

# QtRoot project status and examples of its use at STAR

by  
Valeri Fine  
STAR

Brookhaven National Laboratory



# Overview

Qt Root project comprises the “**Qt-layer, ?Qt Extensions**” and “**Qt Examples**” those used by STAR to facilitate:

- [Online monitoring](#) – Real time
- [Detector simulation](#) – Full-fledged geometry ( hierarchy ) navigation, Selection. Comparison of the different versions.
- [Event reconstruction](#) – Event representation. Event (flat or simple hierarchy) Navigation and Selection on the top of the simplified detector geometry.
- [Data analysis](#) – big statistic, mostly the large number of the histograms.



# Batch and interactive stages

- GUI is not the first concern of the STAR framework. We worry whether we are capable to collect, preserve, re-distribute our hard-earned PBytes.
- However the final stages of the job are mainly interactive.
- The very first steps of the data-taking in the “control rooms” are interactive also



# Abstraction layers

- To satisfy the different requirement mentioned above several abstraction layers are needed to be designed and implemented
- Do not reinvent the wheel – use the proven ROOT design.

ROOT plug-in does allow to do that.

That way we reach several goals in one shot:

- Re-use STAR component across of the application
- Made it useful for ROOT community also



3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CERN



# TGQt – Qt-based implementation of TVirtualX

- The history of the project and the QtRoot technical implementation details were present at ACAT 2002 (Moscow), ACAT 2003 (Tsukuba), ROOT2005 workshop (Geneva), CHEP 2006 (Mumbai).



# Platforms:

- X11 – Unix / MacOS (need volunteers to test it under Cygwin/X11)
- “Native” MacOS
- “Native WIN32”
  
- PyQt

---

Qt-layer does work from a single source as one had promised .



3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CERN



6

# Build

- As ROOT bundle. Configure ROOT with `--enable-qt`
- Stand-alone with the Qt project file provided: `qmake qt.pro`

The later means one can add the “Qt-layer” plug-in (and upgrade it) to the existent ROOT version and upgrade undependably of the ROOT version as well



# “Qt-layer” package

- New singleton “TQtRootSlot class to connect Qt signals with ROOT “slots”:

```
bool QConnectCint(const QObject * sender, const char * signal)
{
    // Connect the Qt signal to the "execute C++ statement" via CINT SLOT
    // The first parameter of the Qt signal must be "const char*"
    return QObject::connect(sender, signal
        , TQtRootSlot::CintSlot(), SLOT(ProcessLine(const char*)));
}
```

- New TGQt method to decorate ROOT TPad’s using the Qt graphics:

```
virtual void TGQt::CopyPixmap(const QPixmap &p, Int_t px1, Int_t py1);
```





# Qt Extensions:

- **qtgui** (shared lib QtRootGui)
  - TContextMenuImp
  - TCanvasImp
  - TBrowserImp
  - TControlBarImp
  - TInspectorImp
- **qtgl**
  - TVirtuasViewer3D
- **qtged**
  - TVirtualPadEditor
- **qtthread**
  - TThreadImp

Since all of those are ROOT plug-in  
no change of the ROOT user code  
is required

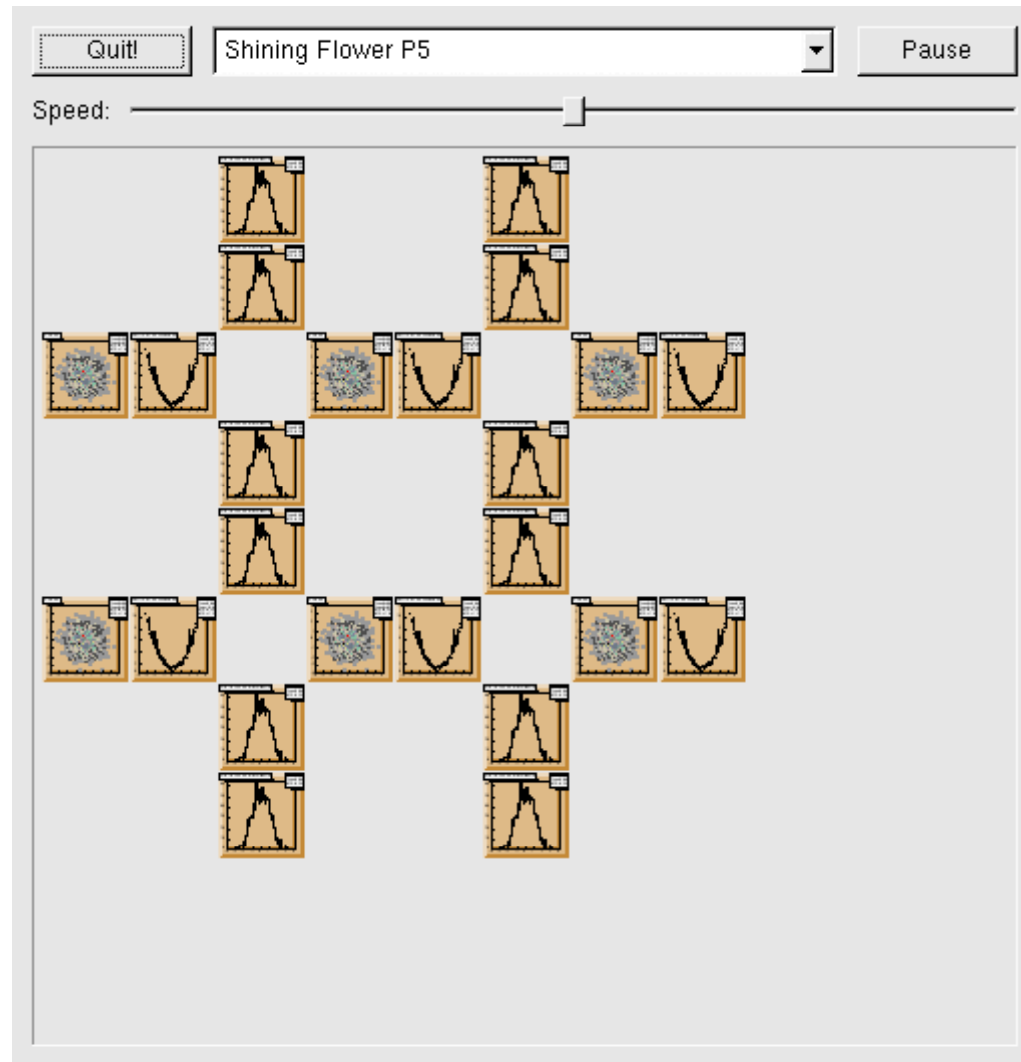


# Qt “Hello Word” Examples

- **HelloWord** - Qt-based ROOT application
- **HelloCanvas** - ROOT-based Qt-application
- **HelloCint(\*)** - ROOT-based shared library
- **QtGBrowser(\*)** - "Advanced version of the Qt application built with the Qt "designer"
- **CustomWidgets**( 7 widgets) – TQtFloatSpinBox, TQtColorSelectButton, TQtPatternSelectButton etc
- **HelloLife** - the example made of the original Qt "life" example
- **HelloPixmap** - the example shows how to decorate ROOT TPad with QPainter
- **CustomCanvasMenu** - It demonstrates how one can customize the automatically generated ROOT Context Menu.
- **macros** - The set of the simple ROOT macros using Qt classes directly



