ROOT Project Status Report

LCG AA Internal Review

18 September 2006

René Brun

CERN

Plan

- There are 3 other talks
  - Mathlibs by Lorenzo Moneta
  - CINT/Reflex by Axel Naumann
  - PROOF by Fons Rademakers
- I will give a short overview of ROOT in general and in particular on the other work packages not covered in these 3 talks
  - Project Organization
  - Releases
  - CORE
  - ROOT I/O and Trees
  - GUI
  - Graphics
  - Geometry

Project Organization

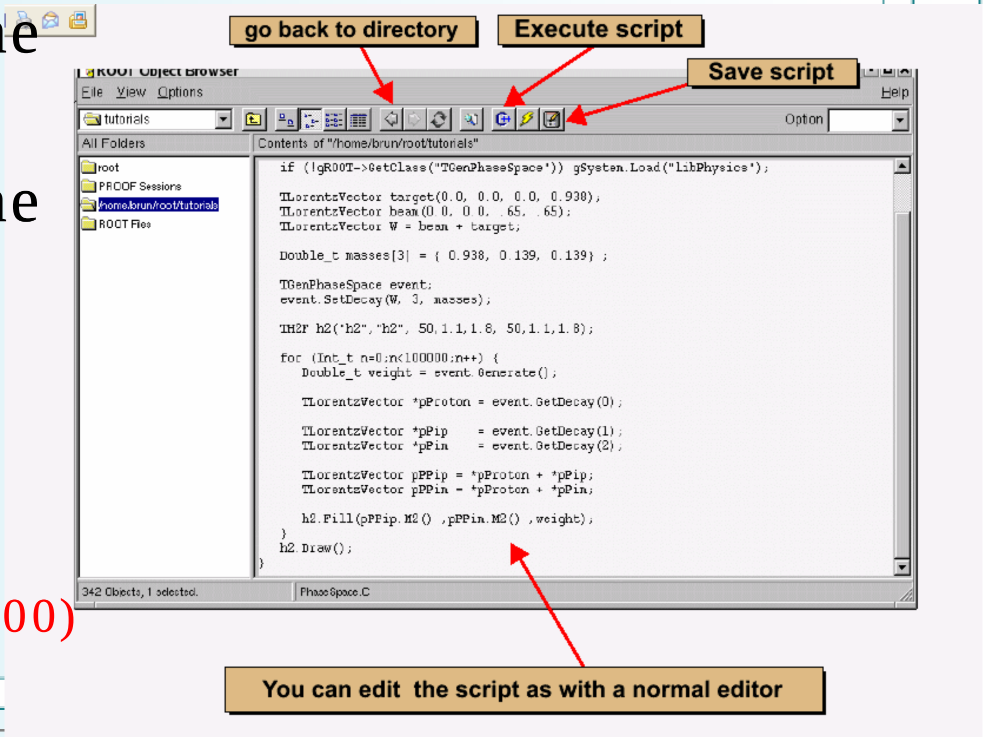
- BASE: Fons Rademakers, Bertrand Bellenot(also SPI)
- I/O: Philippe Canal, Paul Russo(FNAL), Markus Frank(<10%)
- DICT: Philippe Canal, Axel Naumann, Stefan Roiser
- MATH: Lorenzo Moneta, Anna Kreshouk
- GUI: Ilka Antcheva, Bertrand Bellenot
- GRAPHICS: Olivier Couet
- GEOM: Andrei Gheata, Mihaela Gheata (Alice)
- PROOF: Fons, Maarten Ballintijn(MIT), Gerri Ganis, Bertrand , Leandro Franco, Ian Iwaszkiewicz, Andreas Peters(arda)
- SEAL: Lorenzo

Main Events

- ROOT Workshop at CERN October 2005
- PRO version 5.08/00 15 Dec 05
  - http://root.cern.ch/root/Version50800.news.html
- CHEP06 Mumbai: 15 talks/posters
- PRO version 5.10/00 02 Mar 06
  - http://root.cern.ch/root/Version51000.news.html
- PRO version 5.12/00 11 Jul 06 + Users Guide
  - http://root.cern.ch/root/Version51200.news.html
- DEV version 5.13/02 30 Aug 06
  - http://root.cern.ch/root/Version51300.news.html
- DEV version 5.13/04 11 Oct 06
- PRO version 5.14/00 15 Dec 06

Improvements in BASE

- Improvements in plug-in manager and class auto-loader.
- New MonaLisa monitoring plug-in.
- New class TMacro (C++ scripts in memory that can be stored in ROOT files, send through the network).
  - Important for PROOF and Event Displays
- Many improvements in the test suite (roottest)
- Many improvements in the installation system
- Coding conventions. Rule checker
- New Users Guide
- Roottalk(4200) + RootForum(15400) + RootDev(?) + Savannah(700)



Contents of "/home/brun/root/tutorials"

```
   if (!gROOT->GetClass("TGenPhaseSpace")) gSystem.Load("libPhysics");

   TLorentzVector target(0.0, 0.0, 0.0, 0.938);
   TLorentzVector beam(0.0, 0.0, .65, .65);
   TLorentzVector W = beam + target;

   Double_t masses[3] = { 0.938, 0.139, 0.139} ;

   TGenPhaseSpace event;
   event.SetDecay(W, 3, masses);

   TH2F h2("h2","h2", 50,1.1,1.8, 50,1.1,1.8);

   for (Int_t n=0;n<100000;n++) {
      Double_t weight = event.Generate();

      TLorentzVector *pProton = event.GetDecay(0);

      TLorentzVector *pPip    = event.GetDecay(1);
      TLorentzVector *pPim    = event.GetDecay(2);

      TLorentzVector pPPip = *pProton + *pPip;
      TLorentzVector pPPim = *pProton + *pPim;

      h2.Fill(pPPip.M2() ,pPPim.M2() ,weight);
   }
   h2.Draw();
}
```

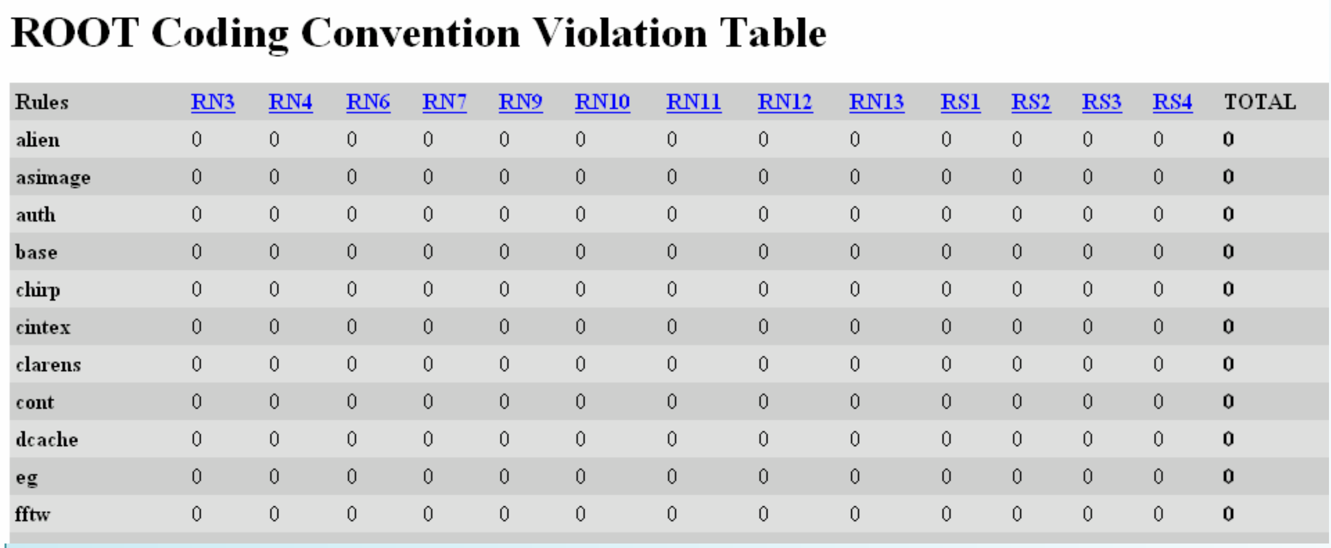
go back to directory

Execute script

Save script

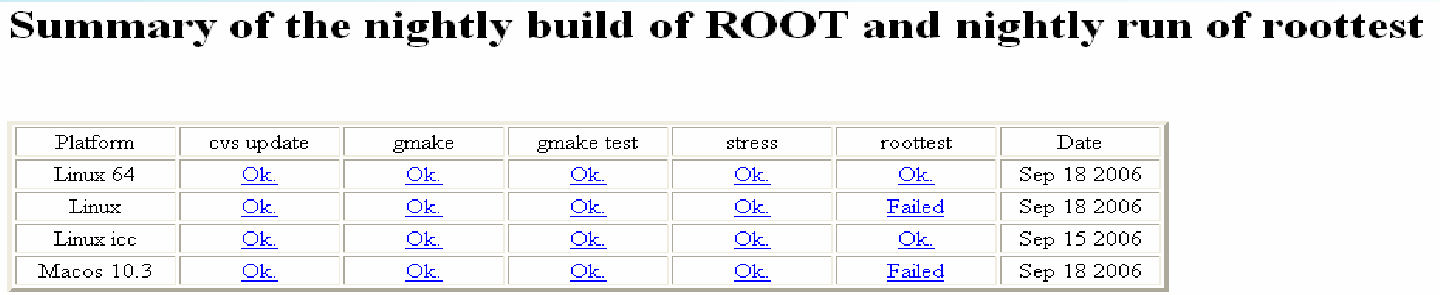
You can edit the script as with a normal editor

