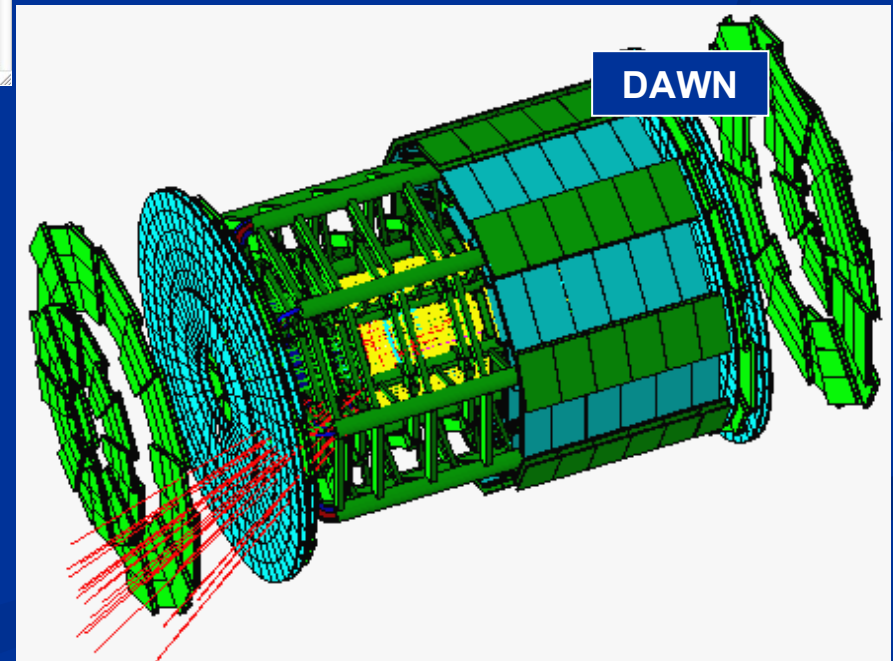
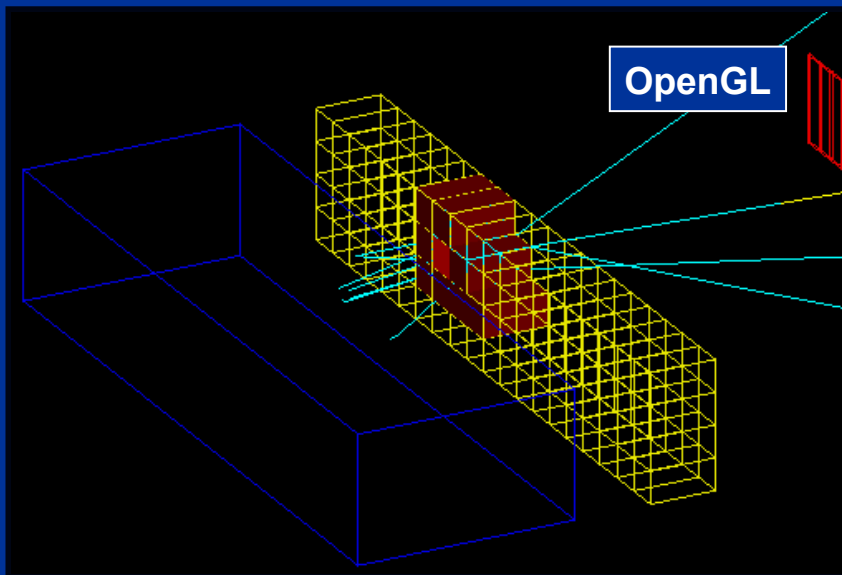
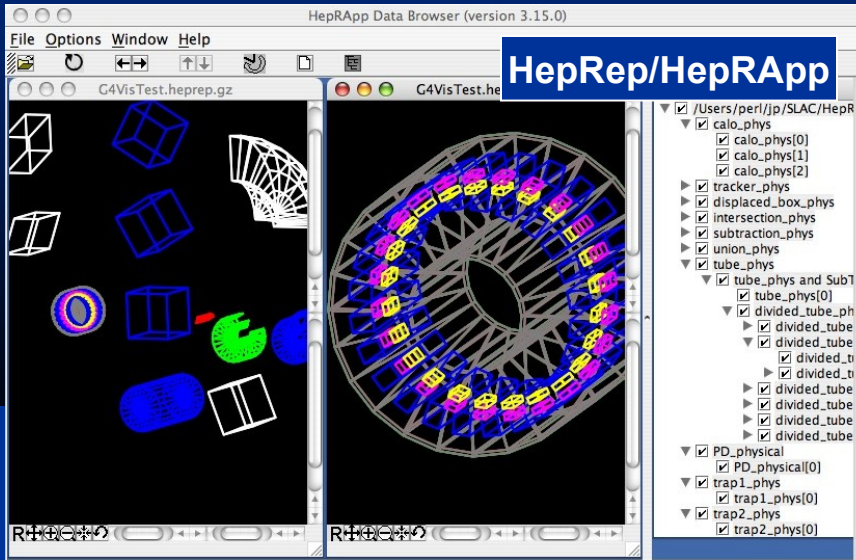


Introduction to Geant4 Visualization

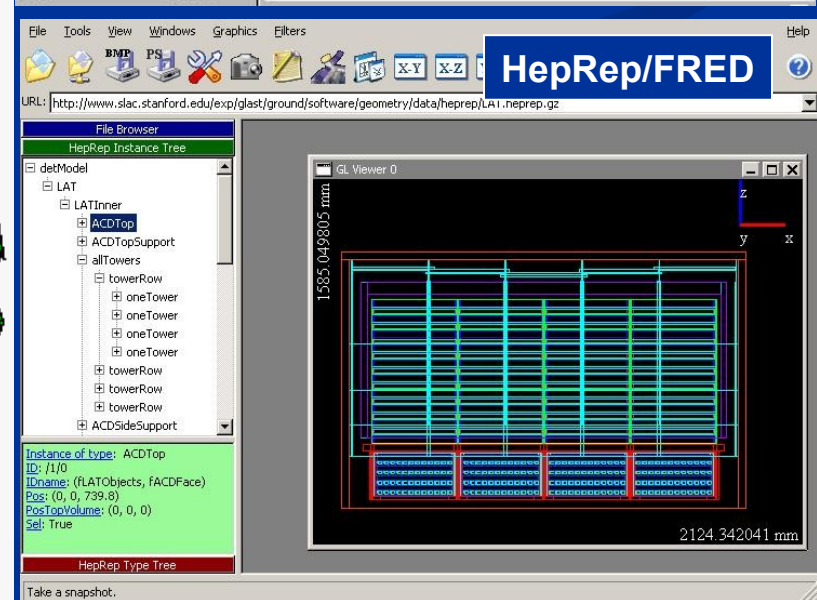
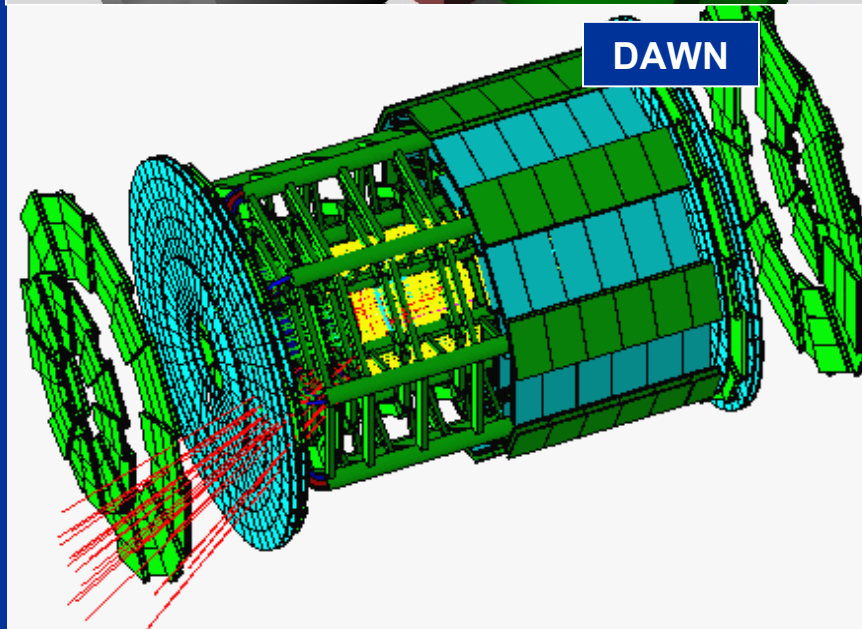
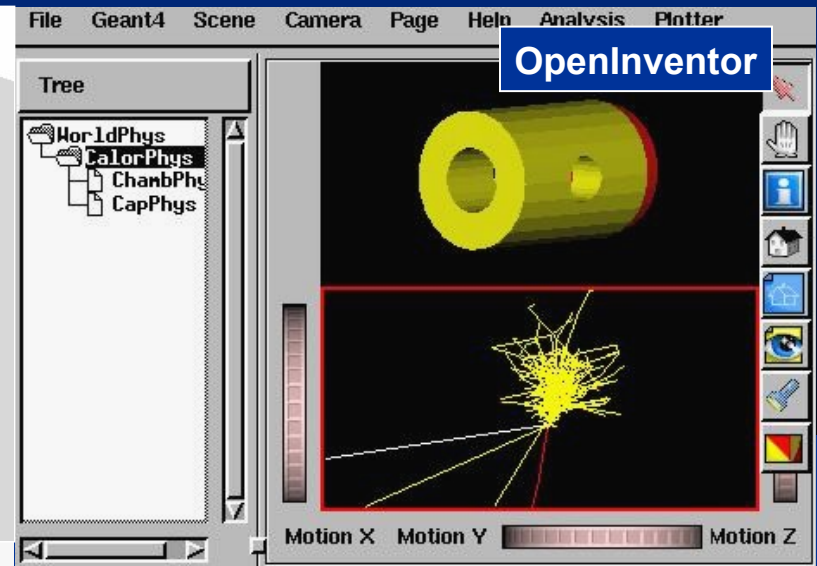
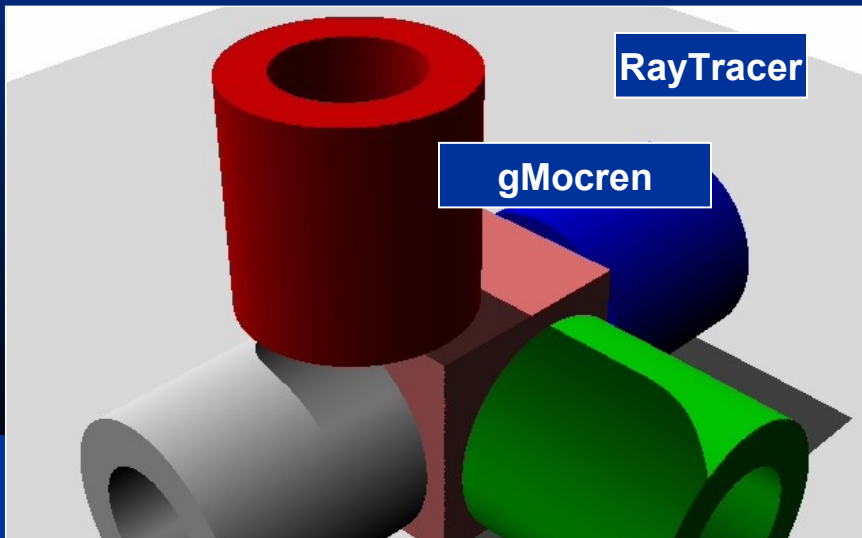
Geant4 v9.4.p01

Hans Wenzel, Fermilab

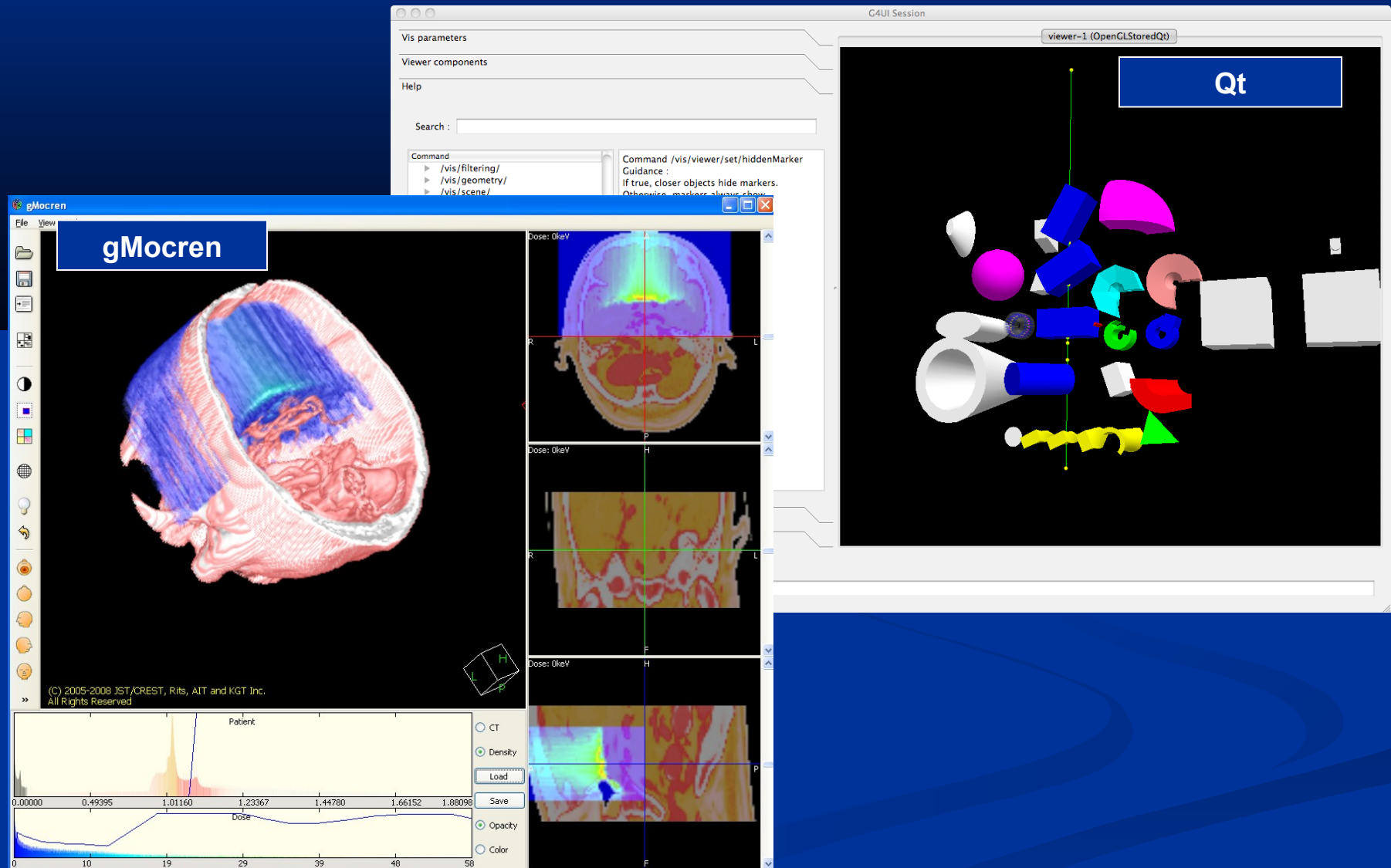
*So many options,
it needs three title
pages*



Introduction to Geant4 Visualization



Introduction to Geant4 Visualization



Thanks

Slides taken from Joseph Perls “Introduction to Geant 4 visualisation” from January 10.

