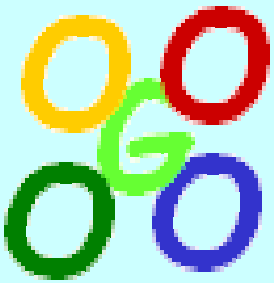


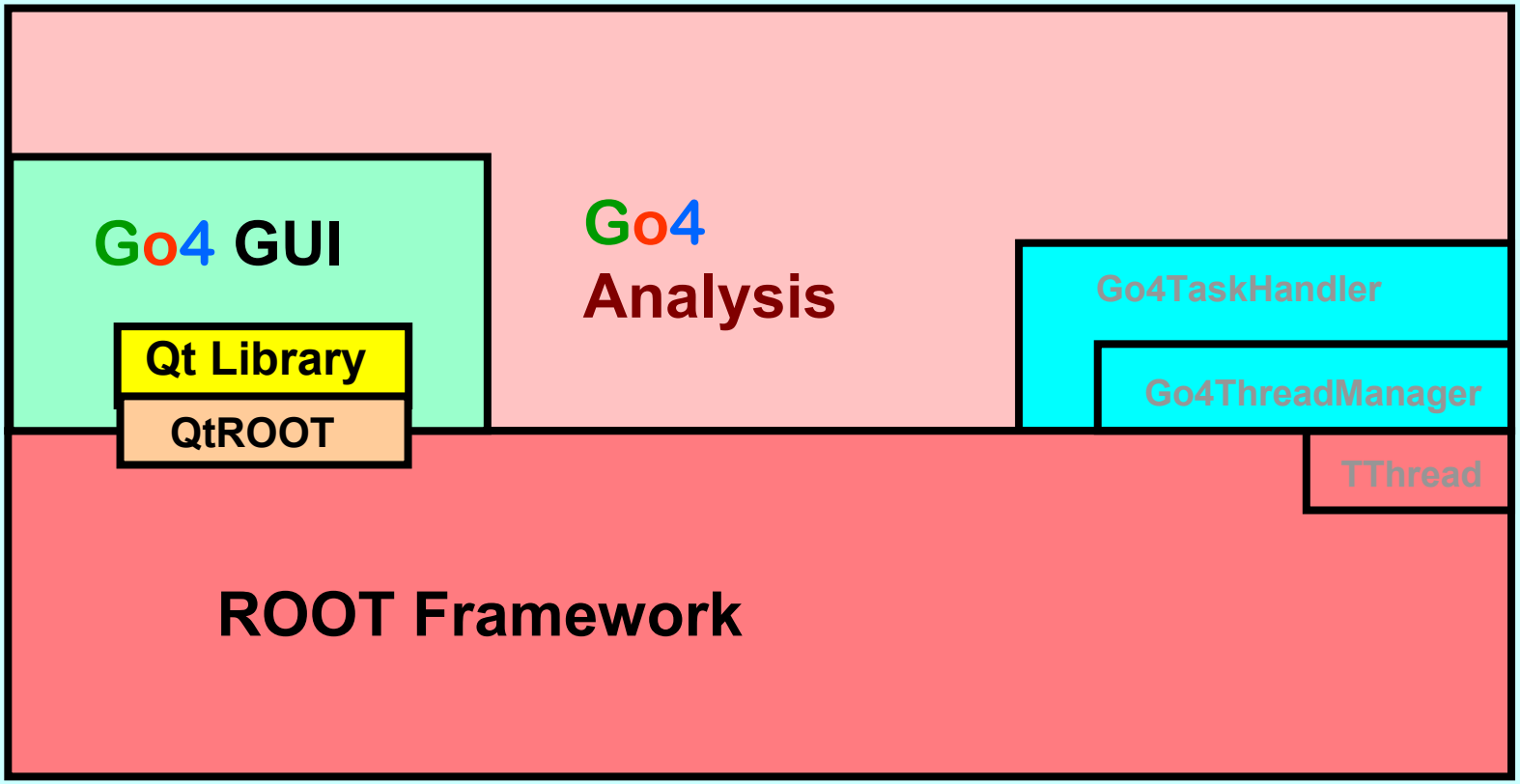
GSI
Online
Offline
Object
Oriented

Go4 v2.2 Status & Overview

J. Adamczewski, M. Al-Turany, D. Bertini, H.G.Essel, S.Linev

CHEP 2003

Package Layers Go4 v2.2



Features of Go4

- Framework for many experiments (AP & NP)
- The analysis is written by the user (unlimited ROOT)
- A GUI controls and steers the analysis
- The GUI is never blocked by analysis
- The analysis may run permanently (e.g. on-line), is never blocked by GUI
- An analysis may update graphics asynchronously
- The GUI provides efficient interactivity
- The analysis may run without change in batch mode

