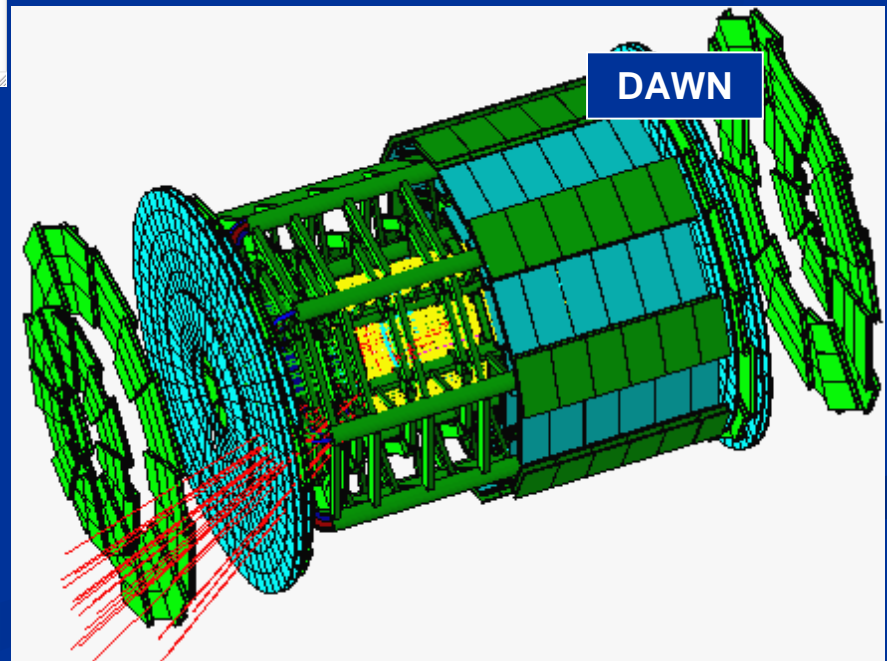
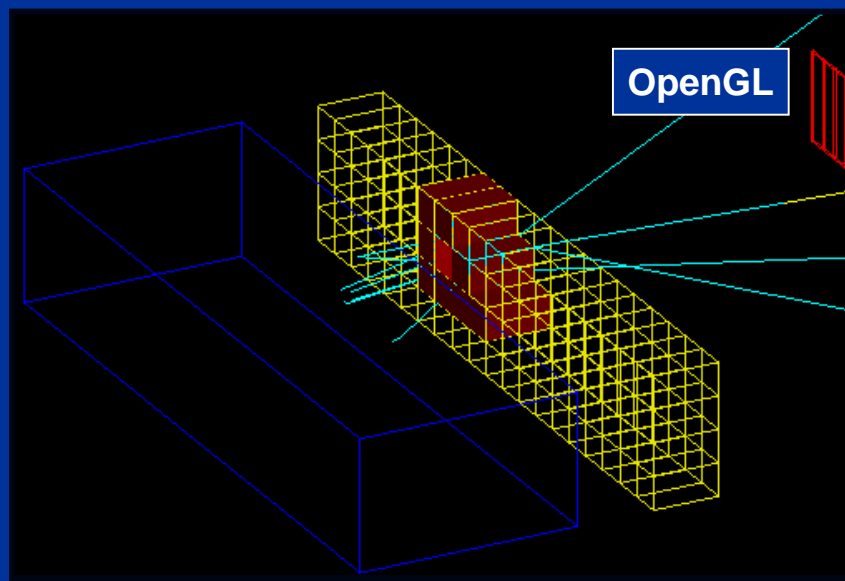
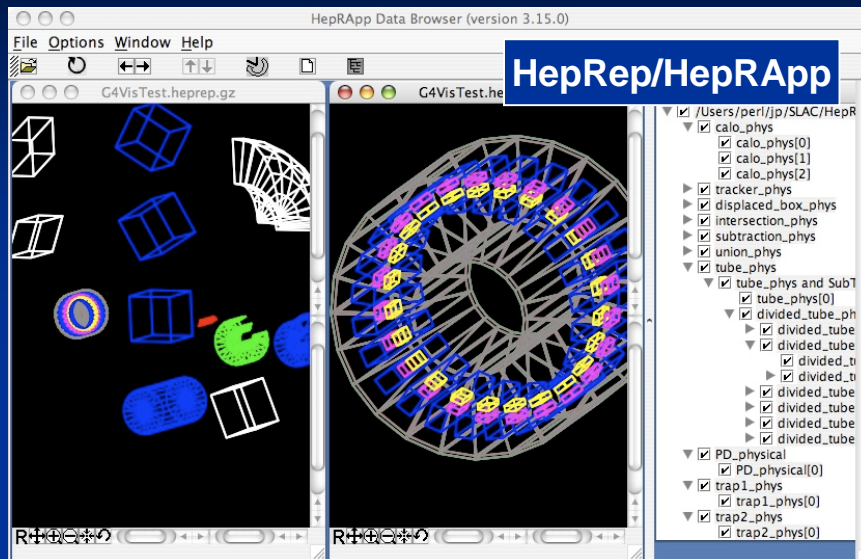


Introduction to Geant4 Visualization

Paul Guèye, HU

Courtesy: Joseph Perl, SLAC

*So many options,
it needs two title
pages*



Purpose of Geant4 Visualization

- n Quick response to study geometries, trajectories and hits
- n High-quality output for publications
- n Flexible camera control to debug complex geometries
- n Tools to show volume overlap errors in detector geometries
- n Interactive picking to get more information on visualized objects

What Can be Visualized

n Simulation data

- n Geometrical components
- n Particle trajectories and tracking steps
- n Hits of particles in the geometry

n Other user defined objects

- n Polylines
 - n such as coordinate axes
- n 3D Markers
 - n such as eye guides
- n Text
 - n descriptive character strings
 - n comments or titles ...

Focus on Usability

- n From the very beginning, Geant4 visualization has been very flexible. The experienced programmer could exploit the full flexibility of C++ and a well-designed series of abstract interfaces to code any desired visualizations.
- n The work of the Geant4 Visualization group over the last few years has been to identify those features that users want most, and make them easily achievable from simple interactive commands.
- n So, in the last year or so we have added:
 - n Trajectory Modeling commands
 - n Trajectory and Hit Filtering commands
 - n Smooth and Rich Trajectory commands
 - n Event Keeping
 - n Time-Development Animation, etc.
- n More great new features will be coming, such as:
 - n Visualization of Fields

Quick Overview of Visualization Driver Choices

- n Geant4 comes with Seven visualization drivers:
 - n OpenGL
 - n OpenInventor
 - n HepRep
 - n DAWN
 - n VRML
 - n RayTracer
 - n ASCIITree
- n Different drivers are appropriate for different needs
- n The set of commands the user sees are consistent regardless of the driver

Design by Interfaces

- n Why Geant4 has so many different visualization systems?
 - n This is a natural result of Geant4 being a toolkit and not a single application.
- n 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.
 - n These interfaces make it straightforward to connect Geant4's core visualization tools to any visualization system
 - n able to drive advanced systems that can natively display complex solids such as Geant4's cut cylinders
 - n able to drive more basic systems that do not understand such solids (system can ask Geant4 visualization to deconstruct complex solids into simpler polygons)
 - n 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.
- n Interfaces are discussed in detail in a paper just finished in May 2007 and currently awaiting publication.

Seven 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

§ Tools to 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 Which Drivers are Available

- n Five of the visualization drivers are always included by default (since they require no external libraries):
 - n HepRepFile
 - n DAWNFILE
 - n VRMLFILE
 - n RayTracer
 - n ASCIITree
- n Other visualization drivers are included only if appropriate environment variables have been set:
 - n 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.
- n You can also add your own visualization driver.
 - n Geant4’s visualization system is modular. By creating just three new classes, you can direct Geant4 information to your own visualization system.