Not covered: gdml and geometry visualisation based on gdml

In General

What Visualisation drivers will be available is configured when you build Geant 4 using the

`./Configure -build`

Command. Your system must be setup up to provide the appropriate libraries. Some visualisation drivers are always present since they don't require extra libraries.

The appropriate environmental variables are then set using the setup script `env.sh` that's created in the build process e.g. on my system:

`source /home/wenzel/geant4.9.4.p01/.config/bin/Linux-g++/env.sh`

To check what's available on your system:

`printenv | grep G4VIS`

e.g. on my System

```
G4VIS_BUILD_VRML_DRIVER=1
G4VIS_BUILD_OIX_DRIVER=1
G4VIS_USE_OPENGLQT=1
G4VIS_BUILD_OPENGLX_DRIVER=1
G4VIS_USE_DAWN=1
G4VIS_BUILD_RAYTRACERX_DRIVER=1
G4VIS_BUILD_OPENGLXM_DRIVER=1
G4VIS_USE_OPENGLX=1
G4VIS_USE_RAYTRACERX=1
G4VIS_USE_OIX=1
G4VIS_BUILD_DAWN_DRIVER=1
G4VIS_USE_VRML=1
G4VIS_USE_OPENGLXM=1
G4VIS_BUILD_OPENGLQT_DRIVER=1
```

Jump Start

more `$G4INSTALL/examples/novice/N03/vis.mac`

Good starting point to play with visualization By default the opengl visualization is used but by uncommenting one can choose the different drivers.

How this Document Fits with Other Tutorial Materials

- **Note: the URLs at the end of this presentation!**
- 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
- This presentation discusses eight visualization drivers:
 - OpenGL
 - OpenInventor
 - HepRep
 - DAWN
 - VRML
 - RayTracer
 - gMocren
 - ASCII Tree

What Can be Visualized

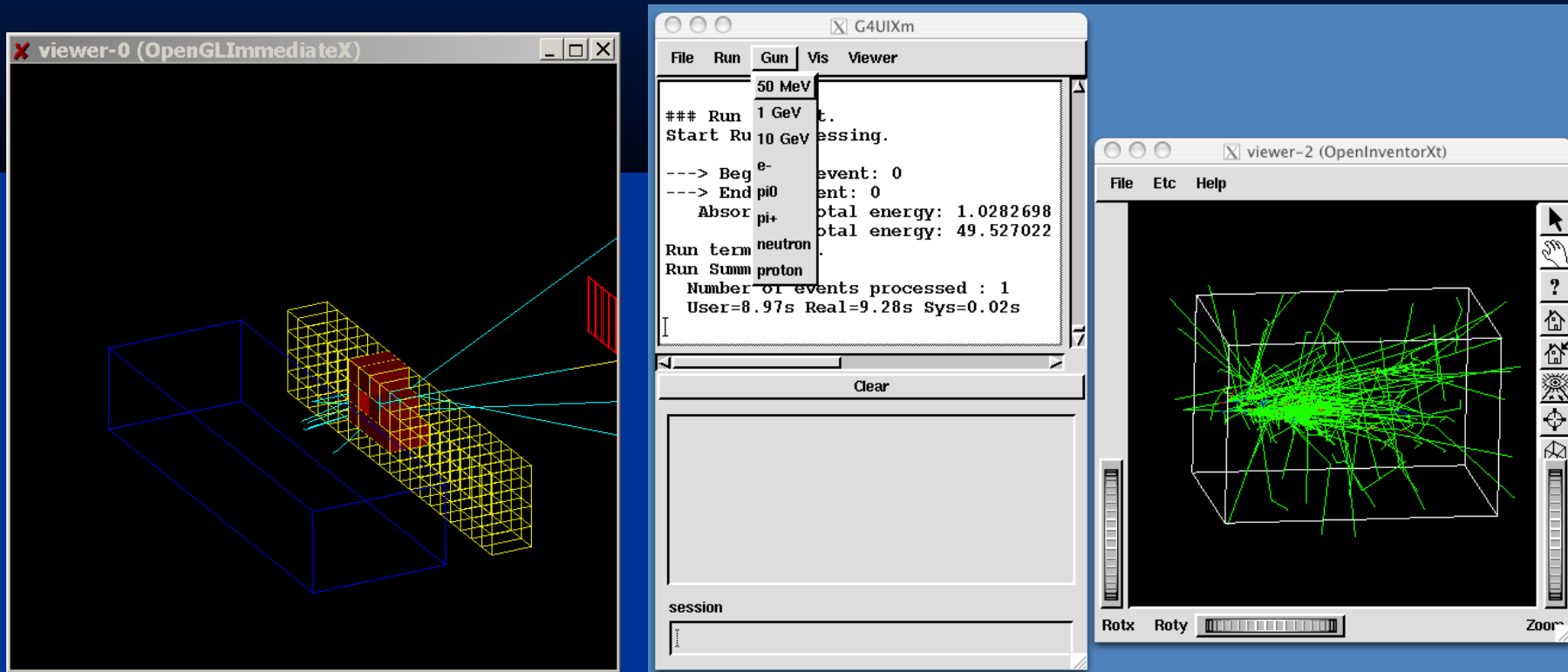
- Simulation data can be visualized:
 - Geometrical components
 - Particle trajectories and tracking steps
 - Hits of particles in the geometry
 - Scored energy, dose, etc.
- Other user defined objects can be visualized:
 - Polylines
 - such as coordinate axes
 - 3D Markers
 - such as eye guides
 - Text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

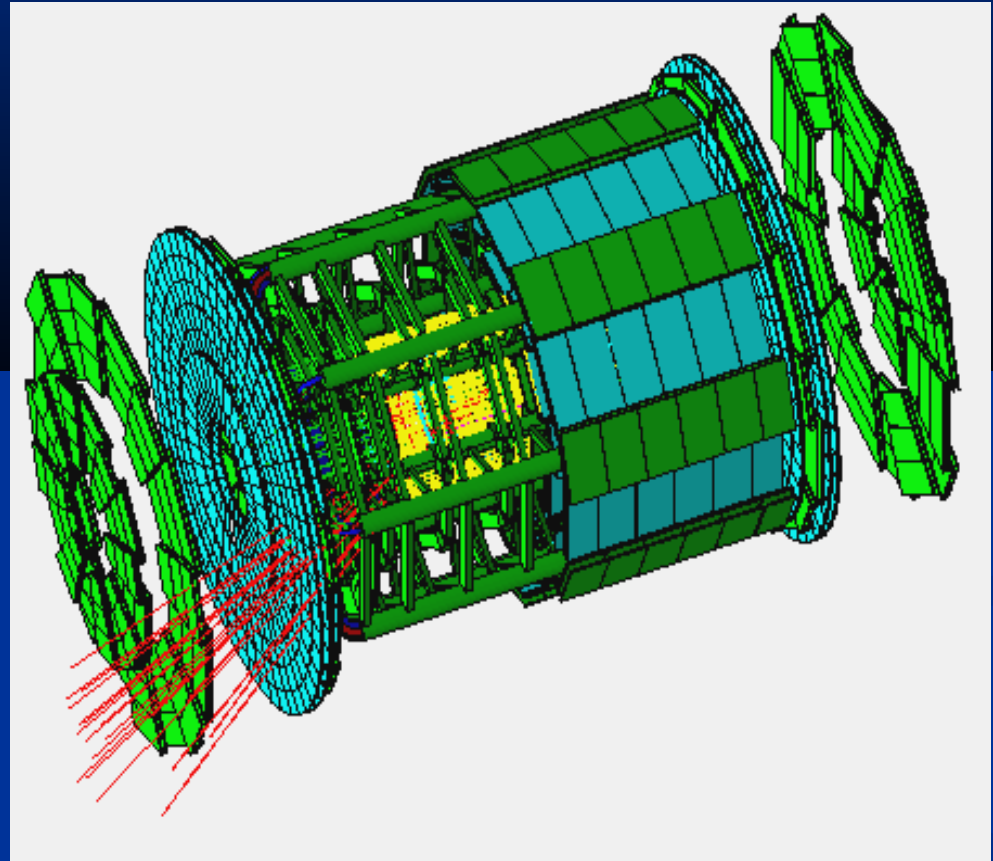
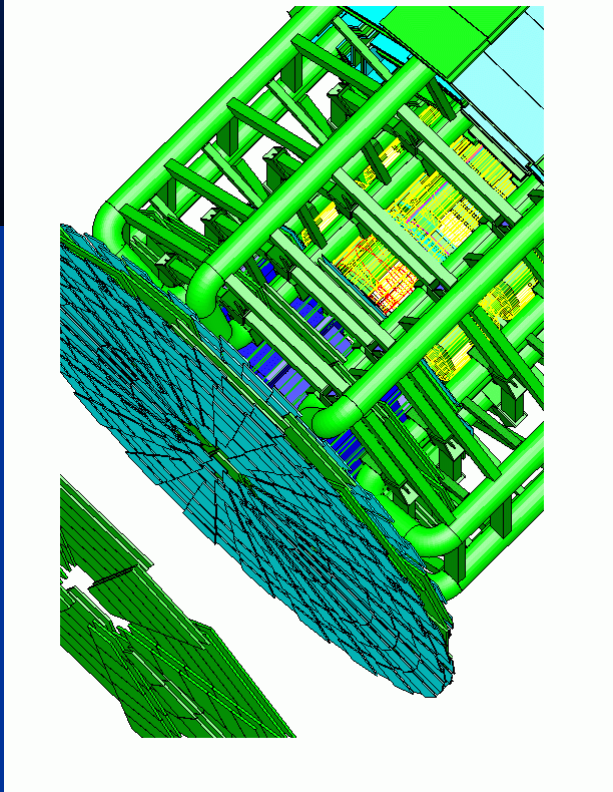
Quick response with flexible camera control to study geometries, trajectories and hits

- OpenGL, Qt and OpenInventor



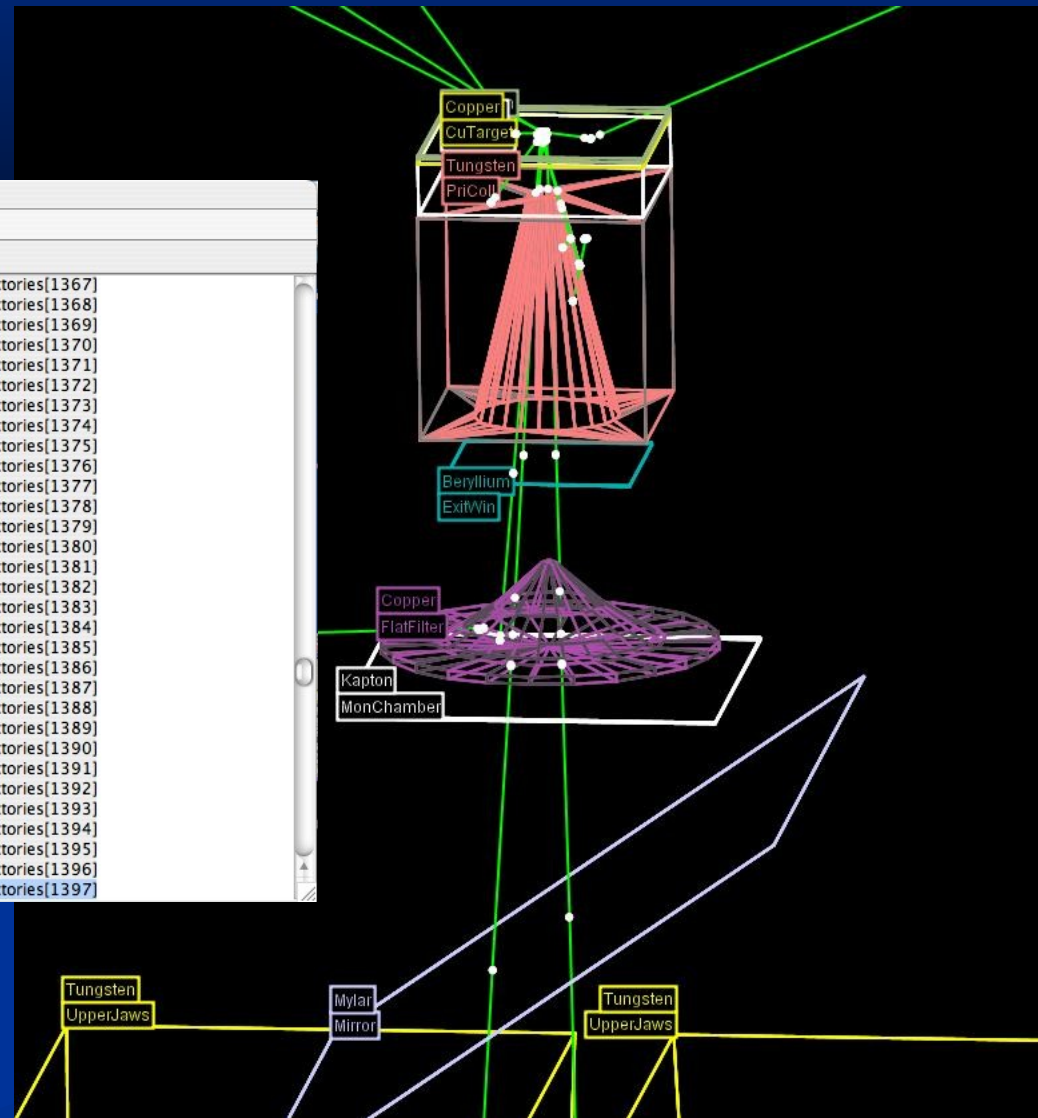
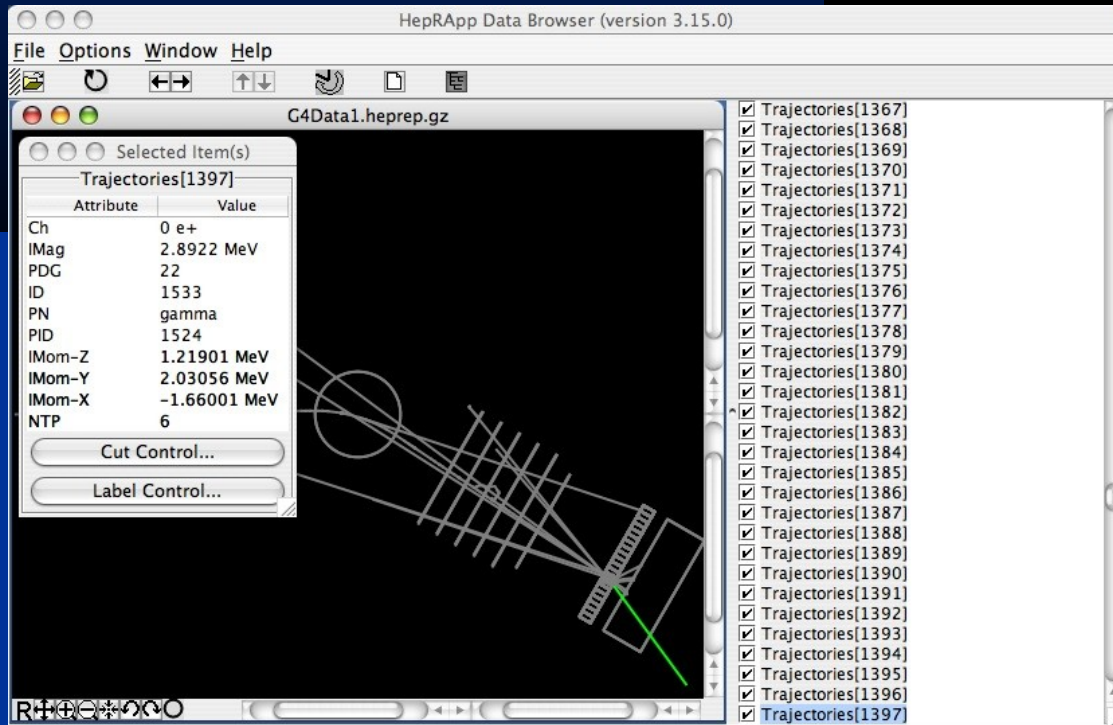
High-quality Output for Publications