Automated Test Suite



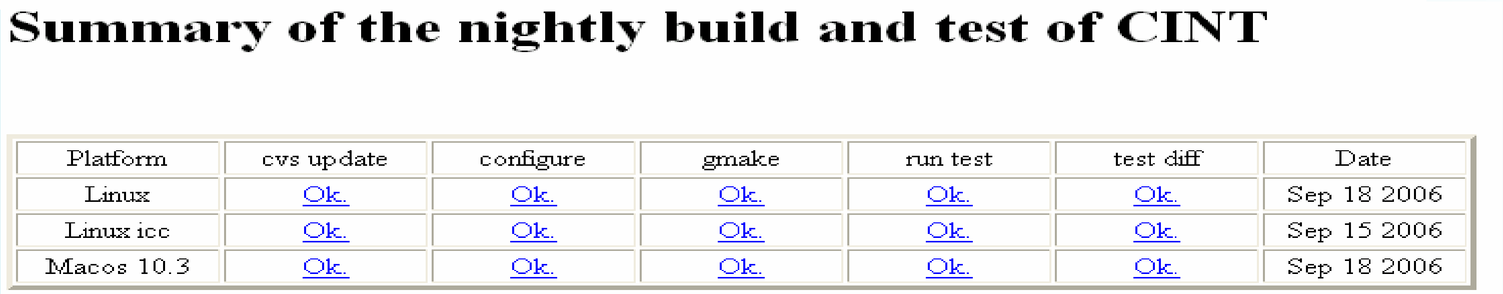
ROOT Coding Convention Violation Table

Rules	RN3	RN4	RN6	RN7	RN9	RN10	RN11	RN12	RN13	RS1	RS2	RS3	RS4	TOTAL
alien	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
asimage	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
auth	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
base	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
chirp	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
cintex	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
clarens	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
cont	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
dcache	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
eg	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
fftw	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**



Summary of the nightly build of ROOT and nightly run of roottest

Platform	cvs update	gmake	gmake test	stress	roottest	Date
Linux 64	Ok.	Ok.	Ok.	Ok.	Ok.	Sep 18 2006
Linux	Ok.	Ok.	Ok.	Ok.	Failed	Sep 18 2006
Linux icc	Ok.	Ok.	Ok.	Ok.	Ok.	Sep 15 2006
Macos 10.3	Ok.	Ok.	Ok.	Ok.	Failed	Sep 18 2006



Summary of the nightly build and test of CINT

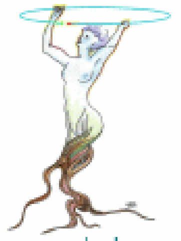
Platform	cvs update	configure	gmake	run test	test diff	Date
Linux	Ok.	Ok.	Ok.	Ok.	Ok.	Sep 18 2006
Linux icc	Ok.	Ok.	Ok.	Ok.	Ok.	Sep 15 2006
Macos 10.3	Ok.	Ok.	Ok.	Ok.	Ok.	Sep 18 2006

I/O Developments in ROOT

Philippe Canal, Paul Russo,

Markus Frank(10%), Leo Franco, Rene

I/O and trees

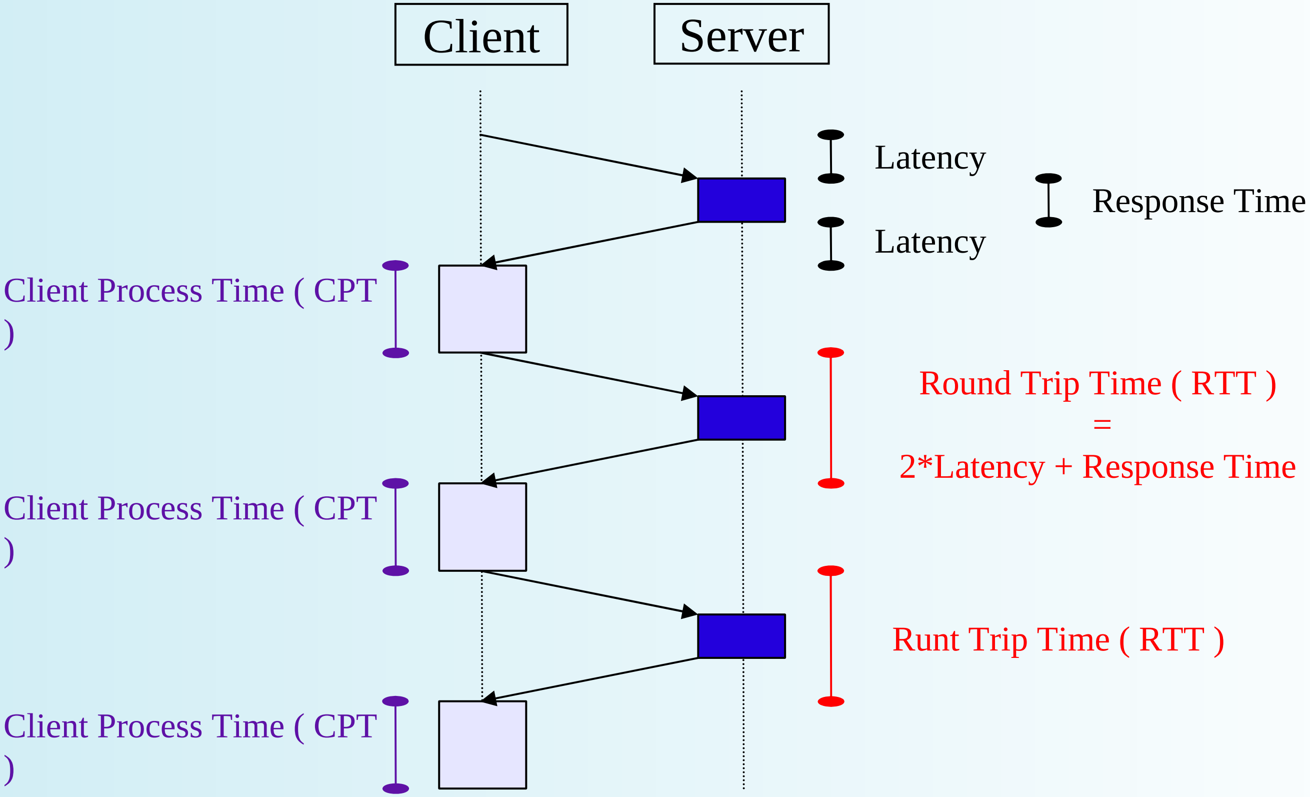


- Added support for more C++ constructs, in particular STL collections and templated classes.
- Generic mechanism for object references.
- Added new cases for automatic schema evolution.
  - More are requested (by Atlas in particular)
- Added support for foreign objects in TDirectory.
- Improved support for multi-threading, but still a lot to do.
- Improved support for SQL data bases (Sergey Linev)
  - TFileSQL introduces a "transparent" access to SQL data bases via standard TFile interface
- New TFileMerger class and support for .zip files
- GFAL interface
- Improved read & write performance for remote files
- Asynchronous open functionality in TFile

I/O and trees (2)

- Implemented fast merging of Trees without unzipping the branch buffers.
- New class TTreeSQL allows the storing and restoring of TTree to and from an SQL database.
- Implemented a TTreeCache with a huge improvement when accessing remote files. (see next slides)
- rootd and xrootd are already able to take advantage of this improvement.
- This improvement opens new possibilities, in particular efficient access to remote files on fast networks (even high latency WAN networks). New ideas for further improvements.