Controlling Visualization

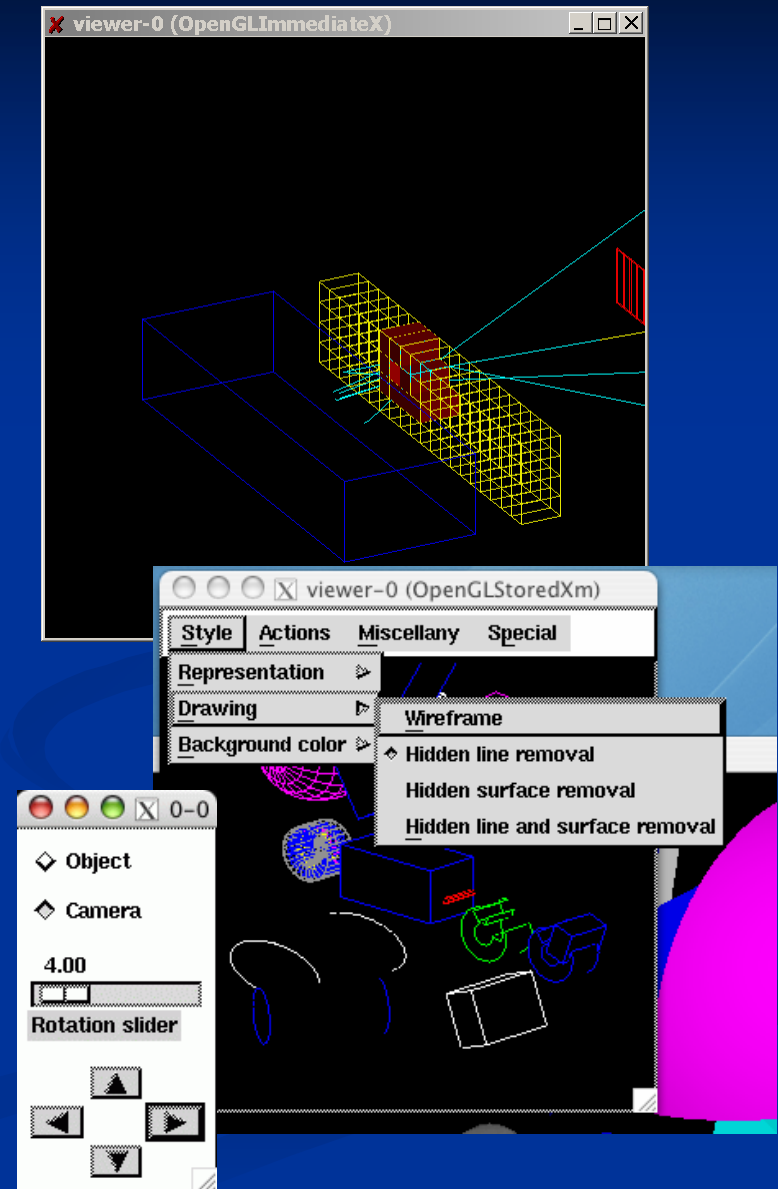
- n Your Geant4 code stays basically the same no matter which driver you use
- n Visualization is performed either with commands or from C++ code
 - n For the present tutorial, we confine ourselves to command-driven visualization.
- n Some visualization drivers work directly from Geant4
 - n OpenGL
 - n OpenInventor
 - n RayTracer
 - n ASCIITree
- n 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)
 - n HepRepFile
 - n DAWNFILE
 - n VRML
- n You can open more than one driver at a time.
 - n For example, do a quick check in OpenGL, then save the same event for a beautiful DAWN plot

Details on the Seven Visualization Drivers

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

OpenGL (Open Graphics Library)

- n /vis/open OGLIX, OGLSWin32 or OGLIXm
- n Features
 - n Control directly from Geant4
 - n Uses GL libraries that are already included on most Linux and Windows systems
 - n Rendered, photorealistic image with some interactive features
 - n zoom, rotate, translate
 - n Fast response (can usually exploit full potential of graphics hardware)
 - n Save as pixel graphics or vector EPS
 - n Live movies
- n Supported by John Allison with Guy Barrand

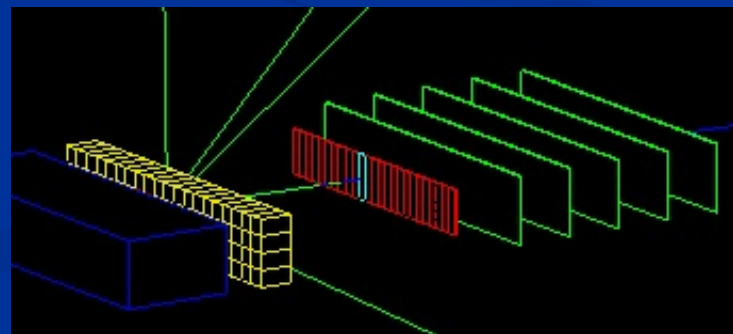
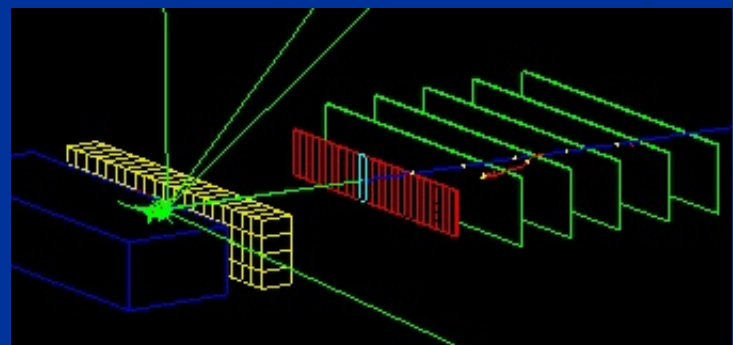
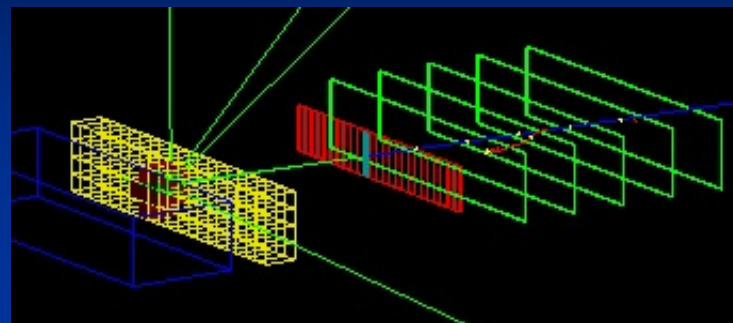


OpenGL Additional Modes

- n There are actually 6 OpenGL drivers - OGLxy:
 - n x = I (Immediate) or S (Stored)
 - n y = X, Xm (Motif) or Win32
- n Immediate mode: draws only to screen, no “memory”; detector can be redrawn after view changes but event data is lost.
- n 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.
- n Also note:
 - n OpenGL supports Smooth shading and Transparency
 - n OpenGL supports non-symmetric scaling:
 - n /vis/viewer/scale <3 vector of scale factors>
 - n or
 - n /vis/viewer/zoom <3 vector of scale factors>