- DAWN



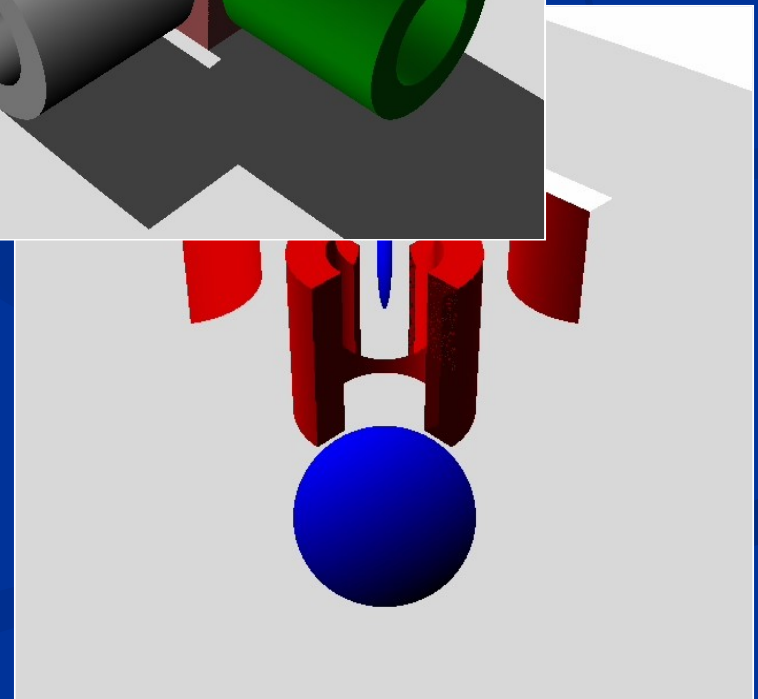
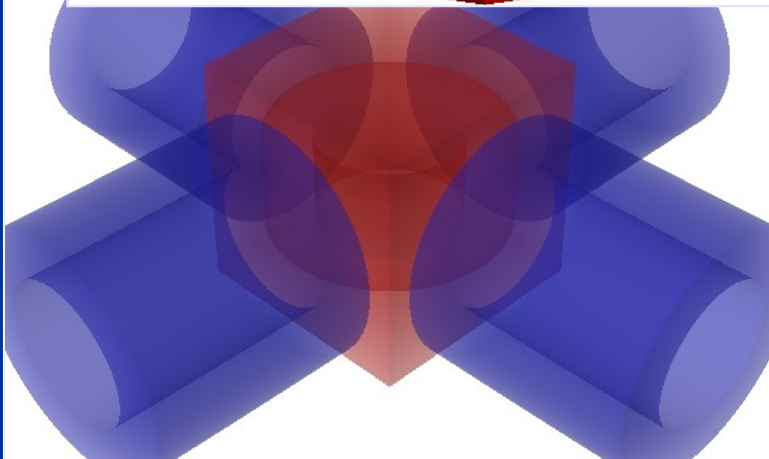
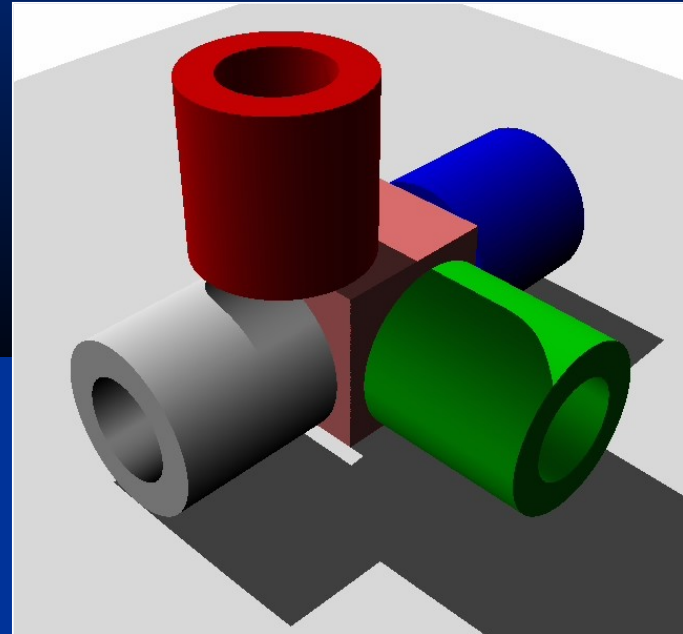
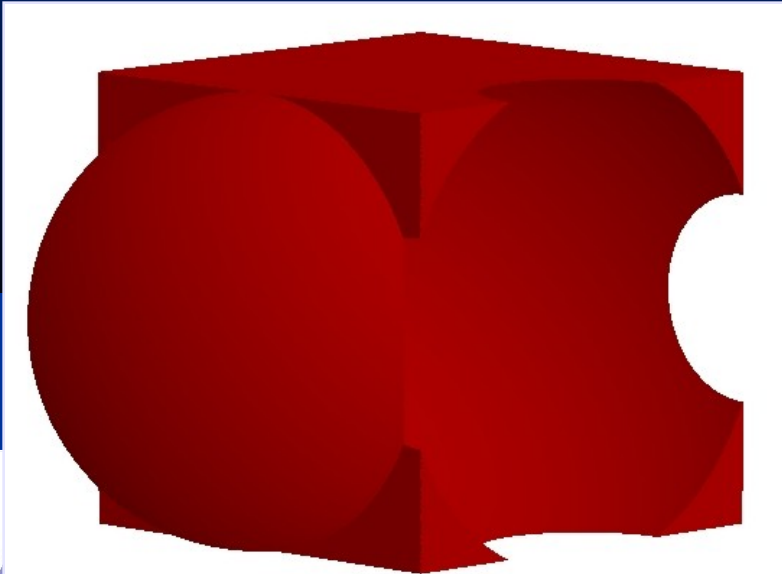
Interactive Picking to Get More Information on Visualized Objects

■ HepRep



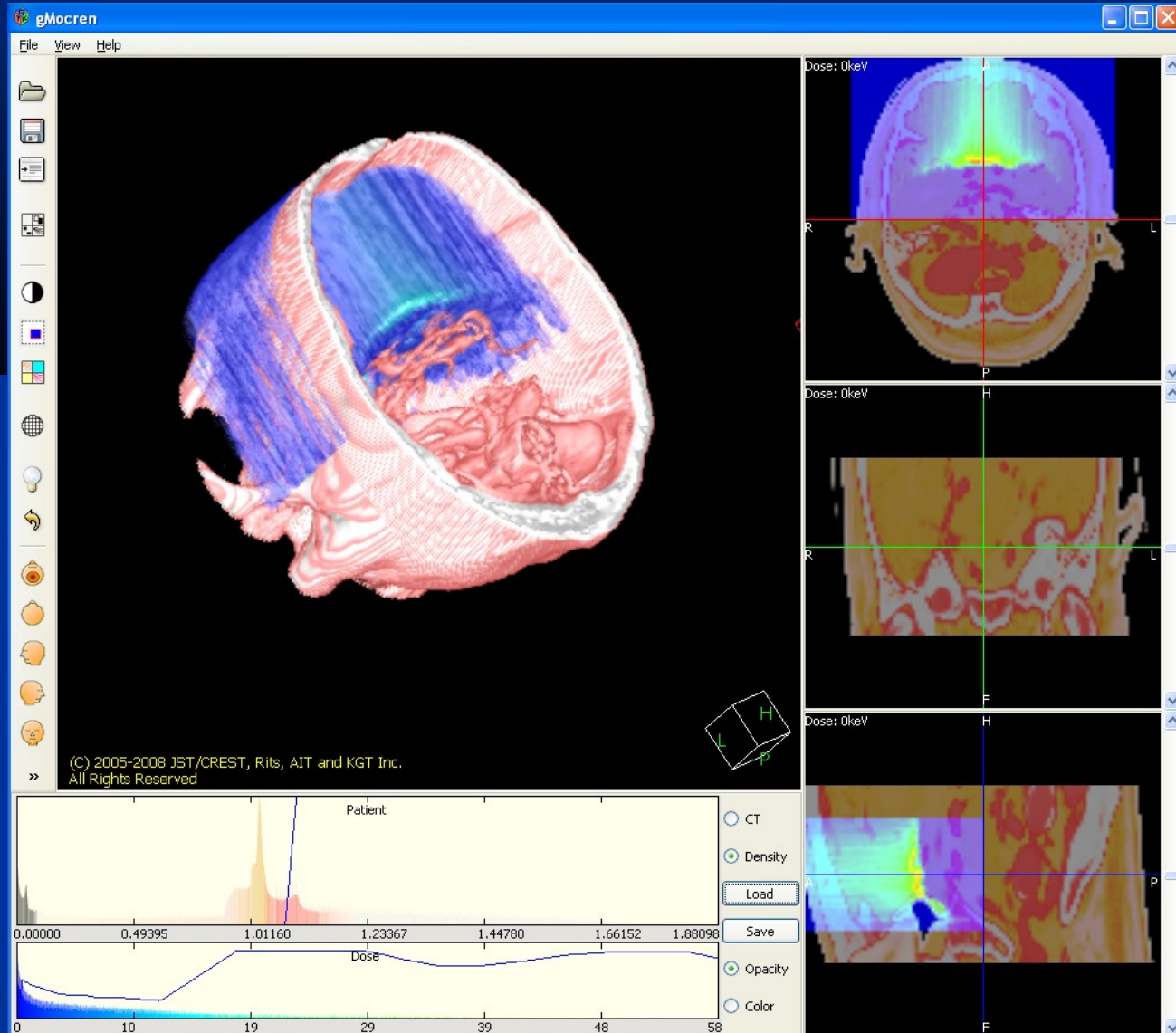
Understand Complex Boolean Solids and Transparent or Reflective Surfaces

- RayTracer



Visualize Volume Data, such as Dose

- gMocren



Understand Geometry Hierarchies

- ASCIITree, HepRep

- /vis/ASCIITree/Verbose 4

- /vis/viewer/flush

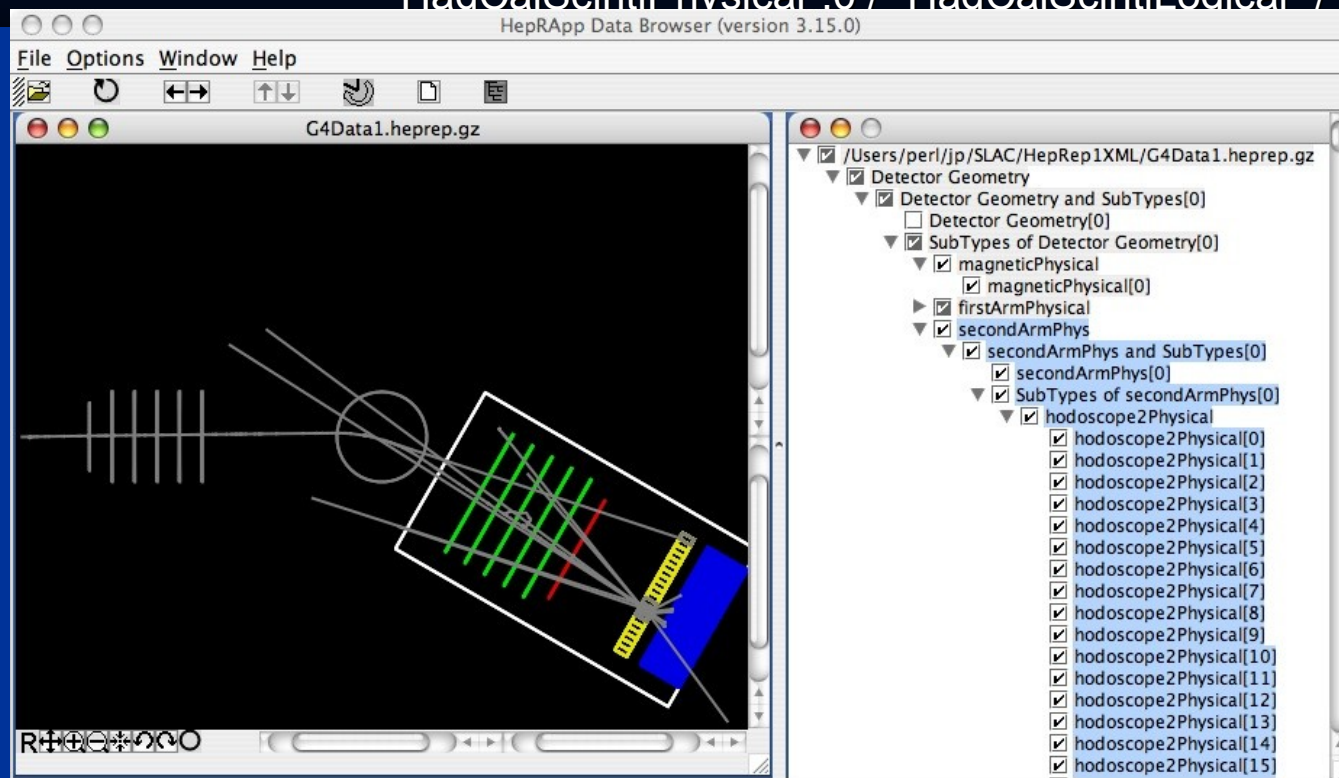
- "HadCalorimeterPhysical":0 / "HadCalorimeterLogical" / "HadCalorimeterBox"(G4Box), 1.8 m3 ,

