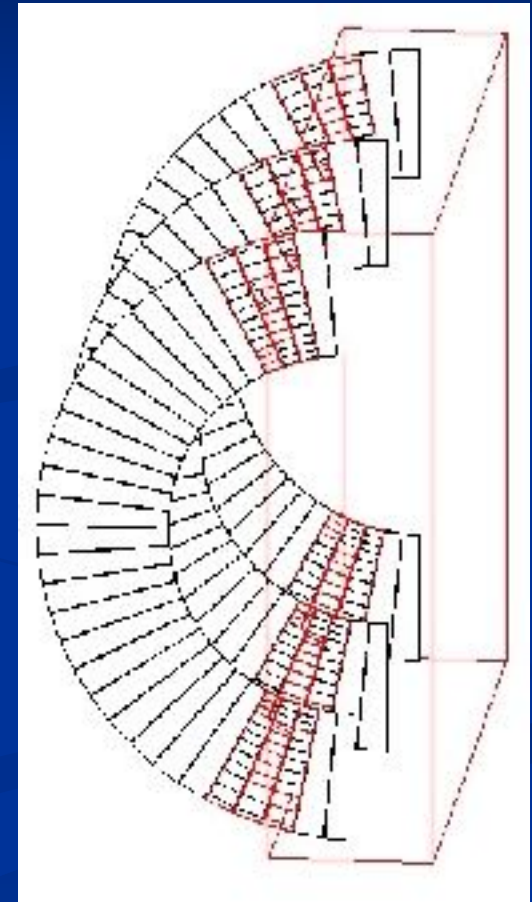
# DAWN Additional Modes

- There are actually two DAWN drivers:
- `/vis/open DAWNFILE`
  - makes a `.prim` file suitable for viewing in the DAWN renderer
- `/vis/open DAWN`
  - establishes a direct connection between Geant4 and the DAWN renderer
  - The DAWN renderer must be linked in at the time that you link Geant4 itself
  - See details in the Application Guide



# DAWNCUT and DAVID

- A standalone program, DAWNCUT, can perform a planar cut on a DAWN image.
  - DAWNCUT takes as input a .prim file and some cut parameters. Its output is a new .prim file to which the cut has been applied.
- Another standalone program, DAVID, can show you any volume overlap errors in your geometry.
  - DAVID takes as input a .prim file and outputs a new .prim file in which overlapping volumes have been highlighted.
- Details at <http://geant4.kek.jp/~tanaka/>



# HepRep and DAWN work through Files

- With HepRep and DAWN, Geant4 creates a file:

## Example .heprep File

```
<heprep
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="HepRep.xsd">
<layerorder="Detector, Event, CalHit, Trajectory, TrajectoryPoint, Hit"/>
<typetree name="G4GeometryTypes" version="1.0">
  <type name="Detector">
    <attvalue name="Layer" showLabel="NONE" type="String" value="Detector"/>
    <attdef category="Physics" desc="Logical Volume" extra="" name="LVol"/>
    <attdef category="Physics" desc="Material Name" extra="" name="Material"/>
    <type name="Detector/World">
      <type name="Detector/World/Calorimeter">
        <type name="Detector/World/Calorimeter/Layer">
          <type name="Detector/World/Calorimeter/Layer/Lead">
            </type>
          </type>
        </type>
      </type>
    </type>
  </typetree>
<typetree name="G4EventTypes" version="1.0">
  <type name="Event">
    <attvalue name="Layer" showLabel="NONE" type="String" value="Event"/>
    <type name="Event/Trajectory">
```

## Example .prim File

```
##G4.PRIM-FORMAT-2.4

##### List of primitives 1 #####
/BoundingBox -1.0 -1.0 -5.0 8.0 4.0 6.0
!SetCamera
!OpenDevice
!BeginModeling

# Box
/Origin 0.0 0.0 0.0
/ColorRGB 1.0 0.0 0.0
/Box 0.5 2.0 4.5

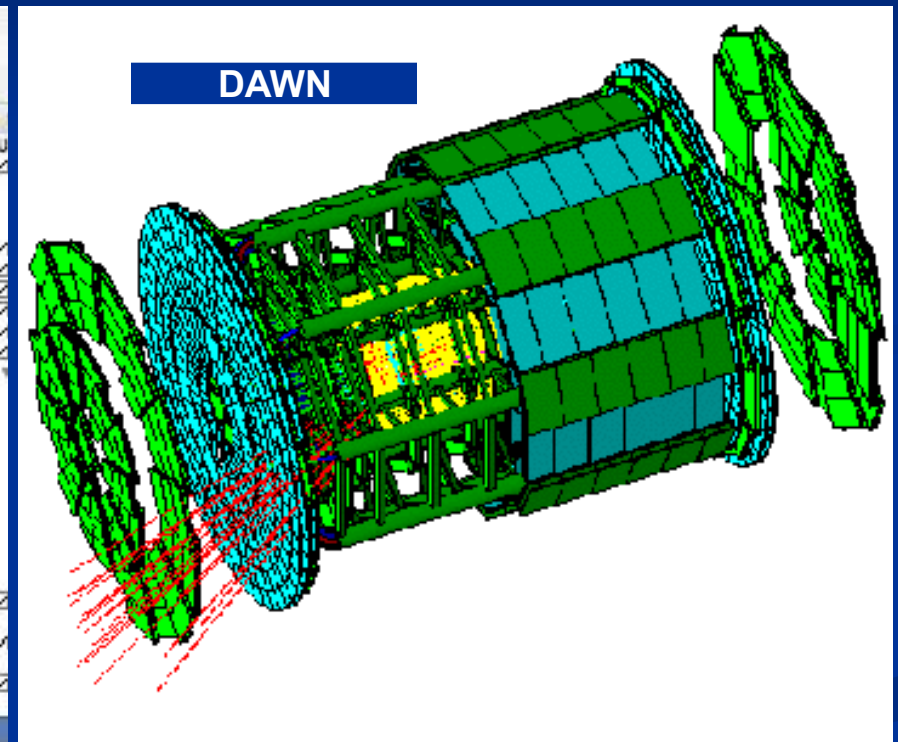
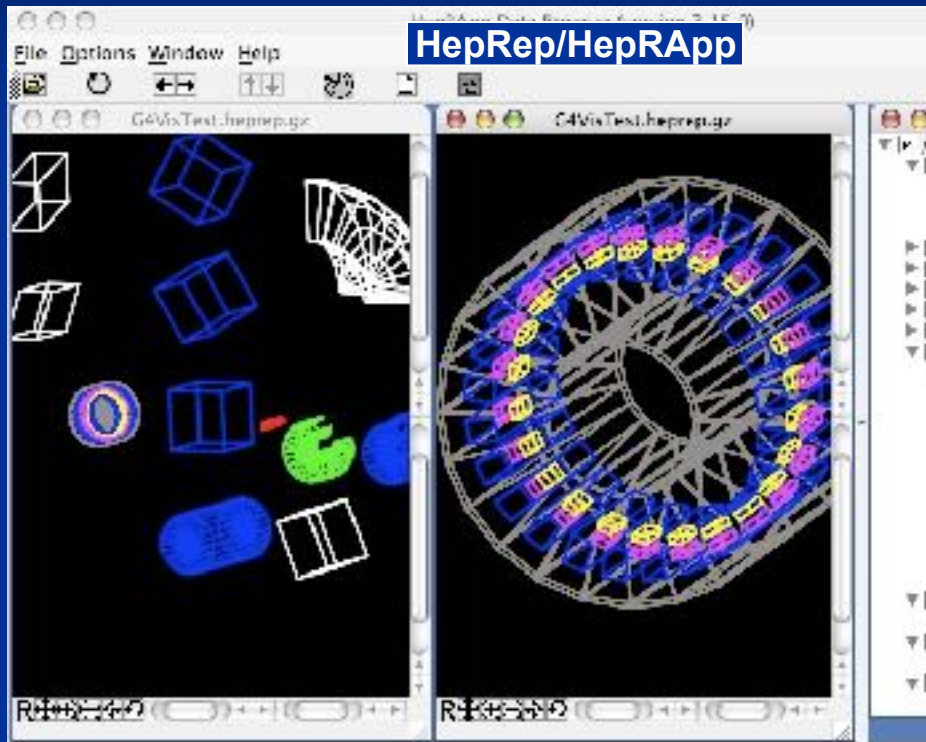
# Column
/Origin 4.0 0.0 0.0
/ColorRGB 0.0 1.0 0.0
/Ndiv 50
/Column 1.5 2.0

# Trd
/Origin 0.0 0.0 0.0
/ColorRGB 0.0 1.0 1.0
/Origin 7.0 0.0 0.0
/Trd 1 0.5 1 0.5 4

# Cone segment
/Origin 1.0 5.0 0.0
/ColorRGB 0.0 1.0 1.0
```