Hidden Line Removal

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

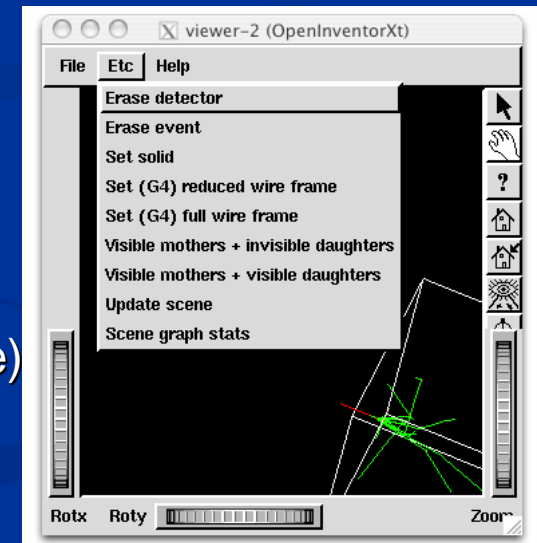
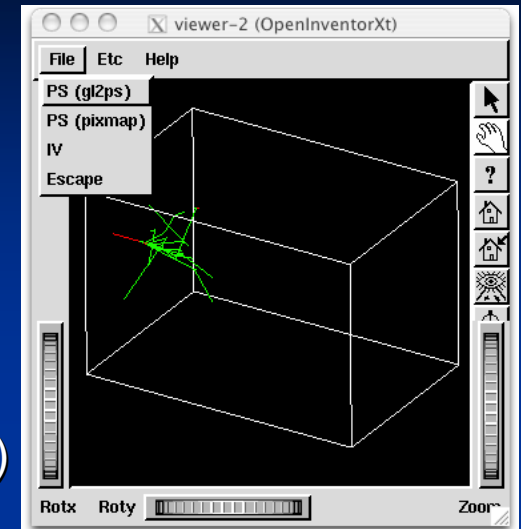


