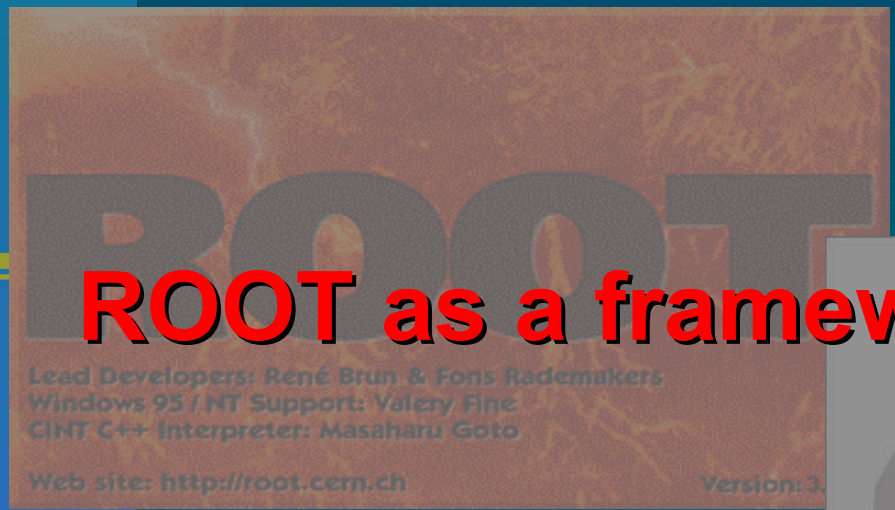
Data Analysis – Team #3



ROOT as a framework for AGATA

What is behind « T3 » ?

Data Storage
Online / offline monitoring and analysis

Everything needed
once
the beam is on the target

Outline

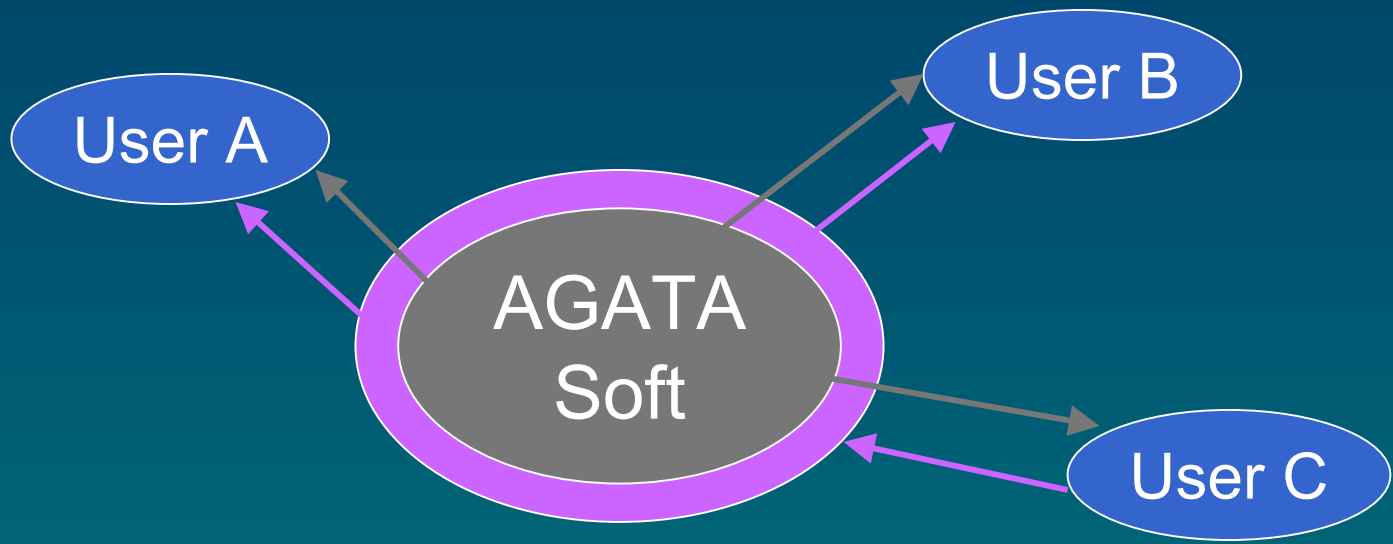
- What is currently done
- What should be done
- Why ROOT can be used
- Conclusions

What is currently done ?