# HepRep and DAWN work through Files

- And you then run an application to visualize that file:



## HepRep and DAWN: complimentary file formats, each with its own strengths

### HepRep

- Hierarchical
- Simple Primitives
- General Purpose
- Representables have Attributes
- No Camera or Lighting Information

### DAWN

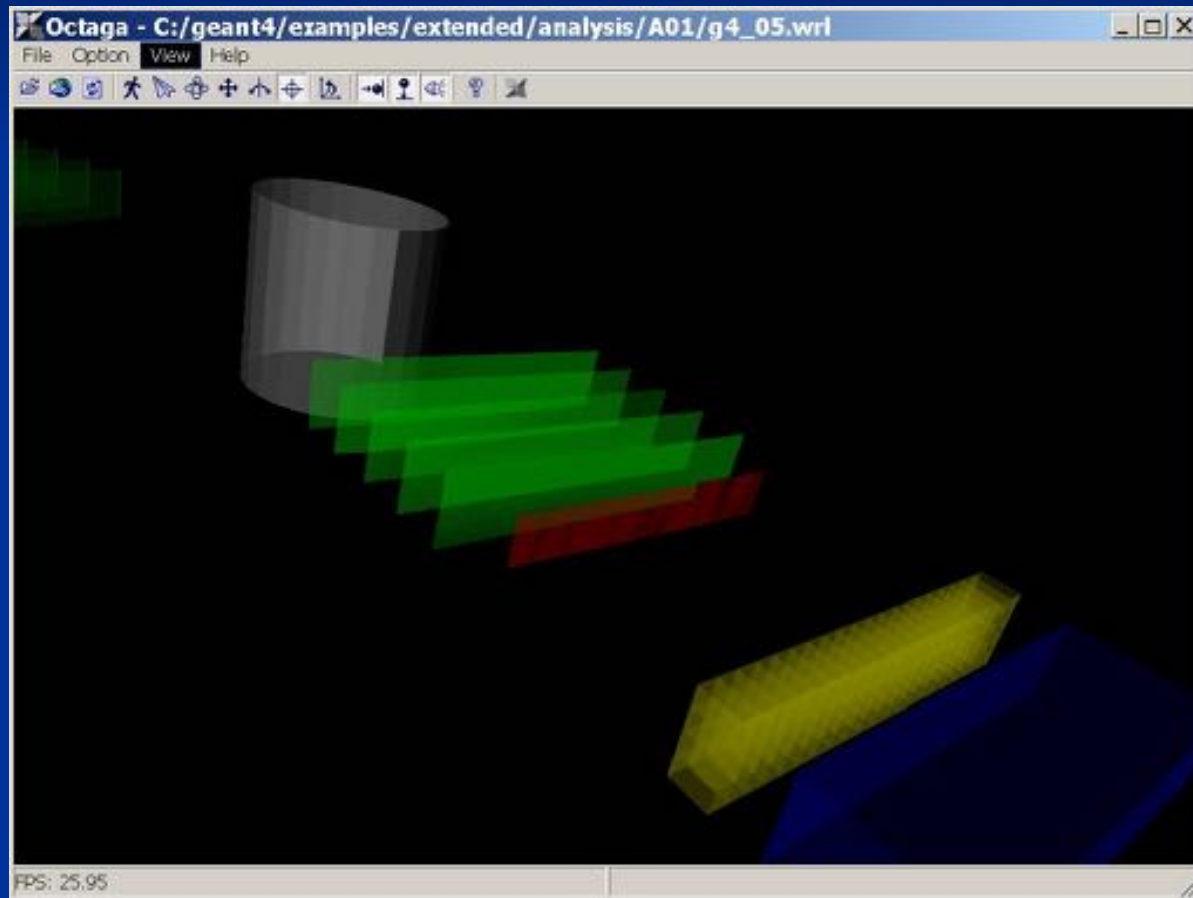
- Flat
- All Geant4 Primitives
- Just for Geant4
- No Attributes
- Camera and Lighting Information