Movies: Time Development of the Event

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

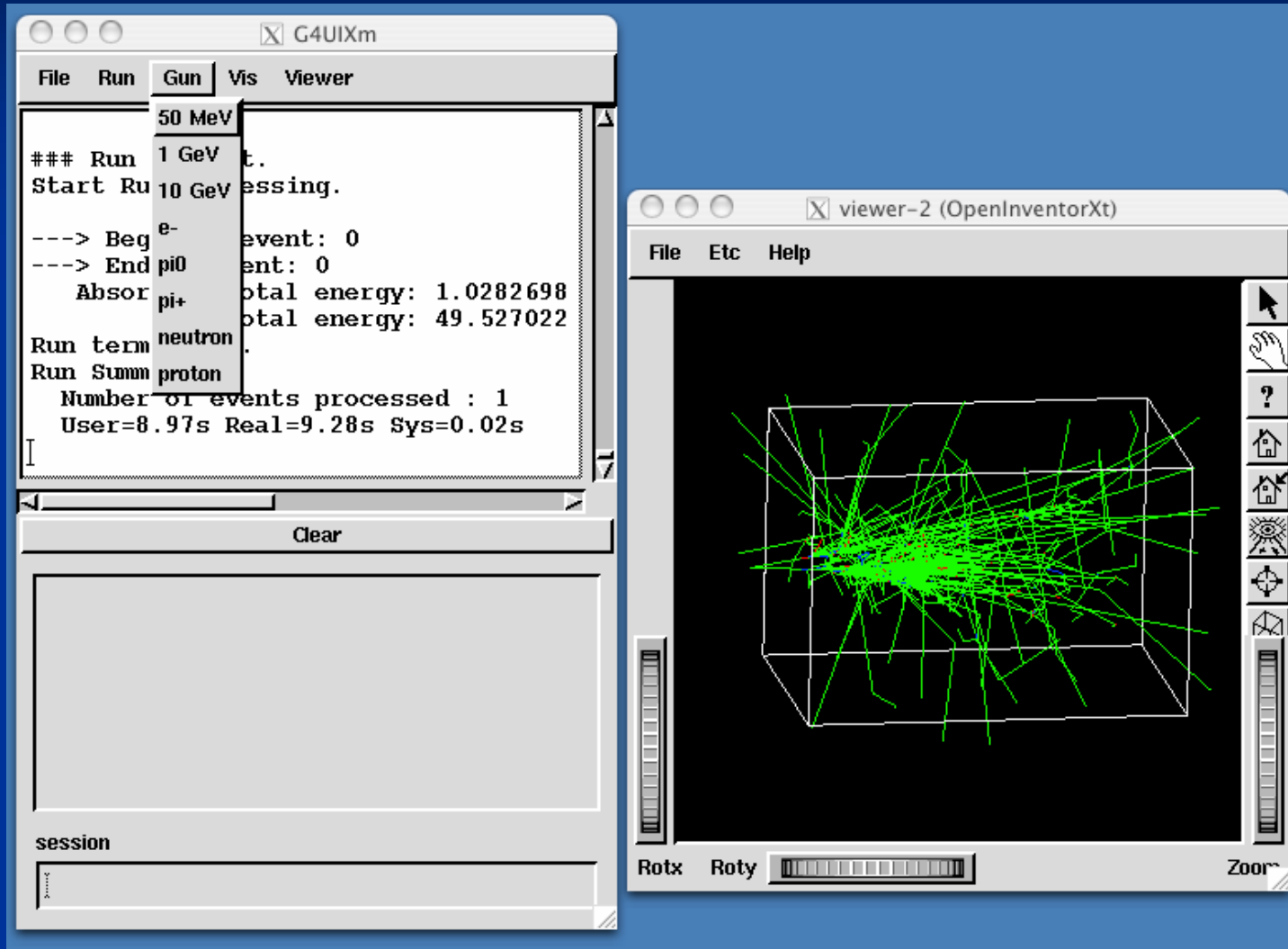
OpenInventor

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



OpenInventor: More GUI Control

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



OpenInventor Details

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

HepRep

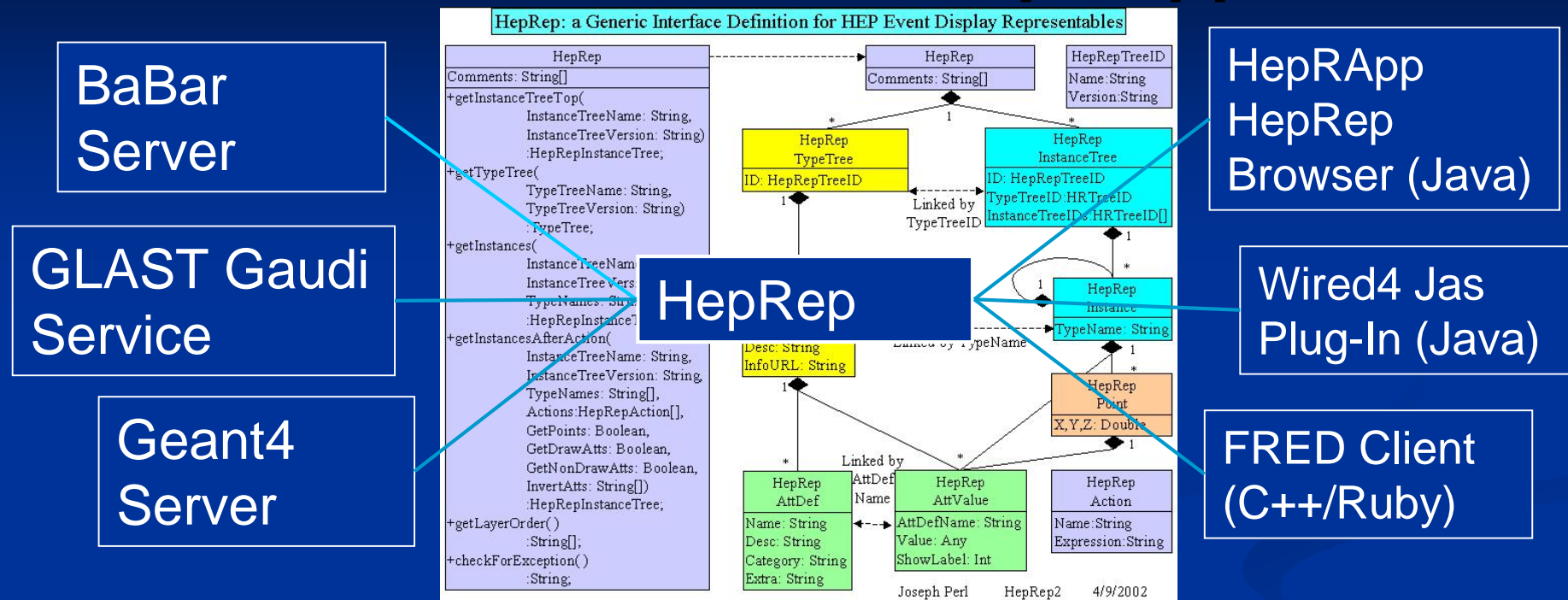
- n /vis/open HepRepFile

- n Features

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

- n Supported by Joseph Perl and Mark Donszelmann

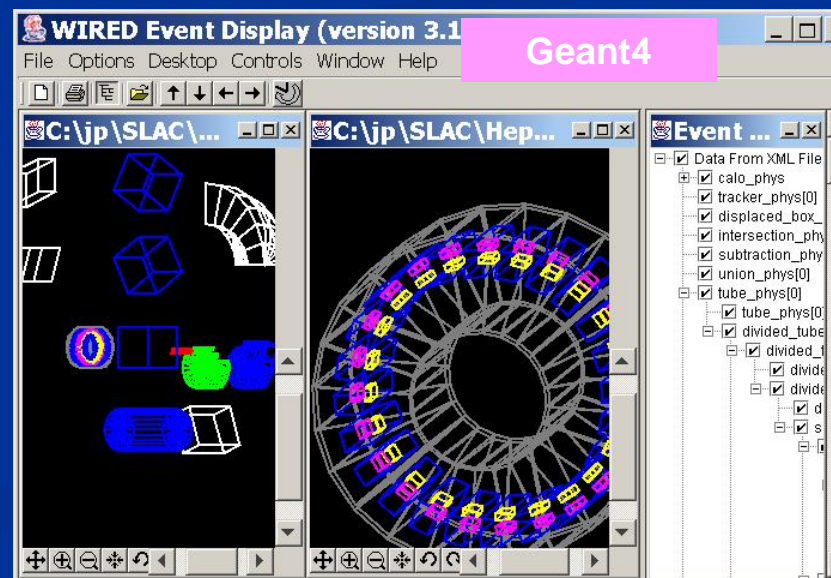
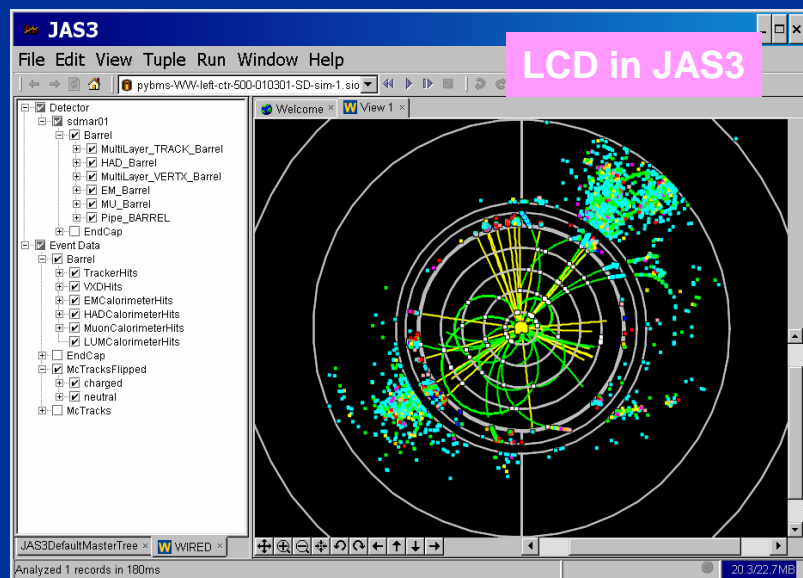
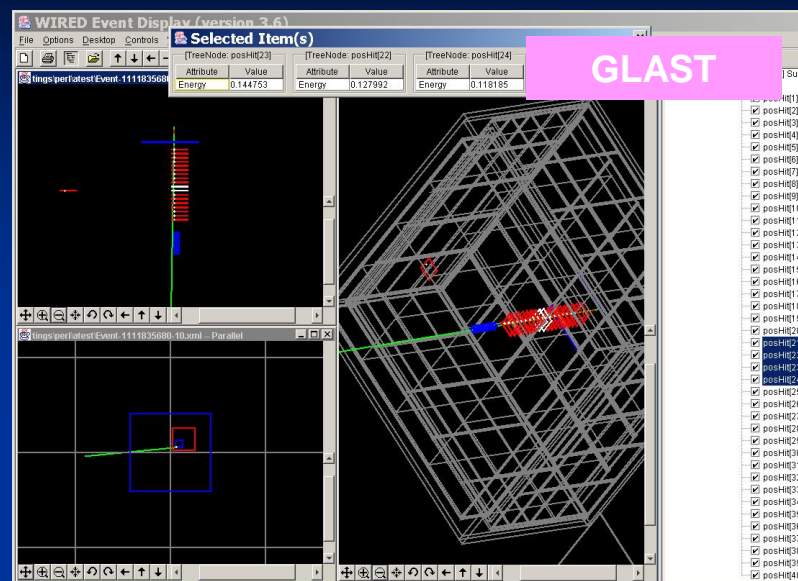
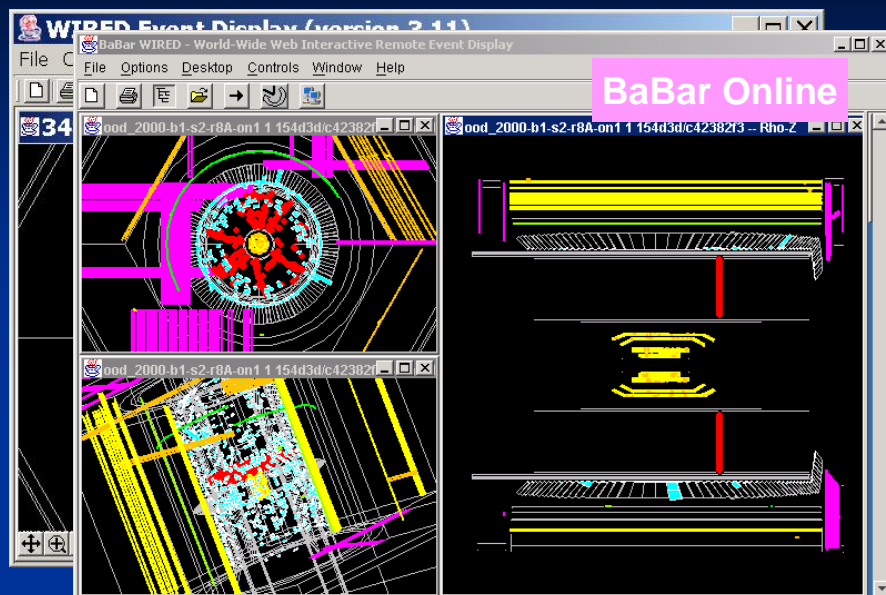
HepRep is Not Just for Geant4 and Not Just for HepRApp