- "HadCalColumnPhysical":-1 (10 replicas) / "HadCalColumnLogical" / "HadCalColumnBox"(G4Box)

- "HadCalCellPhysical":-1 (2 replicas) / "HadCalCellLogical" / "HadCalCellBox"(G4Box)

- "HadCalLayerPhysical":-1 (20 replicas) / "HadCalLayerLogical" / "HadCalLayerBox"(G4Box)

- "HadCalScintiPhysical":0 / "HadCalScintiLogical" / "HadCalScintiBox"(G4Box)



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
 - ASCIITree
- 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

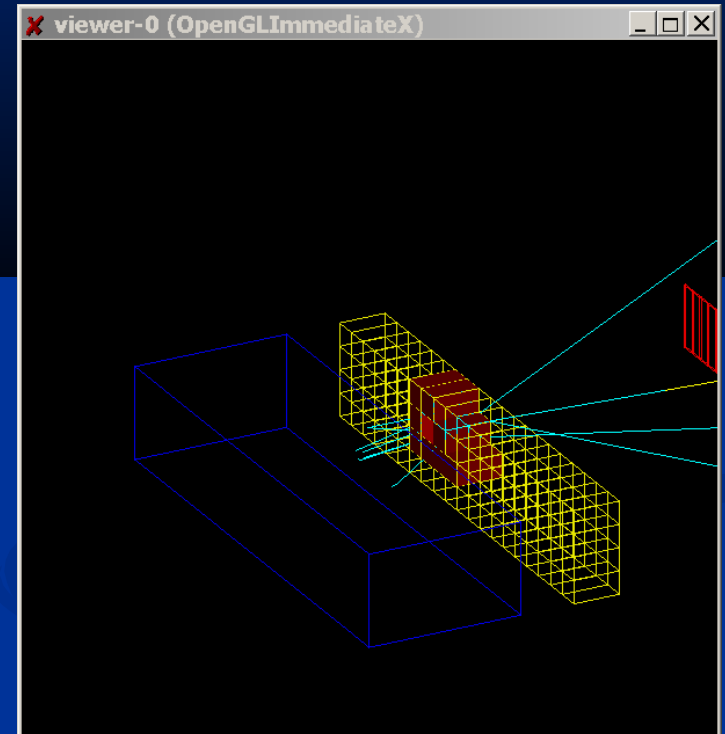
- Visualize a detector using the OpenGL visualization driver:
 - `/vis/open OGL`
 - `/vis/drawVolume`
- Visualize trajectories and hits for 10 events using the HepRep visualization driver:
 - `/vis/open HepRepFile`
 - `/vis/drawVolume`
 - `/vis/scene/add/trajectories`
 - `/vis/scene/add/hits`
 - `/run/beamOn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

Details on the Eight Visualization Drivers

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

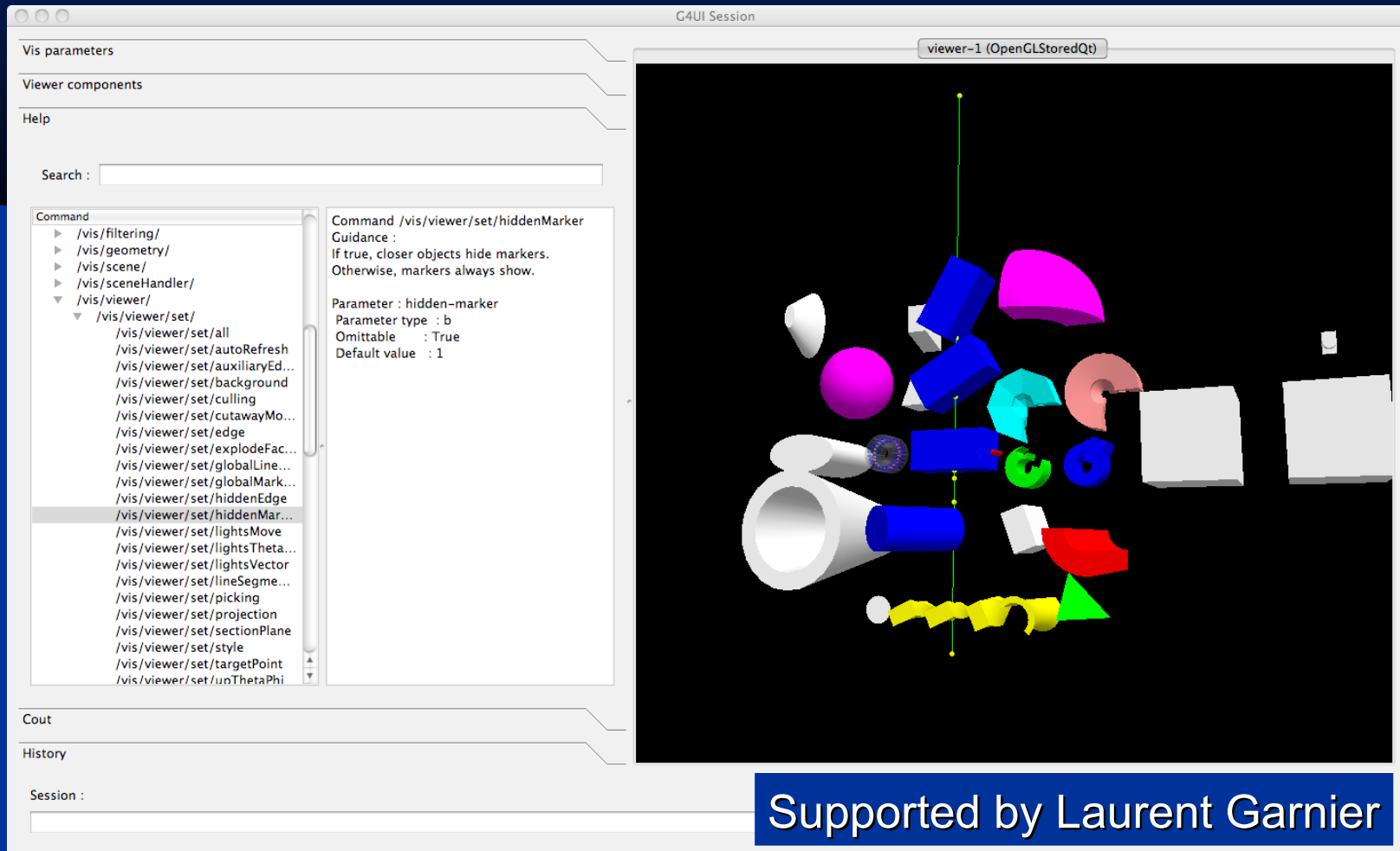
OpenGL

- /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



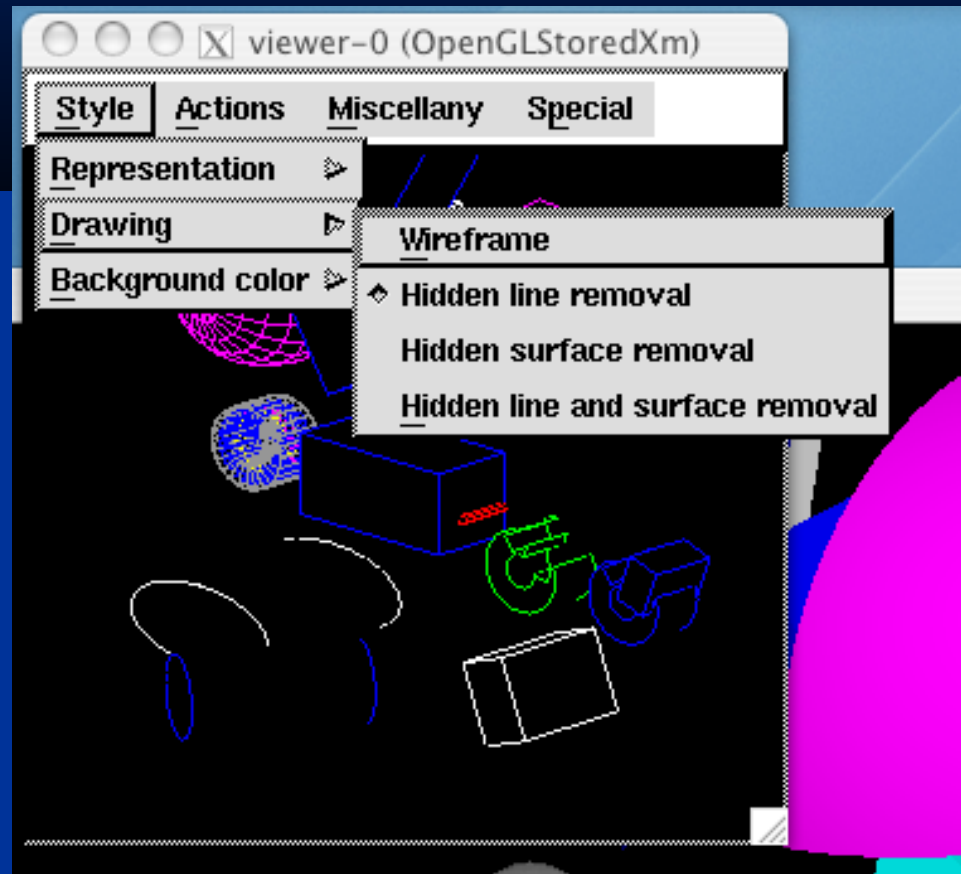
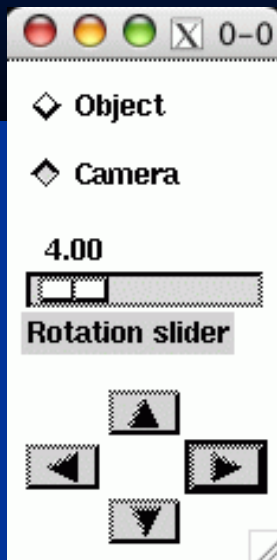
OpenGL with Qt Control

- 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



Postscript Printing

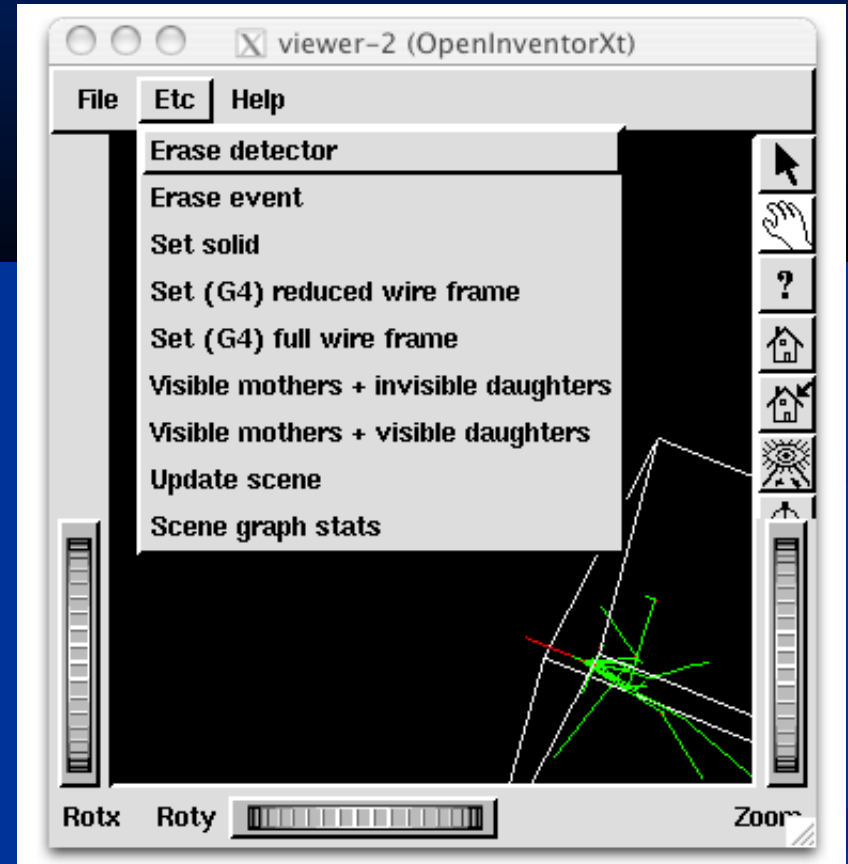
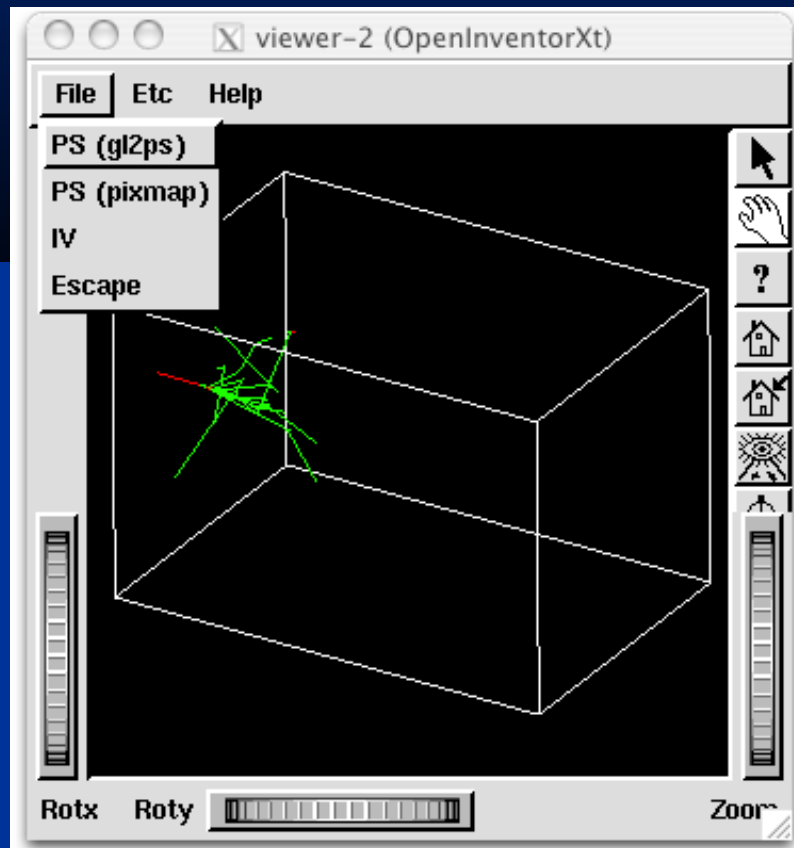
- OpenGL can save the current view to high quality Postscript files
 - `/vis/ogl/printEPS`
- Default output is Vectored, but can also produce Pixmap output
 - `/vis/ogl/set/printMode` vectored or pixmap
- Additional options:
 - `/vis/ogl/set/transparency` True or False