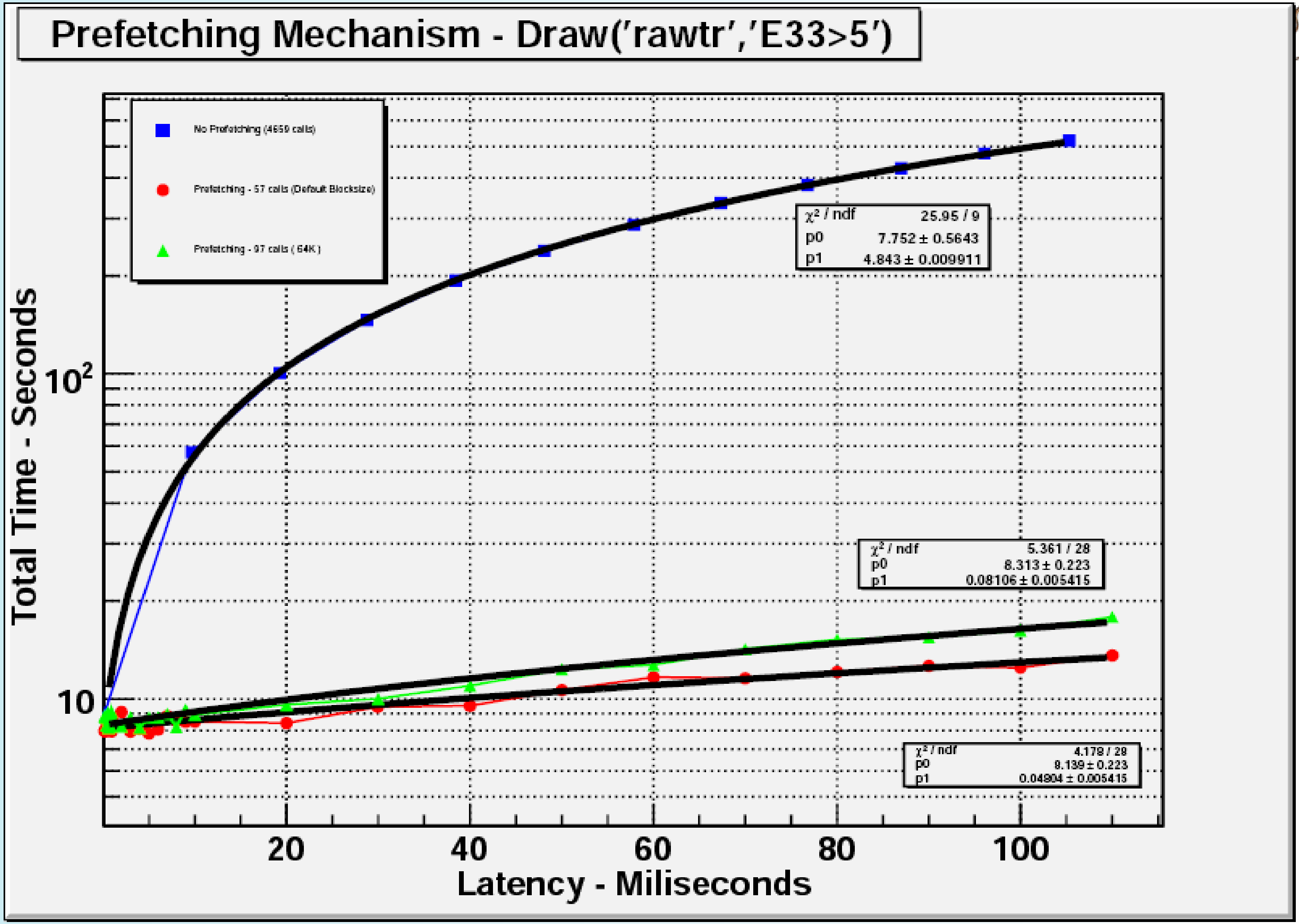
A major problem: network latency


Client
Server
Latency
Response Time
Latency
Client Process Time (CPT)
Round Trip Time (RTT)
=
2*Latency + Response Time
Client Process Time (CPT)
Runt Trip Time (RTT)
Client Process Time (CPT)

Total Time = 3 * [Client Process Time (CPT)] + 3*[Round Trip Time (RTT)]

Total Time = 3* (CPT) + 3 * (Response time) + 3 * (2 * Latency)

Example (h2fast) - Simulated latency (xrootd)



Example of TTreeCache improvement

- The file is on a CERN machine connected to the CERN LAN at at 100MB/s.
- The client **A** is on the same machine as the file (local read)
- The client **B** is on a CERN LAN connected at 100 Mbits/s with a network latency of 0.3 milliseconds (P IV 3 Ghz).
- The client **C** is on a CERN Wireless network connected at 10 Mbits/s with a network latency of 2 milliseconds (Mac Intel Coreduo 2Ghz).
- The client **D** is in Orsay (LAN 100 Mbits/s) connected to CERN via a WAN with a bandwith of 1 Gbits/s and a network latency of 11 milliseconds (P IV 3 Ghz).
- The client **E** is in Amsterdam (LAN 100 Mbits/s) connected to CERN via a WAN with a bandwith of 10 Gbits/s and a network latency of 22 milliseconds (AMD64 280).
- The client **F** is connected via ADSL with a bandwith of 8Mbits/s and a latency of 70 milliseconds (Mac Intel Coreduo 2Ghz).
- The client G is connected via a 10Gbits/s to a CERN machine via Caltech latency 240 ms.
- The times reported in the table are realtime seconds

client	latency(ms)	cachesize=0	cachesize=64KB	cachesize=10MB
A	0.0	3.4	3.4	3.4
B	0.3	22.0	6.0	4.0
C	2.0	11.6	5.6	4.9
D	11.0	124.7	12.3	9.0
E	22.0	230.9	11.7	8.4
F	72.0	743.7	48.3	28.0
G	240.0	>1800	125.4	9.9

One query to a 280 MB Tree I/O = 6.6 MB

User Interface Developments in ROOT

Ilka Antcheva

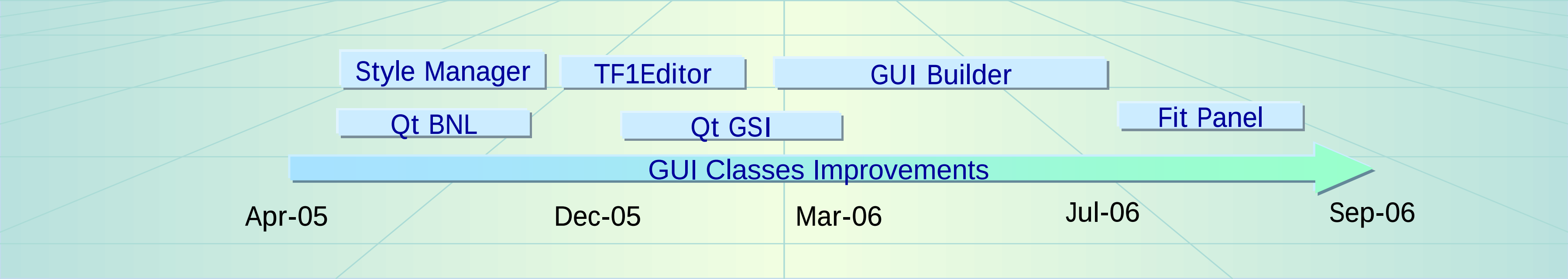
Bertrand Bellenot

Valeriy Onuchin

GUI Work Package

Next Steps (as seen in March 2005):

• GUI Classes	80 % Done
ToDo list http://antcheva.web.cern.ch/antcheva/GuiToDo.html	
• Qt Interface	Qt BNL, Qt GSI
• GUI Builder	New improved design
• Style Manager	Done
• Fit Panel	First prototype coming
• Undo / Redo	To be done



Style Manager

TF1Editor

GUI Builder

Qt BNL

Qt GSI

Fit Panel

GUI Classes Improvements

Apr-05

Dec-05

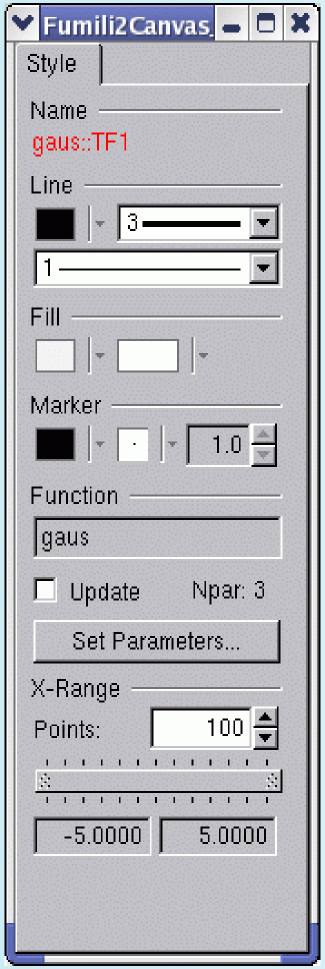
Mar-06

Jul-06

Sep-06

New Widgets

More user-friendly interfaces



Name: gaus::TF1

Line: 3, 1

Fill

Marker: 1.0

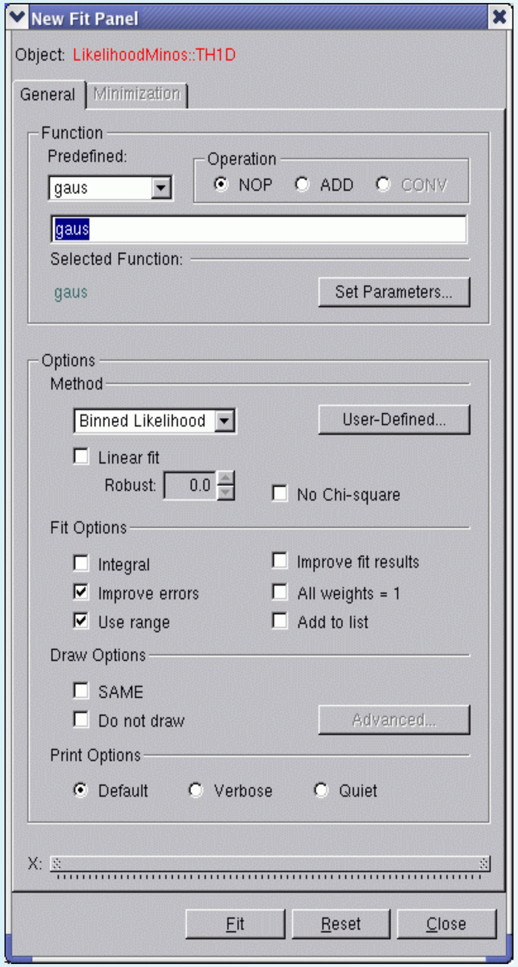
Function: gaus

□ Update Npar: 3

X-Range

Points: 100

-5.0000 5.0000



Object: LikelihoodMinos::TH1D

Function

Predefined: gaus

Operation: ● NOP ○ ADD ○ CONV

gaus

Selected Function: gaus

Options

Method: Binned Likelihood

□ Linear fit

Robust: 0.0 □ No Chi-square

Fit Options

□ Integral □ Improve fit results

☑ Improve errors □ All weights = 1

☑ Use range □ Add to list

Draw Options

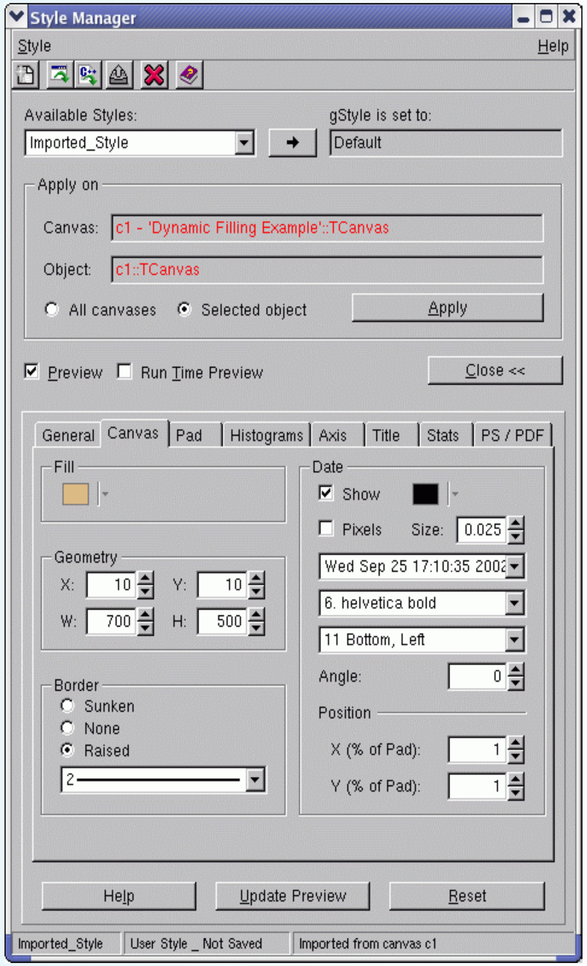
□ SAME

□ Do not draw

Print Options

● Default ○ Verbose ○ Quiet

X:



Available Styles: Imported_Style

gStyle is set to: Default

Apply on

Canvas: c1 - 'Dynamic Filling Example'::TCanvas

Object: c1::TCanvas

○ All canvases ● Selected object

☑ Preview □ Run Time Preview

Fill

Date

☑ Show

□ Pixels Size: 0.025

Wed Sep 25 17:10:35 2002

6. helvetica bold

11 Bottom, Left

Angle: 0

Position

X (% of Pad): 1

Y (% of Pad): 1

Geometry

X: 10 Y: 10

W: 700 H: 500

Border

○ Sunken

○ None

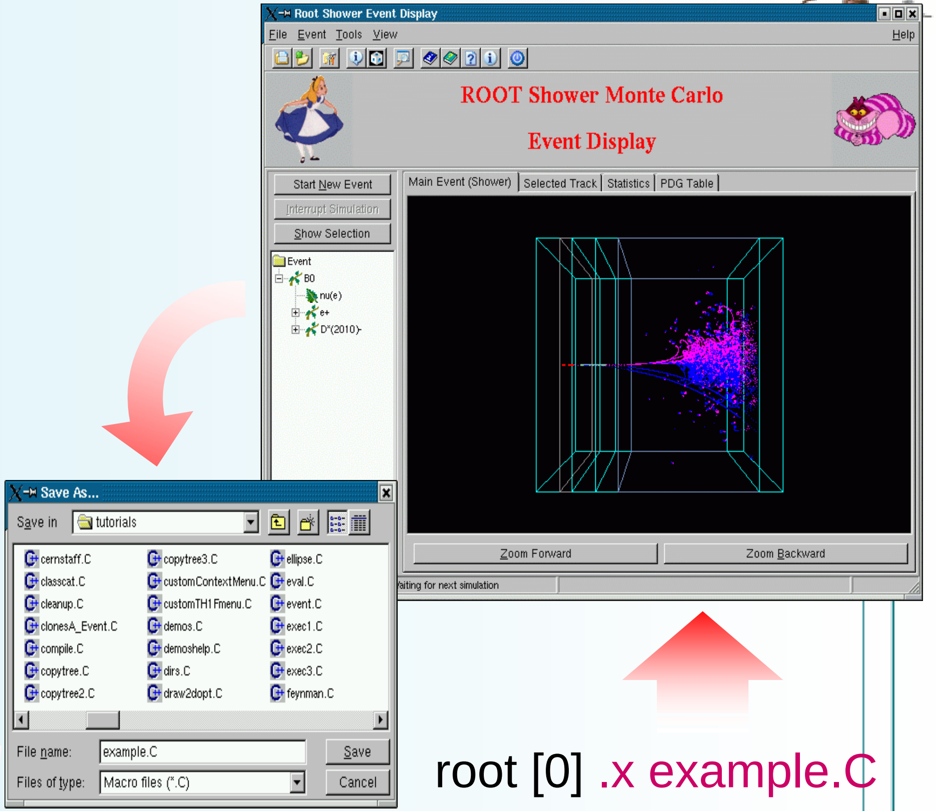
● Raised

2

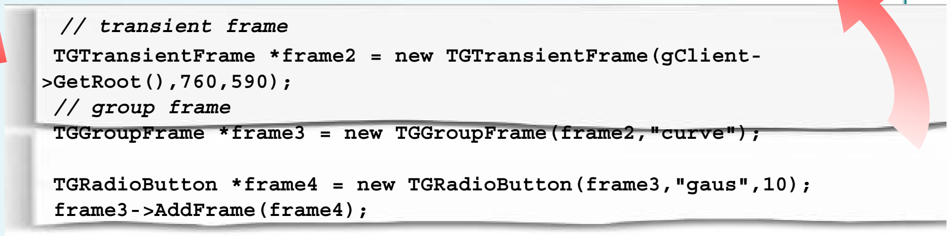
Imported_Style | User Style _ Not Saved | Imported from canvas c1

The GUI Code Generator

- Using *Ctrl+S* any GUI can be ***saved as a C++ macro*** via the SavePrimitive() methods
- This macro can be edited and then can be interpreted or compiled via ACLiC
- Executing the macro restores the complete original GUI
- Signal/slot connections are restored in a global way



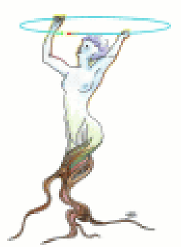
root [0] .x example.C

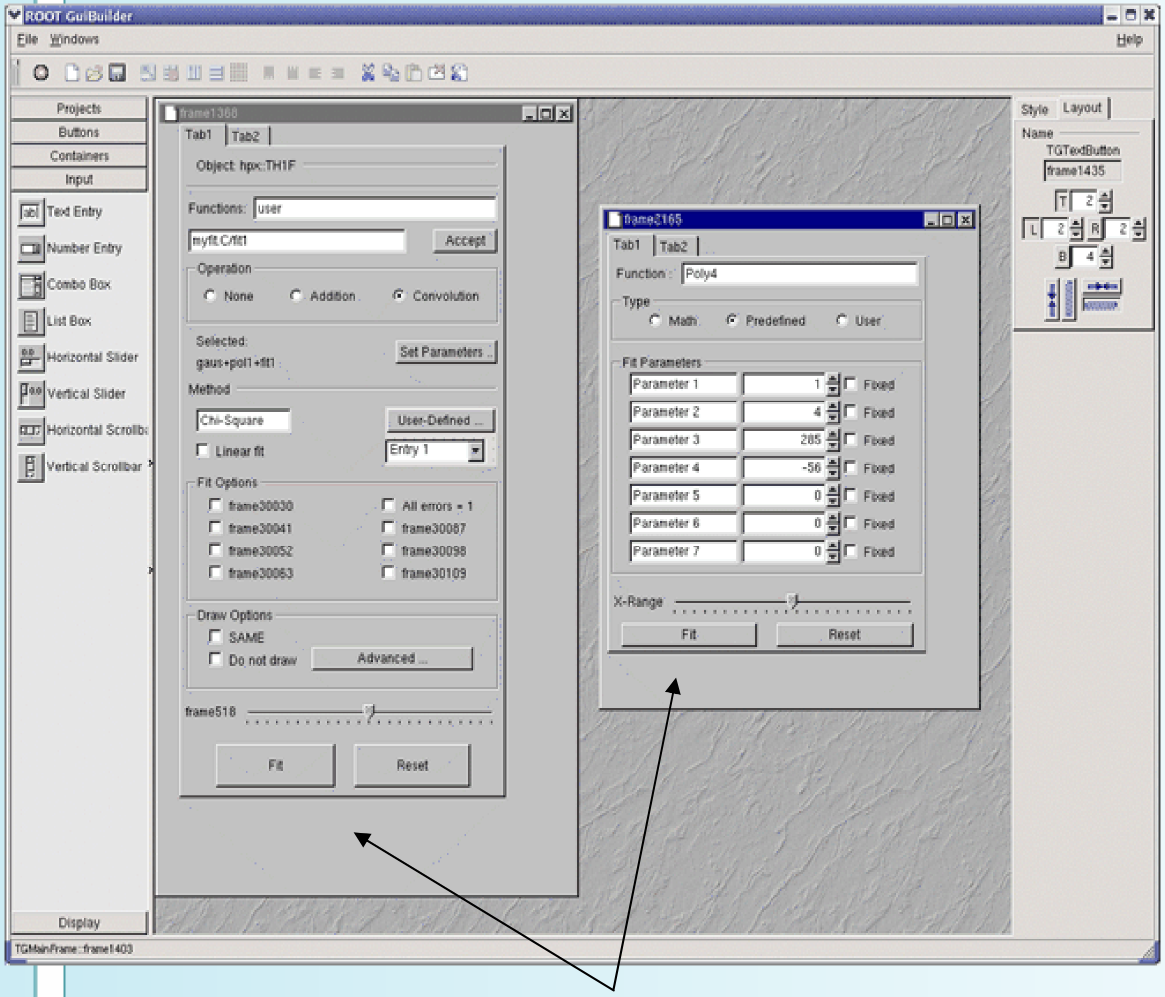


```
 // transient frame
TGTransientFrame *frame2 = new TGTransientFrame(gClient-
>GetRoot(),760,590);
// group frame
TGGroupFrame *frame3 = new TGGroupFrame(frame2,"curve");

TGRadioButton *frame4 = new TGRadioButton(frame3,"gaus",10);
frame3->AddFrame(frame4);
```