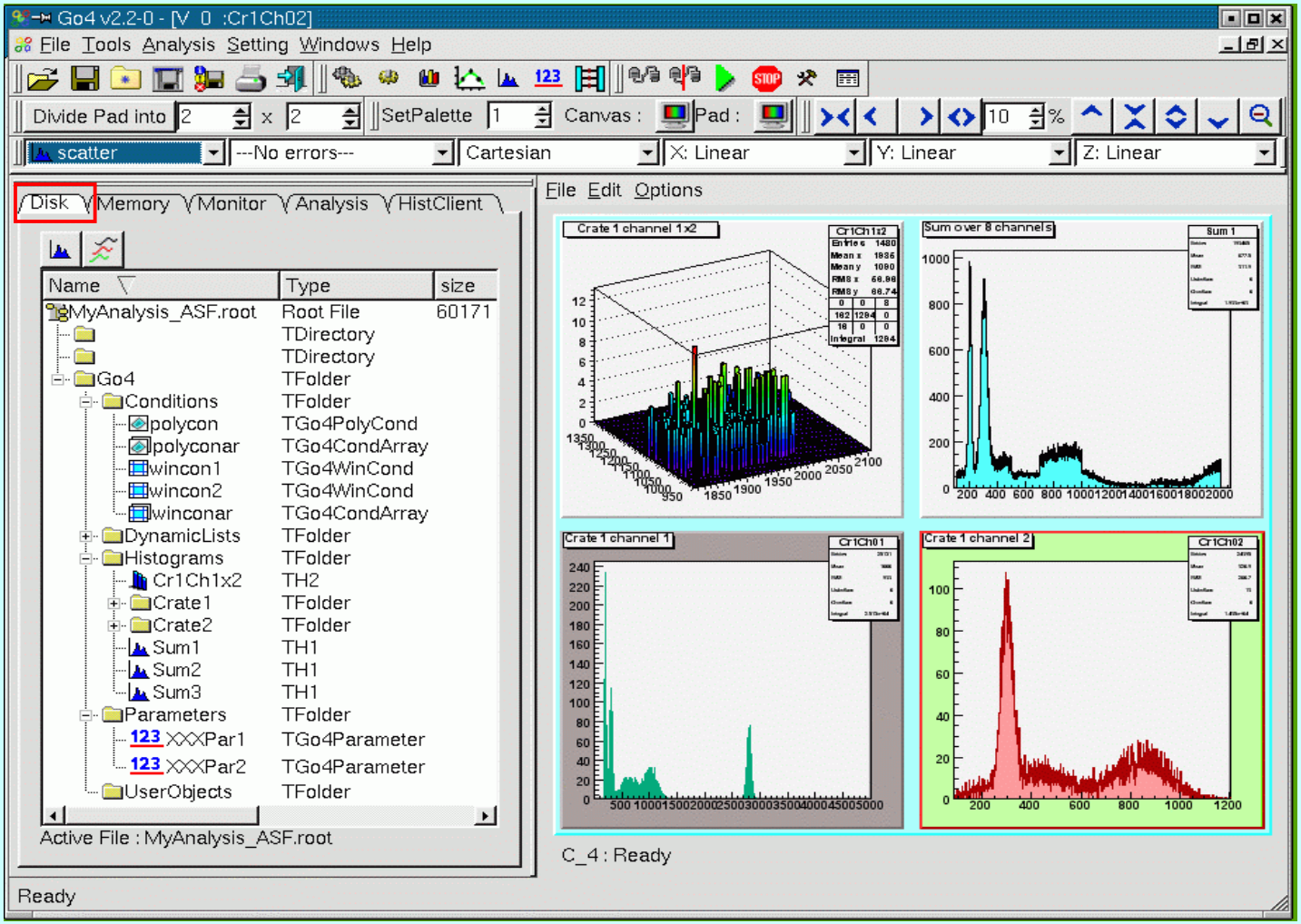
How One Can Use Go4

- Use Go4 GUI as ROOT-file browser / viewer
  - Histograms
  - Trees
  - Fitter
- Connect your ROOT analysis to GUI
  - little efford to adapt
  - dynamic tree draw
  - Example: HADES online analysis
- Develop analysis with Go4 framework
  - Event classes (composite)
  - Event IO (any input, trees, branches)
  - Analysis steps
  - Parameters, conditions + editors
  - Dynamic histogramming

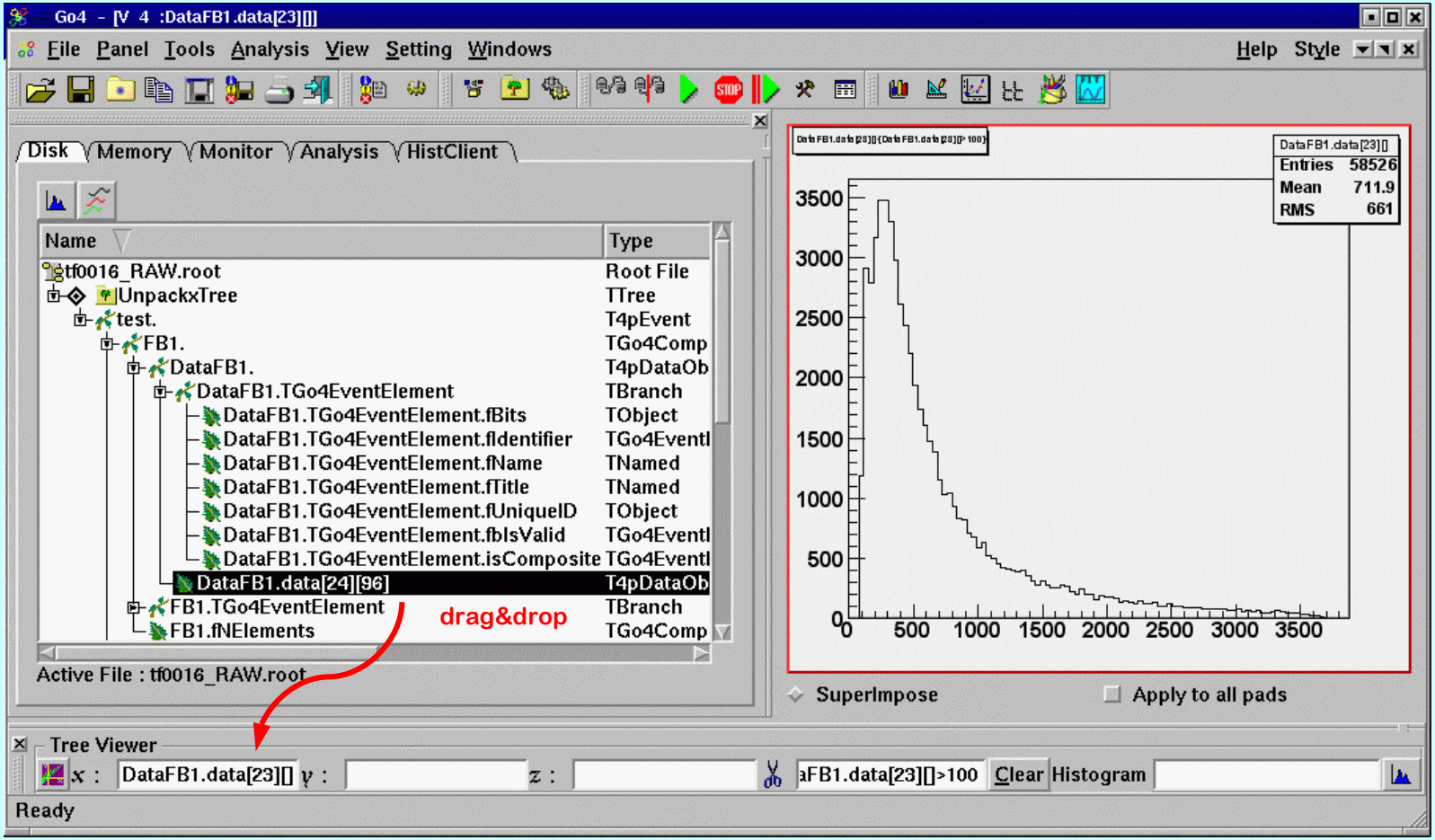
GUI: Stand Alone Features

- Program written in standard ROOT & Qt
- Uses GSI´s QtROOT interface
- Qt and ROOT widgets work
- Developed with Qt designer
- Extended browser (ROOT files)
- Extended tree viewer (unlimited levels for Go4 events)
- Fit (GUI for stand alone fit package)
- Set/save/restore layout