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



June 04-08, 2007 - G4 Paris

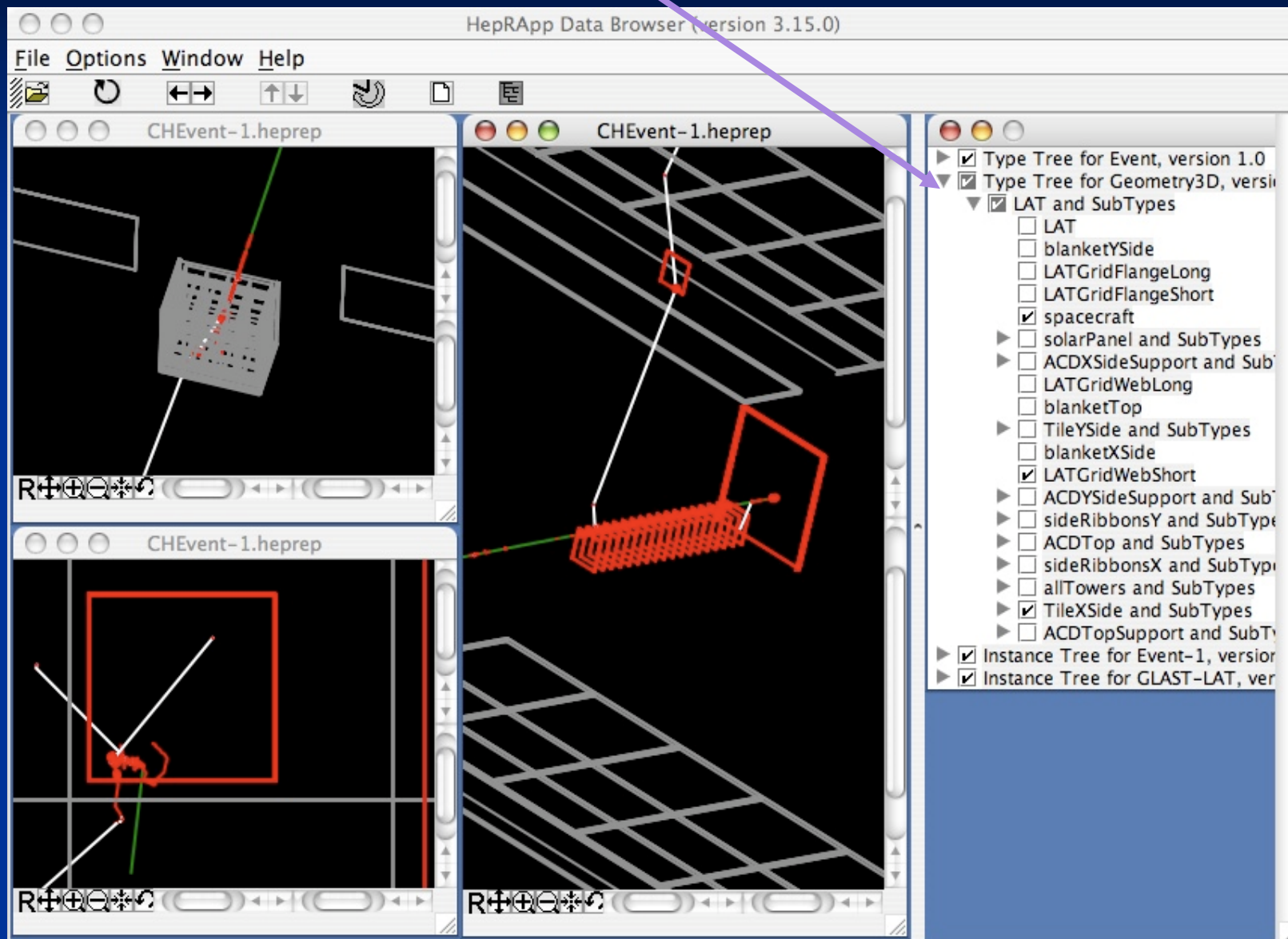
Visualization I

P. Gueye/J. Perl

19

HepRApp: Shows Geometry Hierarchy

Turn visibility on and off from hierarchical control



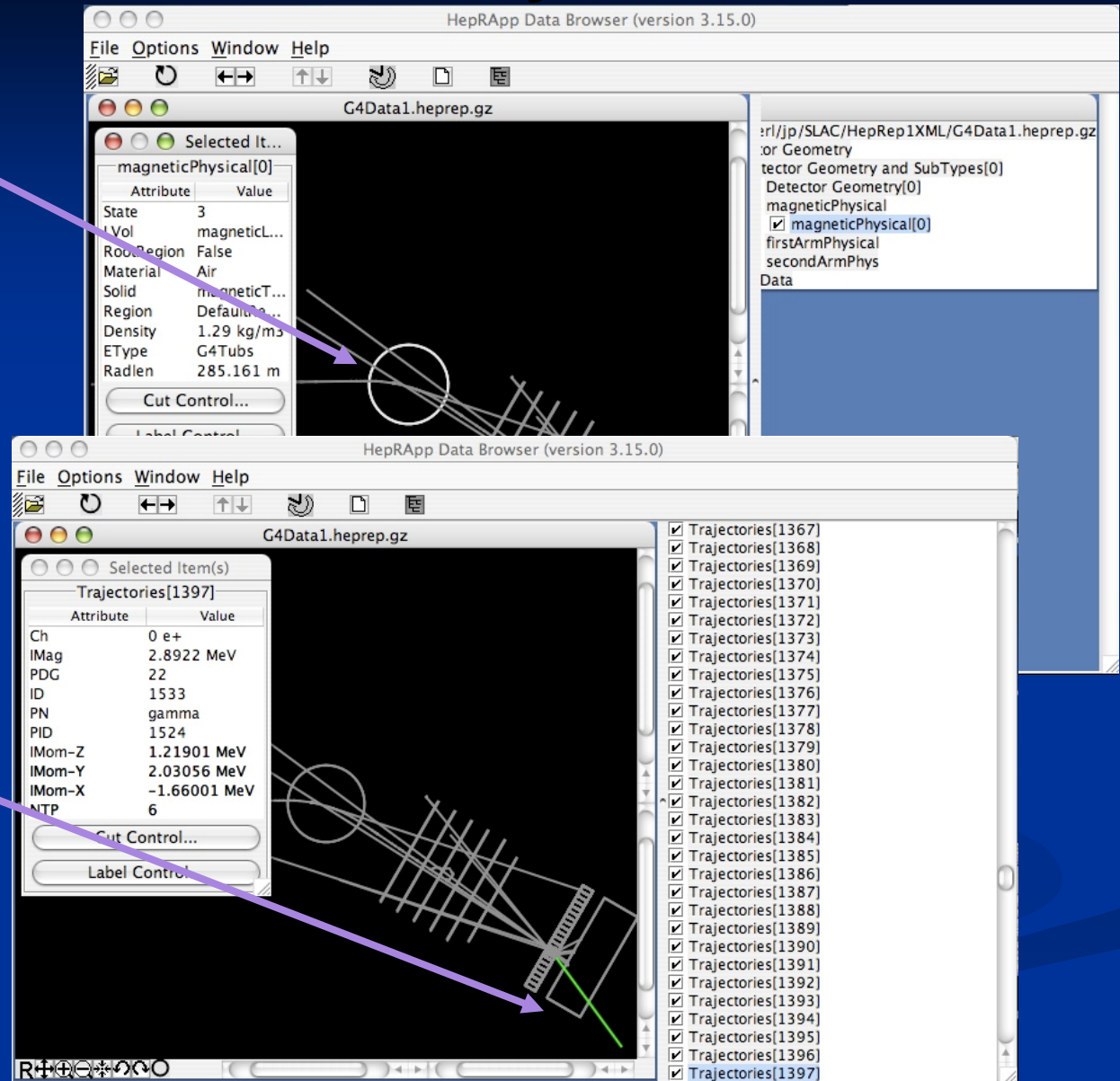
HepRApp: Pick to Show Physics Attributes