- *Data storage*
  DLT, disk, BLUE, StasbourgDB, Radware
- *Analysis*
  MIDAS, Radware, paw, others …
- *Language*
  Fortran, C, C++, MIDAS, Java, others …

Is it the best way to work for AGATA ?

« Framework approach »



Requirements:

Portable / evolutive / well designed

Starting from scratch or from an existing framework ?

Why ROOT ?

From PAW to ROOT …

> "It became time to re-think our approach to large scale data analysis and simulation and at the same time we had to profit from the progress made in computer science over the past 15 to 20 years. Especially in the area of **Object-Oriented** design and development. Thus was born ROOT."

http://root.cern.ch

Overview

- Language C++, Open source
- First version in 1997
  updated version every six months
- Large community involved (4 workshops)
- Portable on many platforms
  unix, linux, windows, OS X
  CC, gcc, icc

Libraries of objects for physicists

- More than 500 classes already defined

  ex: histograms, functions, graphs, minimization classes…
- Many methods (TH1 ~210)

HTML documentation

TH1

class description - source file - inheritance tree

class TH1 : public TNamed, public TAttLine, public TAttFill, public TAttMarker

public:

```
                TH1 TH1()
                TH1 TH1(const char* name, const char* title, Int_t nbinsx, Axis_t xlow, Axis_t xup)
      virtual void ~TH1()
                    .......
      virtual void Add(TF1* h1, Double_t c1 = 1)
      virtual void Add(const TH1* h1, Double_t c1 = 1)
      virtual void Add(const TH1* h1, const TH1* h2, Double_t c1 = 1, Double_t c2 = 1)
      virtual void AddBinContent(Int_t bin)
      virtual void AddBinContent(Int_t bin, Stat_t w)
                    .......
      virtual Int_t Fit(const char* formula, Option_t* option, Option_t* goption, Axis_t xmin = 0, Axis_t xmax = 0)
      virtual Int_t Fit(TF1* f1, Option_t* option, Option_t* goption, Axis_t xmin = 0, Axis_t xmax = 0)
                    .......
      virtual Stat_t Integral(Int_t binx1, Int_t binx2, Option_t* option) const
      virtual Stat_t Integral(Int_t, Int_t, Int_t, Int_t, Option_t*) const
      virtual Stat_t Integral(Int_t, Int_t, Int_t, Int_t, Int_t, Int_t, Option_t*) const
      virtual Double_t KolmogorovTest(TH1* h2, Option_t* option)
                    .......
```

See also

TH1C, TH1D, TH1F, TH1K, TH1S, TH2, TH3

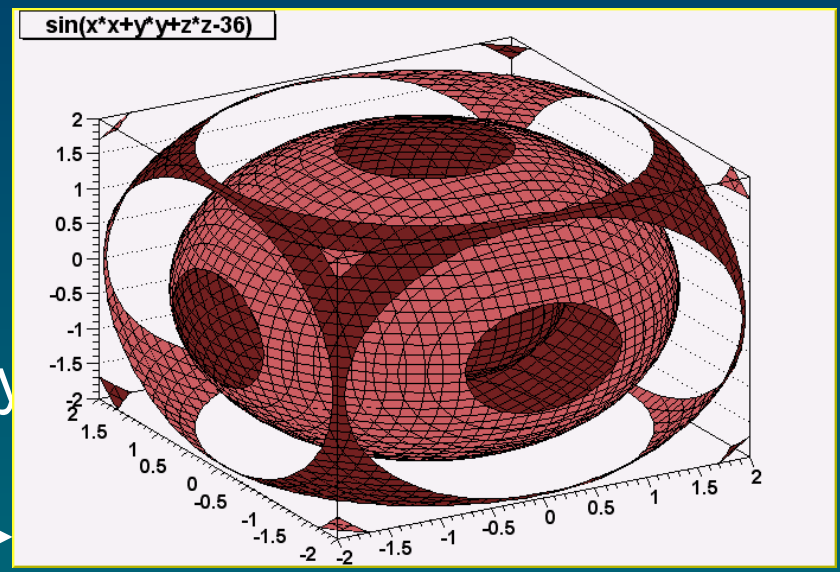
Class Description

```
                The H I S T O G R A M   Classes
                ===============================

   ROOT supports the following histogram types:

    1-D histograms:
       TH1C : histograms with one byte per channel.   Maximum bin content = 255
       TH1S : histograms with one short per channel.  Maximum bin content = 65535
       TH1F : histograms with one float per channel.  Maximum precision 7 digits
       TH1D : histograms with one double per channel. Maximum precision 14 digits
```