# VRML

- `/vis/open VRML1FILE` or `/vis/open VRML2FILE`
- Features
  - Create a file to view in any VRML browser (some as web browser plug-ins).
  - Requires VRML browser (many different choices for different operating systems).
  - Rendered, photorealistic image with some interactive features
    - zoom, rotate, translate
  - Limited printing ability (pixel graphics, not vector graphics)
- Supported by Satoshi Tanaka

# VRML

- Geant4 creates VRML File
  - /vis/open VRML1FILE or /vis/open VRML2FILE
- View file in a VRML Browser
  - Many free options, for example, here is one from octaga.com





# VRML Additional Modes

- There are actually 4 VRML drivers:
  - VRML1FILE
    - makes a file of VRML version 1
  - VRML2FILE
    - makes a file of VRML version 2
  - VRML1
    - establishes a direction connection between Geant4 and your VRML1 renderer
    - the VRML1 renderer must be linked in at the time that you link your Geant4 application
  - VRML2
    - establishes a direction connection between Geant4 and your VRML2 renderer
    - the VRML2 renderer must be linked in at the time that you link your Geant4 application
  - See details in the Application Developer's Guide



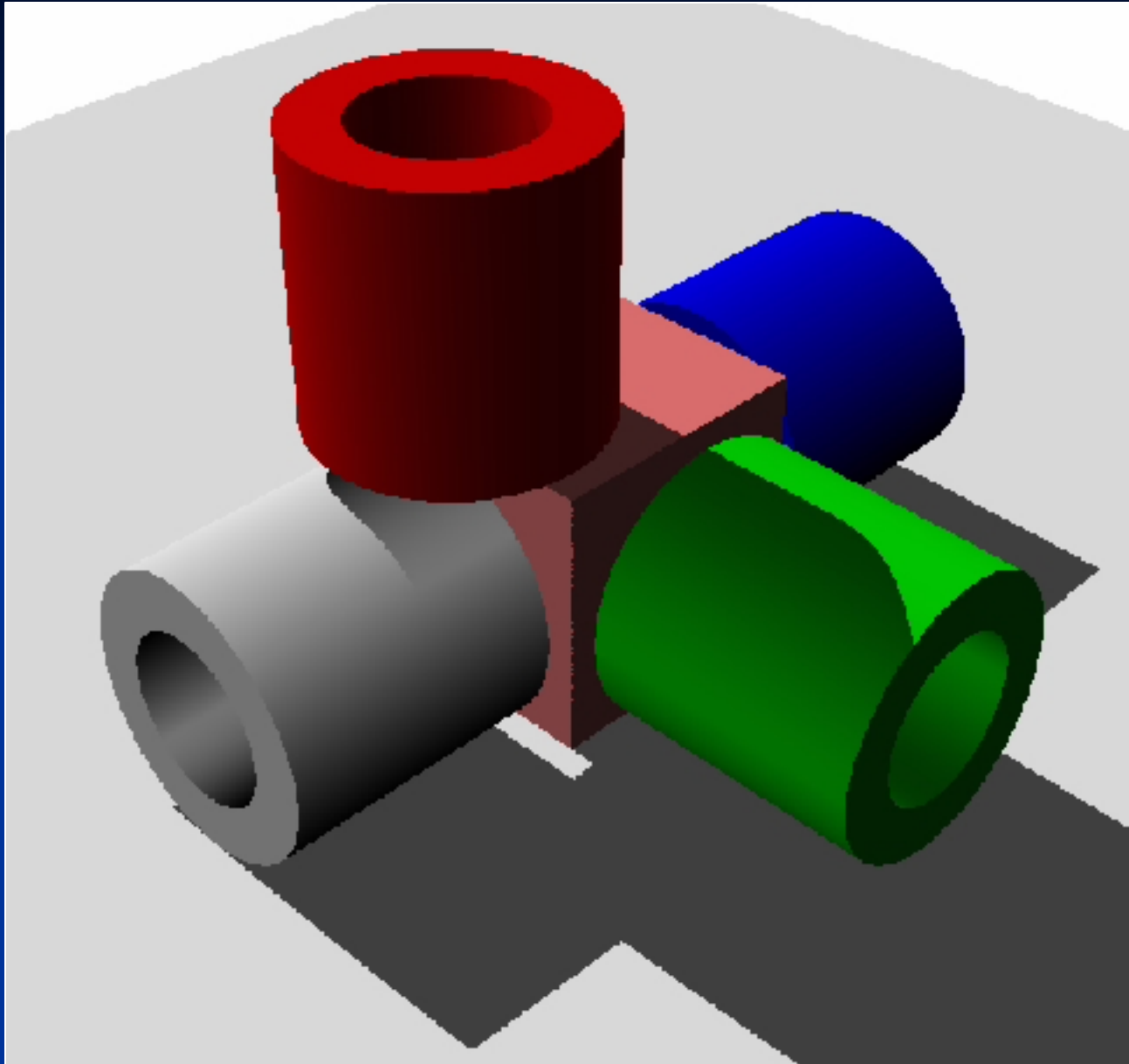
# RayTracer

- /vis/open RayTracer
- Features
  - Create a jpeg file  
(and with RayTracerX option, also draws to x window)
  - Forms image by using Geant4's own tracking to follow photons through the detector
  - Can show geometry but not trajectories
  - Can render any geometry that Geant4 can handle (such as Boolean solids) - no other Vis driver can handle every case
  - Supports shadows, transparency and mirrored surfaces
- Supported by Makoto Asai and John Allison