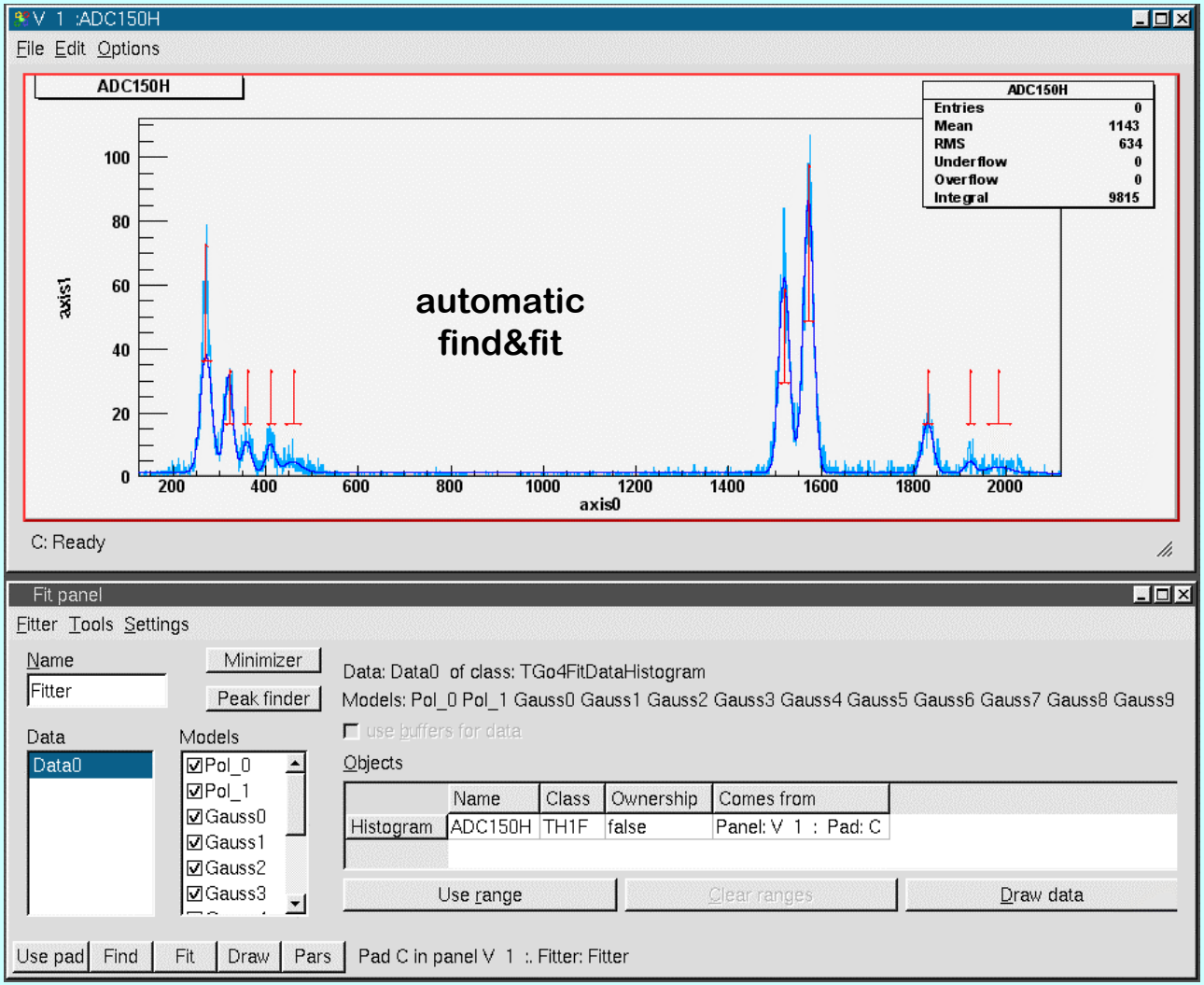
Root File Browser



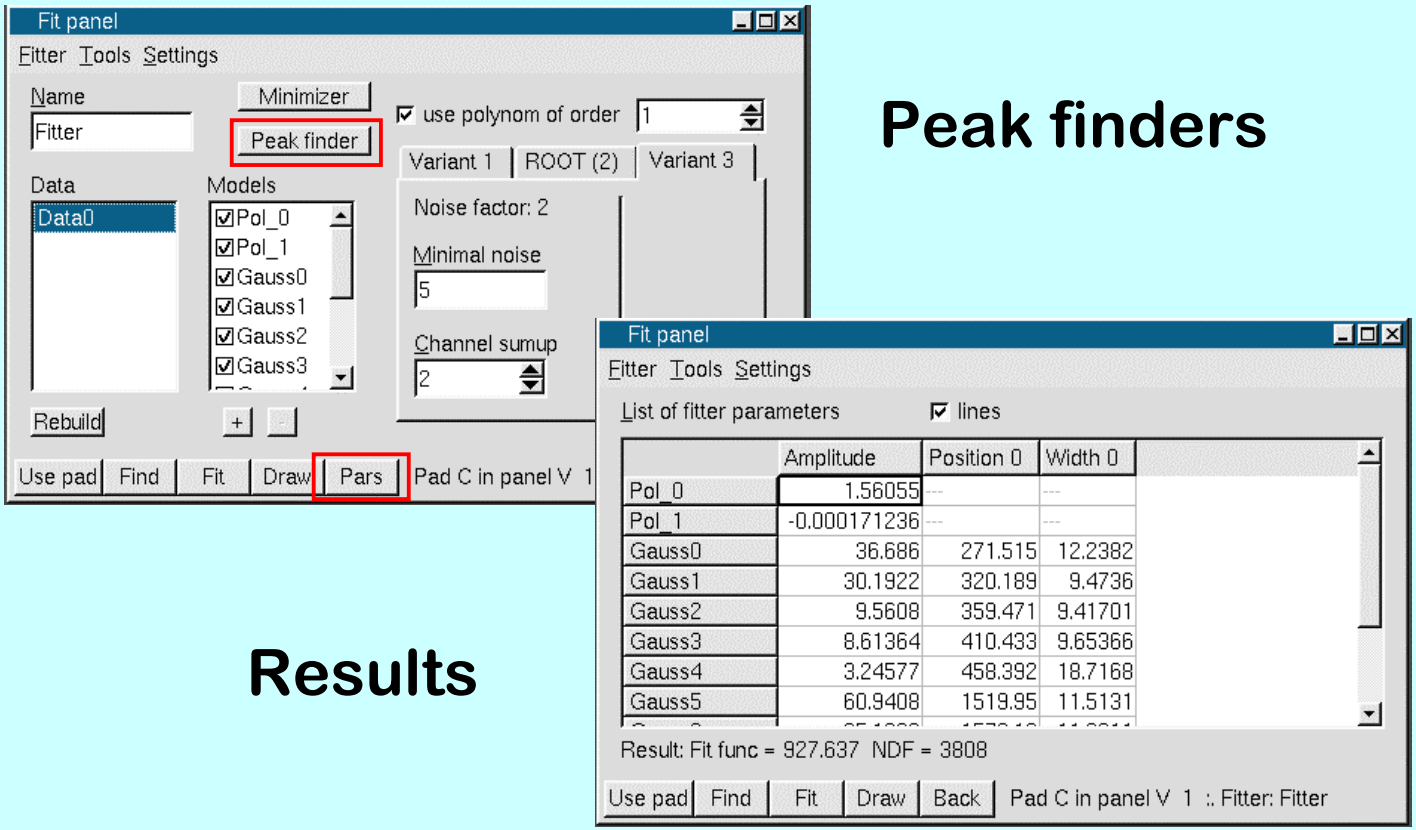
Tree Viewer


Go4 - [V 4 :DataFB1.data[23][]]
File Panel Tools Analysis View Setting Windows
Help Style
Disk Memory Monitor Analysis HistClient
Name
Type
tf0016_RAW.root
Root File
UnpackxTree
TTree
test.
T4pEvent
FB1.
TGo4Comp
DataFB1.
T4pDataOb
DataFB1.TGo4EventElement
TBranch
DataFB1.TGo4EventElement.fBits
TObject
DataFB1.TGo4EventElement.fIdentifier
TGo4Eventl
DataFB1.TGo4EventElement.fName
TNamed
DataFB1.TGo4EventElement.fTitle
TNamed
DataFB1.TGo4EventElement.fUniqueID
TObject
DataFB1.TGo4EventElement.fbIsValid
TGo4Eventl
DataFB1.TGo4EventElement.isComposite
TGo4Eventl
DataFB1.data[24][96]
T4pDataOb
FB1.TGo4EventElement
TBranch
FB1.fNElements
TGo4Comp
drag&drop
Active File : tf0016_RAW.root
DataFB1.data[23][]
Entries 58526
Mean 711.9
RMS 661
3500
3000
2500
2000
1500
1000
500
0
0 500 1000 1500 2000 2500 3000 3500
SuperImpose
Apply to all pads
Tree Viewer
x : DataFB1.data[23][] y :
z :
aFB1.data[23][]>100
Clear
Histogram
Ready