ROOT interpreter

- C/C++ interpreter
- Ex :

  root > TF3 f('name','sin(x*x+y
  root > f.Eval(1,0,1)
  root > f.Draw()

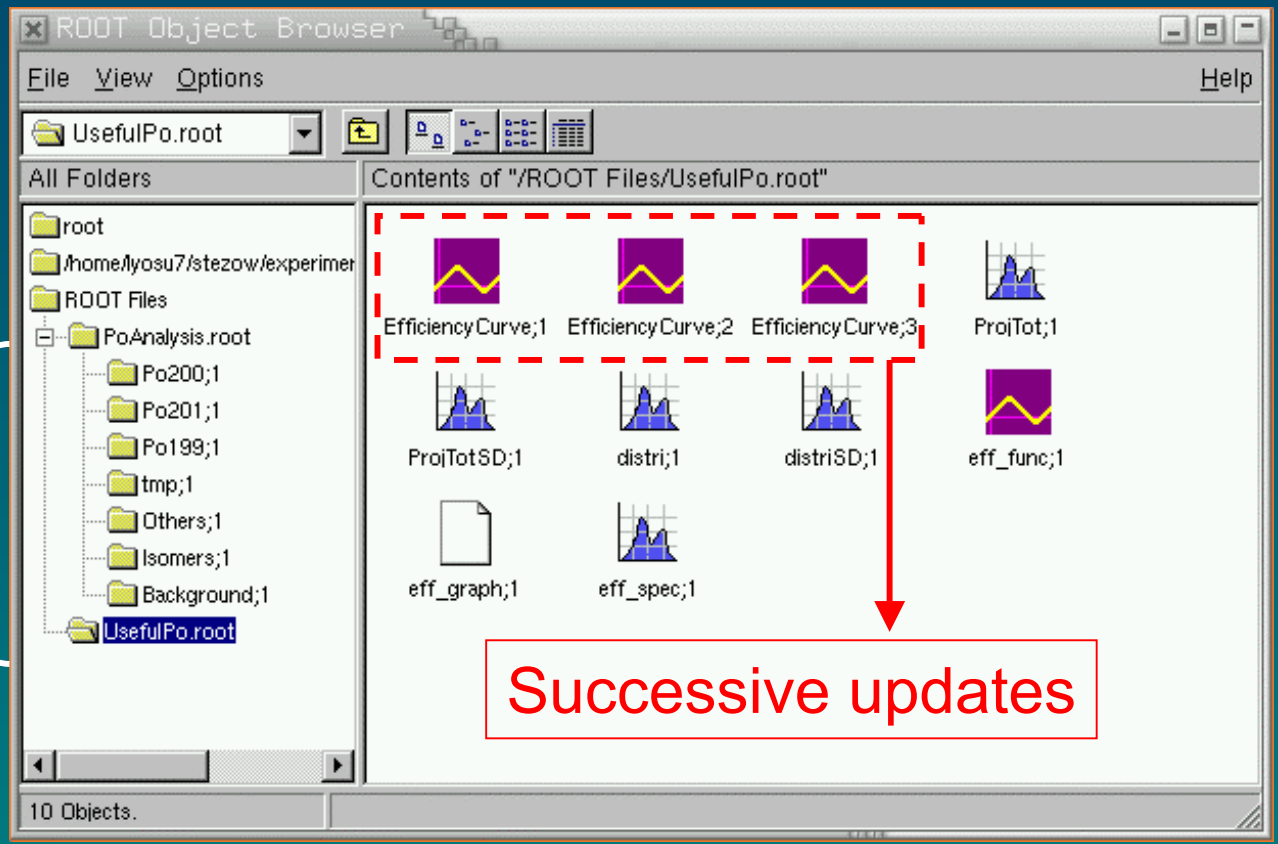


- Complex macros & functions
- There is a way to compile macros
- There is a way to extend the root dictionary