# “Embedded” TPad example:



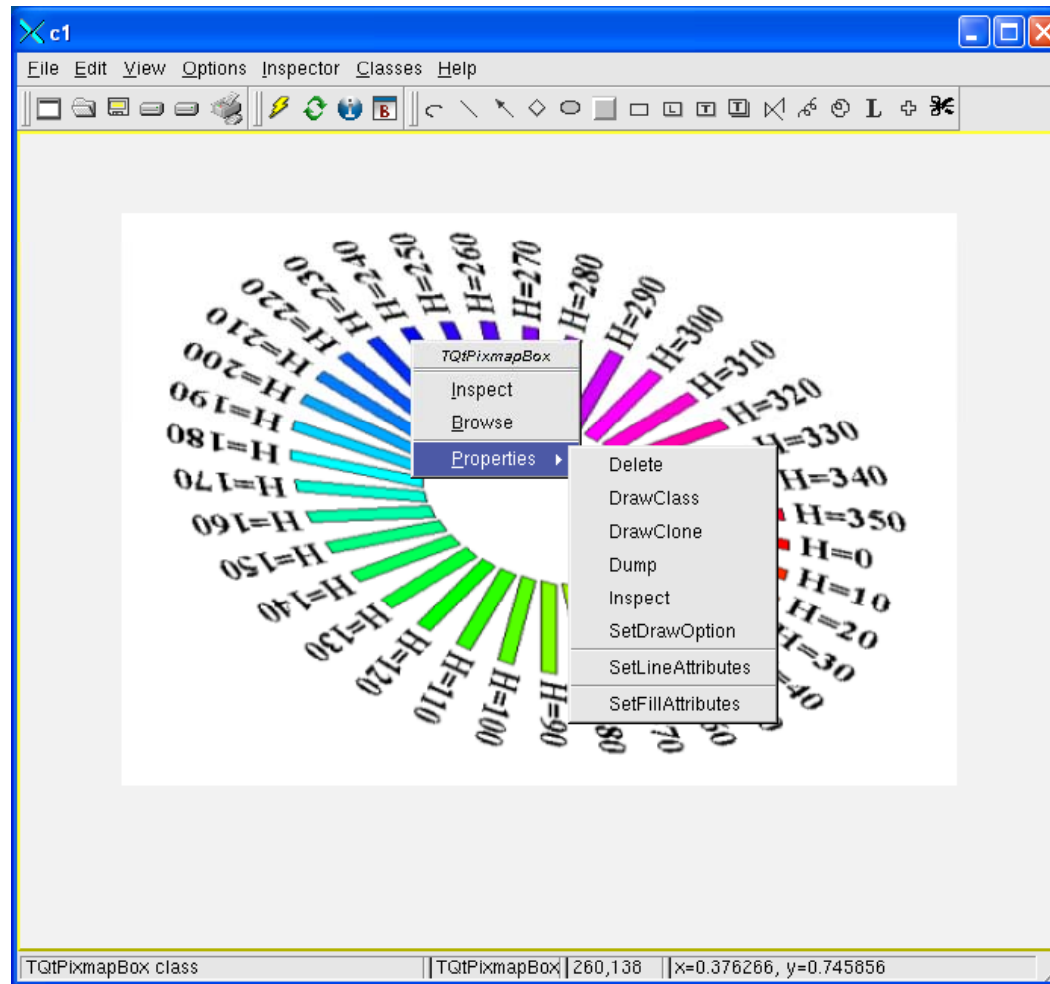
3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CERN

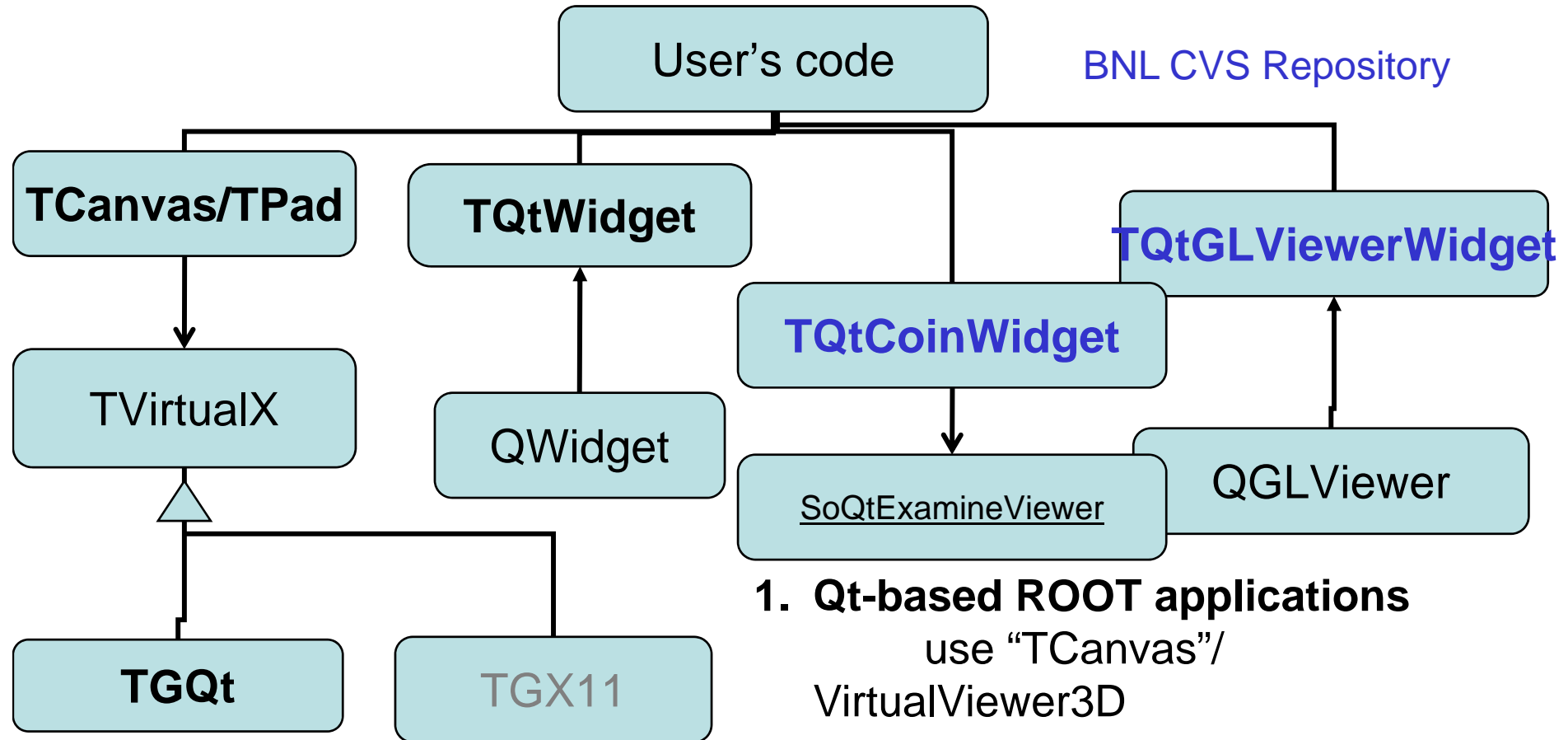


11

# TGQt::CopyPixmap medyhod to decorate ROOT TCanvas



# Two QtRoot end-user “use cases”



BNL CVS Repository

TQtGLViewerWidget

1. **Qt-based ROOT applications**  
use “TCanvas”/  
VirtualViewer3D
2. **ROOT-based Qt application**  
use  
TQtWidget/TQtCoinWidget



3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CERN



13

# 3D ROOT graphics – QtGLViewerWidget

- Can be used with Qt “designer”
- Very simple the end-user interface to pass the “GL list” to be rendered
- Provided with the convenient Qt signals to notify about the GL object selection
- Can be subclassed to meet some customer needs also

The image displays two side-by-side screenshots of Microsoft Internet Explorer. The left window, titled 'Index of QT classes - Microsoft Internet Explorer', shows a webpage for 'TQtGLViewerWidget'. It includes a navigation bar with 'Home', 'Documentation', 'Download', and 'Examples' links. Below the navigation bar is a logo for 'libQGLViewer' and text stating 'Version 2.1 is now available! Supports Qt 4. Dual licensing.' The main content area features a navigation bar with 'Home', 'Documentation', 'Download', and 'Examples' links. Below this is a logo for 'libQGLViewer' and text stating 'Version 2.1 is now available! Supports Qt 4. Dual licensing.' The main content area features a navigation bar with 'Home', 'Documentation', 'Download', and 'Examples' links. Below this is a logo for 'libQGLViewer' and text stating 'Version 2.1 is now available! Supports Qt 4. Dual licensing.' The main content area features a navigation bar with 'Home', 'Documentation', 'Download', and 'Examples' links. Below this is a logo for 'libQGLViewer' and text stating 'Version 2.1 is now available! Supports Qt 4. Dual licensing.'