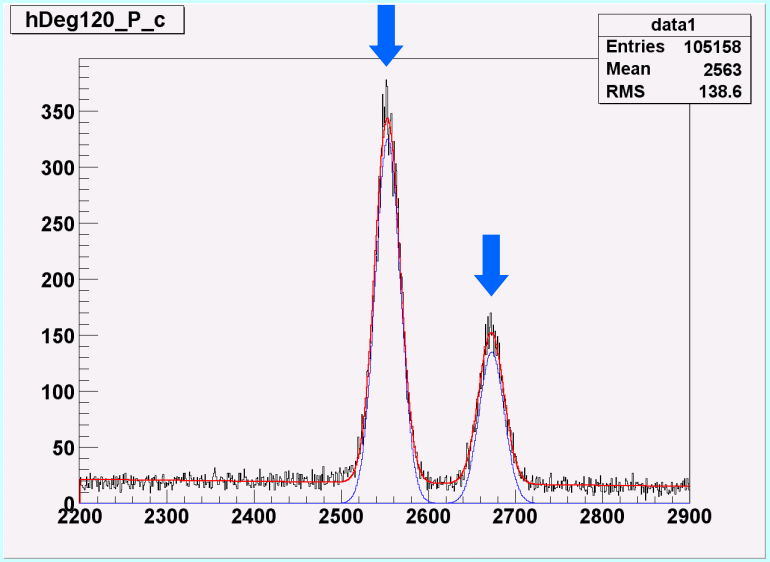
Fitting

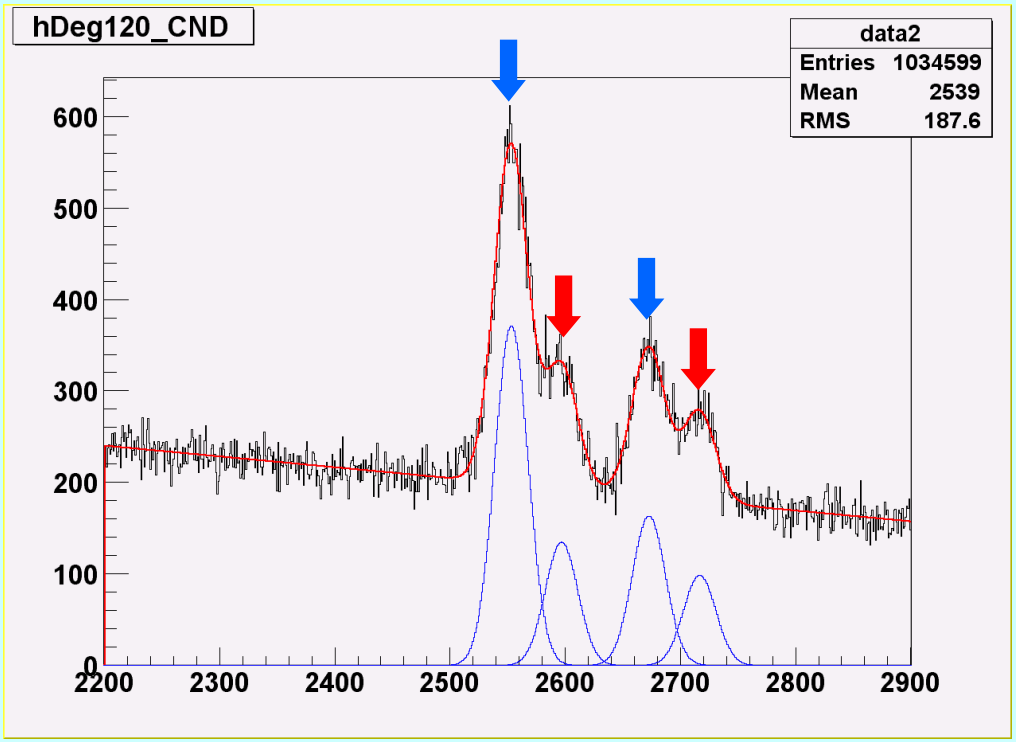

V 1 :ADC150H
File Edit Options
ADC150H
ADC150H
Entries 0
Mean 1143
RMS 634
Underflow 0
Overflow 0
Integral 9815
automatic
find&fit
axis1
axis0
C: Ready
Fit panel
Fitter Tools Settings
Name
Fitter
Minimizer
Peak finder
Data: Data0 of class: TGo4FitDataHistogram
Models: Pol_0 Pol_1 Gauss0 Gauss1 Gauss2 Gauss3 Gauss4 Gauss5 Gauss6 Gauss7 Gauss8 Gauss9
use buffers for data
Data
Data0
Models
Pol_0
Pol_1
Gauss0
Gauss1
Gauss2
Gauss3
Objects
Name
Class
Ownership
Comes from
Histogram
ADC150H
TH1F
false
Panel: V 1 : Pad: C
Use range
Clear ranges
Draw data
Use pad
Find
Fit
Draw
Pars
Pad C in panel V 1 :. Fitter: Fitter

Fitting


Fit panel
Fitter Tools Settings
Name
Fitter
Minimizer
Peak finder
use polynom of order
1
Variant 1
ROOT (2)
Variant 3
Data
Data0
Models
Pol_0
Pol_1
Gauss0
Gauss1
Gauss2
Gauss3
Noise factor: 2
Minimal noise
5
Channel sumup
2
Rebuild
Use pad
Find
Fit
Draw
Pars
Pad C in panel V 1
Peak finders
Fit panel
Fitter Tools Settings
List of fitter parameters
lines
Amplitude
Position 0
Width 0
Pol_0 1.56055
Pol_1 -0.000171236
Gauss0 36.686 271.515 12.2382
Gauss1 30.1922 320.189 9.4736
Gauss2 9.5608 359.471 9.41701
Gauss3 8.61364 410.433 9.65366
Gauss4 3.24577 458.392 18.7168
Gauss5 60.9408 1519.95 11.5131
Result: Fit func = 927.637 NDF = 3808
Use pad
Find
Fit
Draw
Back
Pad C in panel V 1 :. Fitter: Fitter
Results

Fitting





Double fit

Fit library independent of Go4!