Object persistence: ROOT files

Take into account the object evolution – machine independent

Directory like structure

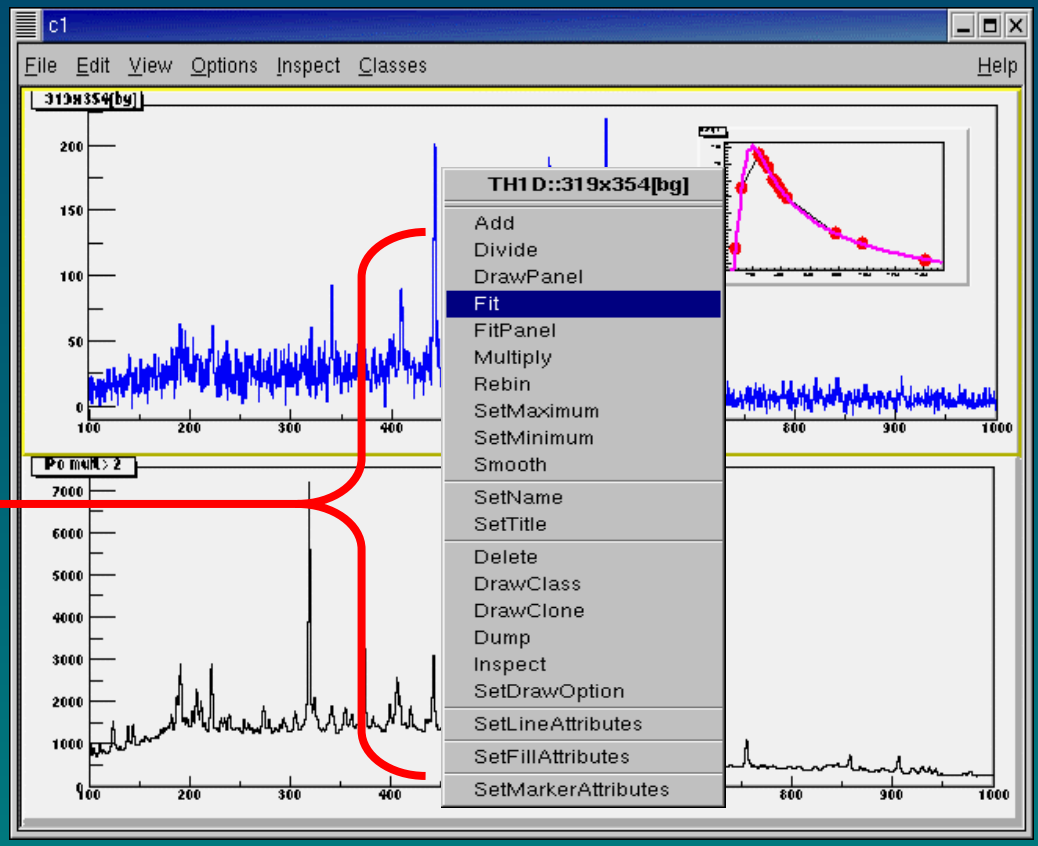


Many graphical features

Windows (canvas/pad) to display any ROOT object

- Easy to divide a window
- Windows saved in:

  ROOT file, postscript, svg, gif



Rich GUI

- **Move, expand, delete**
- **Access to some methods**
- **Way to add new methods**