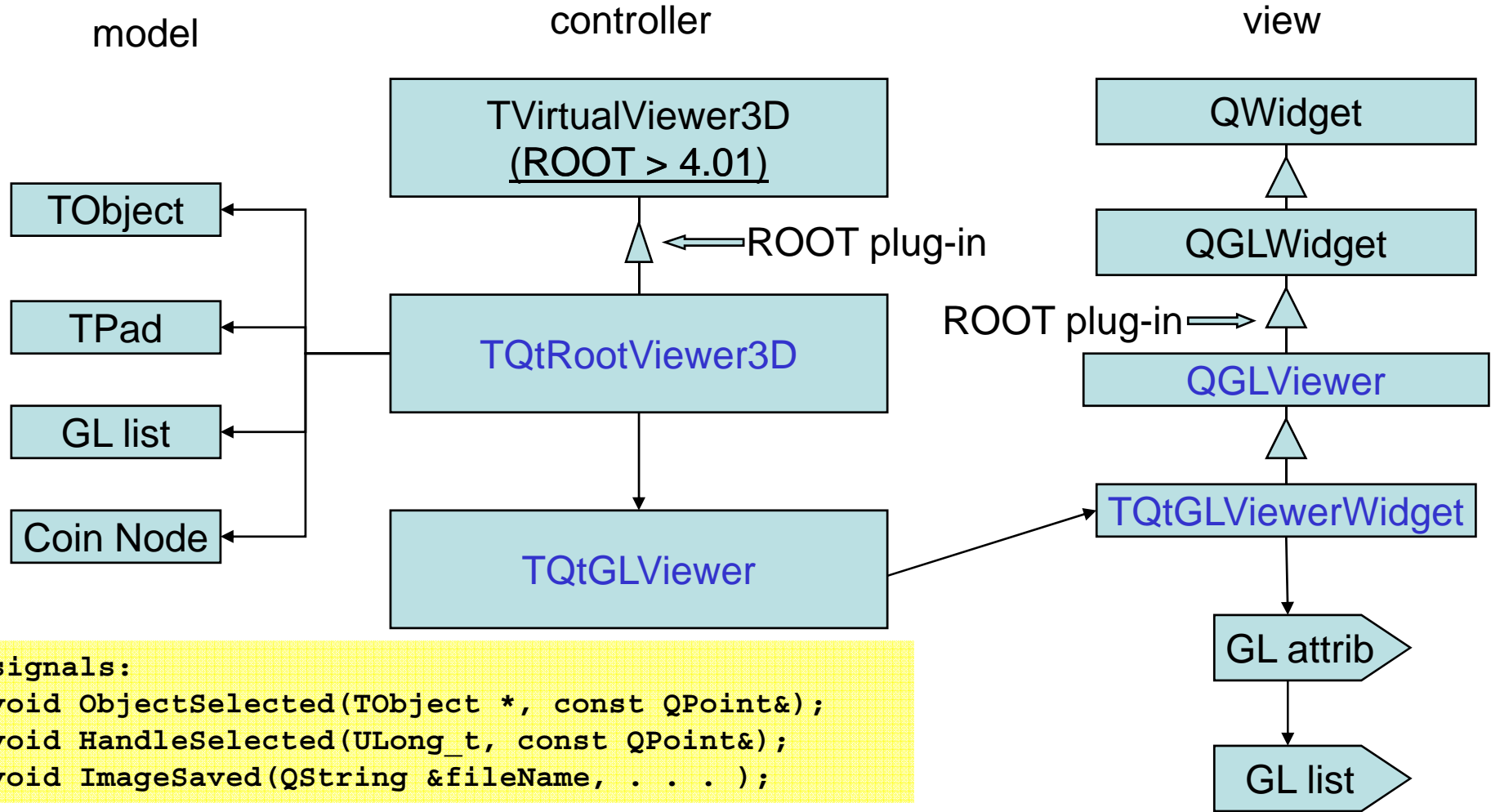
3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CERN

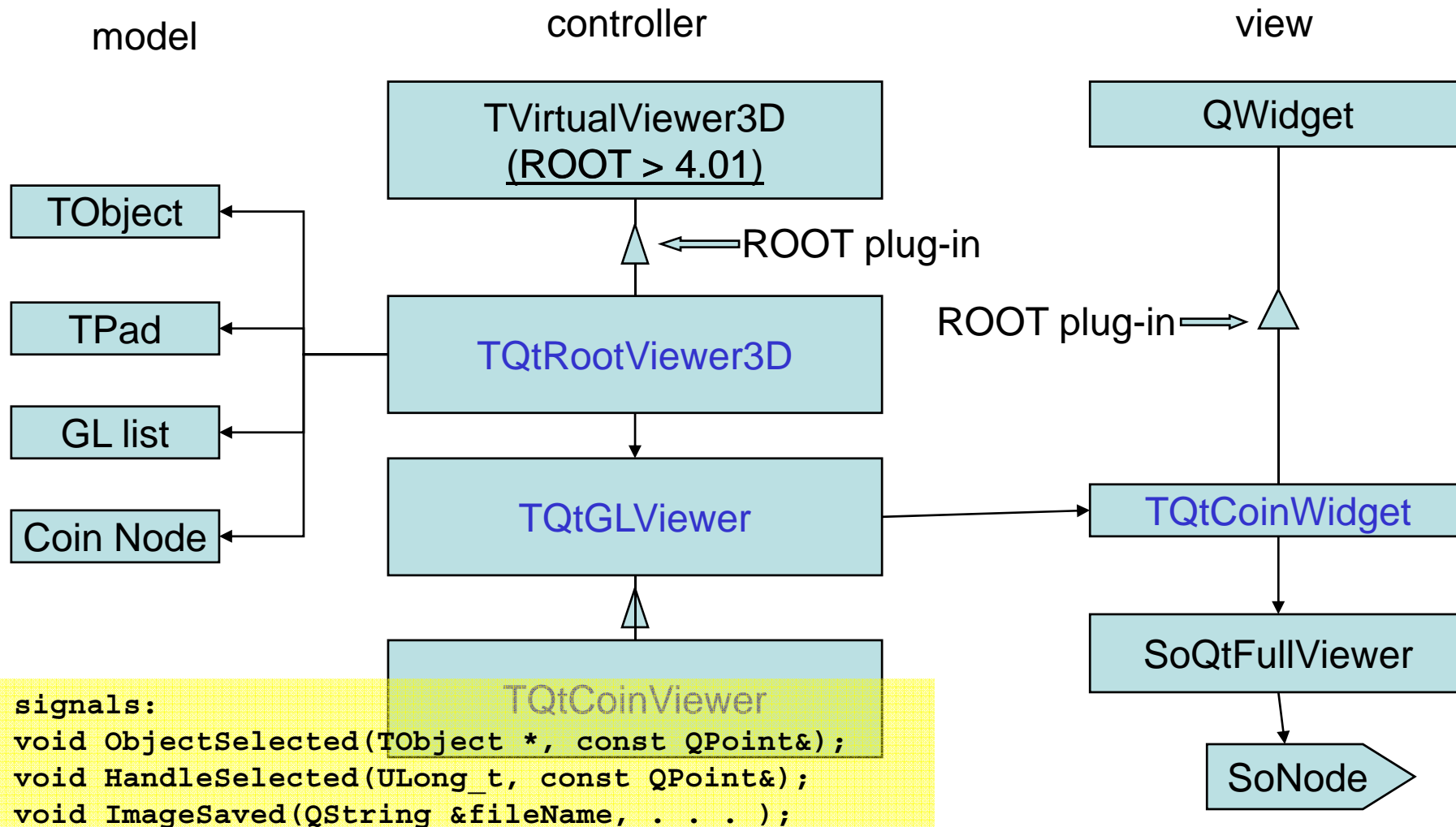


14

# Qt implementation of ROOT 3D viewer interface classical: Model-View-Controller (see ROOT 2005 talk)



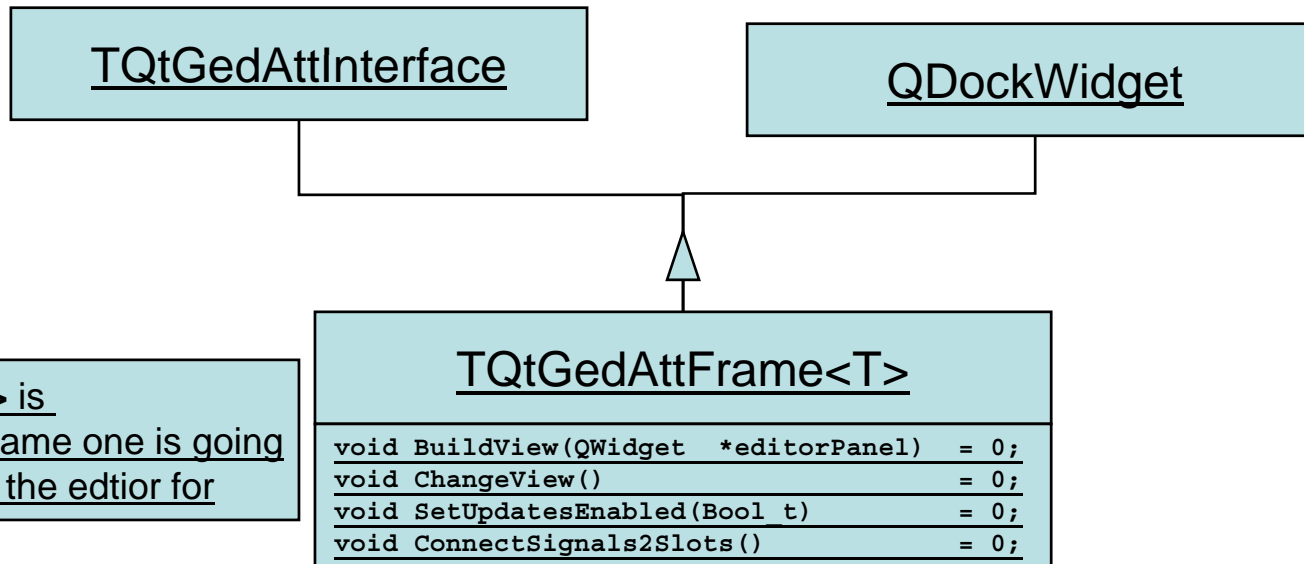
# Coin3D implementation of ROOT 3D viewer interface