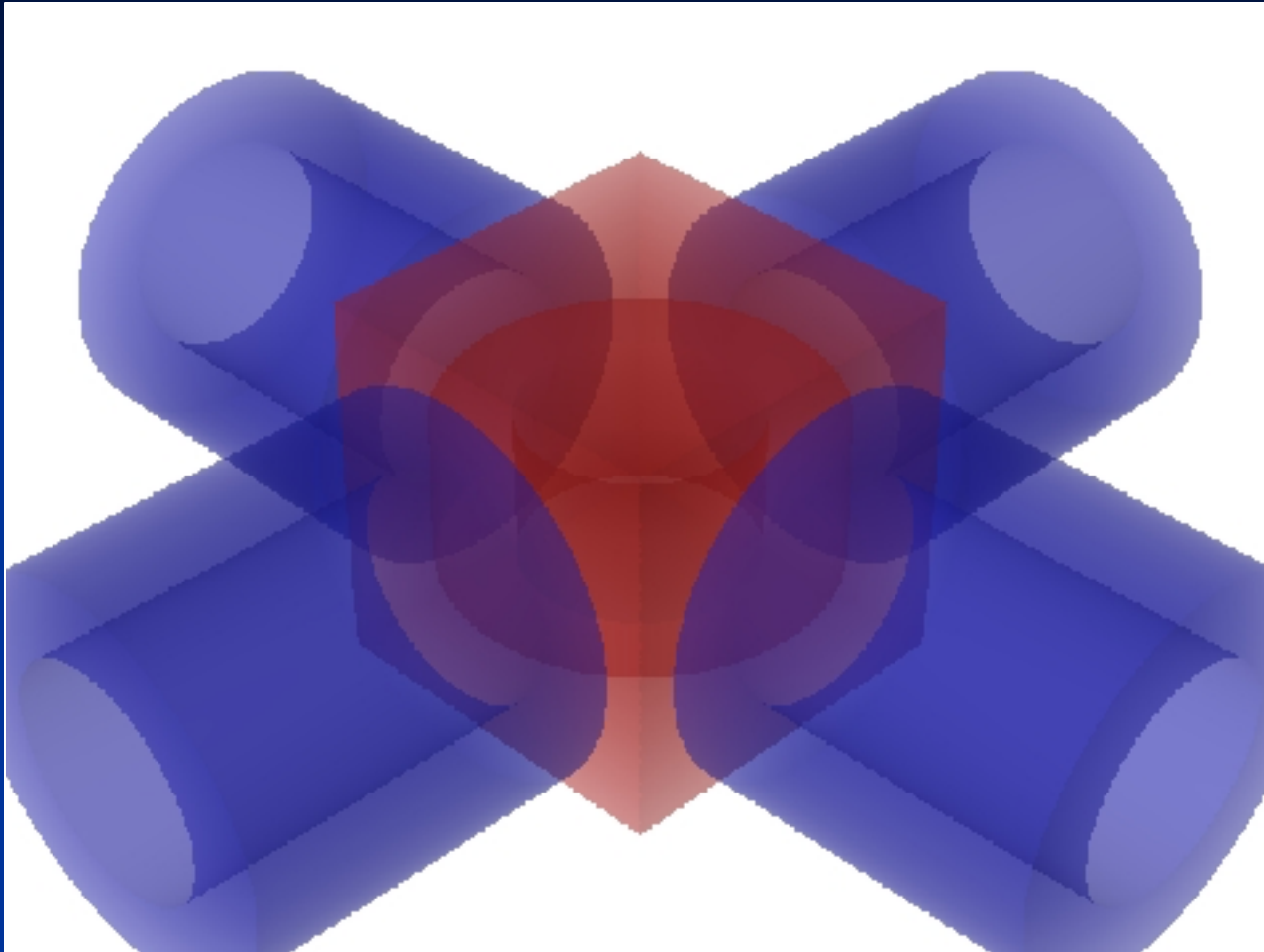
# RayTracer

- RayTracer works by using Geant4's own tracking to shoot photons through the detector onto a sensitive plane. The resulting image is presented as a jpeg file.
  - /vis/open RayTracer
- Some pieces of geometries may fail to show up in other visualization drivers (due to algorithms those drivers use to compute visualizable shapes and polygons), but RayTracer can handle any geometry that the Geant4 navigator can handle.
- RayTracer can not be used to visualize Trajectories.
- Commands:
  - 1) trace \* Start the ray tracing.
  - 2) column \* Define the number of horizontal pixels.
  - 3) row \* Define the number of vertical pixels.
  - 4) target \* Define the center position of the target.
  - 5) eyePosition \* Define the eye position.
  - 6) lightDirection \* Define the direction of illumination light.
  - 7) span \* Define the angle per 100 pixels.
  - 8) headAngle \* Define the head direction.
  - 9) attenuation \* Define the attenuation length for transparent material.
  - 10) distortion \* Distortion effect of the fish eye lens.
  - 11) ignoreTransparency \* Ignore transparency even if the alpha of G4Colour < 1
  - 12) backgroundColour \* Set background colour: red green blue: range 0.->1.