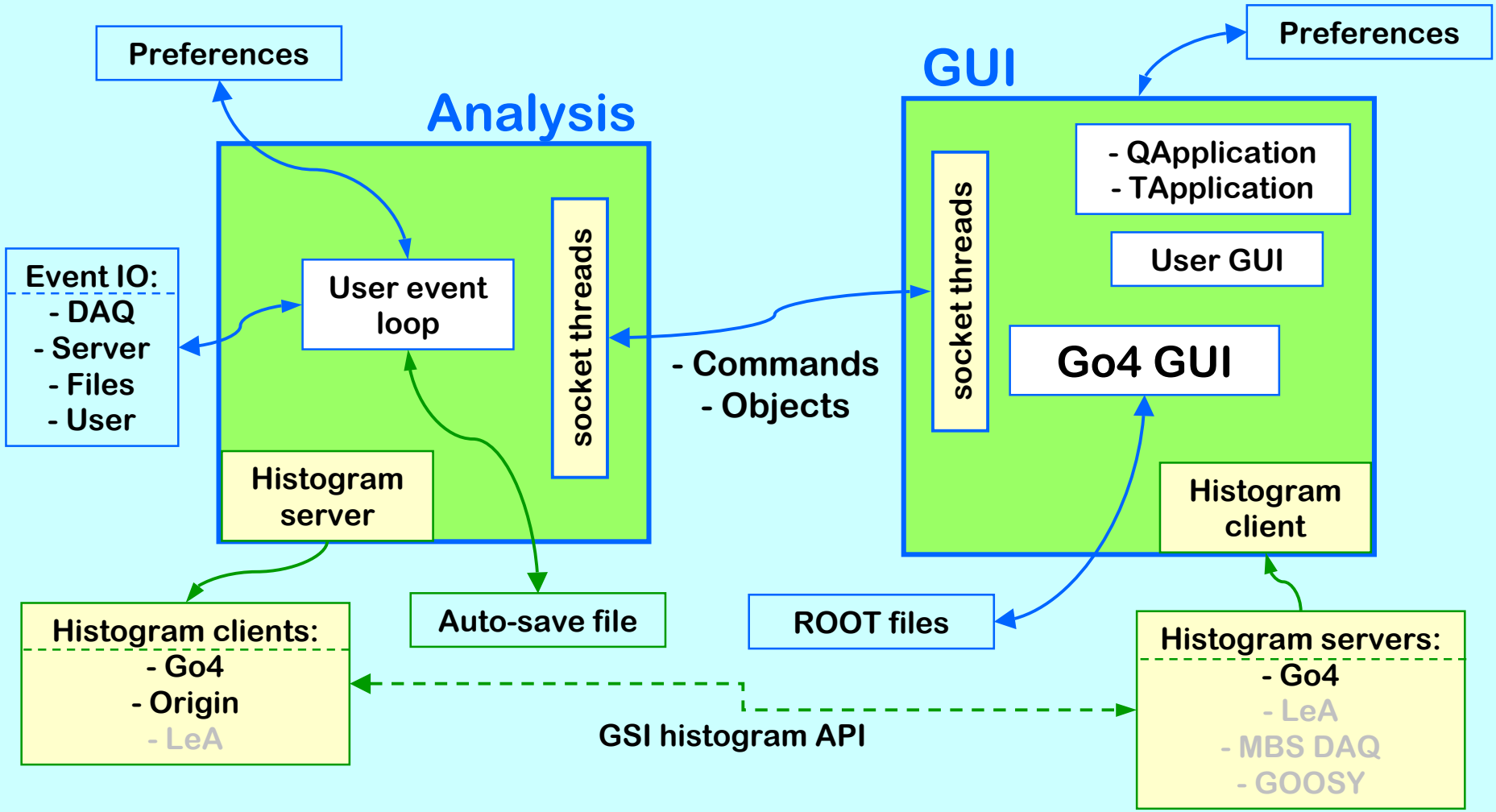
How to Use Go4 cont.

- Use GUI as ROOT-file browser / viewer
  - Histograms
  - Trees
  - Fitter
- **Connect your ROOT analysis to GUI**
  - **little efford to adapt**
  - **dynamic tree draw**
  - **Example: HADES online analysis (+ GUI)**
- Develop analysis with Go4 framework
  - Event classes (composite)
  - Event IO (any input, trees, branches)
  - Analysis steps
  - Parameters, conditions + editors
  - Dynamic histogramming

How to Use Go4 cont.

- Use GUI as ROOT-file browser / viewer
  - Histograms
  - Trees
  - Fitter
- Connect your ROOT analysis to GUI
  - little efford to adapt
  - dynamic tree draw
  - Example: HADES online analysis
- **Develop analysis with Go4 framework**
  - **Event classes (composite)**
  - **Event IO (any input, trees, branches)**
  - **Analysis steps**
  - **Parameters, conditions + editors**
  - **Dynamic histogramming**

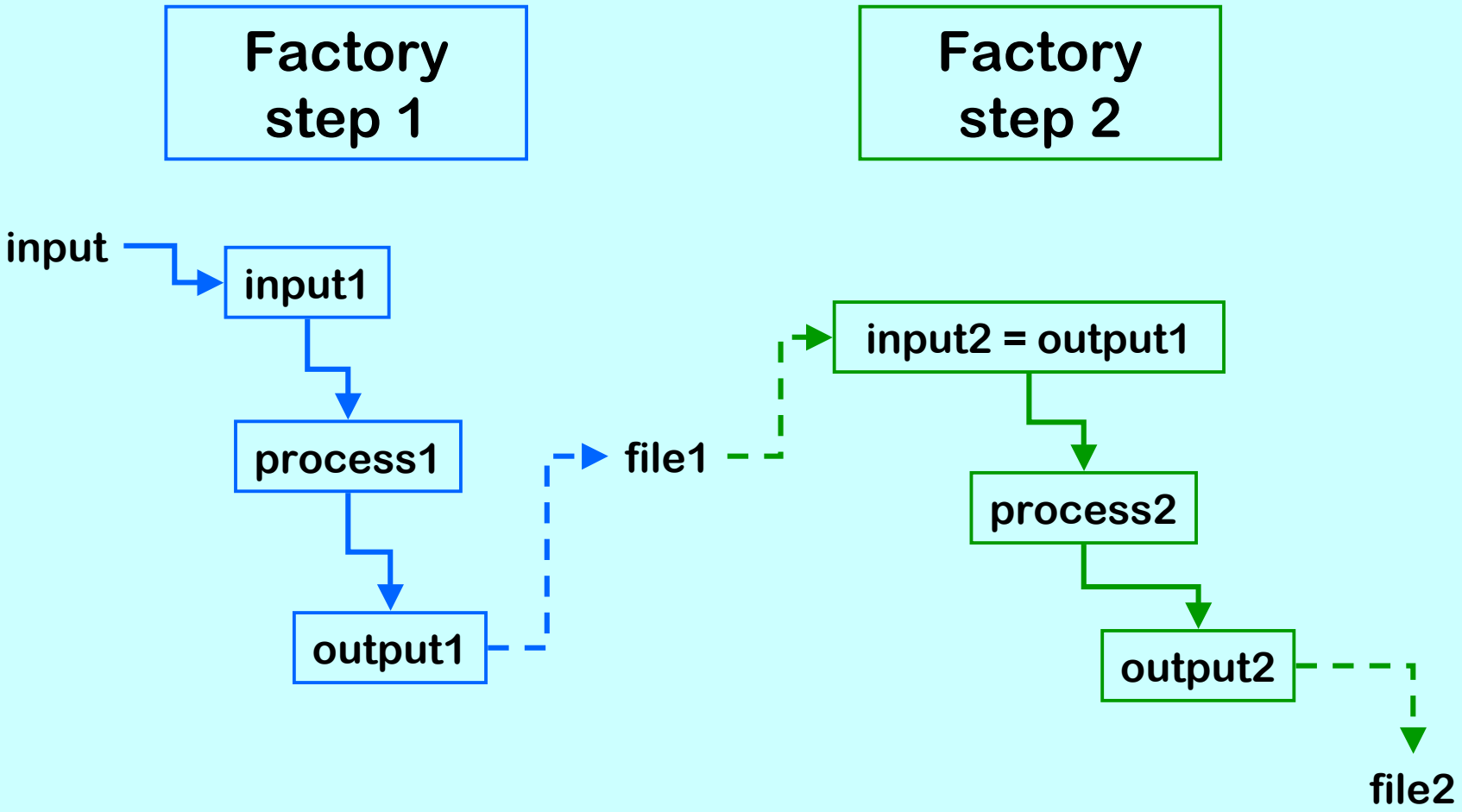
Go4 Tasks: GUI & Analysis



Analysis

- User written (main program)
- Runs in CINT, line mode (batch) or GUI controlled
- Analysis step factory

Analysis Steps (Event Loop)



Analysis cont.