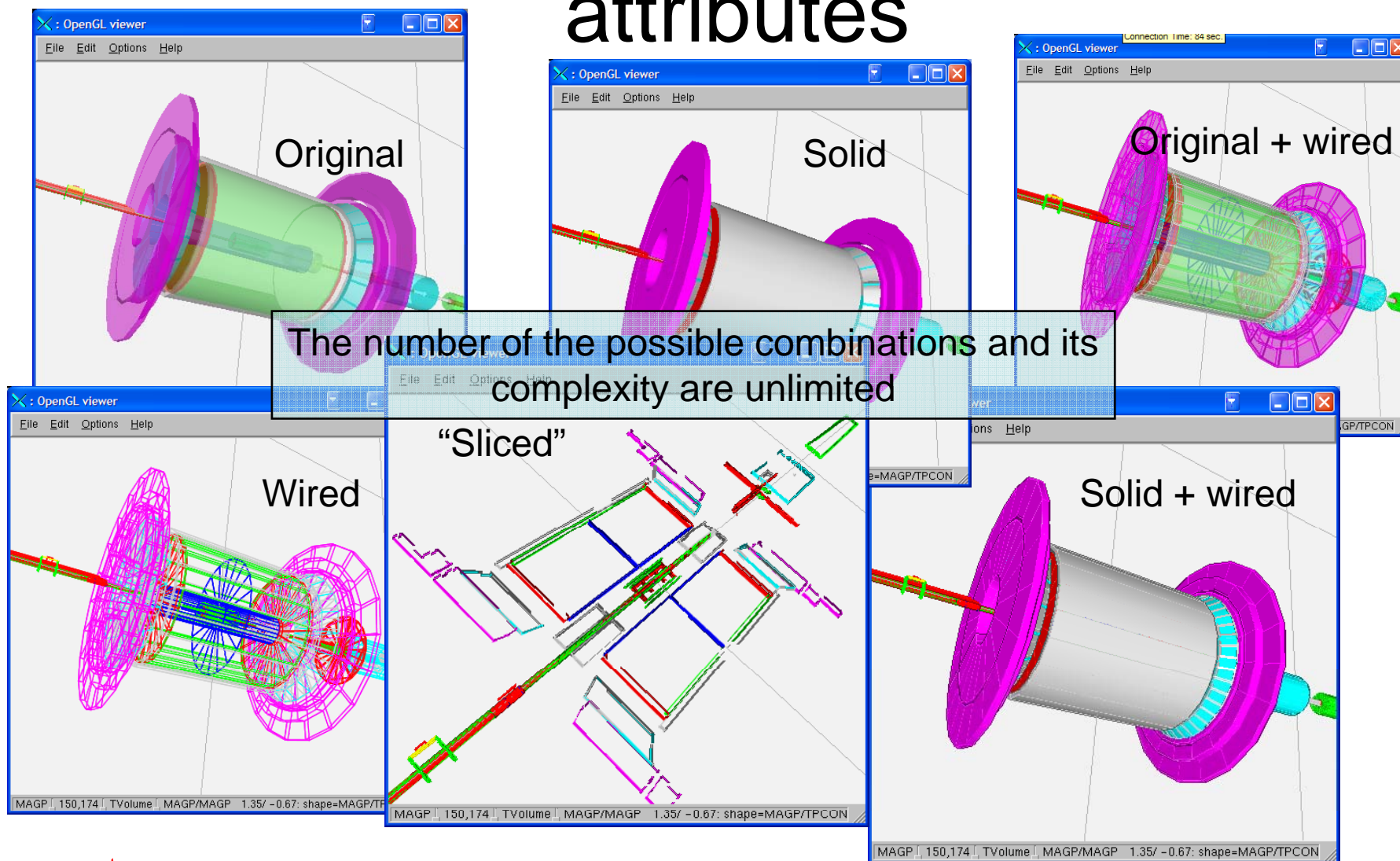


# Qt object editor framework

The design follows one from the ROOT “ged” deliberately  
If one knows how to create the ROOT Object editor one should find no trouble creating the one for Qt\_



# Combination of the different GL attributes



Slide 18

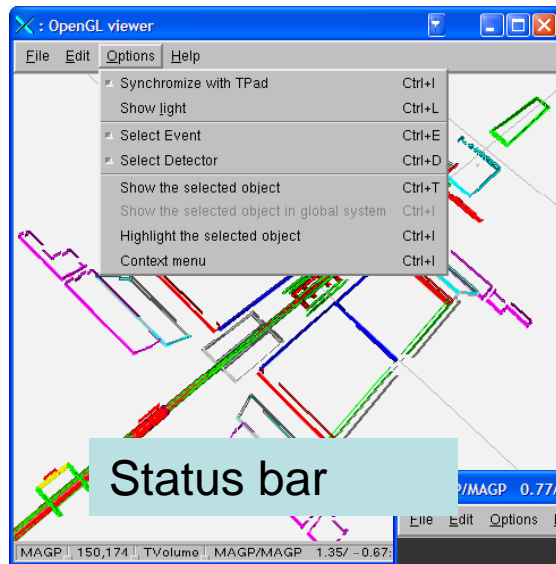
---

vf1

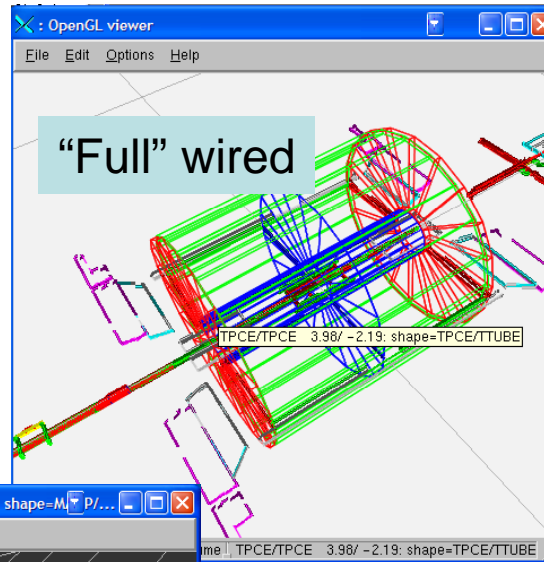
Should be clearer this comes for "free" ...