The GUI Builder





Prototypes related to the new Fit Panel

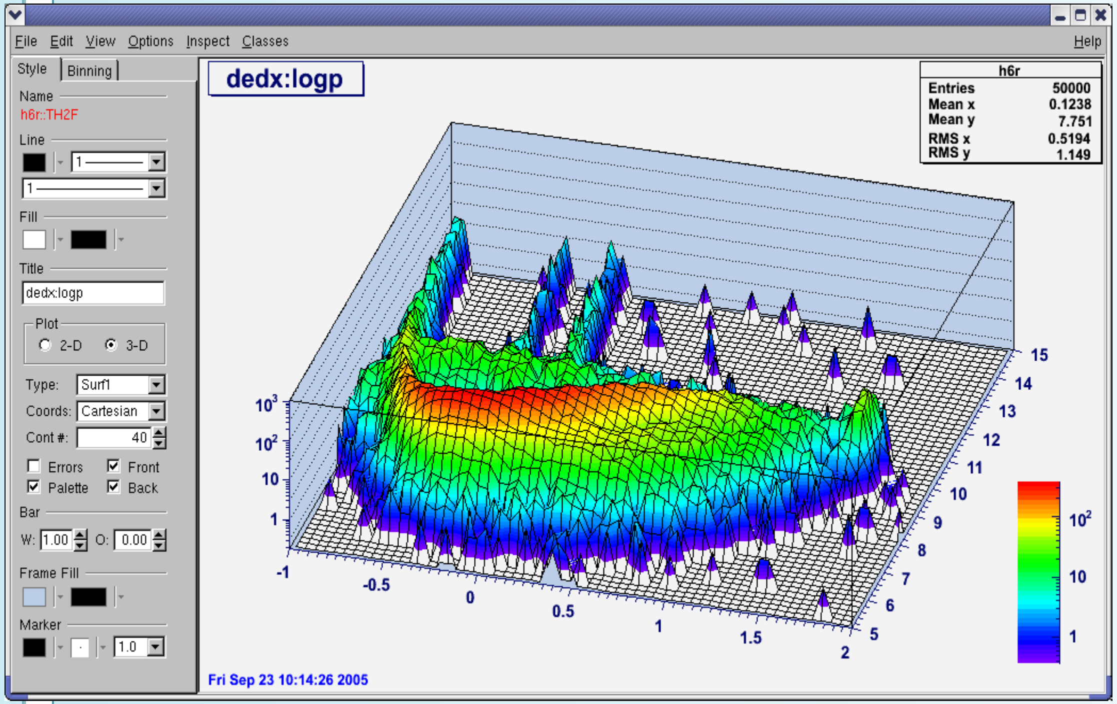
- Save current design in a macro that can be edited and executed via the CINT interpreter:

  root [0] .x example.C

  root [1] .x example.C++

- The design process can start from a macro

GUI: a lot of progress



Name: h6r::TH2F

Title: dedx:logp

Type: Surf1

Coords: Cartesian

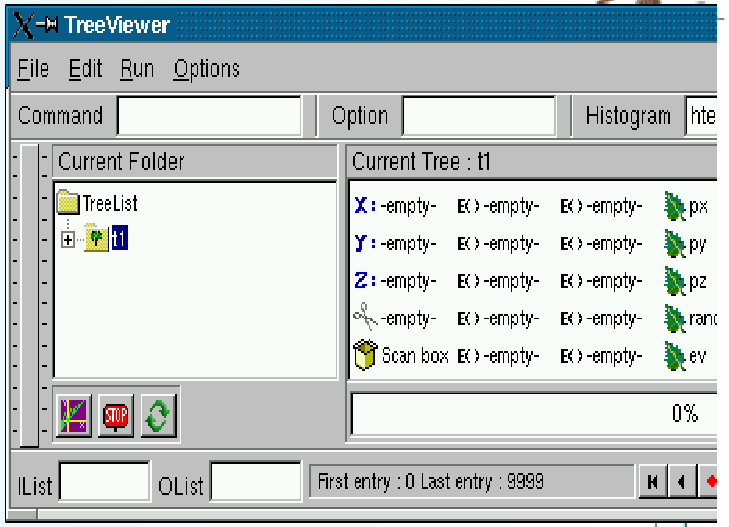
Cont #: 40

W: 1.00 O: 0.00

Marker: 1.0

h6r	
Entries	50000
Mean x	0.1238
Mean y	7.751
RMS x	0.5194
RMS y	1.149

Fri Sep 23 10:14:26 2005

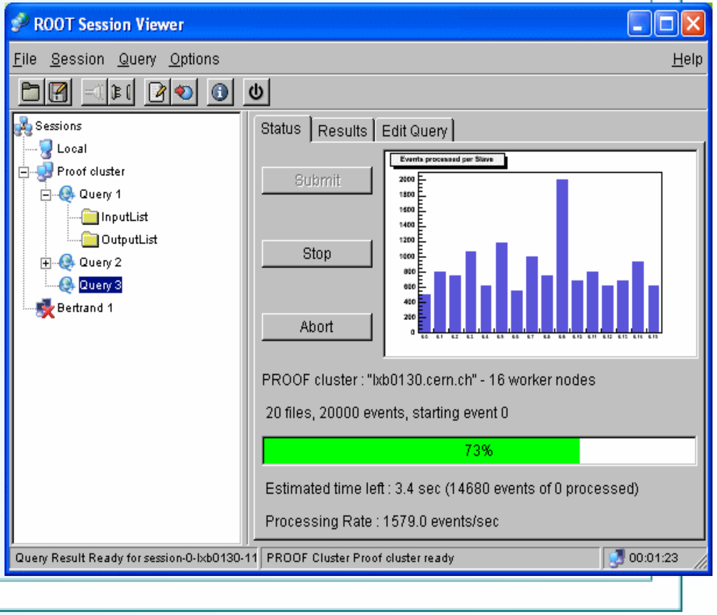

TreeViewer
File Edit Run Options
Command
Option
Histogram
Current Folder
TreeList
t1
X: -empty- E() -empty- E() -empty- px
Y: -empty- E() -empty- E() -empty- py
Z: -empty- E() -empty- E() -empty- pz
-empty- E() -empty- E() -empty- rand
Scan box E() -empty- E() -empty- ev
STOP
IList
OList

Histogram: hte

Current Tree : t1

0%

First entry : 0 Last entry : 9999



PROOF cluster : "lxb0130.cern.ch" - 16 worker nodes

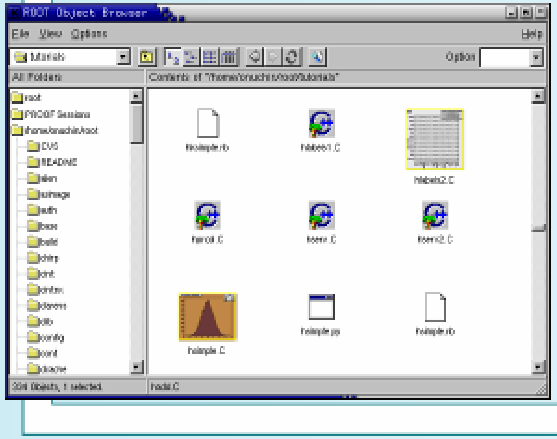
20 files, 20000 events, starting event 0

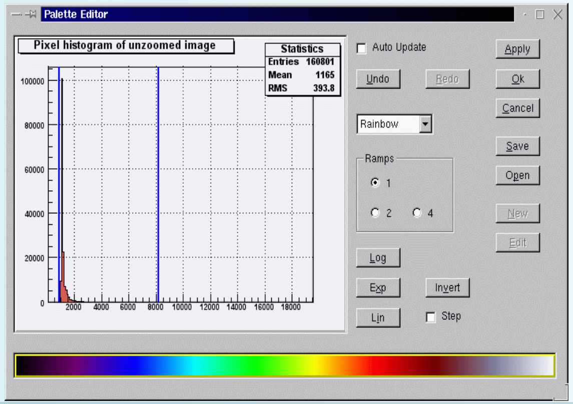
73%

Estimated time left : 3.4 sec (14680 events of 0 processed)

Processing Rate : 1579.0 events/sec

Query Result Ready for session-0-lxb0130-11 PROOF Cluster Proof cluster ready 00:01:23

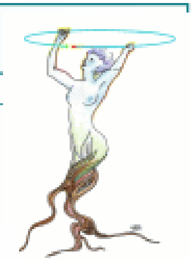




Rainbow

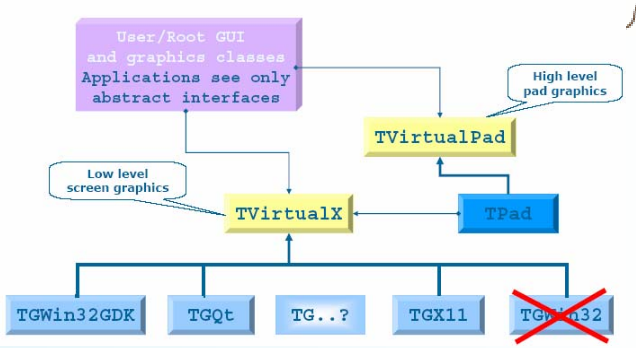
Statistics	
Entries	160801
Mean	1165
RMS	393.8

ROOT and Qt



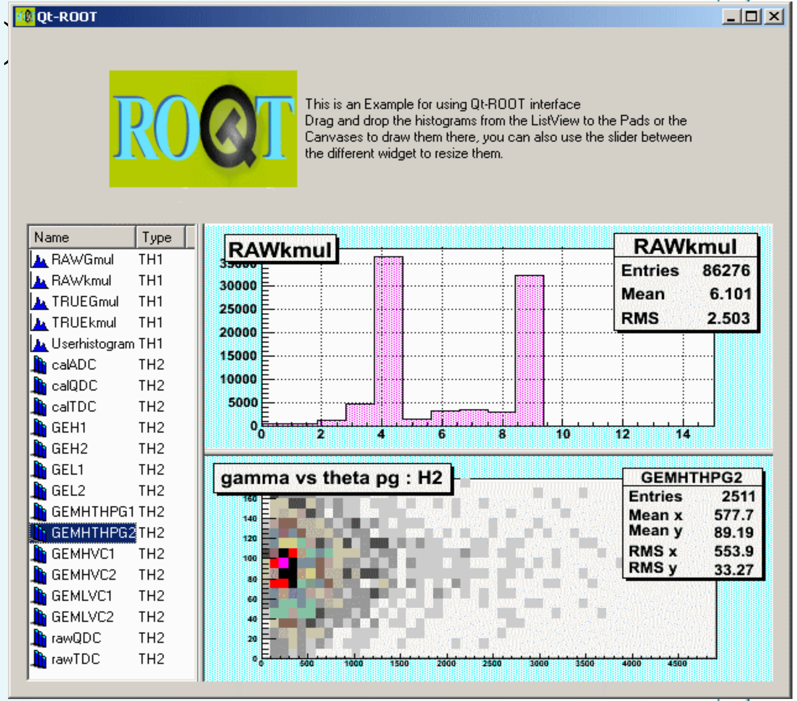
Qt BNL (in ROOT CVS since 2004)