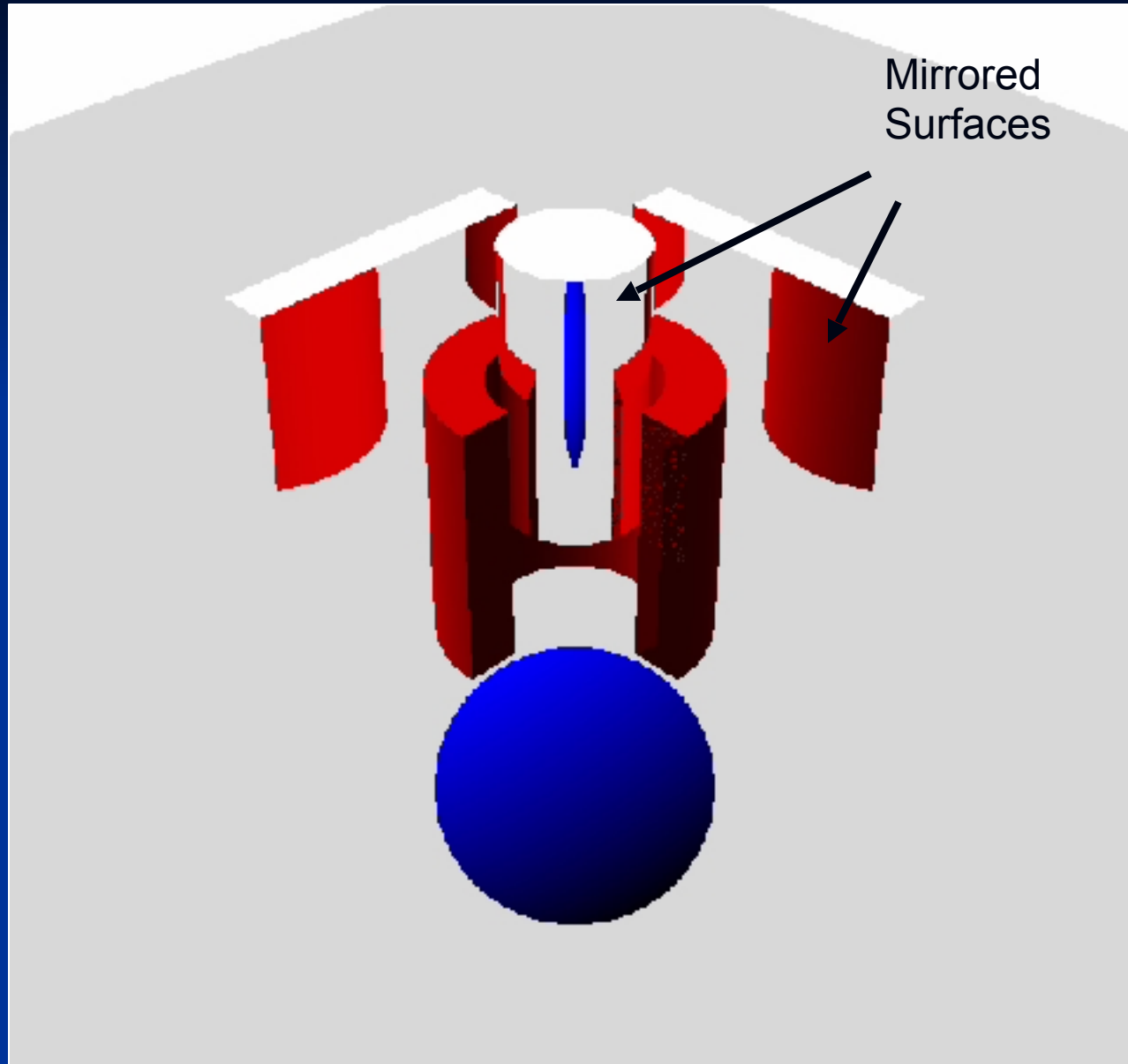
# RayTracer Shows Shadows



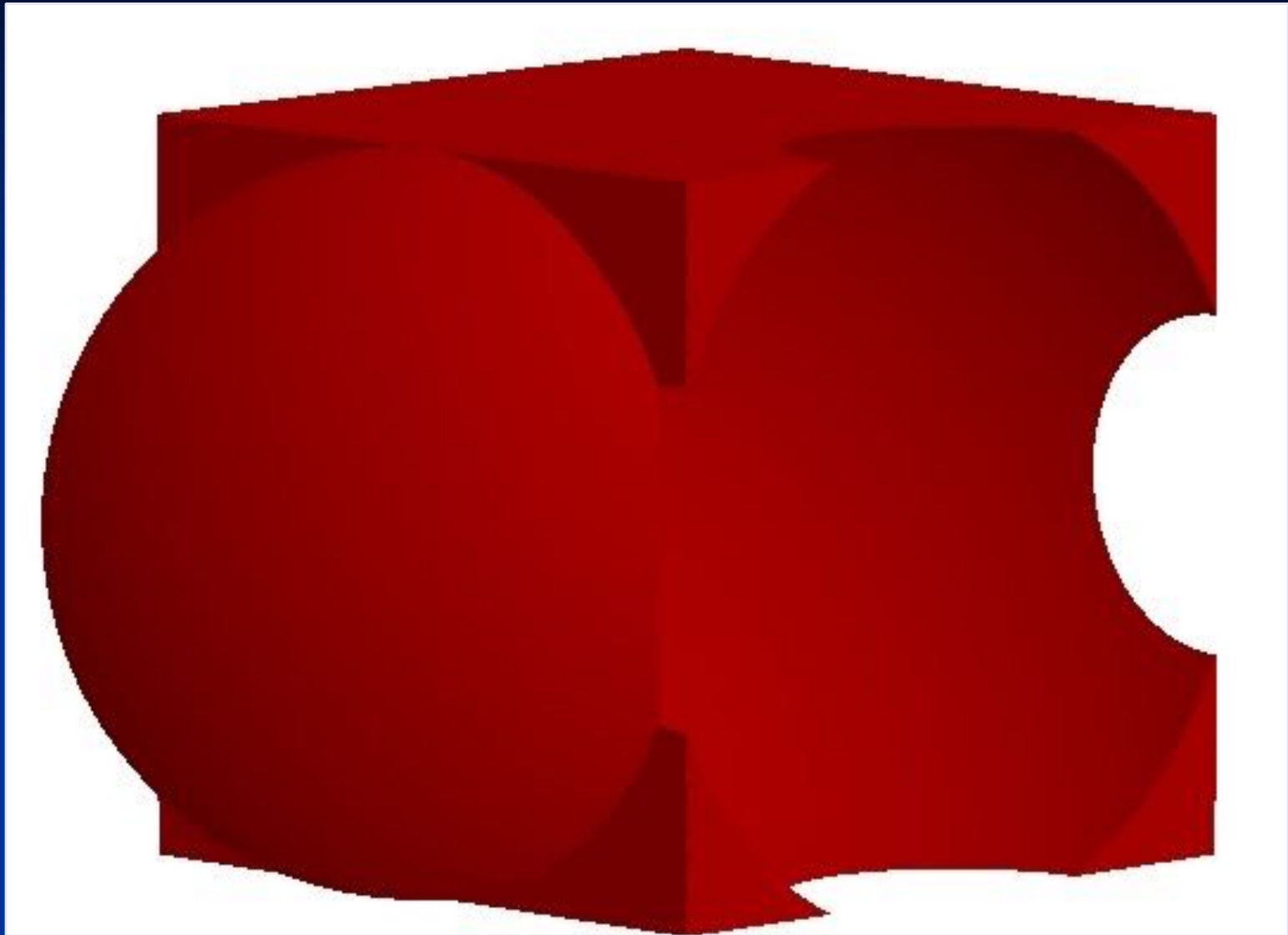
# RayTracer Supports Transparency



# RayTracer Handles Mirrored Surfaces



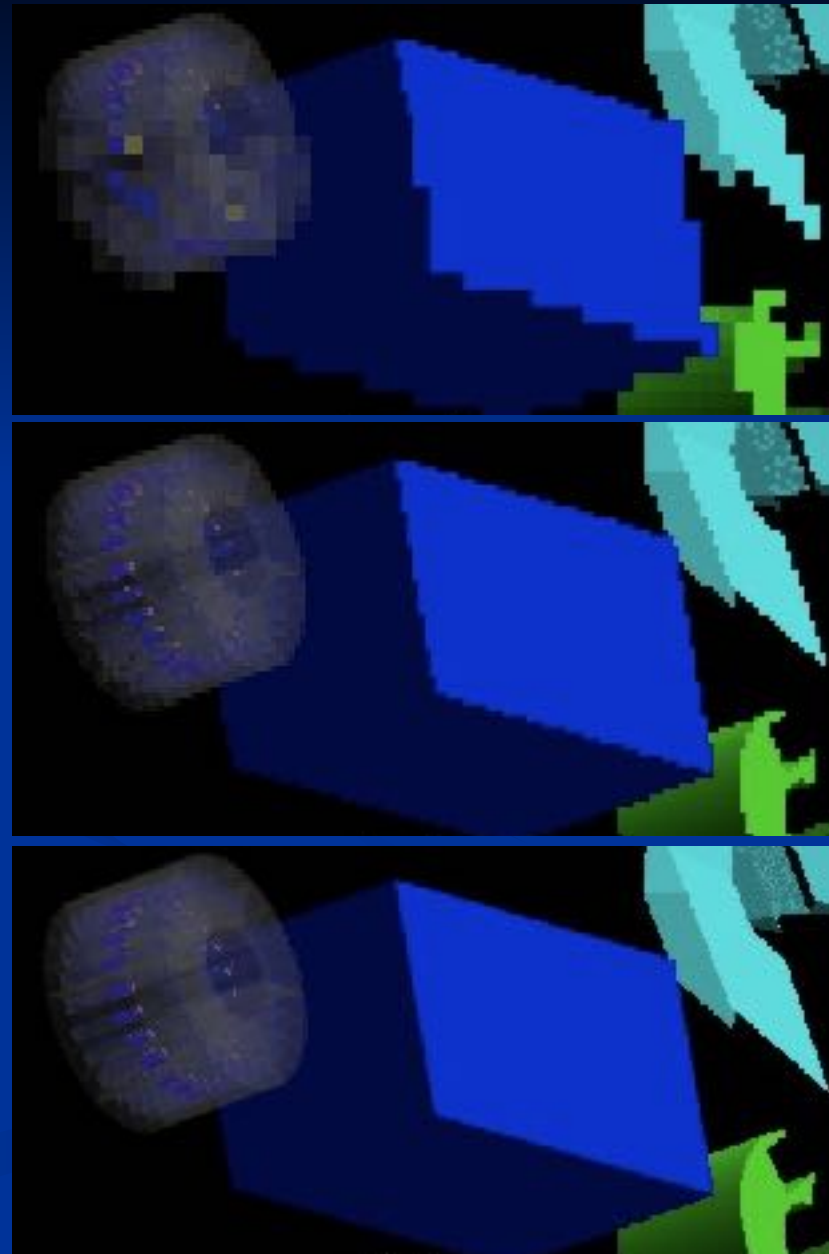
# RayTracer Handles Boolean Solids





# RayTracerX

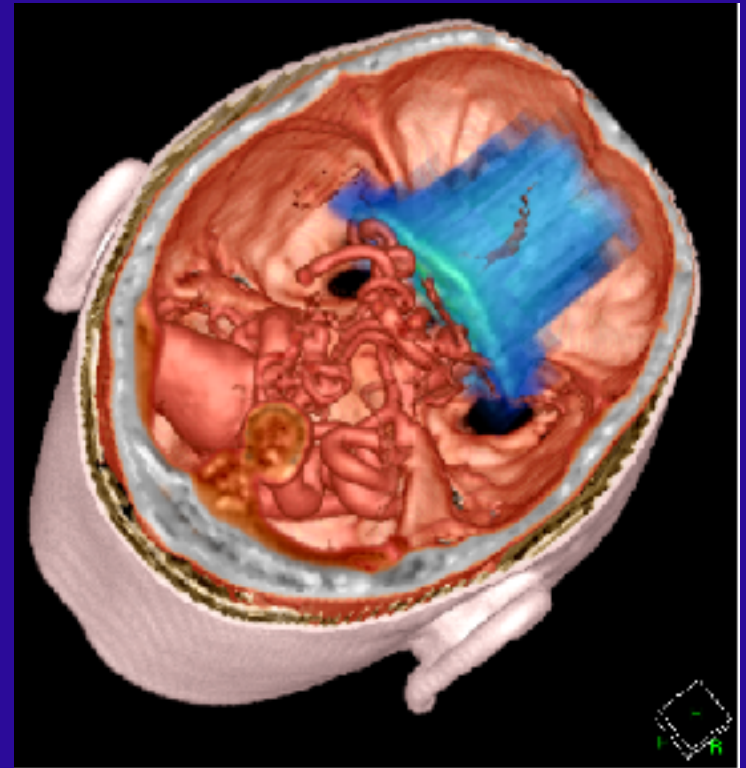
- New since Geant4.8.0
- In addition to
  - `/vis/open RayTracer`
- You have the option of
  - `/vis/open RayTracerX`
- Builds same jpeg file as RayTracer, but simultaneously renders to screen so you can watch as rendering grows progressively smoother.
- Means you can abort and retry the rendering with different view parameters without having to wait for the complete refinement of the image.



# gMocren

Great tool available for volume visualization