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

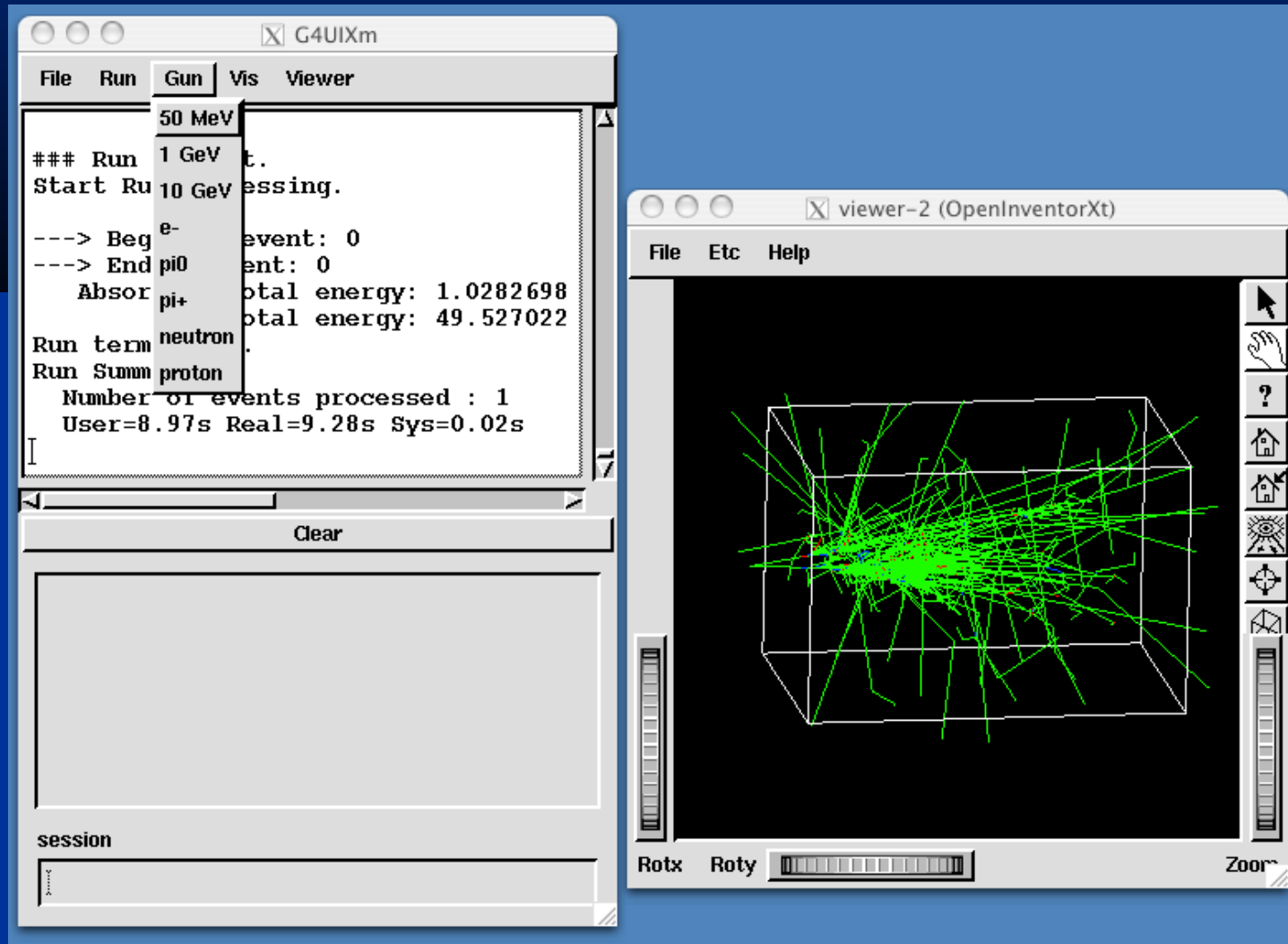
OpenInventor: Start from Geant4

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



OpenInventor: More GUI Control

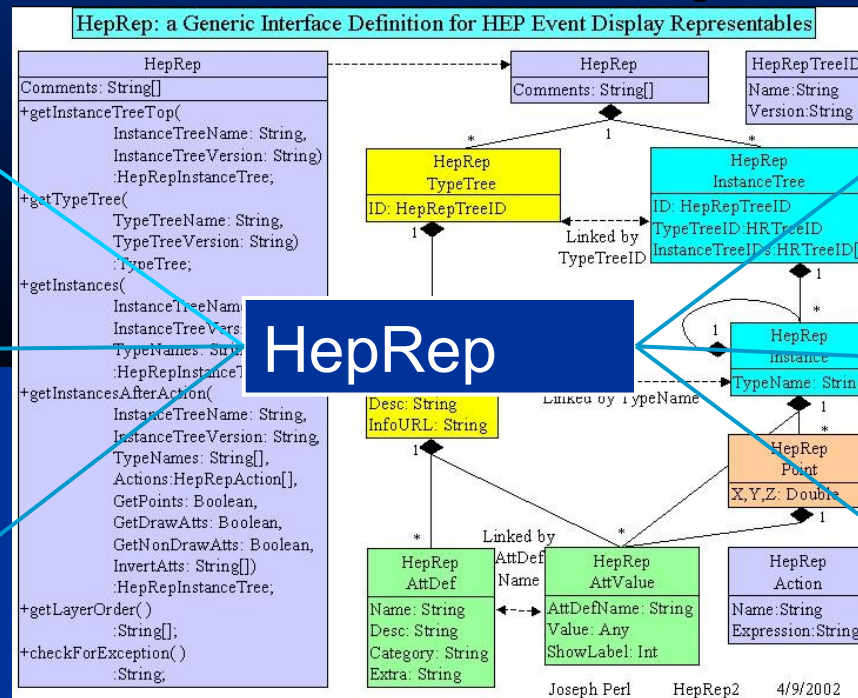
- You can also choose to control the Geant4 run from OpenInventor.



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)

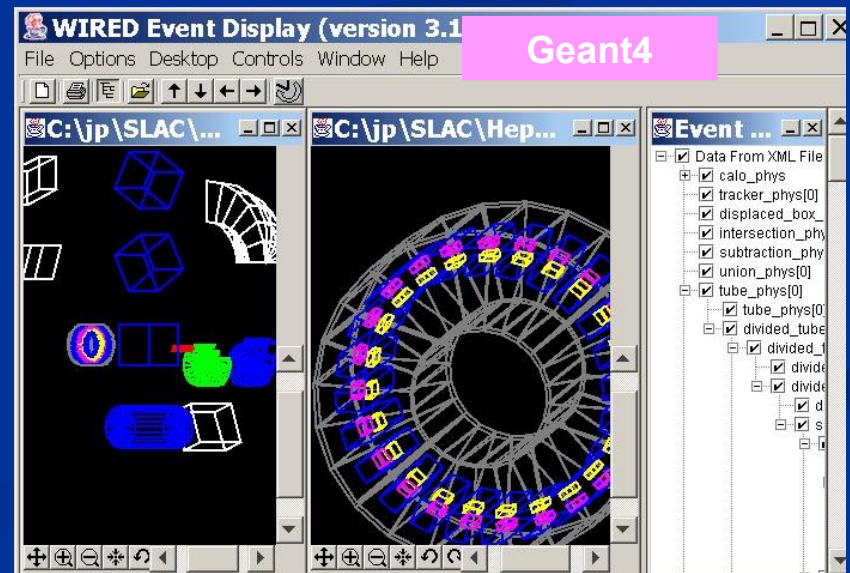
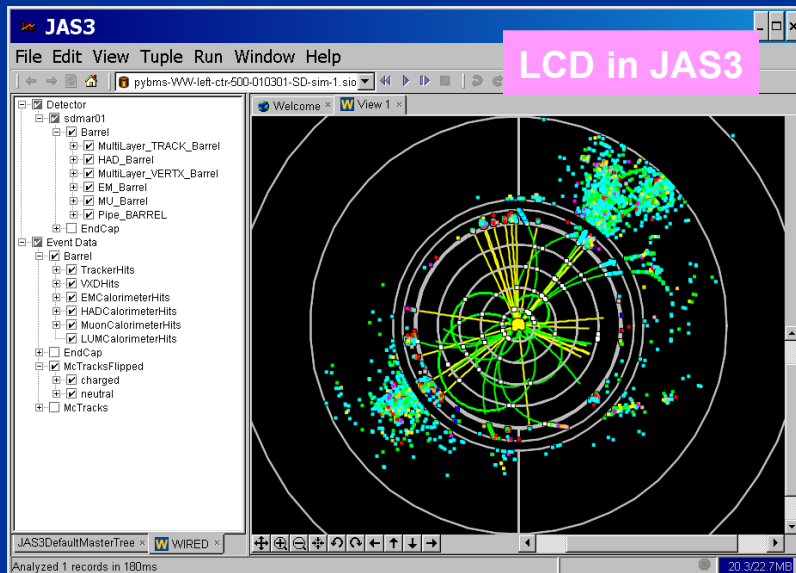
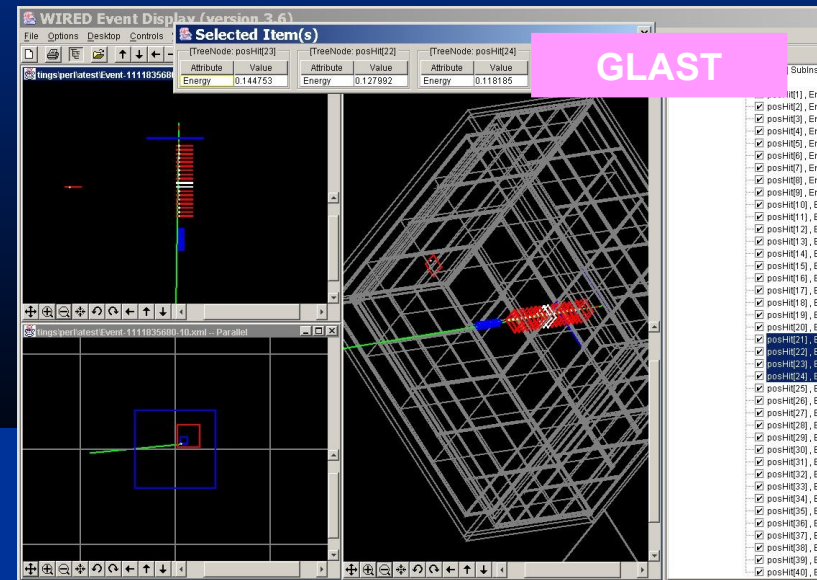
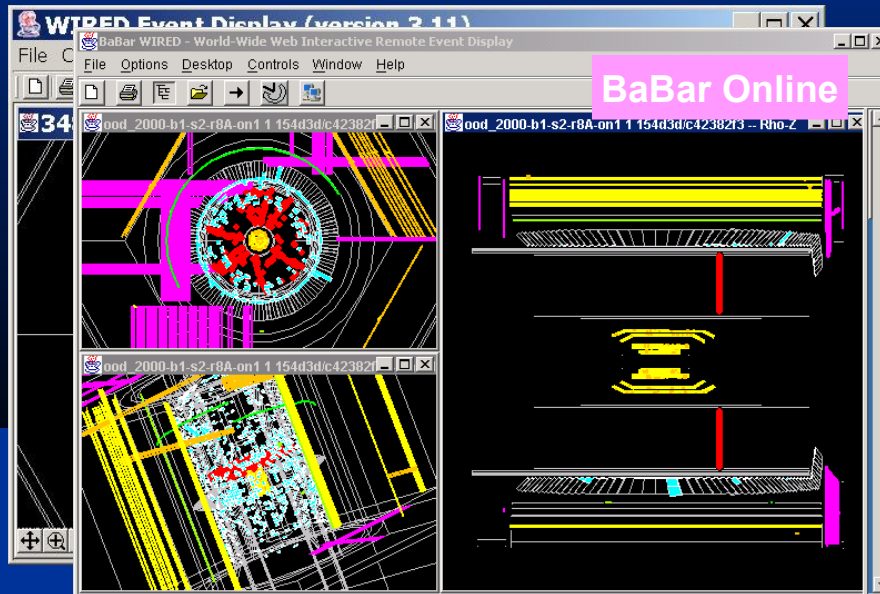
Wired4 Jas
Plug-In (Java)

FRED Client
(C++/Ruby)