Picked on this volume to show

- Material
- Density
- Radlen
- etc

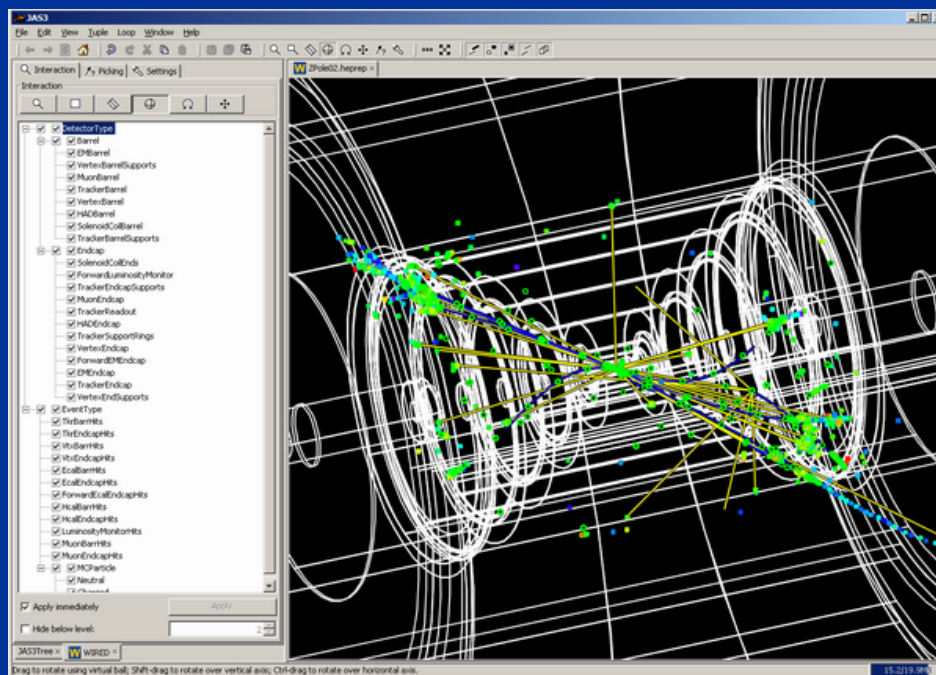
Picked on this trajectory to show

- Particle ID
- Charge
- Momentum
- etc.



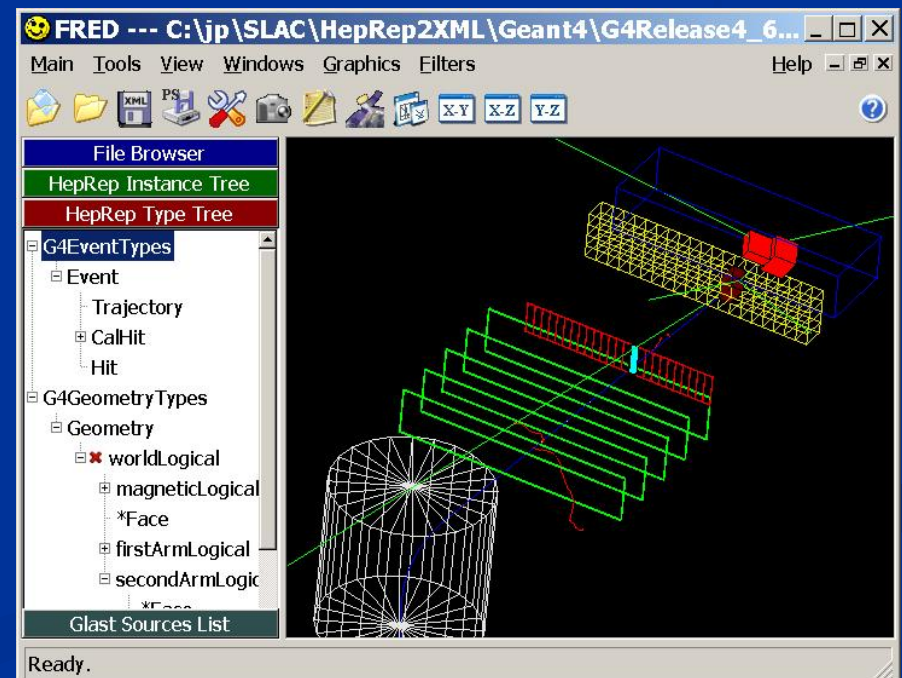
WIRED4: Part of JAS

- n WIRED: World-Wide Web Interactive Remote Display
- n Install JAS (Java Analysis Studio)
- n Then download WIRED4 Plug-in
- n <http://wired.freehep.org>



FRED: Fox Ruby Event Display

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



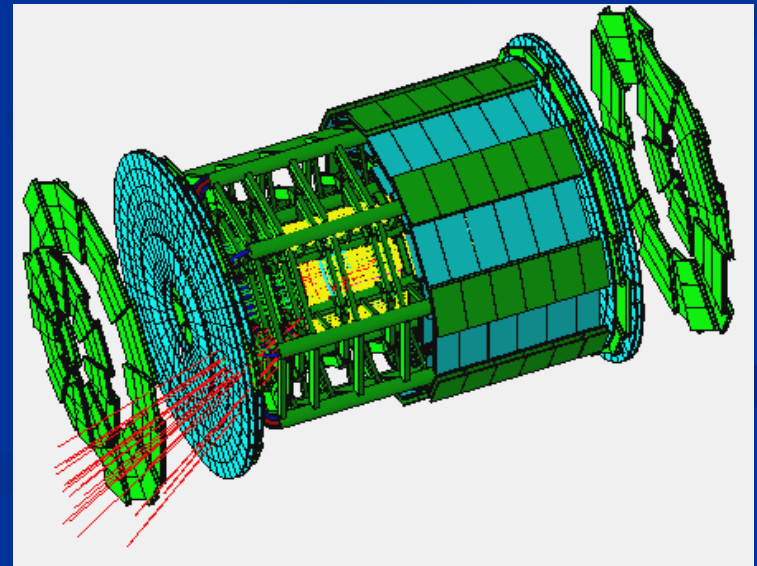
DAWN

- n /vis/open DAWNFILE

- n Features

- n Create a .prim file
 - n Requires DAWN, available for all Linux and Windows systems
 - n DAWN creates a rendered, photorealistic PostScript image
 - n No interactive features once at PostScript stage
 - n Highest quality technical rendering - vector PostScript
 - n View or print from your favorite PostScript application