- User written (main program)
- Runs in CINT, line mode (batch) or GUI controlled
- Analysis step factory
- Composite event classes

Go4 composite event

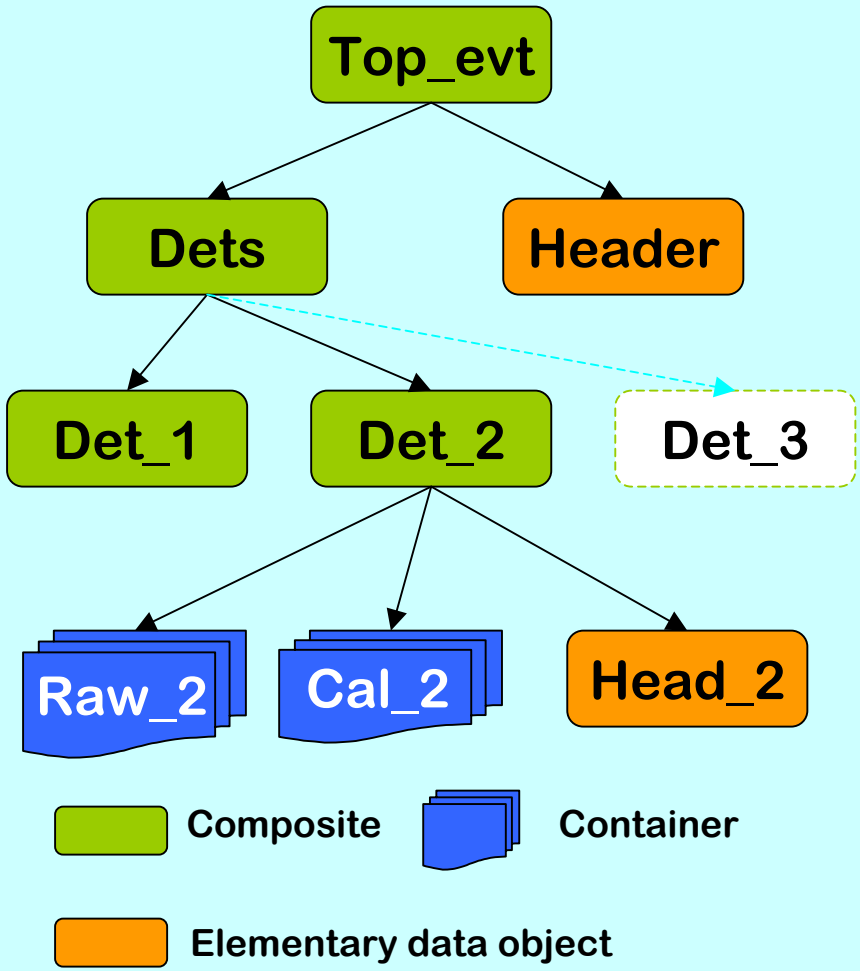
- Build complex event structure out of simple data object components
- Treats single data objects and composition of data objects uniformly
- Easy data store/retrieval to/from file
- Use standard ROOT class for IO mechanism

  (No changes in TTree or TBranchElement)
- ⇨ Recursive method to synchronize the composite structure with TTree

  To retrieve user data, no need for :
  - TTree::MakeCode()
  - TTree::MakeClass()

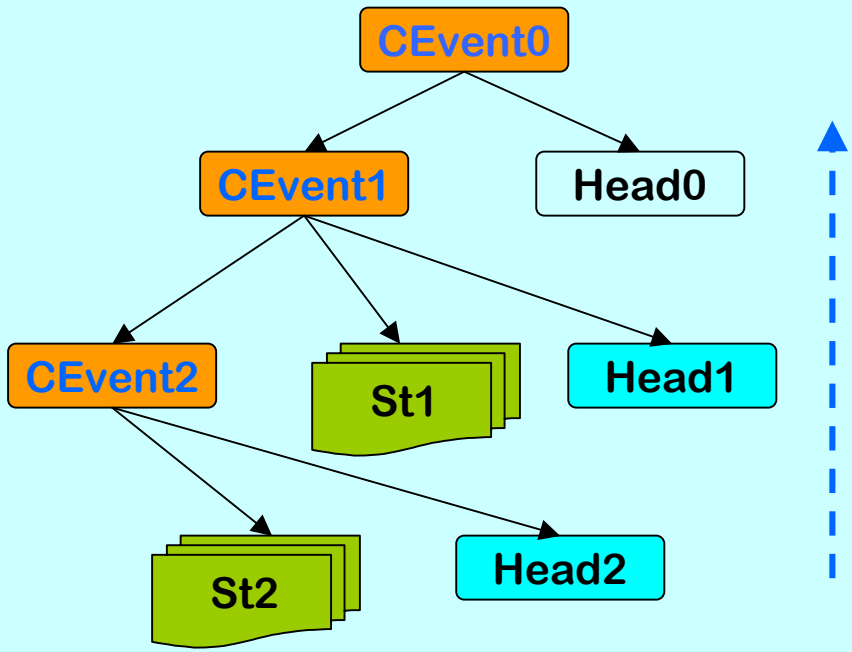
Go4 composite event

- Structural organization of secondary data.
- Represents the natural experimental setup (geometry, detectors, …)
- Fast direct access of components (by index)
- Full or partial IO: map the event object into a ROOT TTree. Use standard ROOT classes for IO mechanism
- ⇨ Recursive method to synchronize the composite structure with TTree at run time
- General design to fit to different needs
- Use TClonesArray, TObjectArray



Go4 composite event

```
{
// Event structure modeling
// second composite
  CEvent2->addElement(head2);
  CEvent2->addElement(st2);
// first composite
  CEvent1->addElement(head1);
  CEvent1->addElement(st1);
  CEvent1->addElement(CEvent2);
// add in top composite event
  CEvent0->addElement(head0);
  CEvent0->addElement(CEvent1);
// set top level branch
  TBranch*b = T->Branch("top",
              "CompositeEvent",&CEvent0);
//compose tree structure
  CEvent0->makeBranch(b,split);
}
```