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

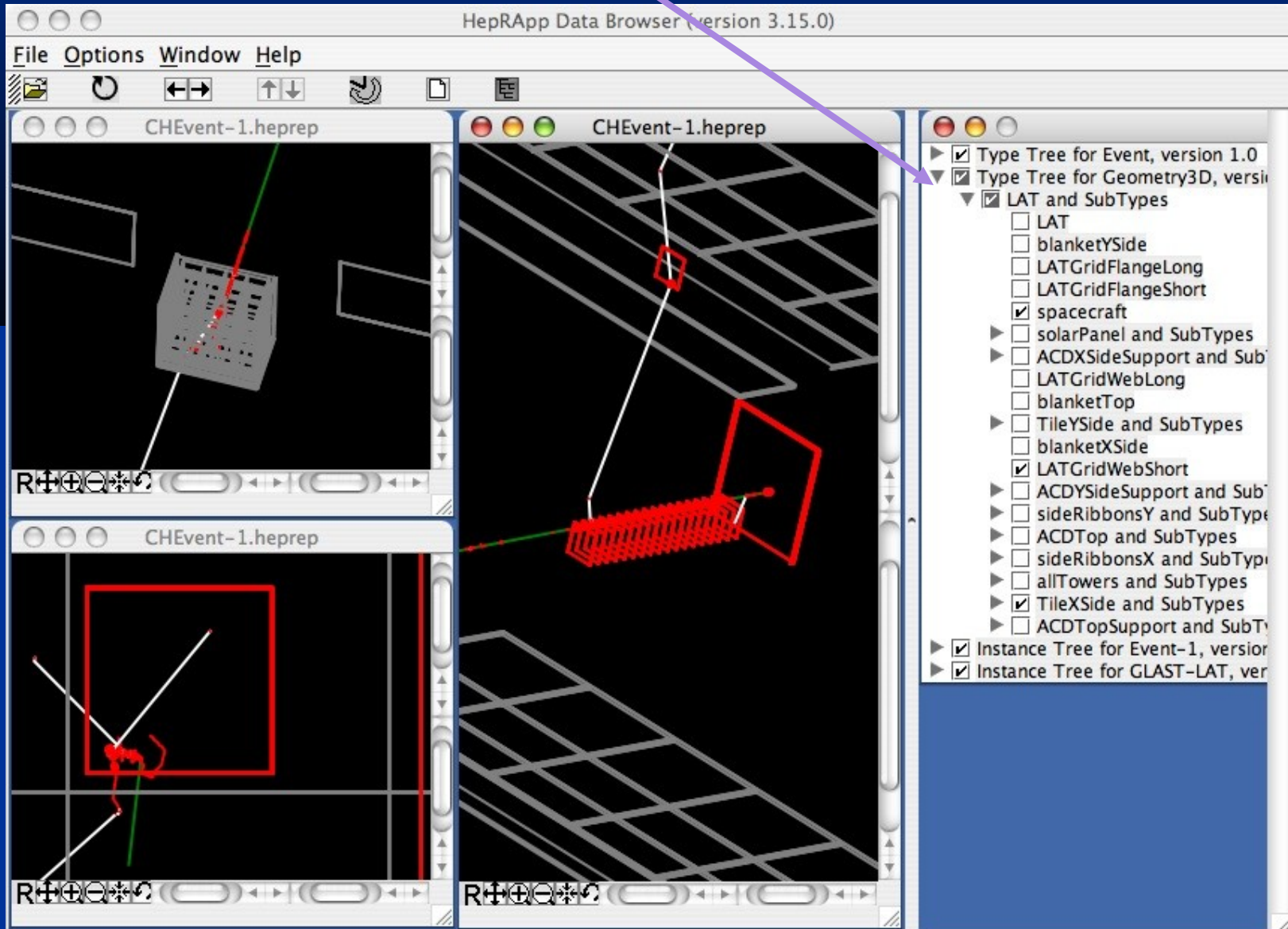
The HepRep format is independent of any one particular language or protocol. It can be used from C++ or Java and can be shipped as Corba, RMI, XML, C++, Java or JNI for consumption by HepRApp, Wired4, FRED or any other HepRep-enabled event display client.

Who's Using HepRep



HepRApp: Shows Geometry Hierarchy

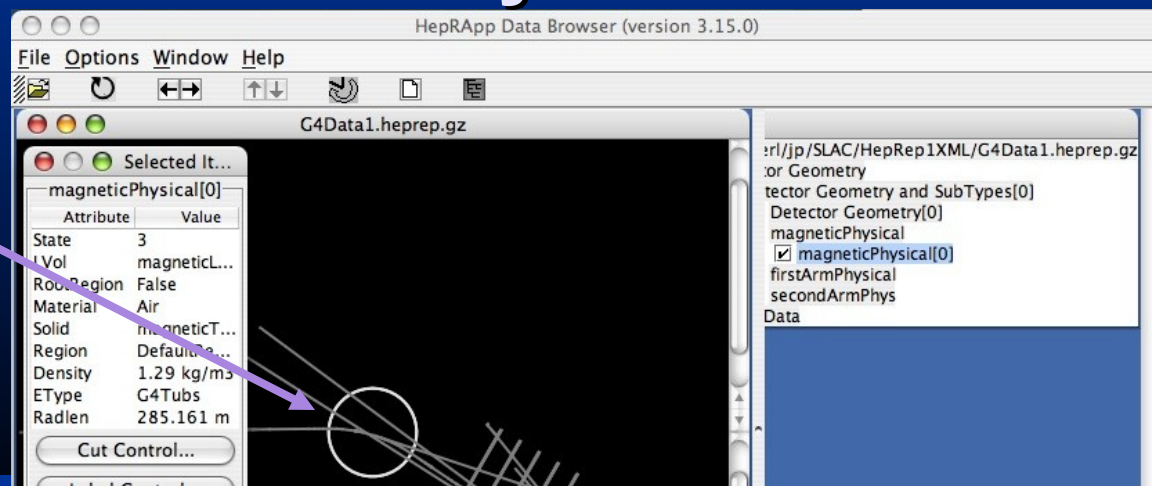
Turn visibility on and off from hierarchical control



HepRApp: Pick to Show Physics Attributes

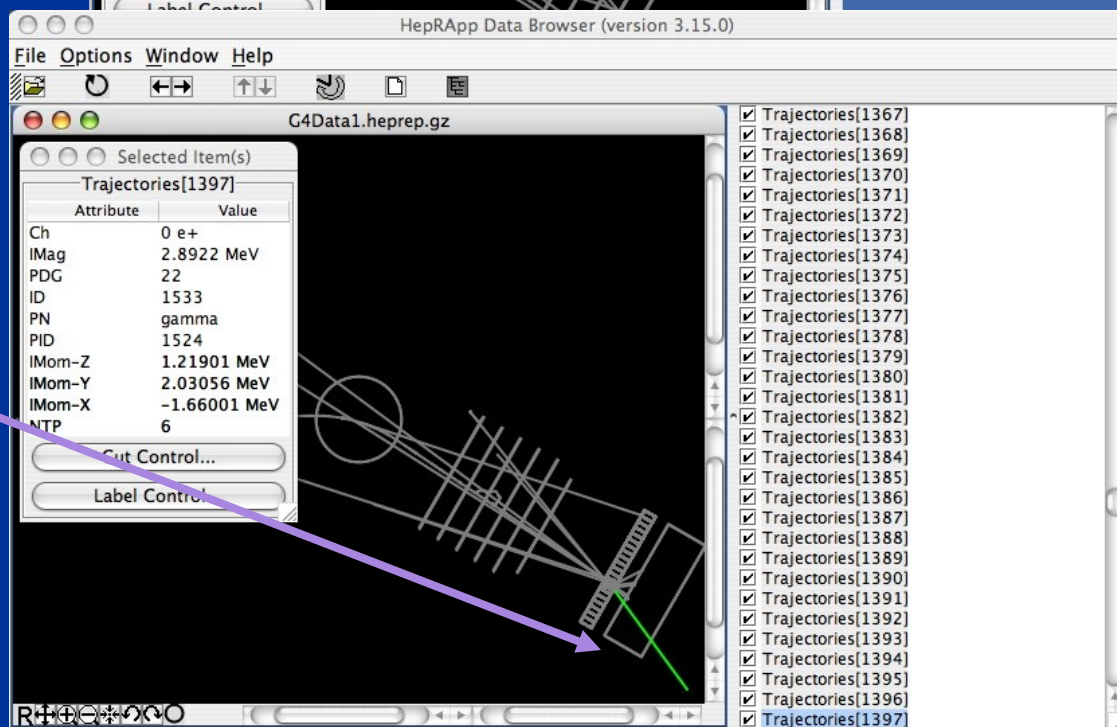
Picked on this volume to show

- Material
- Density
- Radlen
- etc



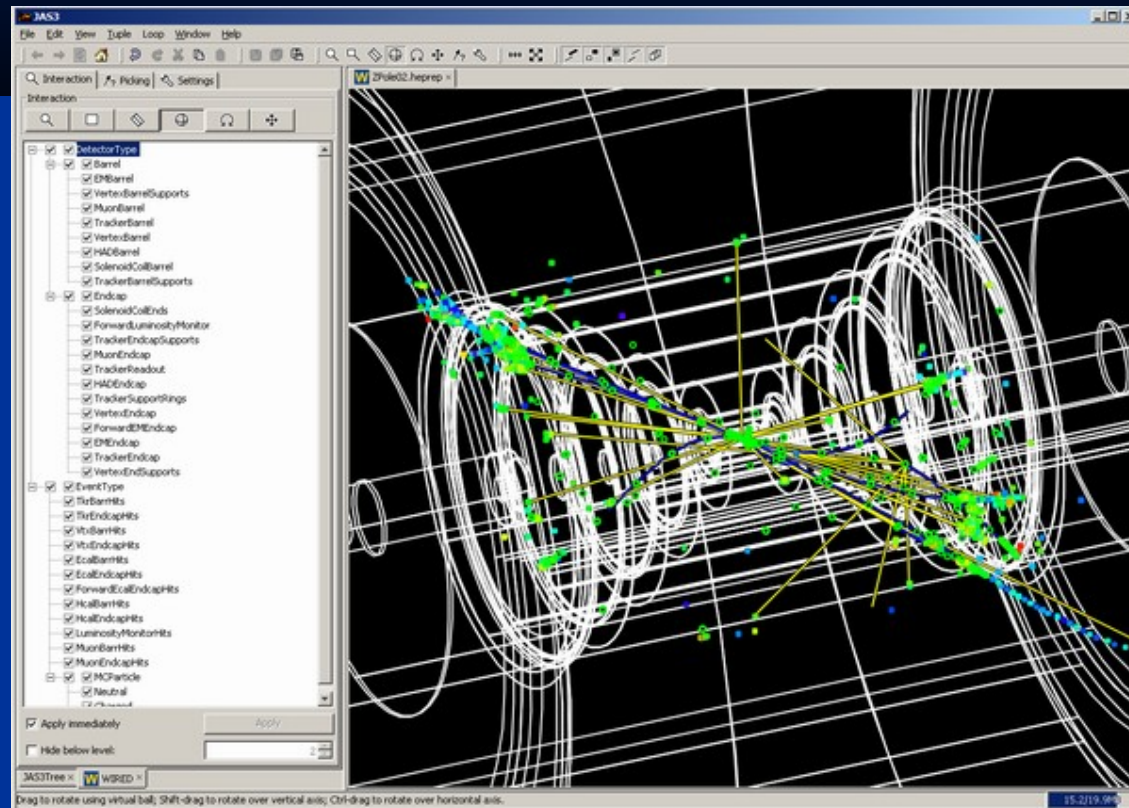
Picked on this trajectory to show

- Particle ID
- Charge
- Momentum
- etc.



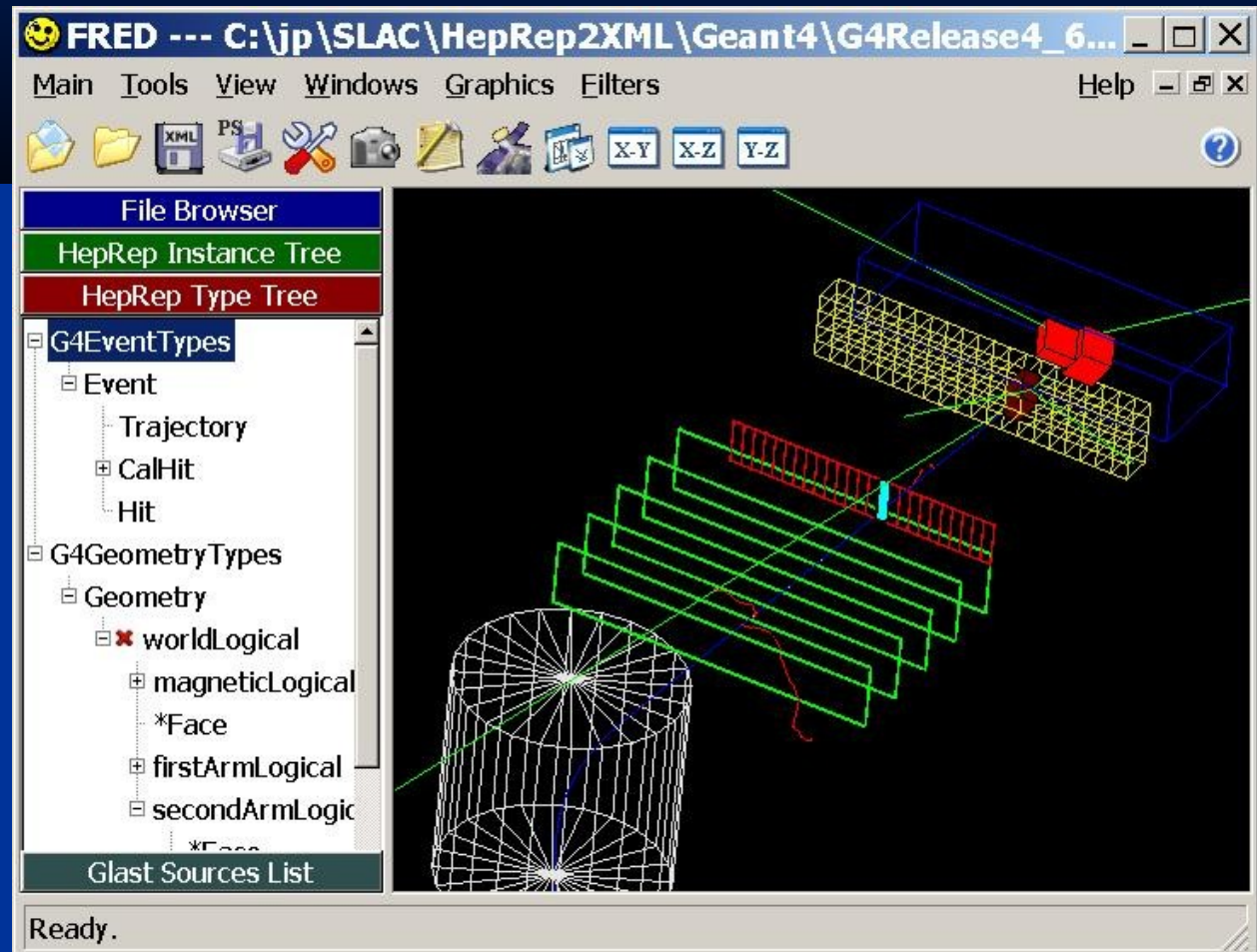
WIRED4: Part of JAS

- 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.