- Uses Qt as a render engine for all ROOT graphics (GUI and Canvas) via TGQt
- ROOT Canvases can be embedded in Qt widgets
- *See talk 158 by Valeri Fine*



Qt GSI (works since many years on Linux, Windows port currently under development)

- Lightweight interface that uses the Qt event loop to drive Qt widgets and the ROOT event loop to handle all ROOT events: GUI, timers, signals, etc.
- Qt widgets are rendered via Qt, ROOT widgets are rendered either via TGX11 or TGWin32GDK
- ROOT Canvases can be embedded in Qt widgets



This is an Example for using Qt-ROOT interface
Drag and drop the histograms from the ListView to the Pads or the Canvases to draw them there, you can also use the slider between the different widget to resize them.

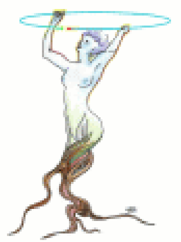
Name	Type
RAWGmul	TH1
RAWkmul	TH1
TRUEGmul	TH1
TRUEkmul	TH1
Userhistogram	TH1
calADC	TH2
calQDC	TH2
calTDC	TH2
GEH1	TH2
GEH2	TH2
GEL1	TH2
GEL2	TH2
GEMHTHPG1	TH2
GEMHTHPG2	TH2
GEMHVC1	TH2
GEMHVC2	TH2
GEMLVC1	TH2
GEMLVC2	TH2
rawQDC	TH2
rawTDC	TH2

RAWkmul	
Entries	86276
Mean	6.101
RMS	2.503

GEMHTHPG2	
Entries	2511
Mean x	577.7
Mean y	89.19
RMS x	553.9
RMS y	33.27

GUI: still a lot to do

- Basic widgets
  - Drag & Drop (browser/canvas)
  - Clipboard between applications
  - Undo/Redo
- High-Level widgets
  - Object Editors
- Integration/Interfaces with other systems
  - QT3 ->QT4
  - Web plug-ins (TkHtml?), Carrot+, Ajax
- Thread Safety
- Recording of GUI events for playback



2-D & 3-DGraphics Developments in ROOT

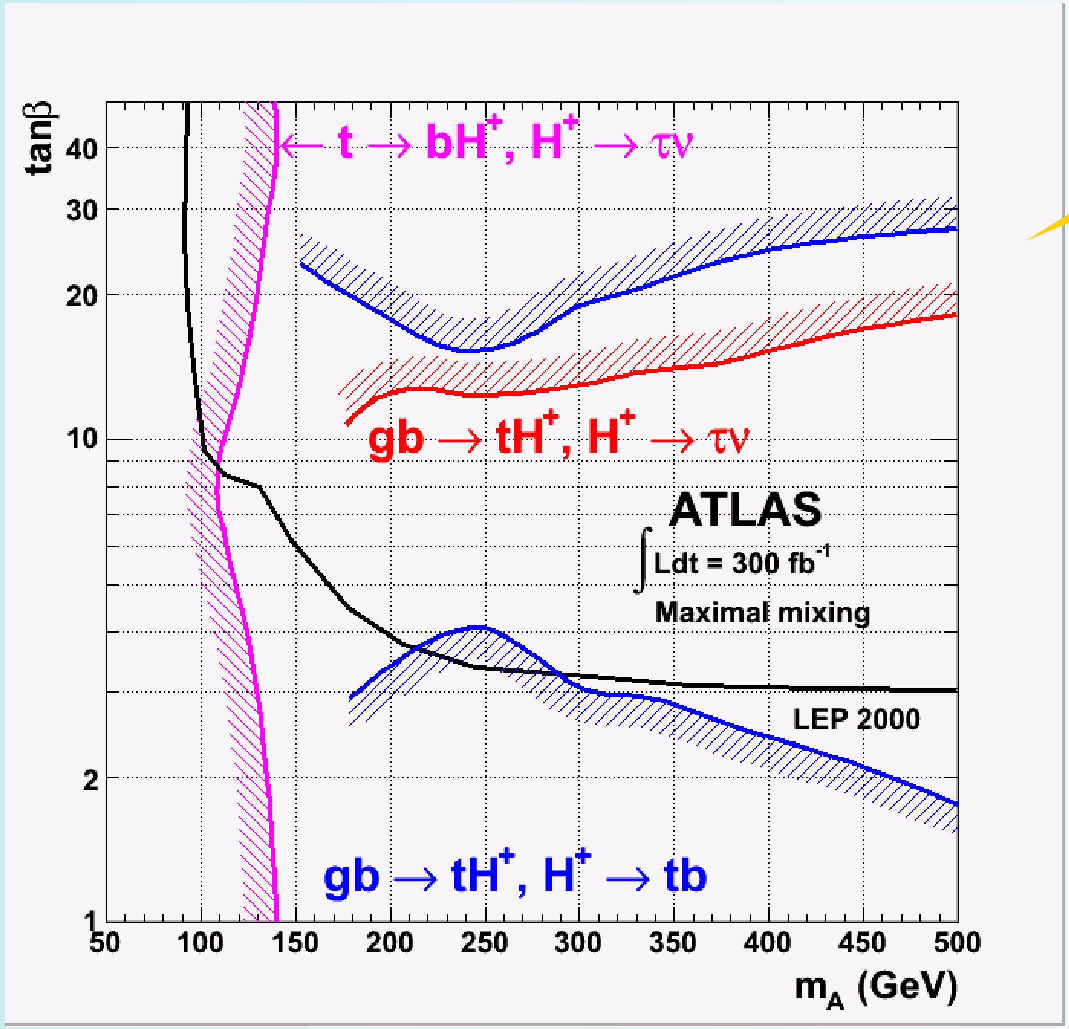
Olivier Couet

Richard Maunder (left in January)

Valery Onuchin (part time from Serpukov)

Timur Pocheptsov (Dubna), Matevz Tadel (Alice)