- From JST/CREST project (Japan) to improve Geant4 for medical physics
- Able to visualize:
  - Volume data  
(including overlay of more than one set)
  - Trajectories
  - Geometry
- Runs on:
  - Windows and Linux
  - Mac will likely happen soon
  - Based on a commercial package but offered freely to all Geant4 users
  - <http://geant4.kek.jp/gMocren>
  - Installation is straightforward, follow the Download link on the above page
    - First run gMocren's one-click installer
    - Then, inside C:\Program Files\gMocren\gtk, you will find the one-click installer for gtk

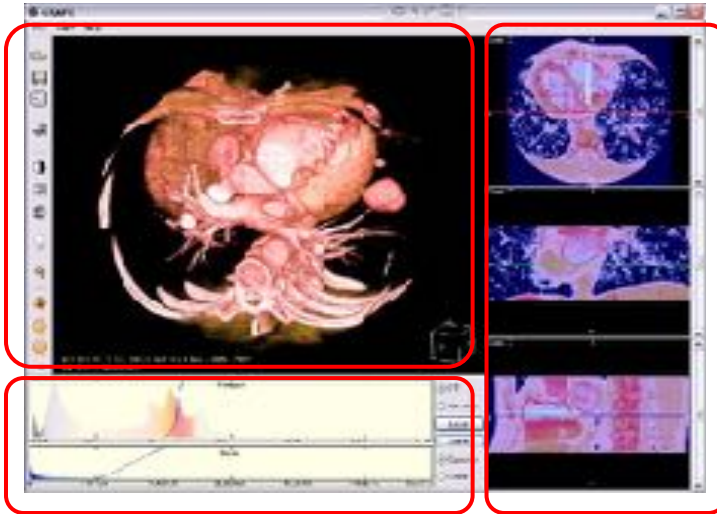


# gMocren : A Visualization Tool

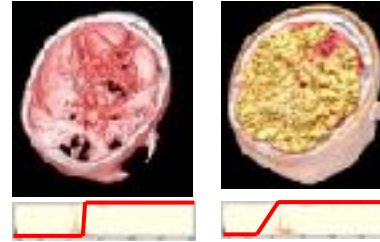
<http://geant4.kek.jp/gMocren/>

3D (ray casting)