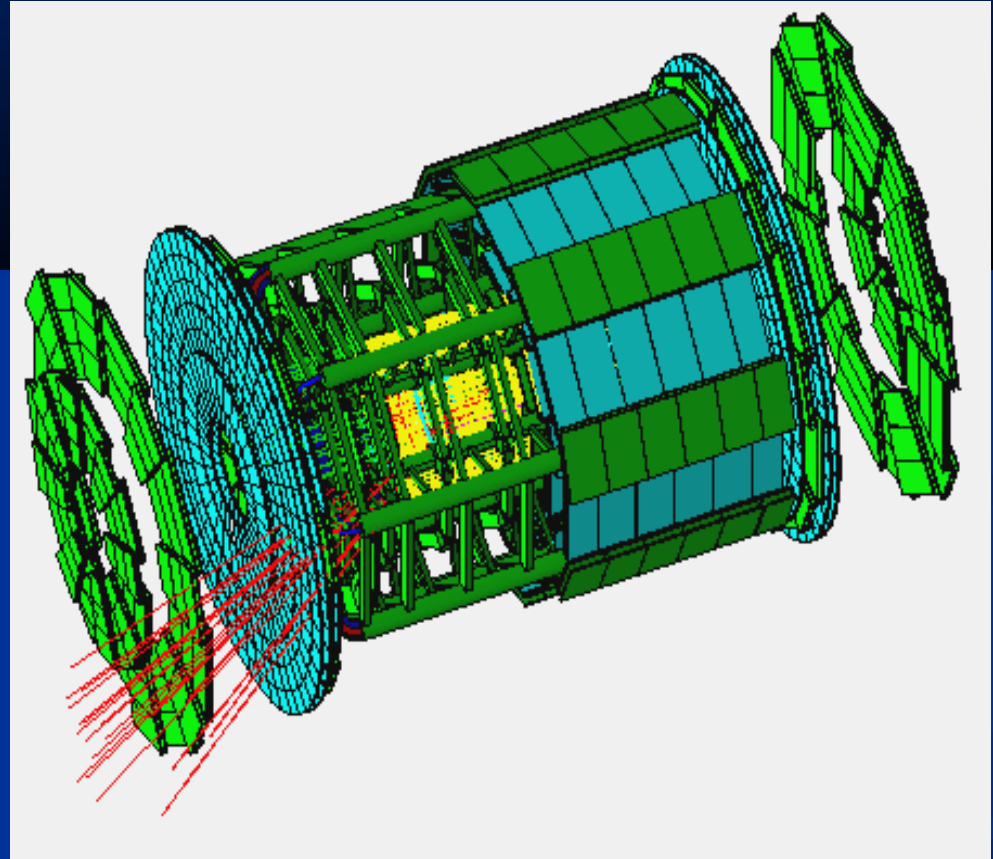
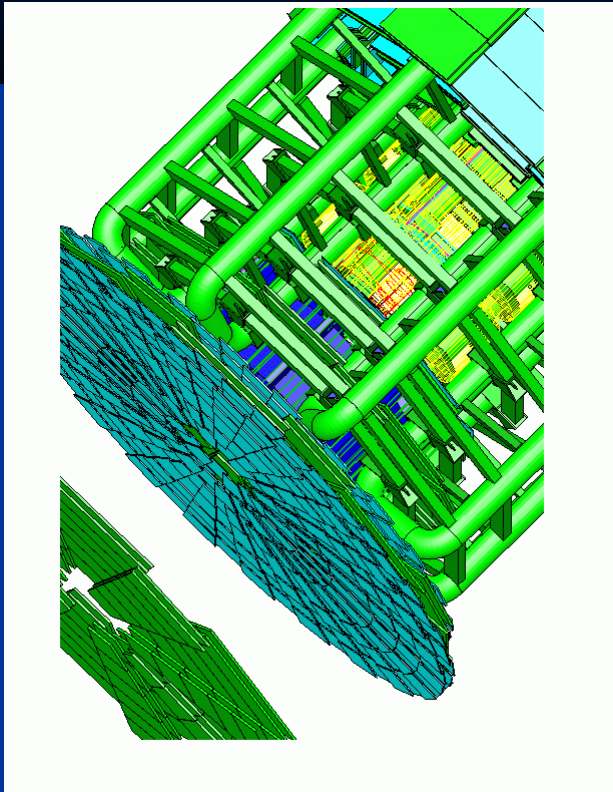


DAWN

- /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

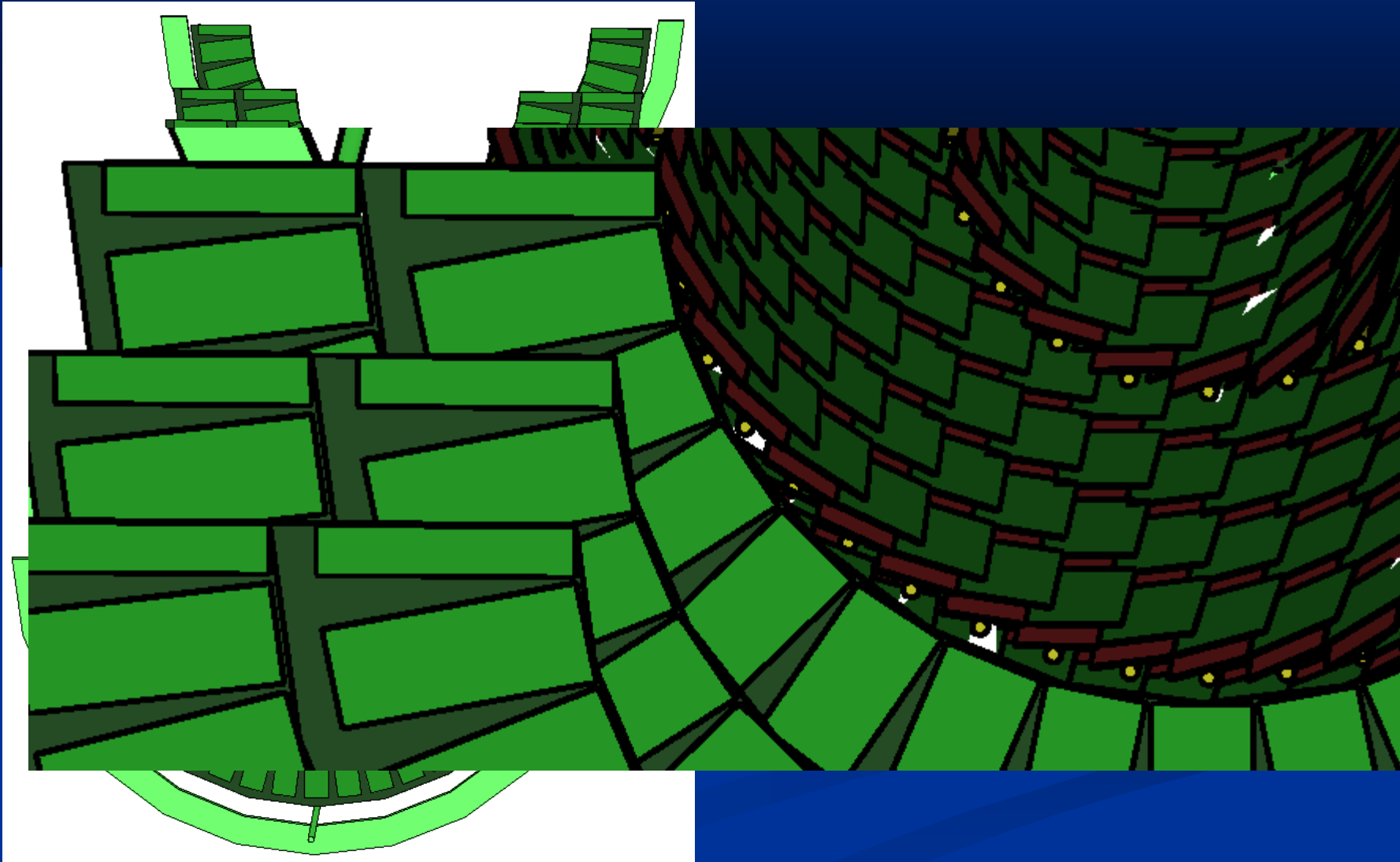
DAWN Examples

- From a repository of beautiful images at
 - http://geant4.kek.jp/~tanaka/GEANT4/ATLAS_G4_GIFFIG/



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

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: complimentary file formats, each with its own strengths

HepRep	DAWN
<ul style="list-style-type: none">■ Hierarchical■ Simple Primitives	<ul style="list-style-type: none">■ Flat■ All Geant4 Primitives
<ul style="list-style-type: none">■ General Purpose■ Representables have Attributes■ No Camera or Lighting Information	<ul style="list-style-type: none">■ Just for Geant4■ No Attributes■ Camera and Lighting Information

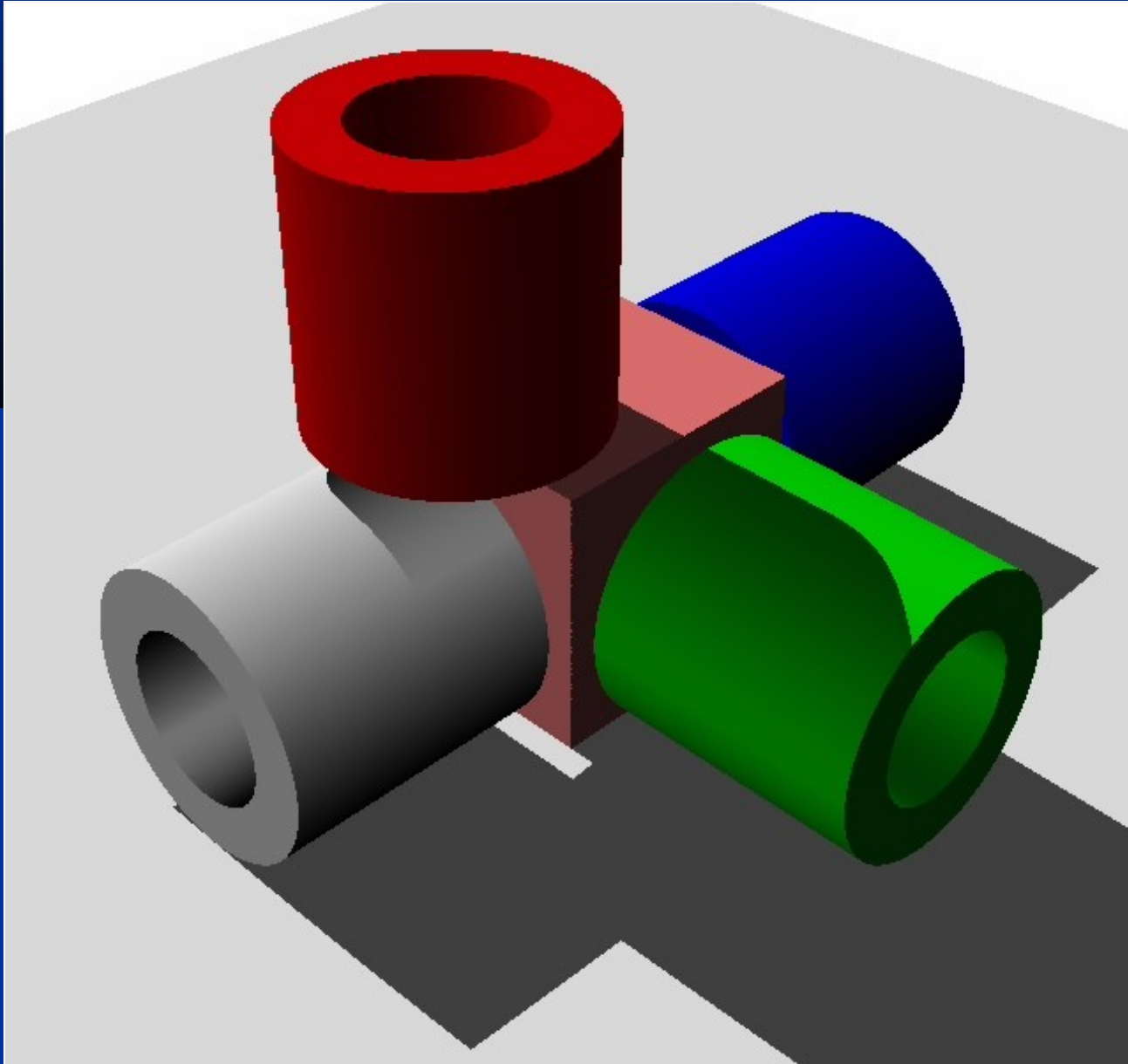
RayTracer

- /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

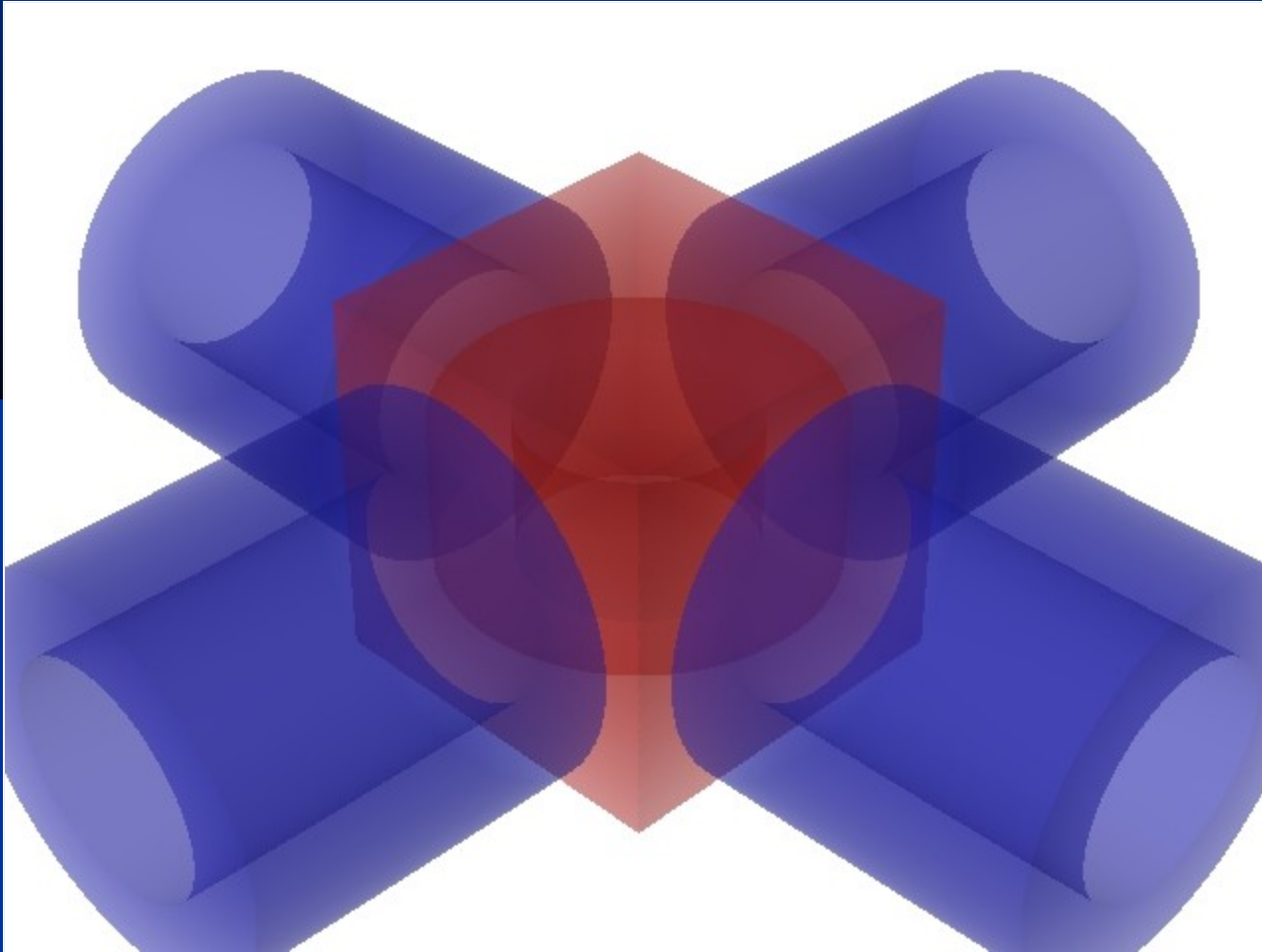
RayTracer

- 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.