- n Supported by Satoshi Tanaka



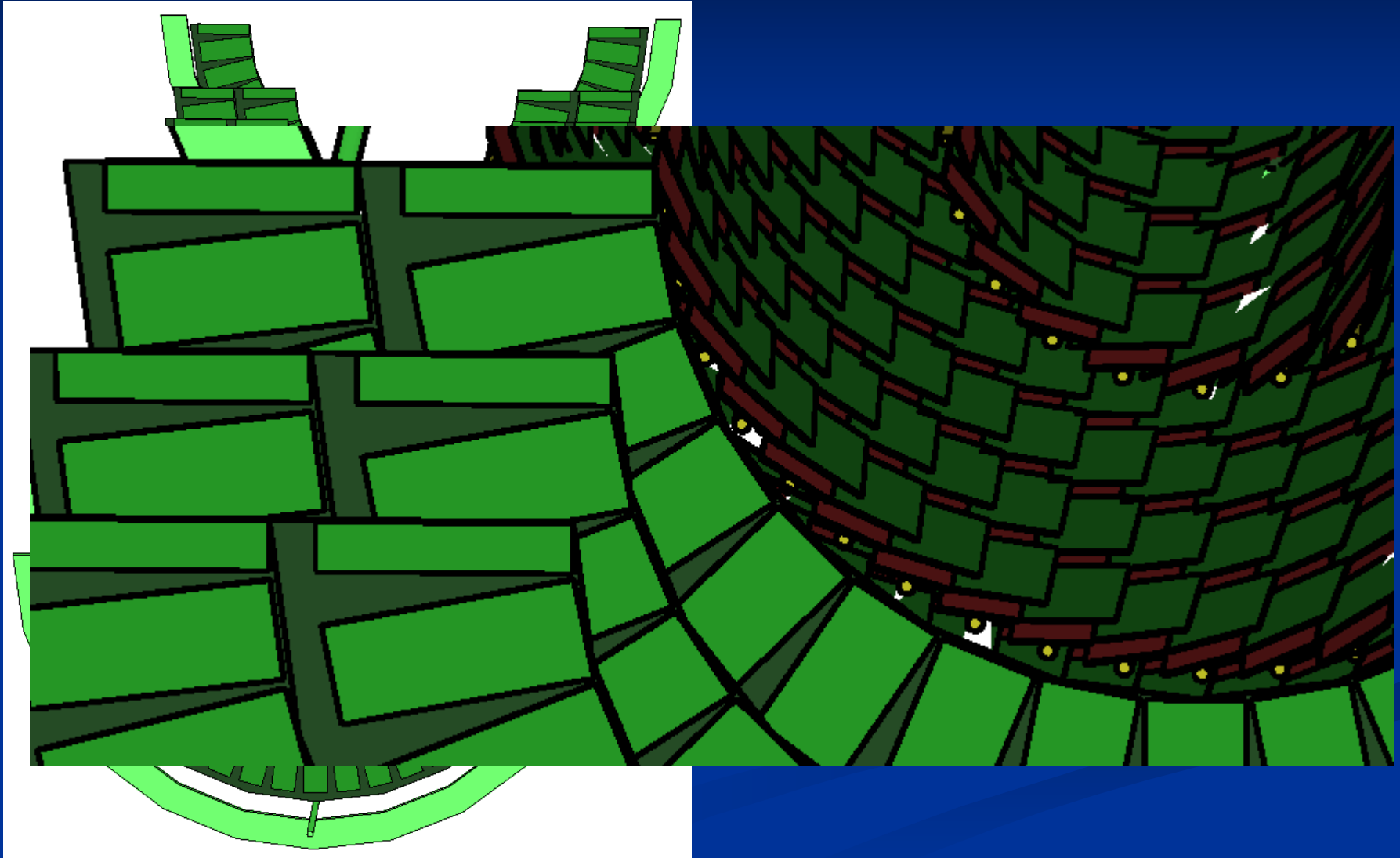
Origins of DAWN

Fukui Renderer DAWN (Drawer for Academic WritINgs).

- n A vectorized 3D PostScript processor with analytical hidden line/surface removal intended for precise technical drawing of complicated objects.
- n Specifically designed for Geant4.
- n Primitives set is same as Geant4 primitives set.
- n Produces device-independent vectorized graphics for high quality technical applications.
- n From a repository of beautiful images at
 - n http://geant4.kek.jp/~tanaka/GEANT4/ATLAS_G4_GIFFIG/

DAWN makes True Vector PostScript

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



DAWN Additional Modes

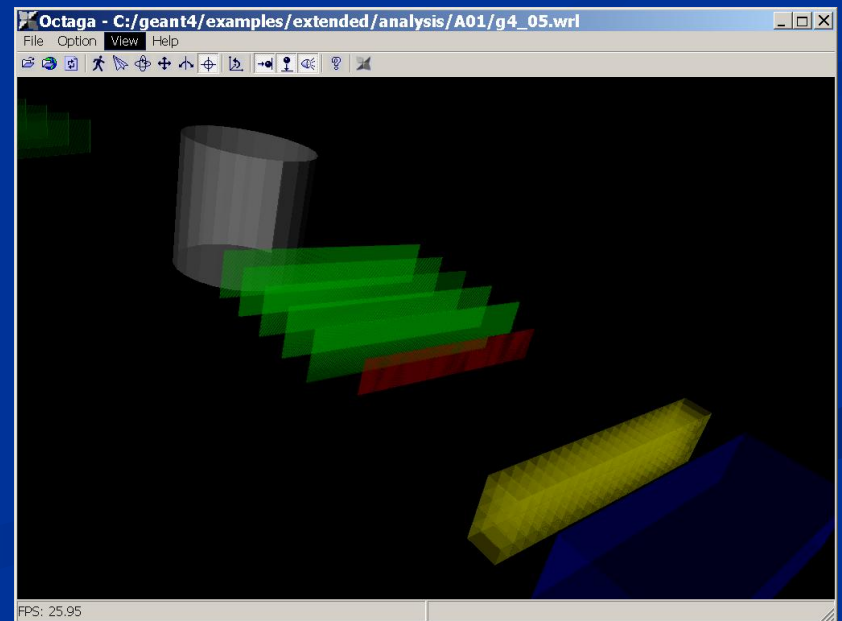
- n There are actually two DAWN drivers:
- n `/vis/open DAWNFILE`
 - n makes a .prim file suitable for viewing in the DAWN renderer
- n `/vis/open DAWN`
 - n establishes a direct connection between Geant4 and the DAWN renderer
 - n The DAWN renderer must be linked in at the time that you link Geant4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

DAWNCUT and DAVID

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

VRML (Virtual Reality Modeling Language)

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



VRML Additional Modes

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

RayTracer

- n /vis/open RayTracer

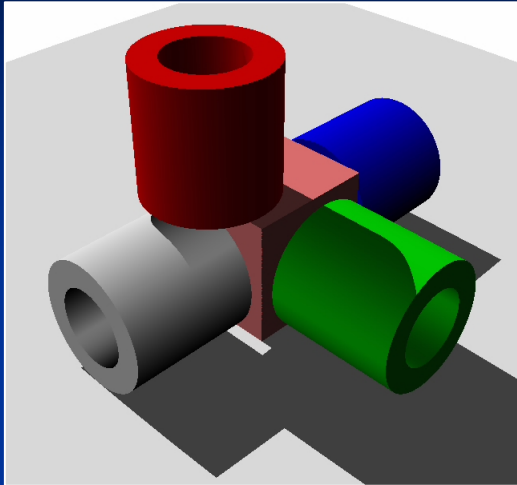
- n Features

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

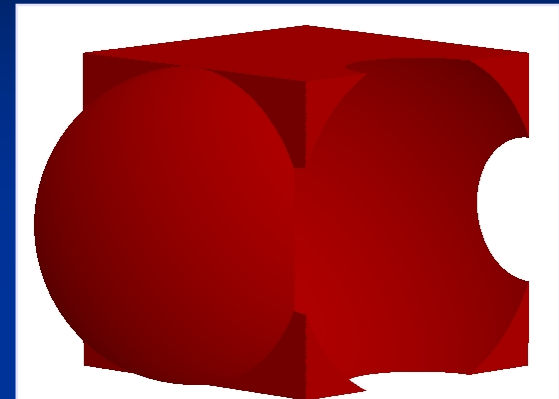
- n Supported by Makoto Asai and John Allison