2D (MPR)

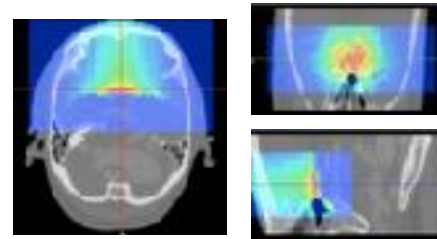


Opacity curve and color map editor



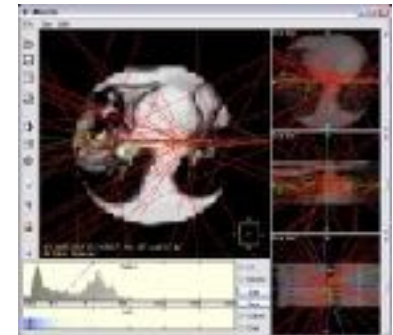
free hand or templates with WW&WL editing

Calculated dose distribution

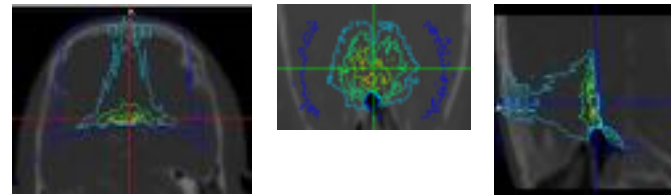


color mapping

Particle trajectories



Trajectory information in the simulation is available.



contour plot

Opacity curve and color map editor

## Functionality Requirements :

- To visualize
  - the modality image used by the simulation,
  - the calculated dose distribution and
  - the particle trajectories
- in an agreeable speed
- Transfer function editor
- Multi-platform

*gMocren and utility software are freely available.*

## Supported system :

- Windows 2k/XP or PC Linux OS
- Pentium 4 or faster
- more than 1 GB (recommend)

# Sample Data available for Download

Copyright © 2006-2008  
A. Kimura,  
K. Hasegawa,  
A. Saitoh, S. Tanaka,  
and JST.  
All rights reserved.

**Top Page**  
**News & Updates**  
**System Requirements**  
**Manual (English)**  
**Manual (Japanese)**  
**Tutorial (English)**  
**Tutorial (Japanese)**

**Sample Data**

**We provide sample data in this page.**

GDD files

- [a chest data \(5.2 MB\)](#)
- [a head data \(1.9 MB\)](#)
- [a chest data with color trajectories \(0.5 MB\)](#)

(to download a data file, right-click the link and click 'save as')

# gMocren File Driver

- To export Geant4 visualization to gMocren files:
  - `/vis/open gMocrenFile`
  - `/vis/scene/add/psHits`
  - `/vis/viewer/flush`
  - many other options available with `/score/draw ...` commands

# ASCIITree

- `/vis/open ATree`
- Features
  - Text dump of the geometry hierarchy
  - Not graphical
  - Control over level of detail to be dumped
  - Can calculate mass and volume of any hierarchy of volumes
- Supported by John Allison



# ASCIITree

- ASCIITREE is a visualization driver that is not actually graphical, but that dumps the hierarchy as a simple text tree.
  - `/vis/open ATree`
- `/vis/viewer/flush`
  - `"worldPhysical":0`
  - `"magneticPhysical":0`
  - `"firstArmPhysical":0`
  - `"hodoscope1Physical":0`
  - `"hodoscope1Physical":1` (repeated placement)
  - `"hodoscope1Physical":2` (repeated placement)
  - `"hodoscope1Physical":3` (repeated placement)
  - `"hodoscope1Physical":4` (repeated placement)
- Can be set to various levels of detail
  - `/vis/ASCIITree/verbose <verbosity>`
  - 0: prints physical volume name.
  - 1: prints logical volume name.
  - 2: prints solid name and type.
  - 3: prints volume and density of solid.
  - 4: calculates and prints mass(es) of volume(s) in scene.
  - By default, shows only daughters of first placement and not repeat replicas.
  - Add 10 to the above to also show repeated placements and replicas.

# ASCIITree: Calculate Volume and Mass

- At verbosity level 4, ASCIITree calculates the mass of the complete geometry tree taking into account daughters up to the depth specified for each physical volume.
- The calculation involves subtracting the mass of that part of the mother that is occupied by each daughter and then adding the mass of the daughter, and so on down the hierarchy.
- `/vis/ASCIITree/Verbose 4`
- `/vis/viewer/flush`
- `"HadCalorimeterPhysical":0 / "HadCalorimeterLogical" / "HadCalorimeterBox"(G4Box), 1.8 m3, 11.35 g/cm3`
  - `"HadCalColumnPhysical":-1 (10 replicas) / "HadCalColumnLogical" / "HadCalColumnBox"(G4Box), 180000 cm3, 11.35 g/cm3`
    - `"HadCalCellPhysical":-1 (2 replicas) / "HadCalCellLogical" / "HadCalCellBox"(G4Box), 90000 cm3, 11.35 g/cm3`
      - `"HadCalLayerPhysical":-1 (20 replicas) / "HadCalLayerLogical" / "HadCalLayerBox"(G4Box), 4500 cm3, 11.35 g/cm3`
        - `"HadCalScintiPhysical":0 / "HadCalScintiLogical" / "HadCalScintiBox"(G4Box), 900 cm3, 1.032 g/cm3`
- Calculating mass(es)...
  - Overall volume of "worldPhysical":0, is 2400 m<sup>3</sup>
  - Mass of tree to unlimited depth is 22260.5 kg

# Summary and Further Resources

# Choose the Driver that Meets Your Needs

- If you want very responsive photorealistic graphics (and have the OpenGL libraries installed)
  - OpenGL is a good solution
- If you want the above plus more interactivity (and have Qt or OpenInventor libraries installed)
  - OpenGL Qt or OpenInventor are good solutions
- If you want GUI control, want to be able to pick on items to inquire about them (identity, momentum, etc.), perhaps want to render to vector formats, and a wireframe look will do
  - HepRep will meet your needs
- If you want to render highest quality photorealistic images for use in a poster or a technical design report, and you can live without quick rotate and zoom
  - DAWN is the way to go
- If you want to render to a 3D format that others can view in a web browser plug-ins
  - VRML is the way to go
- If you want to visualize a geometry that the other visualization drivers can't handle, or you need transparency or mirrors, and you don't need to visualize trajectories
  - RayTracer will do it
- If you want to visualize volume data, such as dose to tissue in medical physics
  - gMocren will do it
- If you just want to quickly check the geometry hierarchy, or if you want to calculate the volume or mass of any geometry hierarchy
  - ASCIITree will meet your needs

# Details

- Some drivers are always present, others require setting of environment variables (since they require external libraries)
  - The ./Configure procedure helps you set these variables
- You can explore the online command guidance to learn the extensive set of visualization commands
  - Just type “help” at the Geant4 idle prompt
- You can also add your own visualization driver.
  - Geant4’s visualization system is modular. By creating just three new classes, you can direct Geant4 information to your own visualization system.
- You can even open more than one driver at a time.
  - For example, do a quick check in OpenGL, then save a beautiful DAWN plot. See talk on Visualization Commands.