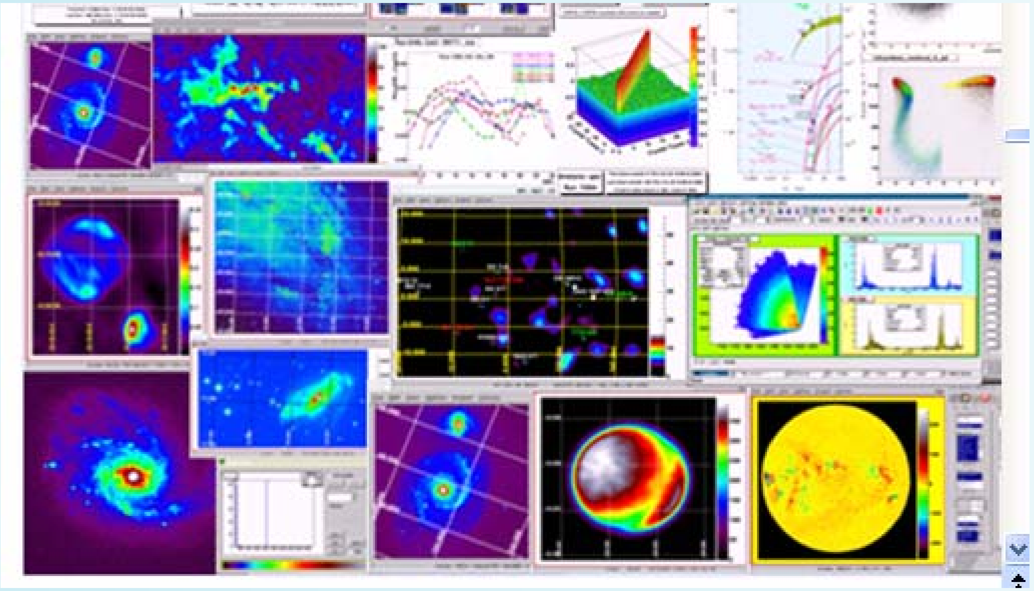
Many new 2-D classes



Exclusion plots

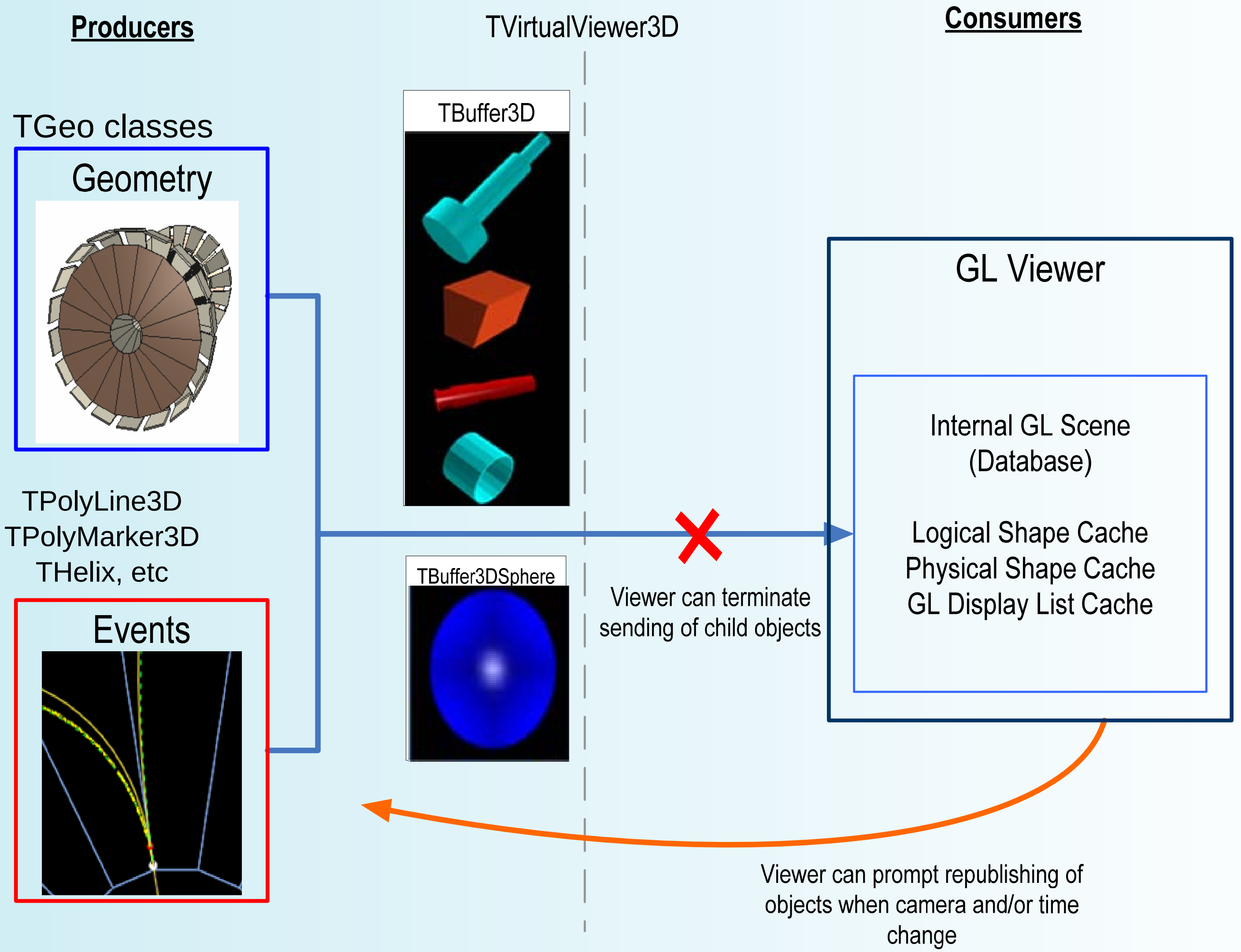
Extensions to image processing classes

- Pictures in most formats (jpg,gif,png,ps,eps,pdf can be imported and exported from/to the pad.
- Filters/transforms can be applied to the pictures



from the new puzzle.C tutorial

ROOT-GL viewer: a General Architecture



Producers

TVirtualViewer3D

Consumers

TGeo classes

Geometry

TBuffer3D

GL Viewer

Internal GL Scene
(Database)

Logical Shape Cache
Physical Shape Cache
GL Display List Cache

TPolyLine3D
TPolyMarker3D
THelix, etc

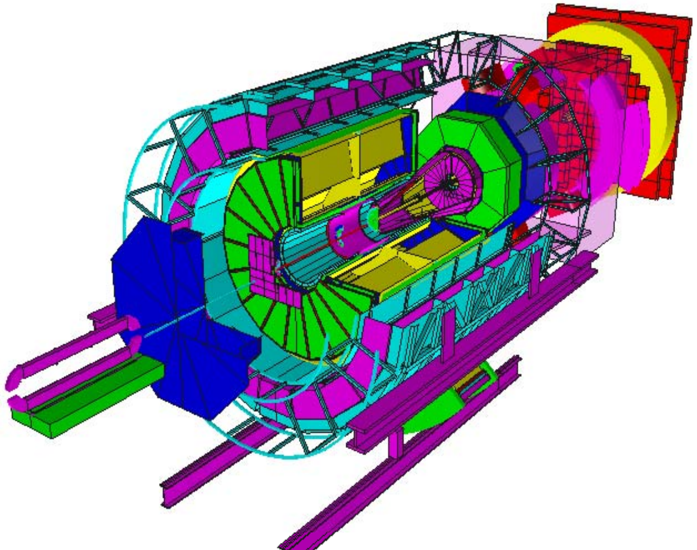
TBuffer3DSphere

Viewer can terminate
sending of child objects

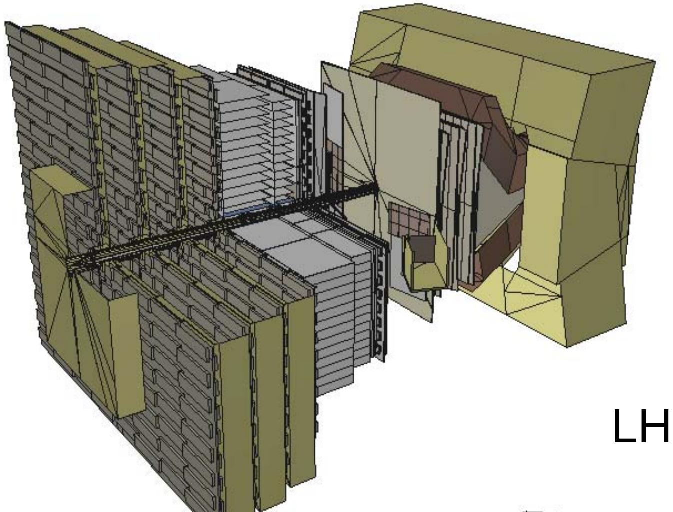
Events

Viewer can prompt republishing of
objects when camera and/or time
change

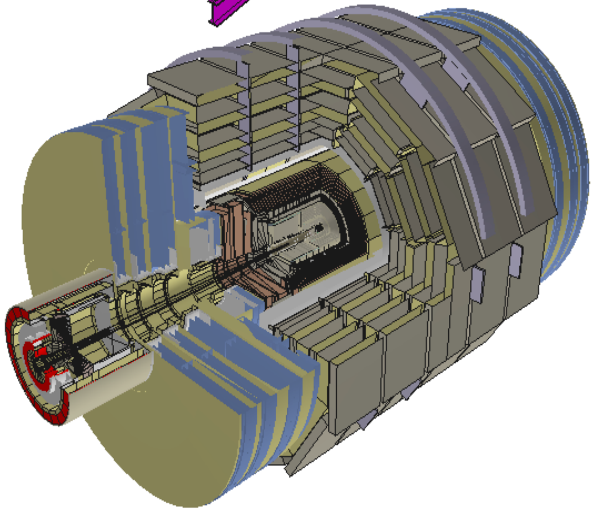
Detector Geometries



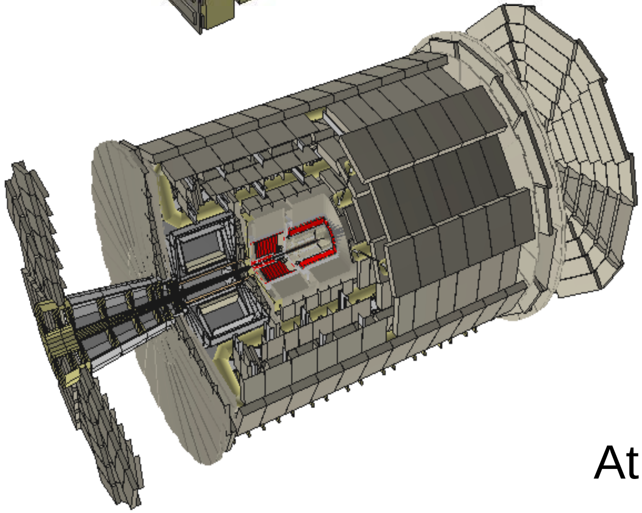
Alice



LHCb



CMS



Atlas

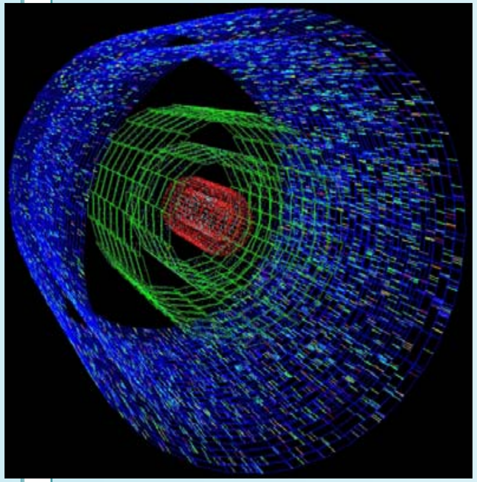
The Geometry package

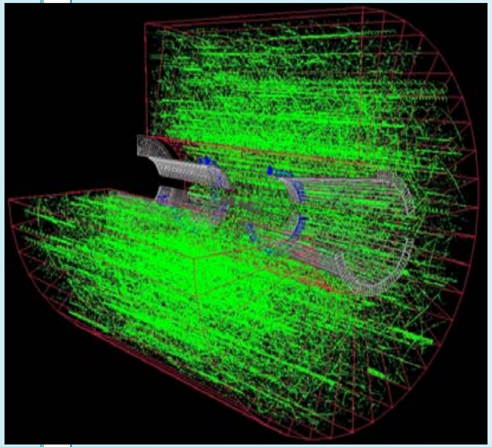
- Developed and supported by Alice (Andrei Gheata)
- Interfaces to Geant3 and Fluka validated
- Interface to Geant4 available in the coming weeks
- Used for simulation, reconstruction, event displays.