RayTracer Features

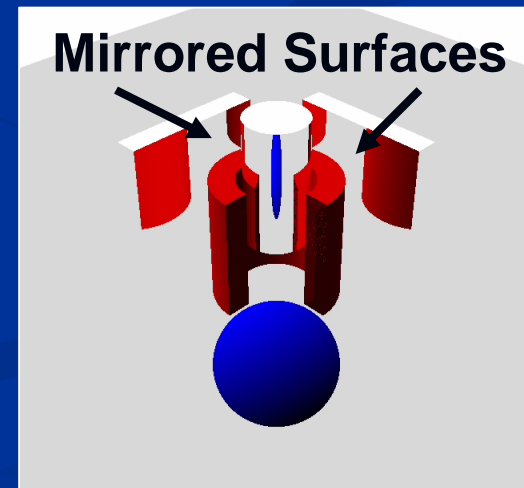
Shows Shadows



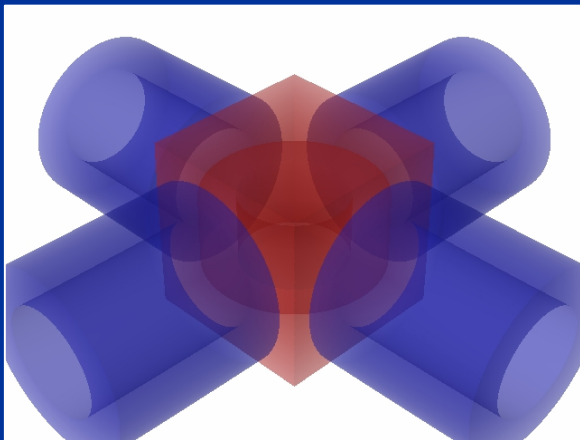
Handles Boolean Solids



Handles Mirrored Surfaces

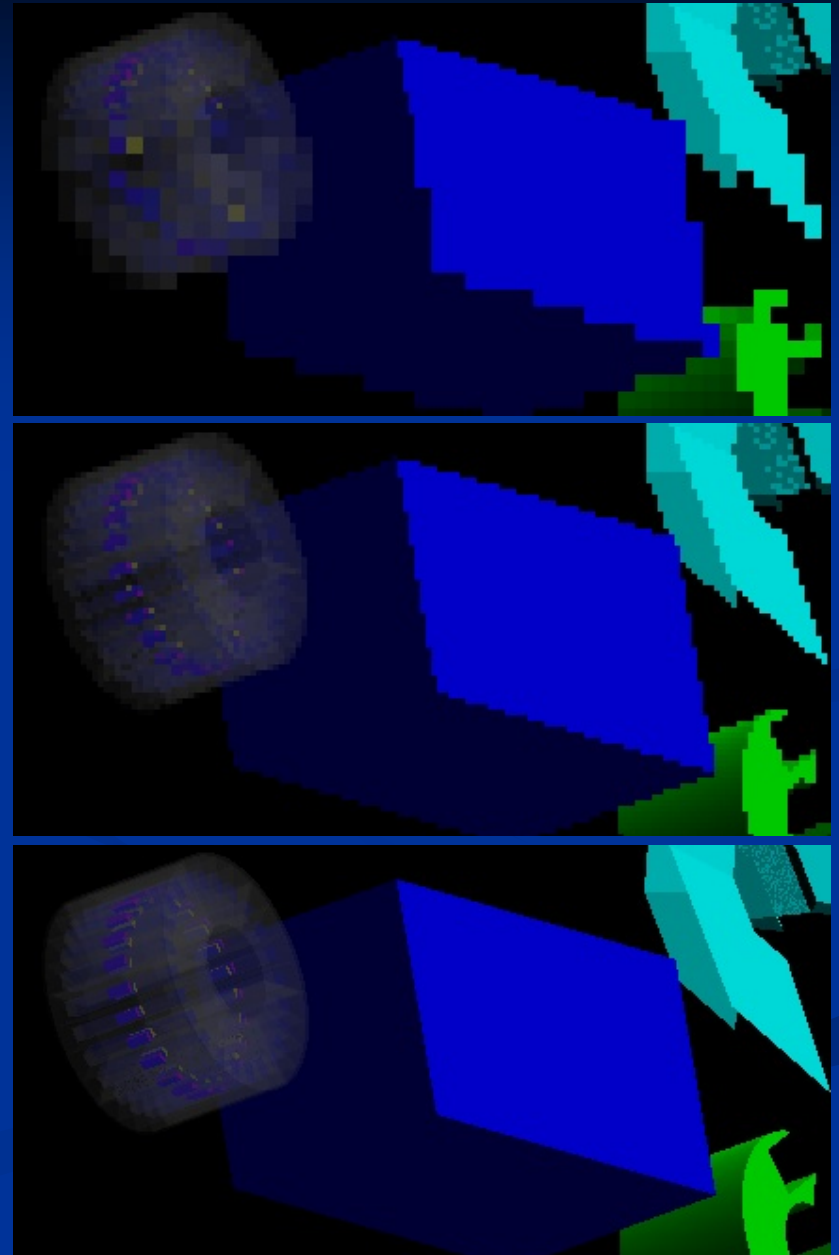


Supports Transparency



RayTracerX

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



ASCIITree

- n /vis/open ATree

- n Features

- n Text dump of the geometry hierarchy
 - n Not graphical
 - n Control over level of detail to be dumped
 - n Can calculate mass and volume of any hierarchy of volumes

- n Supported by John Allison

ASCIITree

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

Choose the Driver that Meets Your Needs

- n If you want very responsive photorealistic graphics (and have the OpenGL libraries installed)
 - n OpenGL is a good solution
(if you have the Motif extensions, this also gives GUI control)
- n If you want very responsive photorealistic graphics plus more interactivity (and have the OpenInventor libraries installed)
 - n OpenInventor is a good solution
- n 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
 - n HepRep will meet your needs
- n 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
 - n DAWN is the way to go
- n If you want to render to a 3D format that others can view in a variety of commodity browsers (including some web browser plug-ins)
 - n VRML is the way to go
- n 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
 - n RayTracer will do it
- n If you just want to quickly check the geometry hierarchy, or if you want to calculate the volume or mass of any geometry hierarchy
 - n ASCIITree will meet your needs

Details

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

Two Page Summary of Visualization Drivers (1)

n OpenGL

- n Control directly from Geant4
- n Uses GL libraries that are already included on most Linux and Windows systems
- n Rendered, photorealistic image with some interactive features
 - n zoom, rotate, translate
- n Fast response (can usually exploit full potential of graphics hardware)
- n Print as pixel graphics or vector EPS
- n Movies

n OpenInventor

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

n HepRep

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

Two Page Summary of Visualization Drivers (2)

n DAWN

- n Create a file to view in the DAWN Renderer
- n Requires DAWN, available for all Linux and Windows systems.
- n Rendered, photorealistic image
- n No interactive features once at PostScript stage
- n Highest quality technical rendering - vector PostScript
- n View or print from your favorite PostScript application

n VRML

- n Create a file to view in any VRML browser (some as web browser plug-ins).
- n Requires VRML browser (many different choices for different operating systems).
- n Rendered, photorealistic image with some interactive features
 - n zoom, rotate, translate
- n Limited printing ability (pixel graphics, not vector graphics)

n RayTracer

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

n ASCIITree

- n Text dump of the geometry hierarchy
- n Not graphical
- n Control over level of detail to be dumped
- n Can calculate mass and volume of any hierarchy of volumes

Geant4 Visualization Resources

Geant4 Installation Guides

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

Hands on HepRApp Tutorial

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

Hands on DAWN Tutorial

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

Hands on OpenGL Tutorial

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

Geant4 Visualization Commands

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

Geant4 Advanced Visualization

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

How to Make a Movie

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

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

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

List of Visualization Commands:

Øhttp://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/AllResources/Control/UIcommands/_vis_.html

For Questions or Comments: Geant4 Visualization Online Forum:

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

References

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