Valeri Fine, 3/27/2007

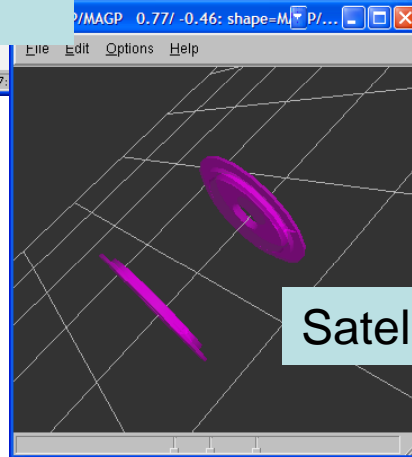
# 3D selection and highlighting



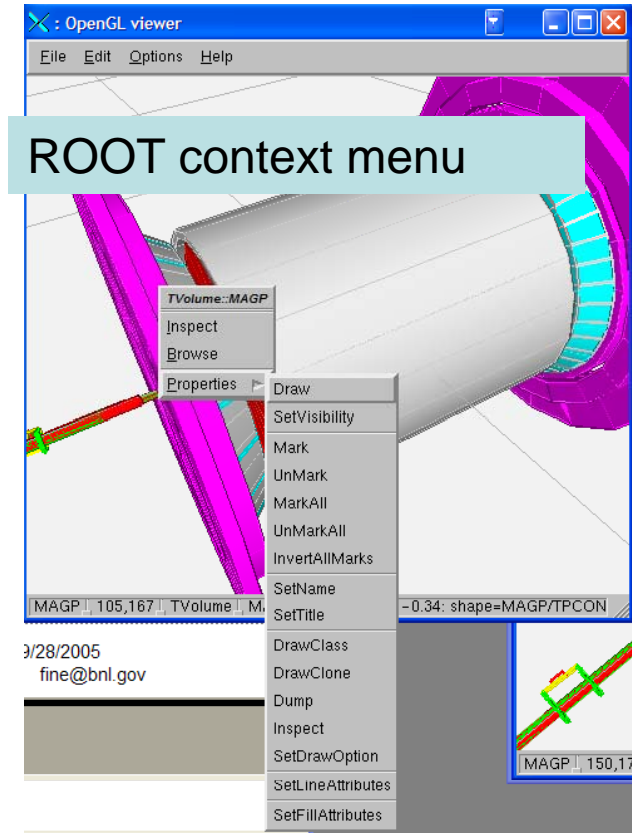
Status bar



"Full" wired



Satellite widget




ROOT context menu



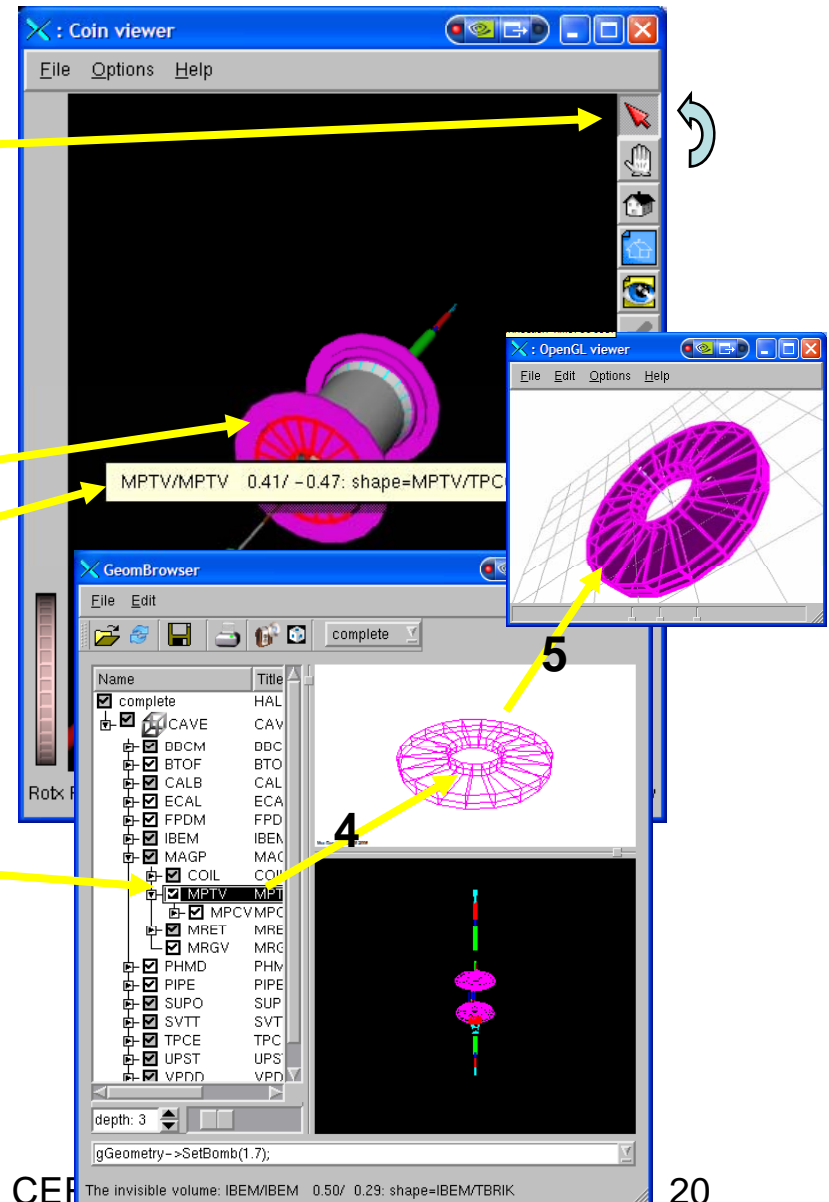
# The Default Selection Slot

To select object one has to

- Turn the Coin viewer “selection” tool on 
- Use the left mouse button to point the image on the 3D view

The browser should:

1. Highlight the selected shape
2. Popup the label with the text provided by the selected object TObjec::Info method
3. Find the object in the left tree pane of the Geometry Browser and select the found object there
4. The object selected in the “tree” list should be painted in the right upper TCanvas widget



3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CE

BROOKHAVEN  
NATIONAL LABORATORY

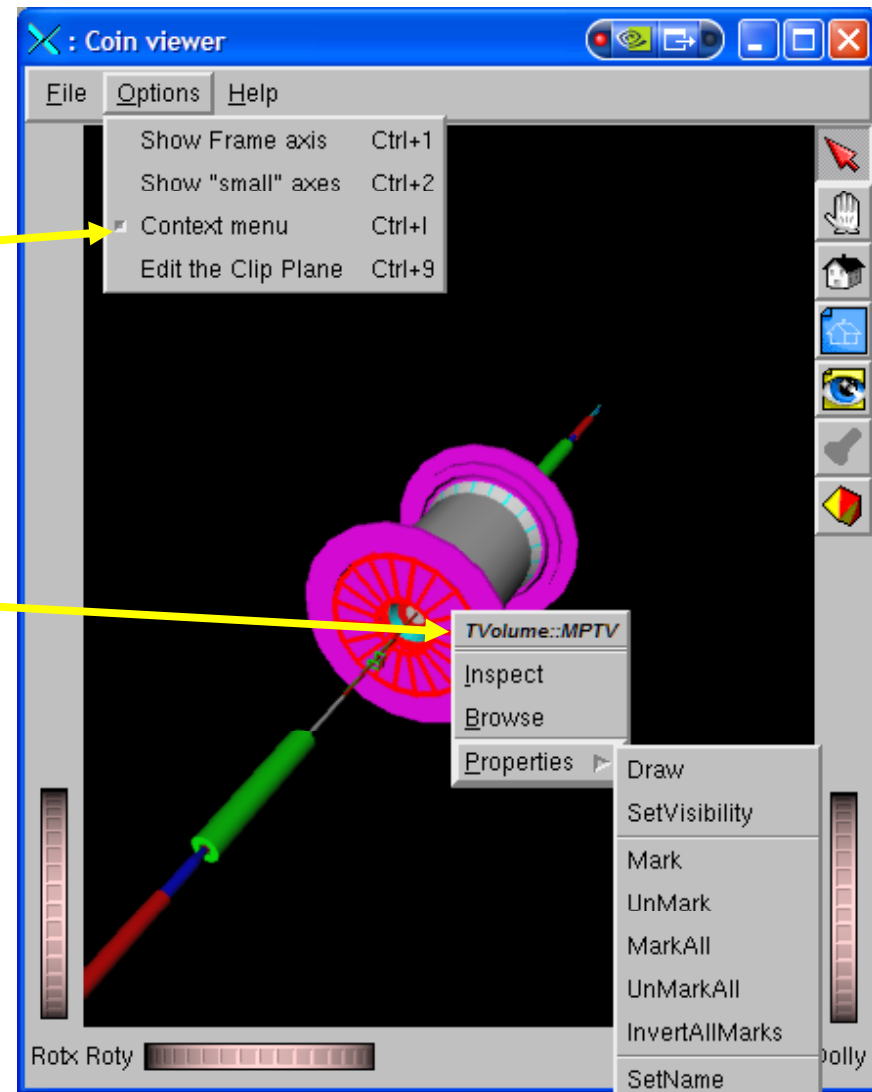
20

# The “Menu” Selection Slot

The Browser provides another “selection” slot.

The SLOT is activated via the drop-down “Options” menu.

To select the object user should follow the previous slide. The only difference, instead of the label the browser will pop the ROOT “Context menu”