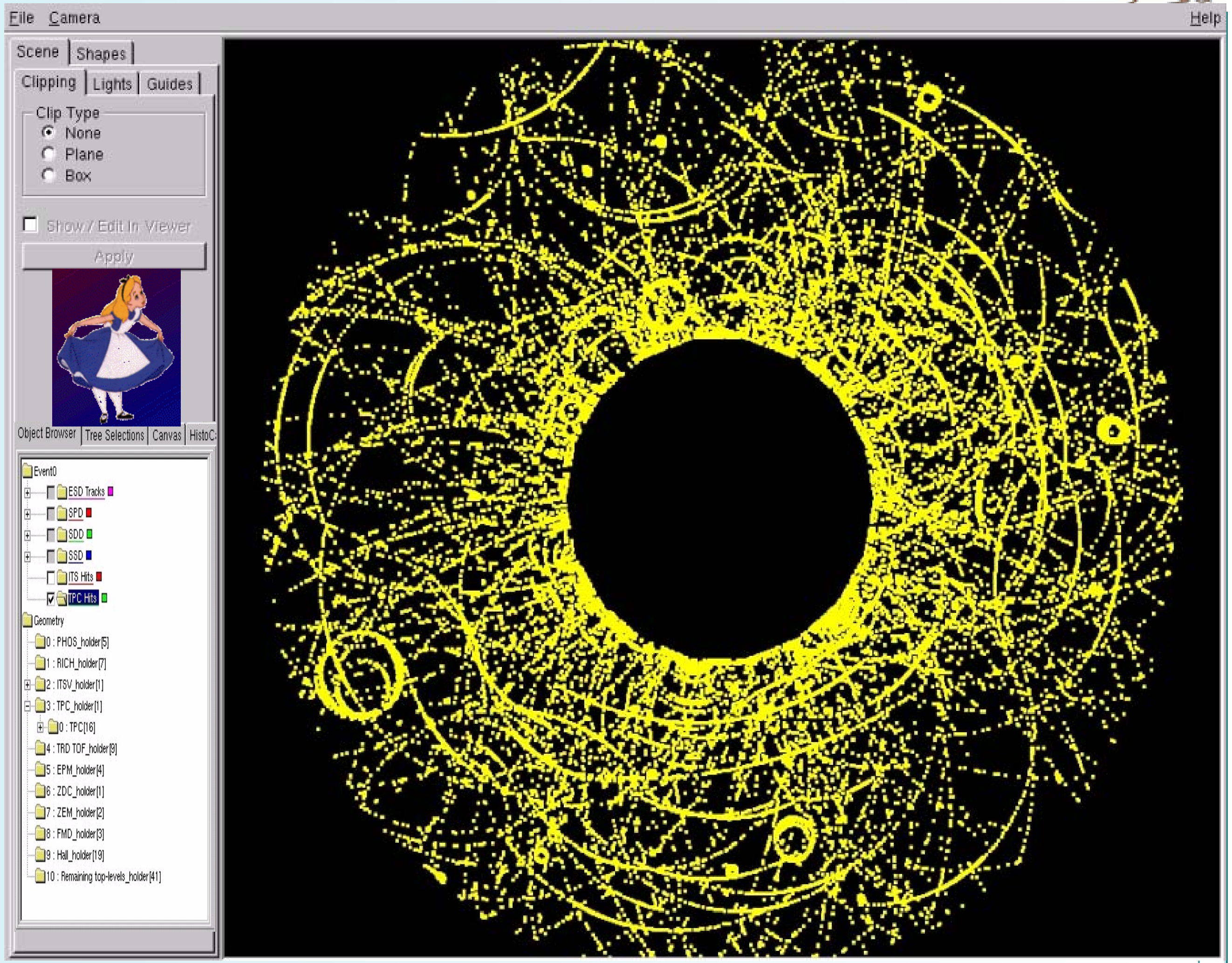
- An alignment facility part of the package.

- A GUI for geometry editor has been developed by Mihaela Gheata (in development release).

Infrastructure for event displays







Clip Type

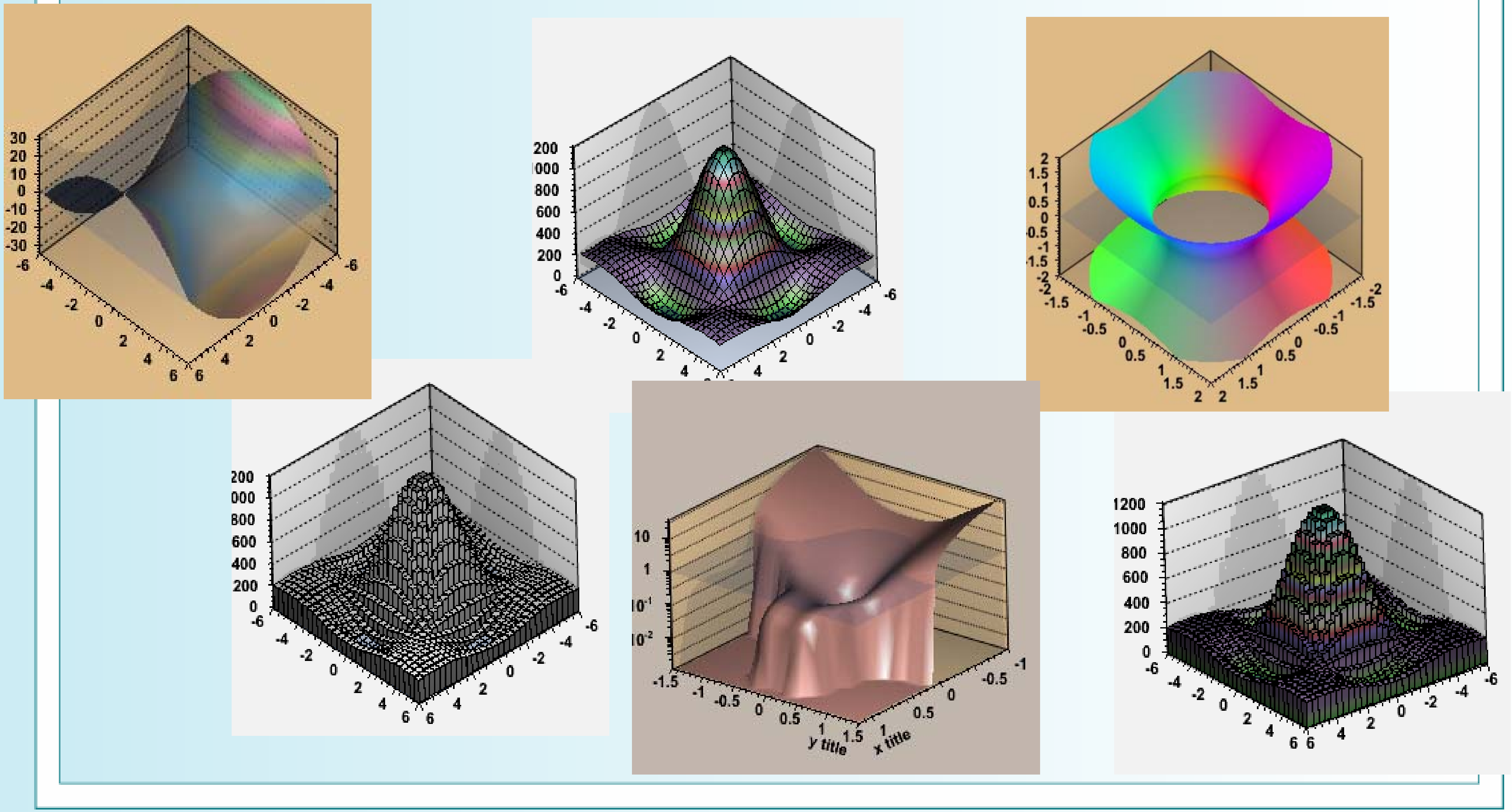
● None
○ Plane
○ Box

□ Show / Edit In Viewer

- Event0
  - □ ESD Tracks 🟪
  - □ SPD 🟥
  - □ SDD 🟩
  - □ SSD 🟦
  - □ ITS Hits 🟥
  - ☑ TPC Hits 🟩
- Geometry
  - 0 : PHOS_holder[5]
  - 1 : RICH_holder[7]
  - 2 : ITSV_holder[1]
  - 3 : TPC_holder[1]
    - 0 : TPC[16]
  - 4 : TRD TOF_holder[9]
  - 5 : EPM_holder[4]
  - 6 : ZDC_holder[1]
  - 7 : ZEM_holder[2]
  - 8 : FMD_holder[3]
  - 9 : Hall_holder[19]
  - 10 : Remaining top-levels_holder[41]

GL in Pad

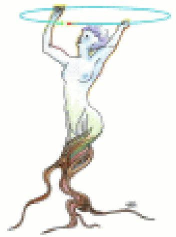
- The ROOT 2-d, 3-d histograms and functions can be viewed via GL in the pad, offering real time zoom , projections, slices and rotations.



Graphics summary

- Continuous requests for zillions of tiny improvements (histograms, graphs, functions)
- Growing importance of GL (driven by hardware)
- Many more developments required for efficient Event Displays
- We must investigate the post X11 era: XGL, Glucose

ROOT: Next Events



- Version 5.14 in December with a first version of CINT+Reflex
- PROOF validation process continuation
- More work with remote file caching
- ROOT using Reflex
- ROOT with a web plug-in ?
- Infrastructure for Event Displays
- Working for the post X11 era
- Progress with BOOT. Requires progress with:
  - remote files caching
  - Reflex persistency
  - direct access to ABI (no C++ stubs)
  - Understanding memory structures used by compilers

- ROOT workshop in March 2007 (CERN?)

ROOT: general summary

- Many new features added in the past 18 months.
- Big effort on robustness and code quality, but we still have to make a few pro patch releases.
- No major problems expected in the coming months.
- ROOT/CINT/Reflex integration: still many unknowns.
- Huge progress with PROOF.
- Main risk is 2008 when the LCG contracts will terminate.