# Two Page Summary of Visualization Drivers (1)

- OpenGL
  - Control directly from Geant4
  - Uses GL libraries that are already included on most Linux and Windows systems
  - Rendered, photorealistic image with some interactive features
    - zoom, rotate, translate
  - Fast response (can usually exploit full potential of graphics hardware)
  - Print as pixel graphics or vector EPS
  - Additional interactive control if you have the Qt or Motif libraries installed and configured
  - Movies
- OpenInventor
  - Control from the OpenInventor GUI
  - Requires addition of OpenInventor libraries (freely available for most Linux and Windows systems).
  - Rendered, photorealistic image
  - Many interactive features
    - zoom, rotate, translate
    - click to “see inside” opaque volumes
    - click to show attributes (momentum, etc., dumps to standard output)
  - Fast response (can usually exploit full potential of graphics hardware)
  - Expanded printing ability (vector and pixel graphics)
- HepRep
  - Create a file to view in the HepRApp HepRep Browser, WIRED4 Jas Plugin or FRED Event Display
  - Requires one of the above browsers (freely available for all systems)
  - Wireframe or simple area fills (not photorealistic)
  - Many interactive features
    - zoom, rotate, translate
    - click to show attributes (momentum, etc.)
    - special projections (FishEye, etc.)
    - control visibility from hierarchical (tree) view of data
  - Hierarchical view of the geometry
  - Export to many vector graphic formats (PostScript, PDF, etc.)



# Two Page Summary of Visualization Drivers (2)

- DAWN
  - Create a file to view in the DAWN Renderer
  - Requires DAWN, available for all Linux and Windows systems.
  - Rendered, photorealistic image
  - No interactive features once at PostScript stage
  - Highest quality technical rendering - vector PostScript
  - View or print from your favorite PostScript application
- VRML
  - Create a file to view in any VRML browser (some as web browser plug-ins).
  - Requires VRML browser (many different choices for different operating systems).
  - Rendered, photorealistic image with some interactive features
    - zoom, rotate, translate
  - Limited printing ability (pixel graphics, not vector graphics)
- RayTracer
  - Create a jpeg file (and with RayTracerX option, also draws to x window)
  - Forms image by using Geant4's own tracking to follow photons through the detector
  - Can show geometry but not trajectories
  - Can render any geometry that Geant4 can handle (such as Boolean solids)
    - no other Vis driver can handle every case
  - Supports shadows, transparency and mirrored surfaces
- gMocren
  - Create a file to view in the gMocren browser.
  - Requires gMocren, available for all Linux and Windows systems (with Mac coming soon)
  - Can overlay patient scan data (from DICOM) with Geant4 geometry, trajectories and dose
- ASCII Tree
  - Text dump of the geometry hierarchy (not graphical)
  - Control over level of detail to be dumped
  - Can calculate mass and volume of any hierarchy of volumes

# Geant4 Visualization Resources

## Geant4 Installation Guides

➤ <http://geant4.slac.stanford.edu/installation>

## Hands on HepRApp Tutorial

➤ <http://geant4.slac.stanford.edu/Presentations/vis/G4HepRAppTutorial/G4HepRAppTutorial.html>

## Hands on DAWN Tutorial

➤ <http://geant4.slac.stanford.edu/Presentations/vis/G4DAWNTutorial/G4DAWNTutorial.html>

## Hands on OpenGL Tutorial

➤ <http://geant4.slac.stanford.edu/Presentations/vis/G4OpenGLTutorial/G4OpenGLTutorial.html>

## Geant4 Visualization Commands

➤ <http://geant4.slac.stanford.edu/Presentations/vis/G4VisCommands.ppt> (and .pdf)

## Geant4 Advanced Visualization

➤ <http://geant4.slac.stanford.edu/Presentations/vis/G4VisAdvanced.ppt> (and .pdf)

## How to Make a Movie

➤ <http://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.ppt> (and .pdf)

## Visualization Chapter of the Geant4 User's Guide for Application Developers

➤ <http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/>

## List of Visualization Commands:

➤ [http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/AllResources/Control/UIcommands/\\_vis\\_.html](http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/AllResources/Control/UIcommands/_vis_.html)

## For Questions or Comments: Geant4 Visualization Online Forum:

➤ <http://geant4-hn.slac.stanford.edu:5090/HyperNews/public/get/visualization.html>

# References

- OpenScientist Home Page  
<http://openscientist.lal.in2p3.fr>
- HepRep: a generic interface definition for HEP event display representables  
<http://www.slac.stanford.edu/~perl/heprep>
- HepRApp HepRep Browser  
<http://www.slac.stanford.edu/~perl/HepRApp>
- Wired4 JAS Plug-In  
<http://wired.freehep.org>
- Fred: oh no, another event display (a HepRep client)  
<http://www.fisica.uniud.it/~glast/FRED>
- DAWN Home Page  
[http://geant4.kek.jp/~tanaka/DAWN/About\\_DAWN.html](http://geant4.kek.jp/~tanaka/DAWN/About_DAWN.html)
- DAWNCUT Home Page  
[http://geant4.kek.jp/~tanaka/DAWN/About\\_DAWNCUT.html](http://geant4.kek.jp/~tanaka/DAWN/About_DAWNCUT.html)
- DAVID Home Page  
[http://geant4.kek.jp/~tanaka/DAWN/About\\_DAVID.html](http://geant4.kek.jp/~tanaka/DAWN/About_DAVID.html)
- Satoshi Tanaka's GEANT4 Ritsumeikan University Group Home Page (more information on DAWN, sample PRIM files, images, etc.)  
<http://geant4.kek.jp/~tanaka/>
- gMocren Home Page  
<http://geant4.kek.jp/gMocren>