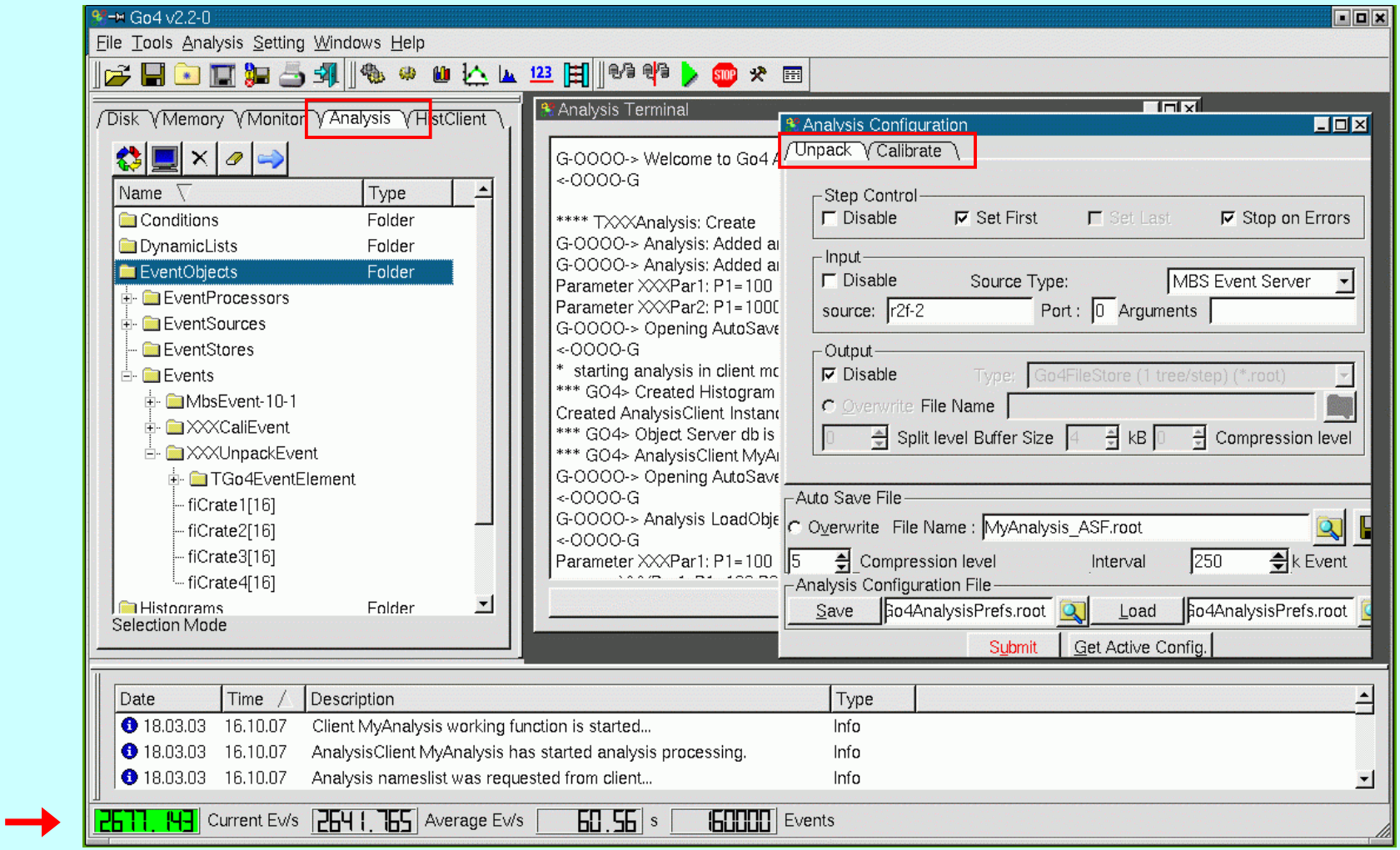
Analysis cont.

- User written (main program)
- Runs in CINT, line mode (batch) or GUI controlled
- Analysis step factory
- Composite event classes
- Event IO (GSI sources, others)
- Analysis control from GUI

GUI for Analysis

- Analysis control (start, stop, ...)
- Analysis step control (built from analysis)
- Visualize analysis status (from analysis)
- Remote browser (objects from analysis)
- Remote tree draw (tree from memory of analysis)
- Condition editing (window, polygon from analysis)
- Object editing (from analysis)
- Adding user GUIs
- Event displays (user), can be initiated from analysis