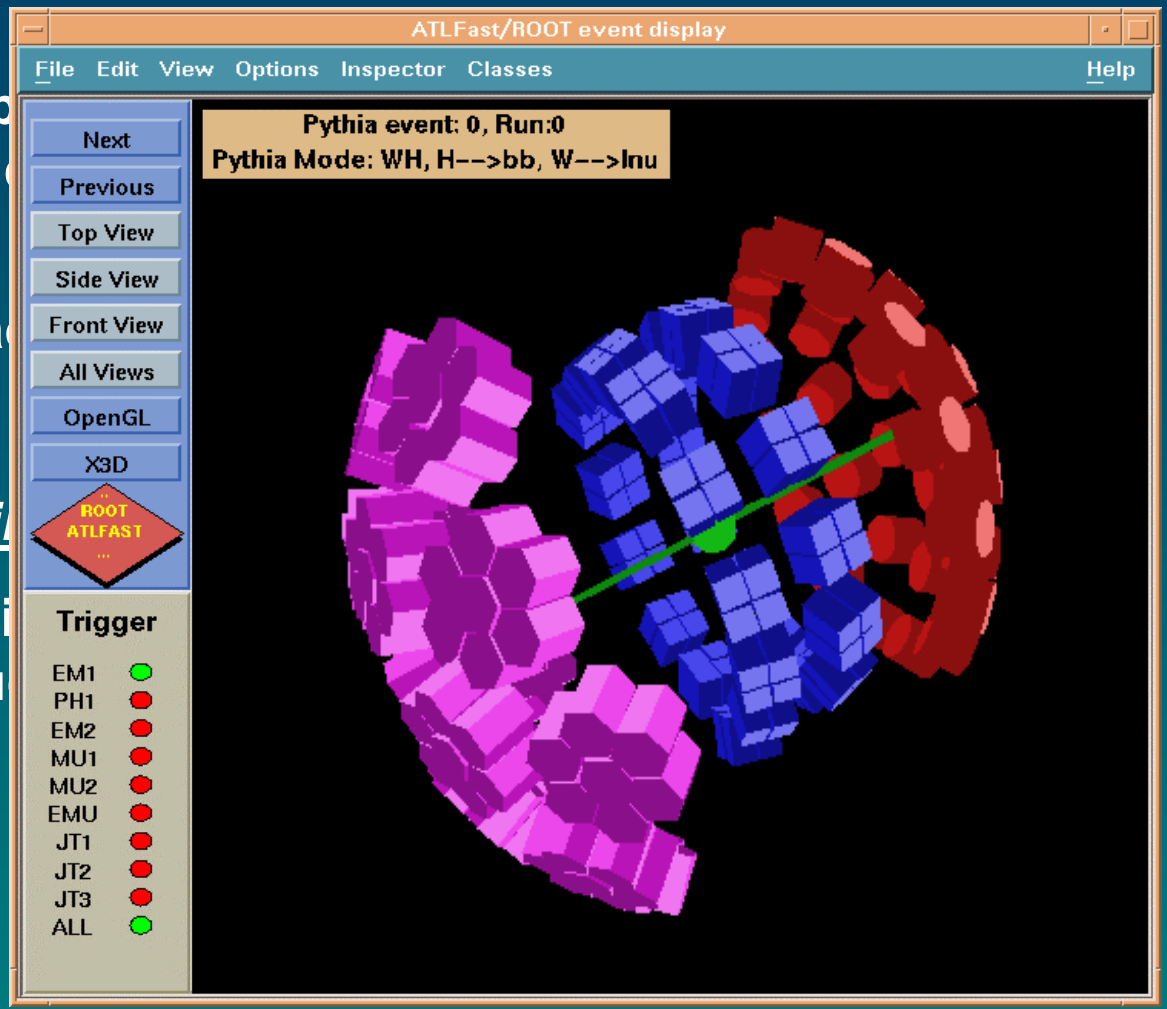
Event display

The ROOT geometry p
browsing, tracking an

The goal is to be able
purposes, such as tra

Concerning the tracki

ROOT does not track i
is done by what is plu



Others

- Interface to the operating system
- Generation of random numbers (MC)

  ex: random number from an histogram
- Many ways to fit
- Base classes to built new GUI
- Thread support
- Map files
- …..

TTree: special object to store events

- Each branch can be splitted
- Methods to apply cuts
- Treeviewer to display distributions
- Mechanism to chain trees

Proof and GRID

request

Master proof

out/obj

Proof
Parallel ROOT Facility

proof 1

proof 2

.....

proof n

ROOT is part of the LCG (LHC Computing Grid) projects

Data management - What is needed ?

How many parallel processes are needed to go through all the data set

40 TB - Data access time 100 MB/sec

1 seconde	1 minute	1 hour	1 day
$4\ 10^5$	$7\ 10^3$	116	5

GRID !!

What is done in our field

Sofware solutions developed to speed up the gating process

BLUE, StrasbourgDB, Radware

Radware next generation developed for GRETA

- Data ~50 GB + Table ~50 GB
- 10 stations running at the same time
- GRETA liked simulated events (simple ones)
- fold gating ~ 6-7

gated spectrum in seconds !!
50 GB [50 MB/s], reduction by a factor 10^3

Data management - What is needed ?

How many parallel processes are needed to go through all the interesting events

40 TB - Data access time 100 MB/sec

1 seconde	1 minute	1 hour	1 day
$4\ 10^5$	$7\ 10^3$	116	5
400	7	1	1

GRID ??

Questions to be answered

- Data storage?
  Disk (cheap, fast), tapes
- What structure for the parallel processing?
  - « Cluster of Universities » for each experiment
  - AGATA center
  - GRID

Avoid too much data transfert between computers !!

- It costs « probably » too much time
- It becomes difficult to handle

Conclusions

- ROOT: solid framework for AGATA
- Specific developments for Nuclear Physics
  Our objects: NuclearLevel, Gamma …
  Our methods: Gating, DCO, …
- Any suggestions are welcome !
  Meeting tomorrow & AGATA-ANALYSIS

Database (Oracle, MySQL) of level scheme