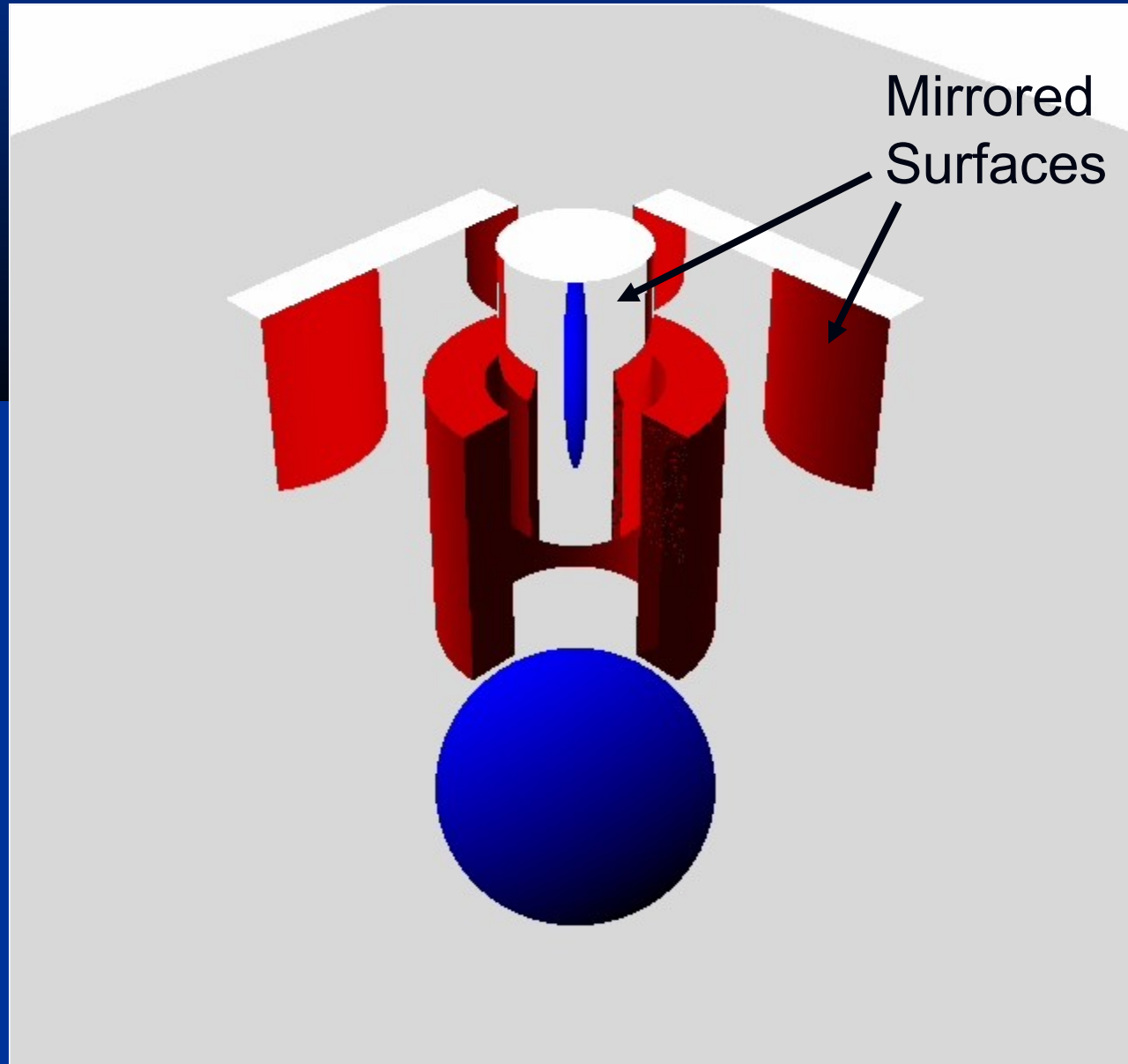
RayTracer Shows Shadows



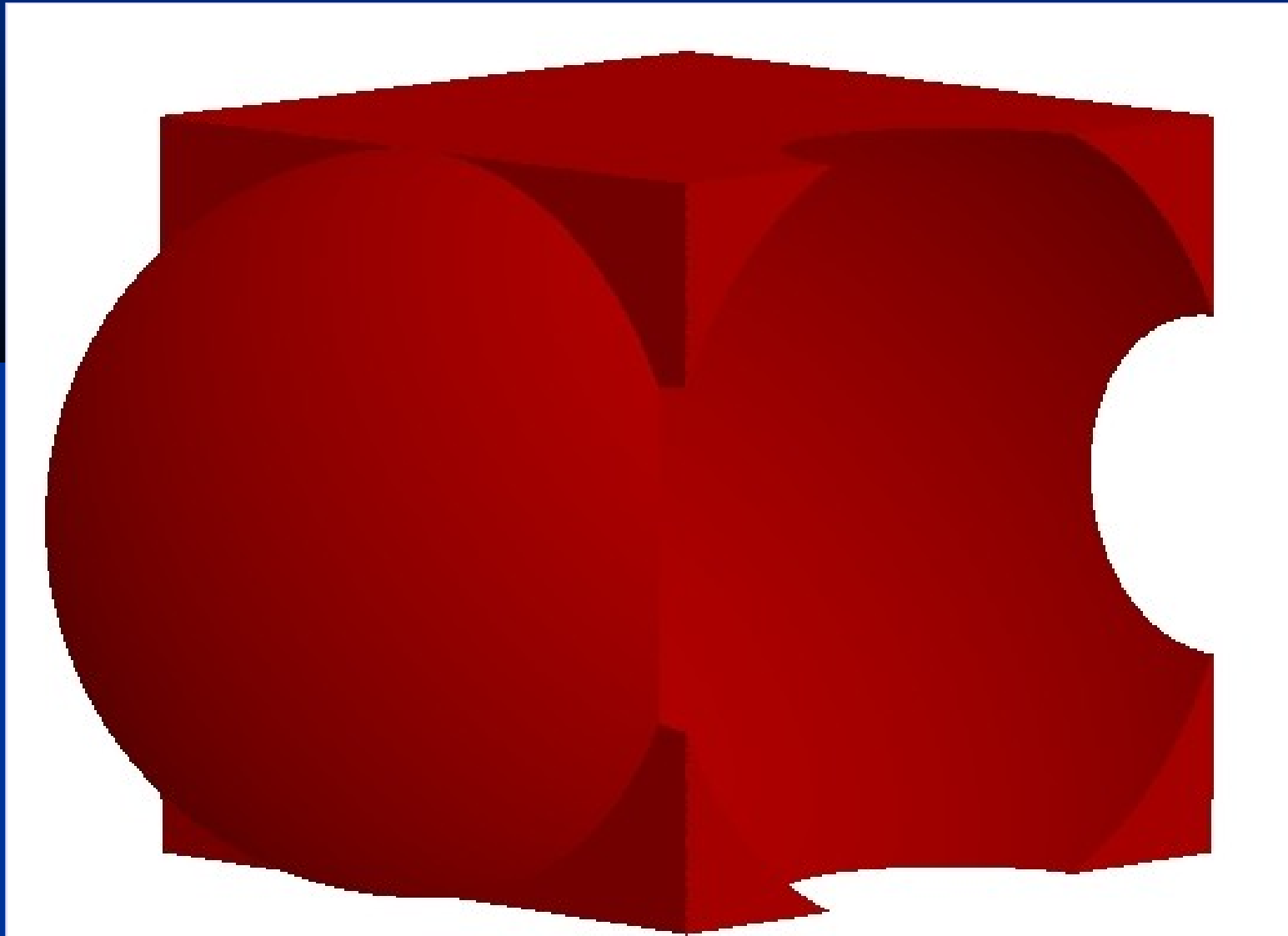
RayTracer Supports Transparency



RayTracer Handles Mirrored Surfaces

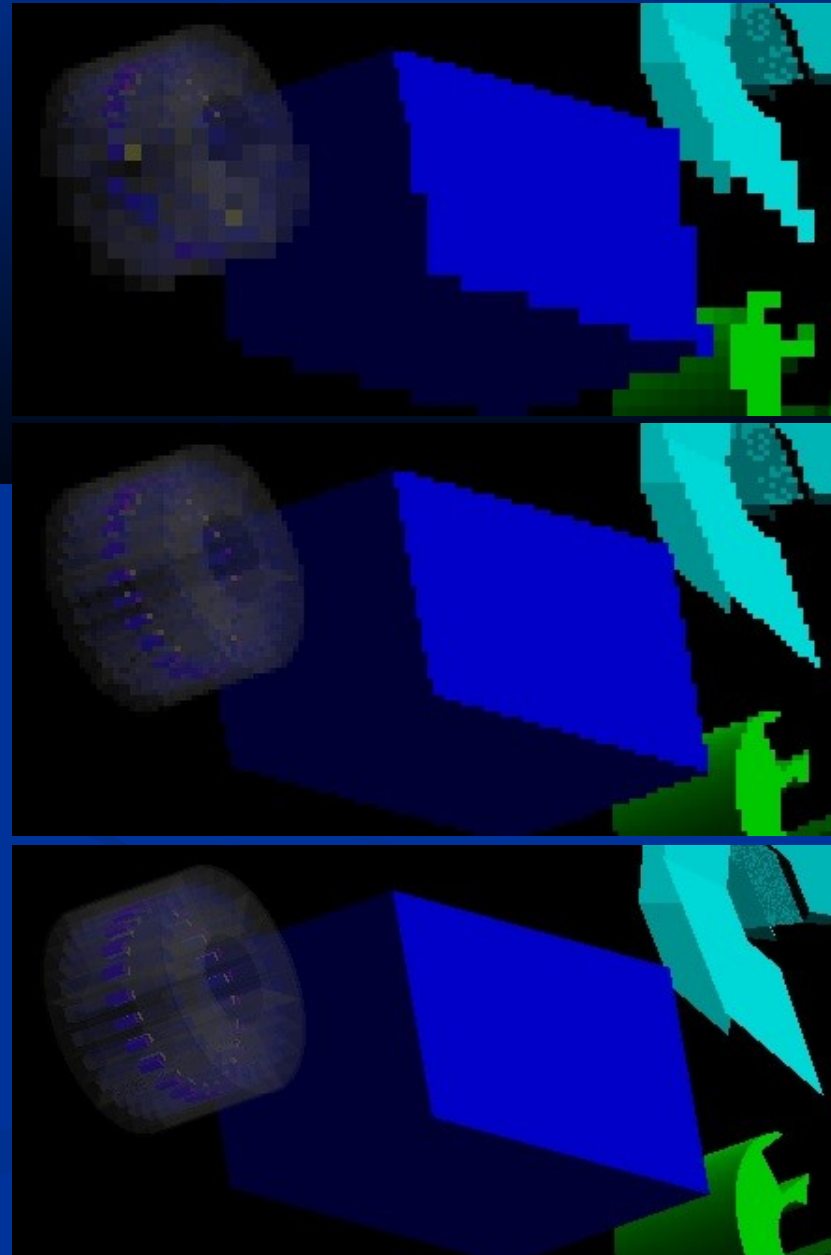


RayTracer Handles Boolean Solids



RayTracerX

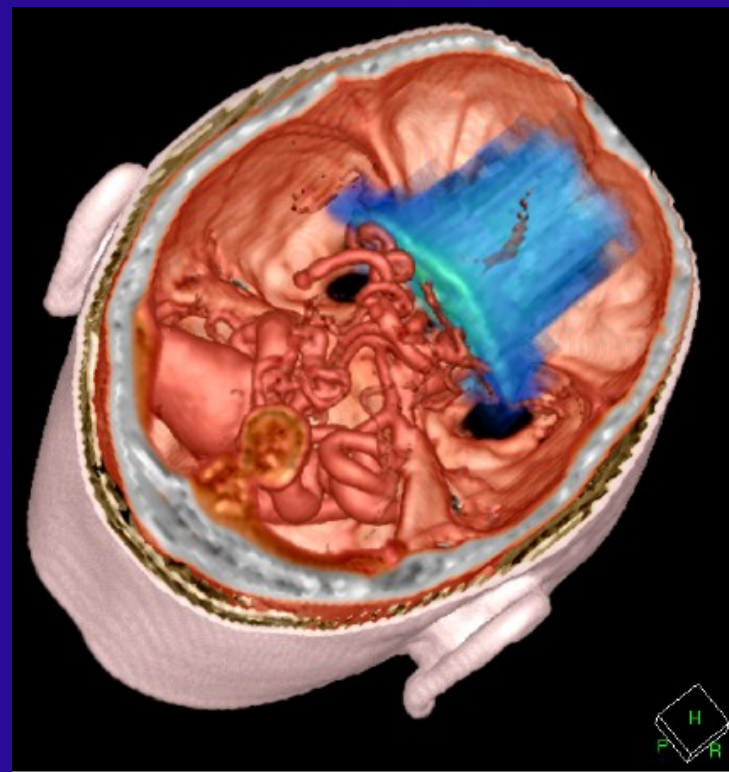
- New since Geant4.8.0
 - In addition to
 - /vis/open RayTracer
 - You have the option of
 - /vis/open RayTracerX
-
- Builds same jpeg file as RayTracer, but simultaneously renders to screen so you can watch as rendering grows progressively smoother.
 - Means you can abort and retry the rendering with 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



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 m3
 - 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

■ ASCIITree

- 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

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
- DAWNCUT Home Page
http://geant4.kek.jp/~tanaka/DAWN/About_DAWNCUT.html
- DAVID Home Page
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>