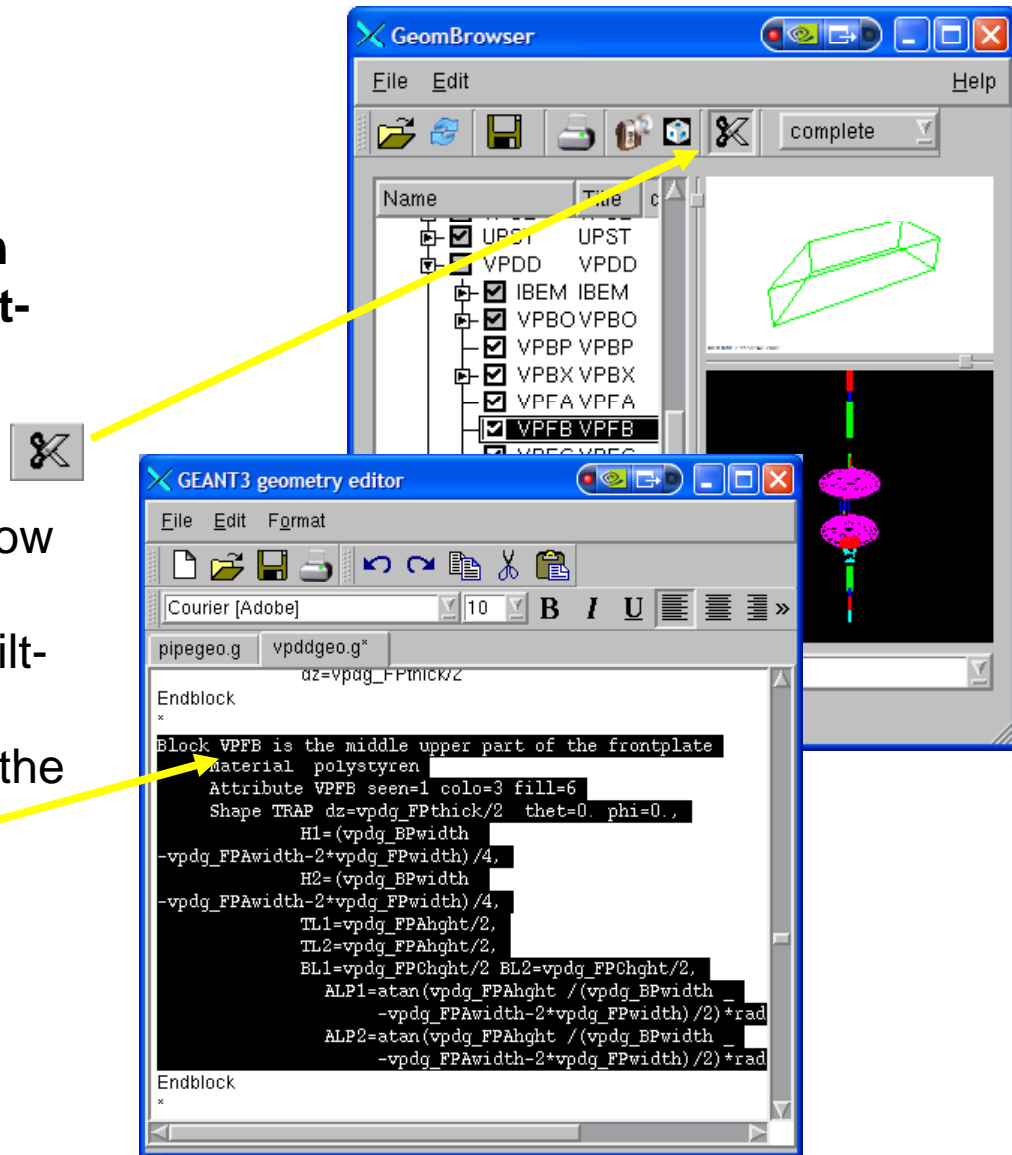


# The “GeomBrowse” Custom Selection

The STAR Geometry Browser provides the custom selection slot in addition to the two built-in slots described above

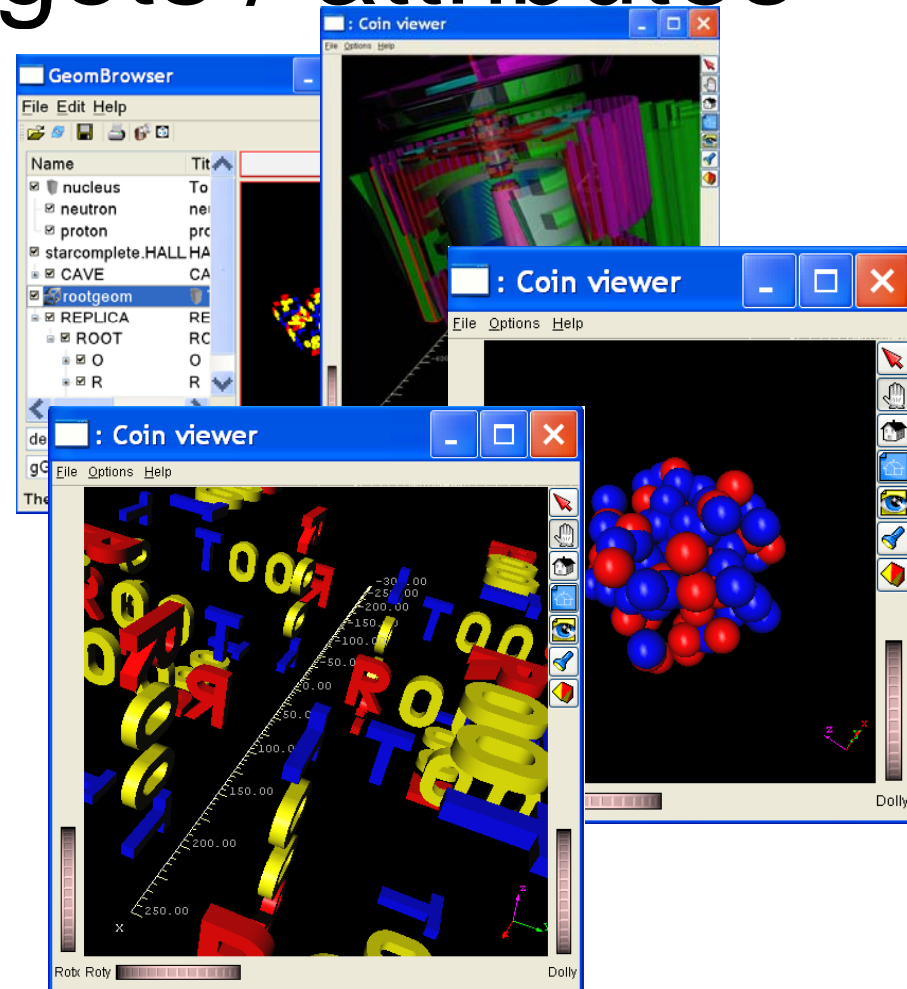
The SLOT is activated via the icon. 

To select the object user should follow the previous slides. The only difference, **in addition** to the built-in action, the application is to popup the text edit window with the source code of the selected volume highlighted.



# Multiply 3D widgets / attributes

The Qt-based ROOT 3D plug-ins do allow creating as many OpenGL widgets as your local video hardware can sustain and set the video attributes for each widget separately as the [short animation](#) on the right demonstrates:



3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CERN

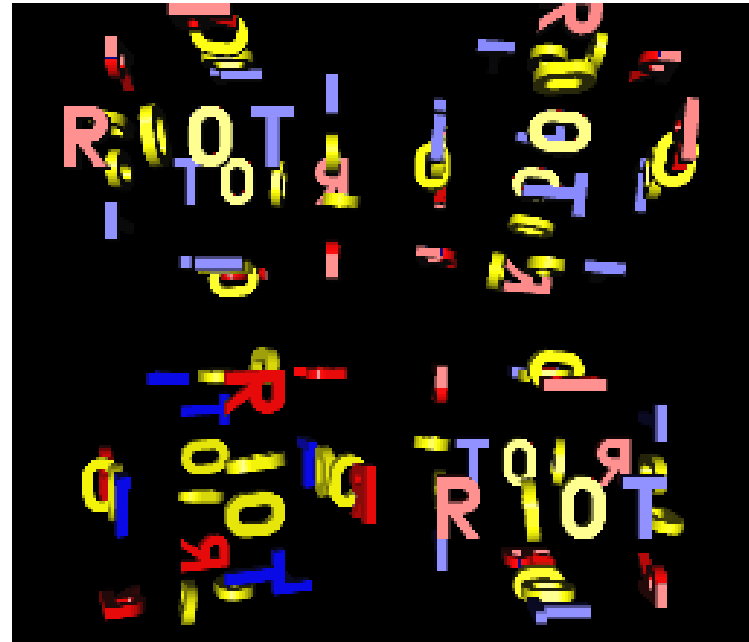


23



# Mixed ROOT / Coin3D 3D scene

It is simple to create the ROOT objects with the mixing ROOT 3D and Coin3D objects. For example the ROOT objects define the geometry (shapes, color and positions) and Coin3D define the animations, lighting, clipping, 3D labels etc. The mixed objects can be saved/restored using the regular either ROOT (root file) or Coin3D (see slide 4) I/O



The `rootgeom_coin.c` ROOT macro creates such object and renders it with non-Coin3D TCanvas and Coin3D ROOT plug-in.



3/27/2007  
fine@bnl.gov

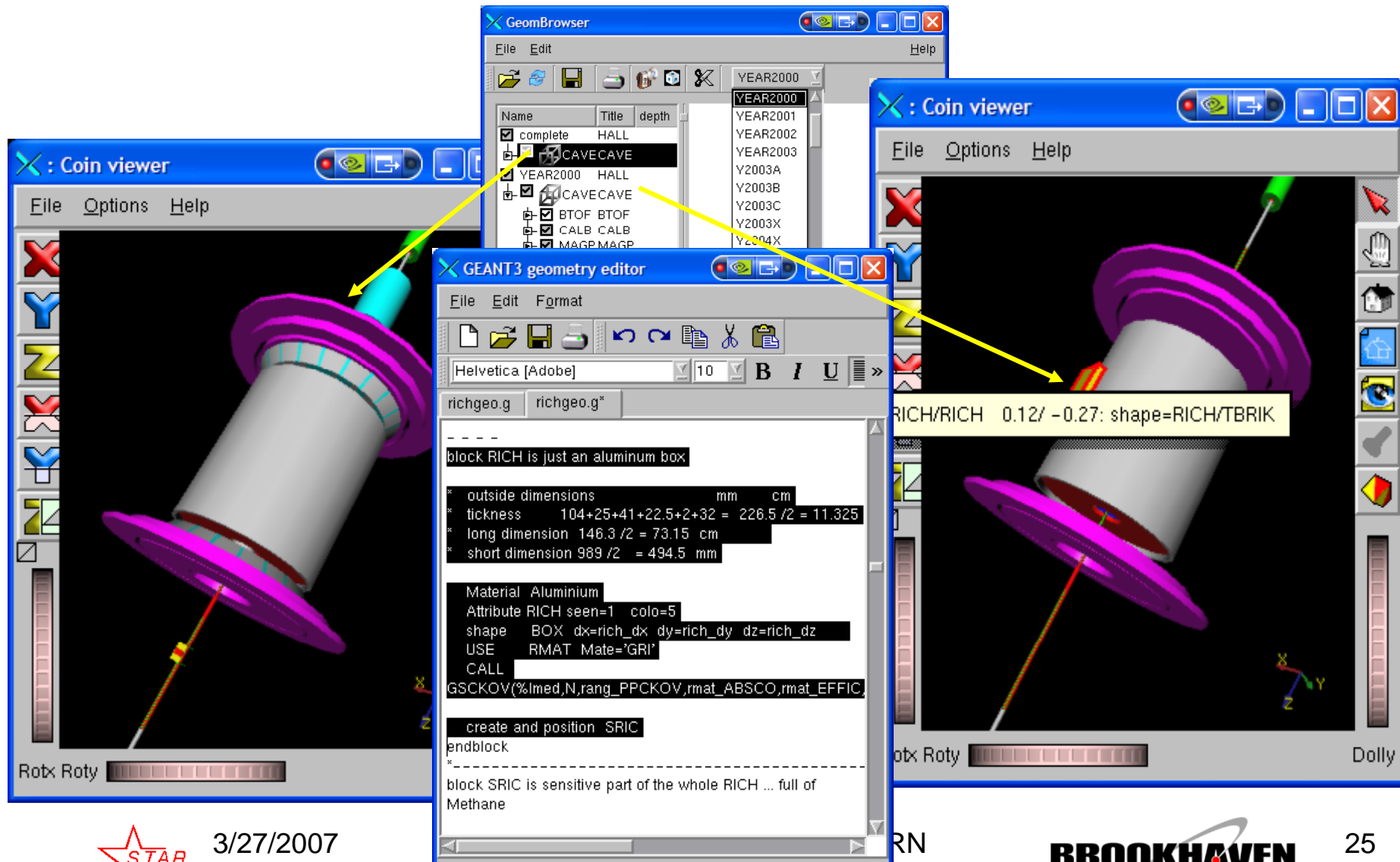
ROOT 2007 Workshop, CERN



24

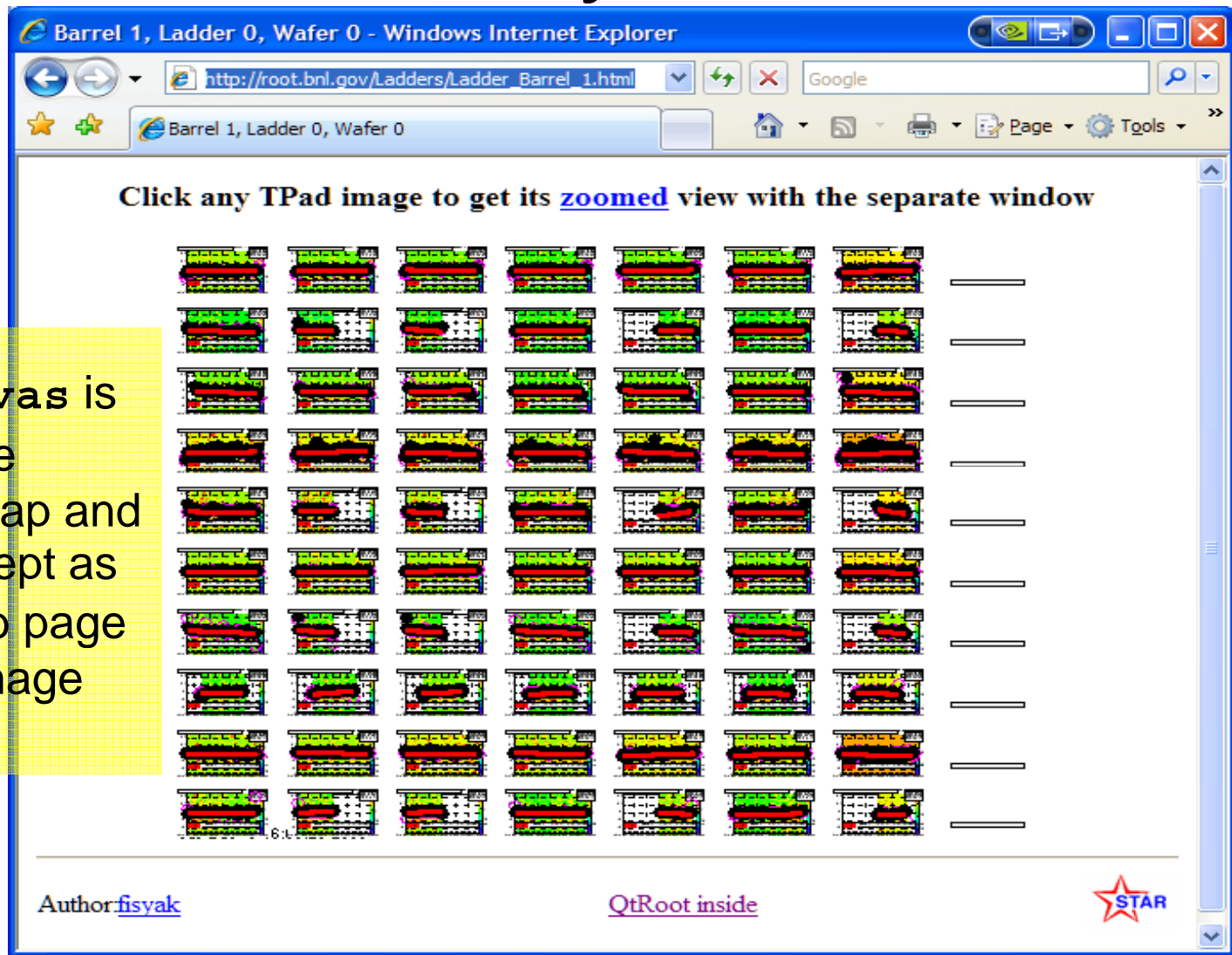
# Detector simulation

## Compare two versions of the STAR



3/27/2007  
fine@bnl.gov

# Data Analysis



One click: TCanvas is converted to the HTML image map and each TPad is kept as a separate Web page linked by the image map.



3/27/2007  
fine@bnl.gov

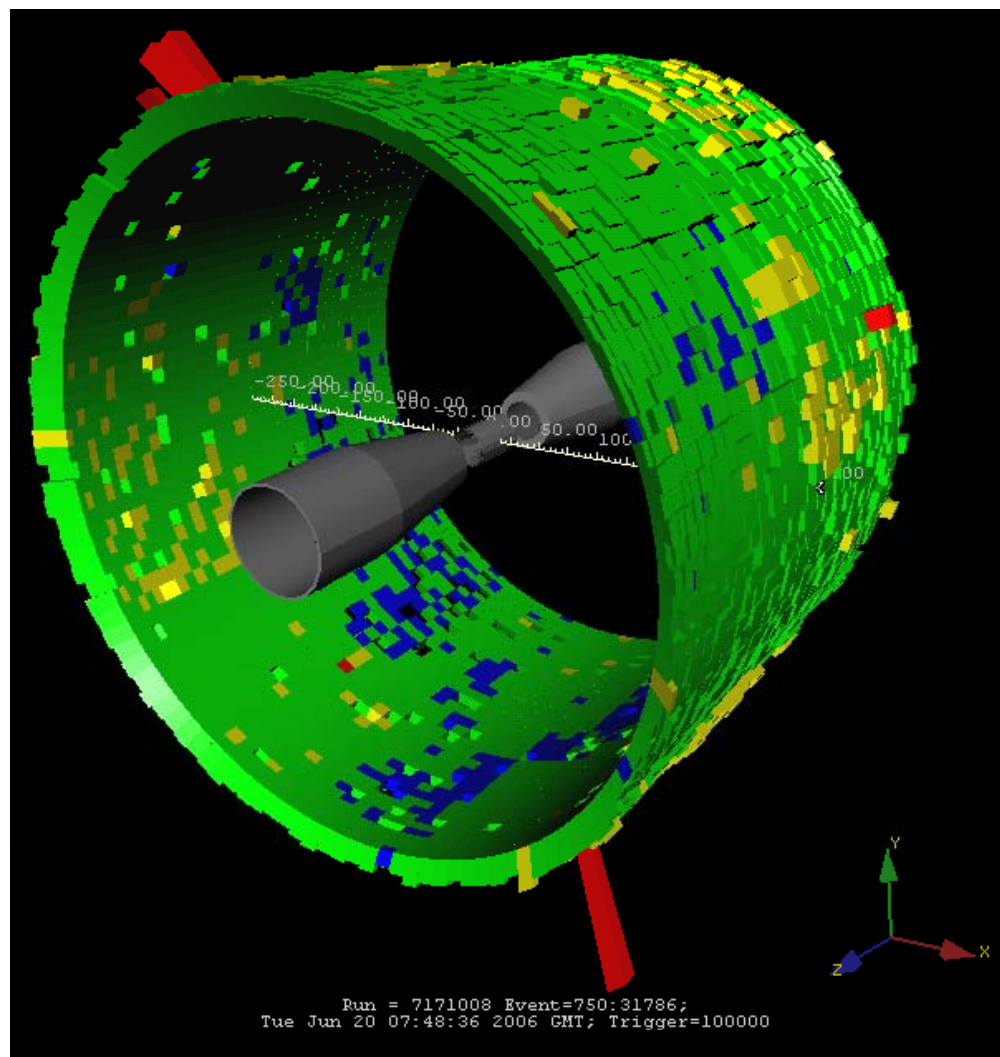
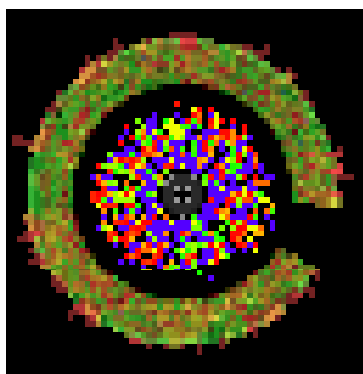
ROOT 2007 Workshop, CERN

**BROOKHAVEN**  
NATIONAL LABORATORY

26

# Online monitoring

During this Run we will be using the 3d generation of the Monitor



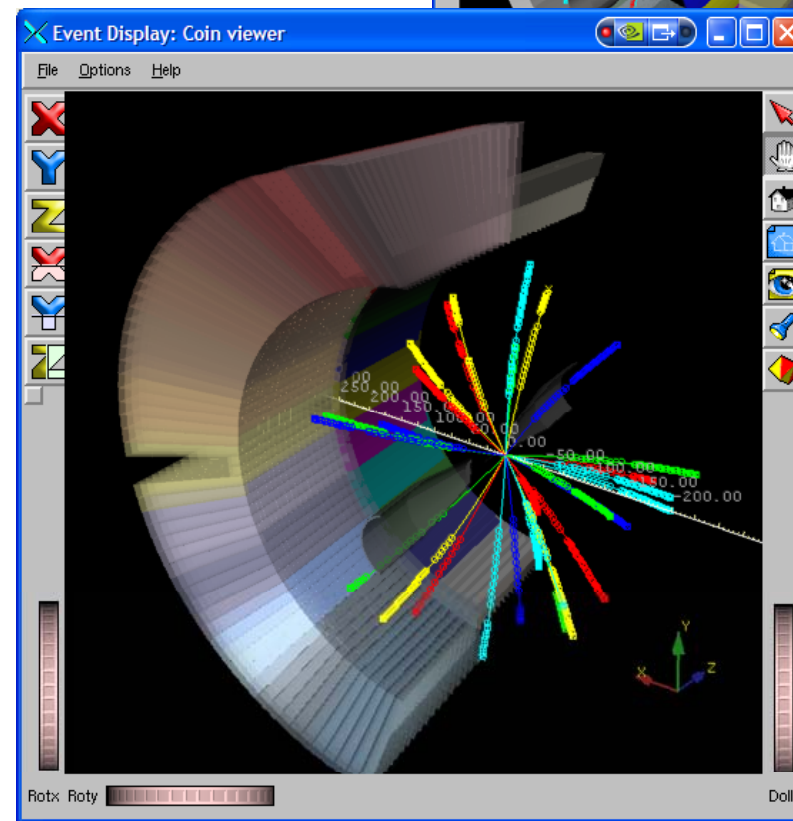
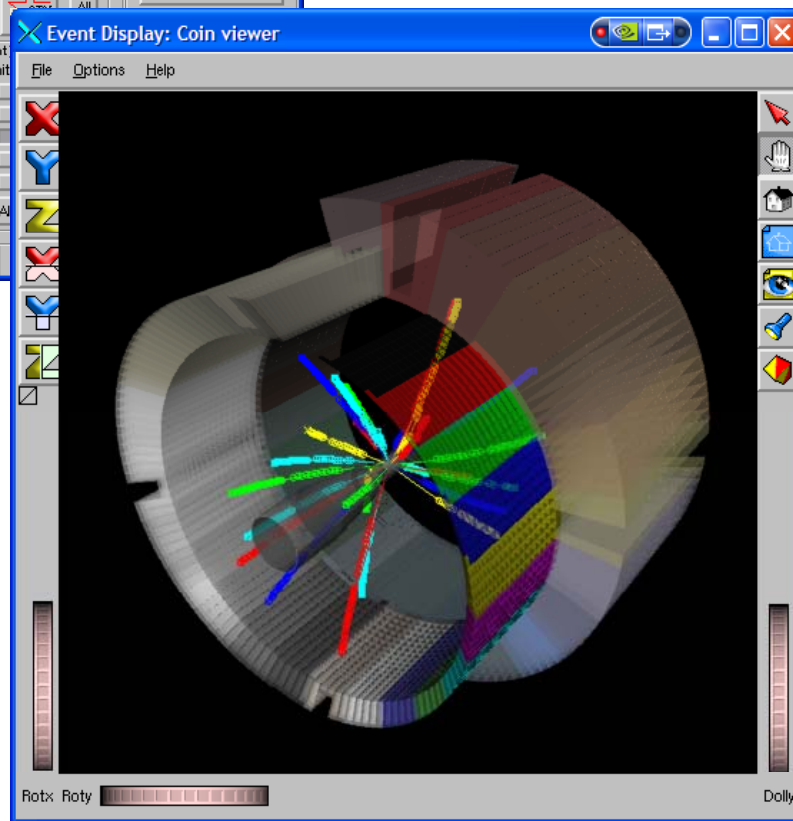
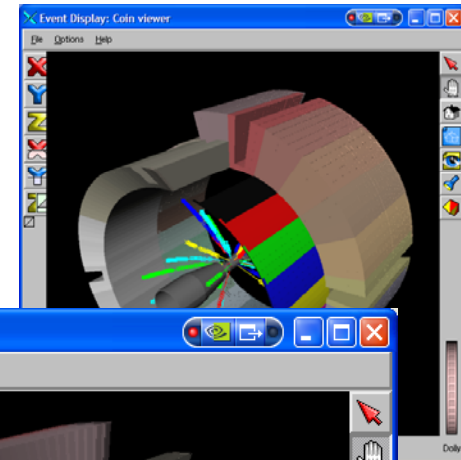
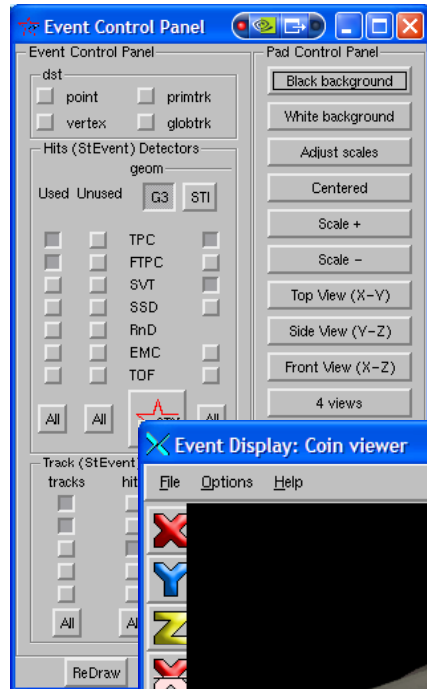
3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CERN



# Event reconstruction

## (cont)



3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CERN

**BROOKHAVEN**  
NATIONAL LABORATORY

28

# Upcoming

- Qt4 is imminent.
- STAR plans to complete it by July, 2007
- Needs to understand how long we should keep the Qt3 brand if any.
  - Ideally, would like to switch to qt4 support (only) in 2008
  - Will support longer depending on community needs (please, feedback)



3/27/2007  
fine@bnl.gov

ROOT 2007 Workshop, CERN



29