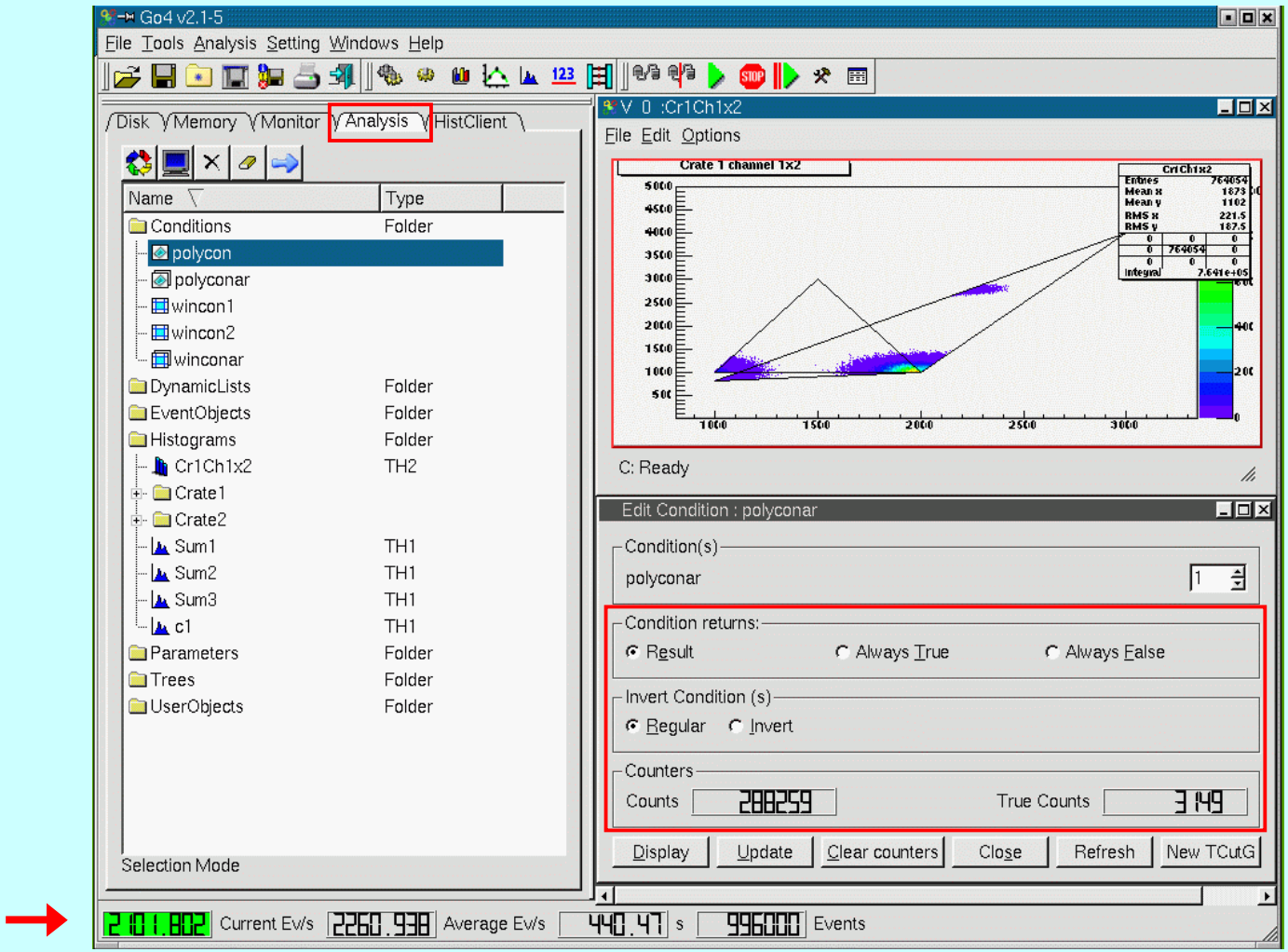
Analysis started



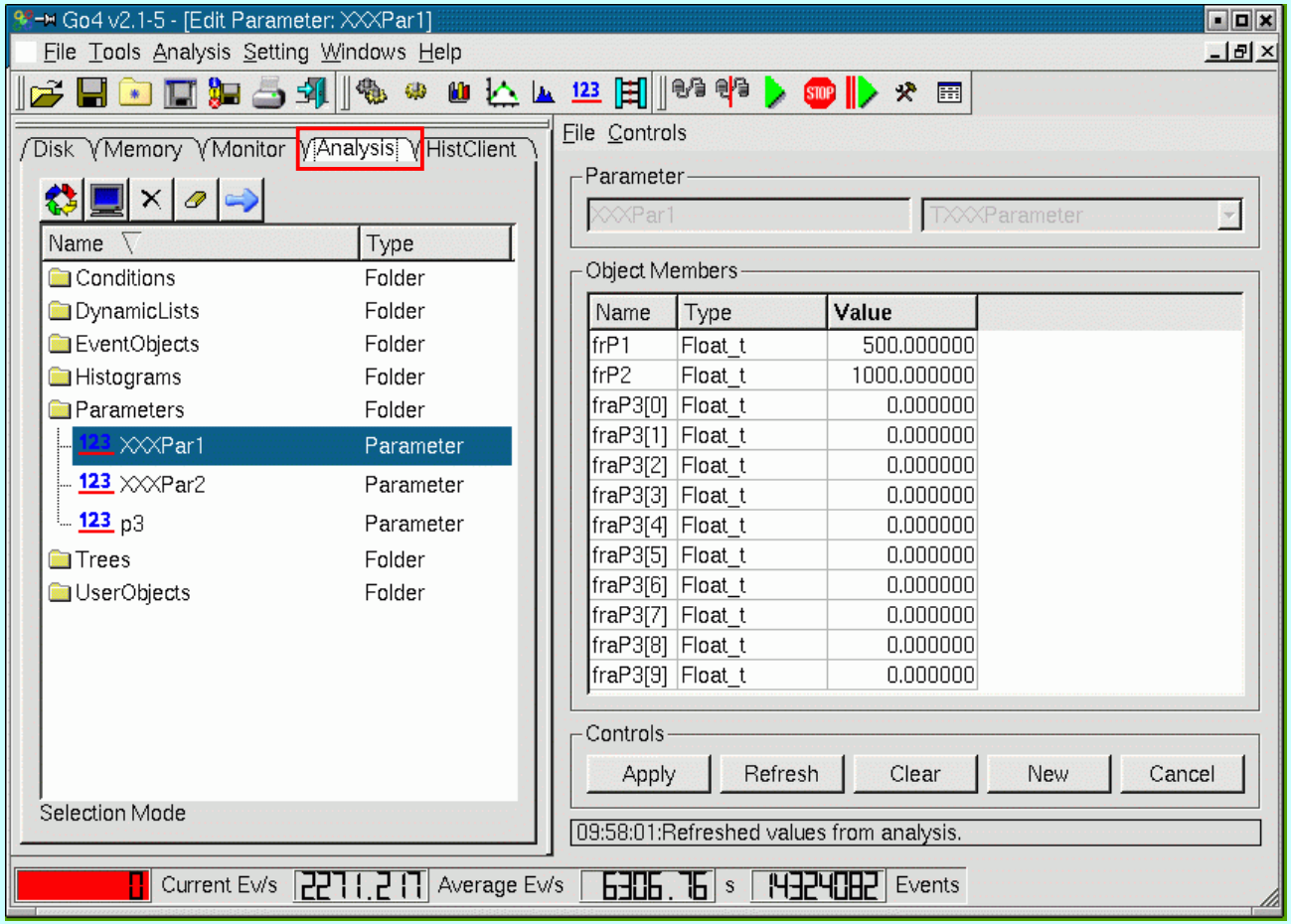
Analysis cont.

- User written (main program)
- Runs in CINT, line mode (batch) or GUI controlled
- Analysis step factory
- Composite event classes
- Event IO (GSI sources, others)
- Analysis control
- Conditions (windows, polygons)
- Parameter editor
- Dynamic histogramming
- Persistency through autosave

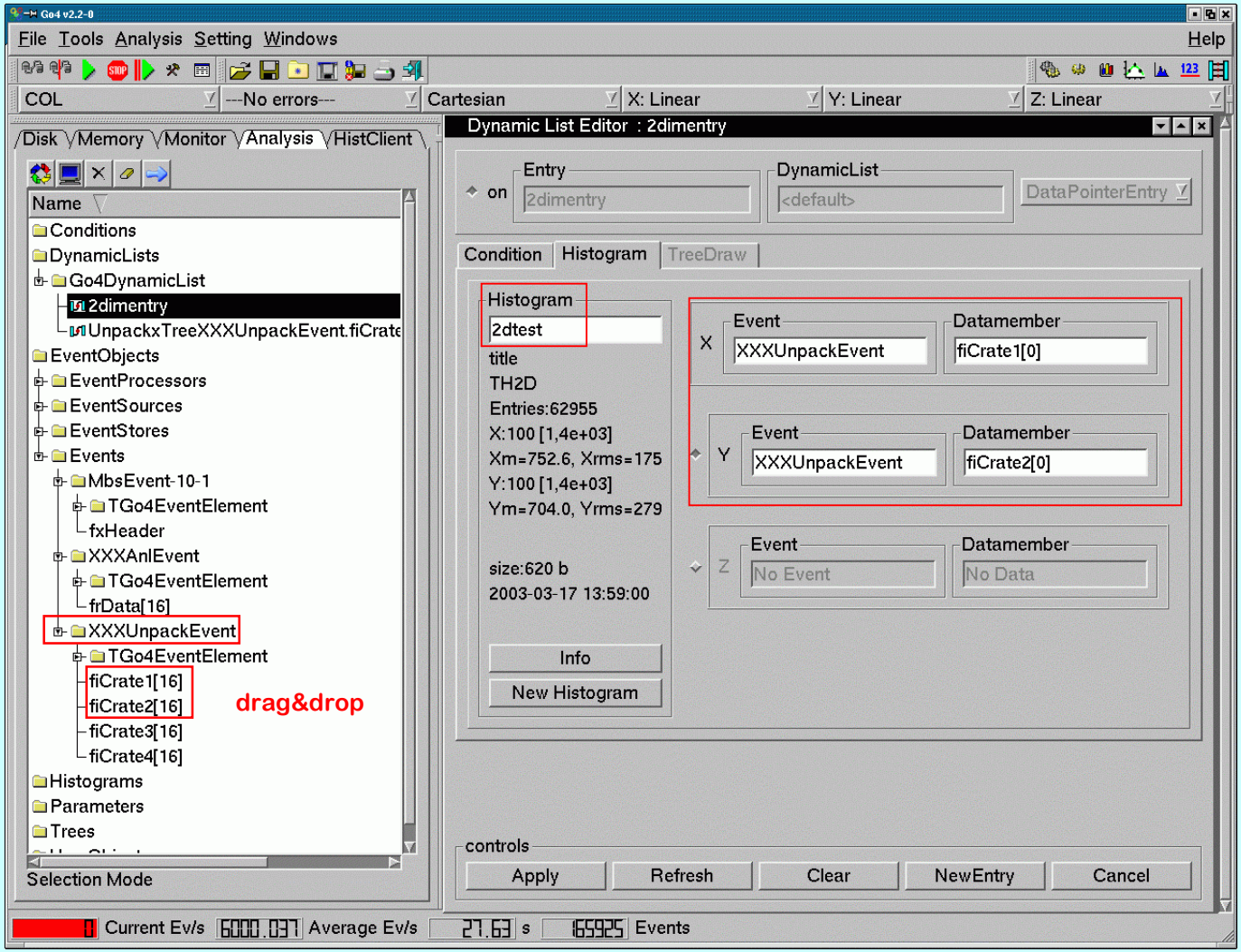
Condition editor



Parameter editor

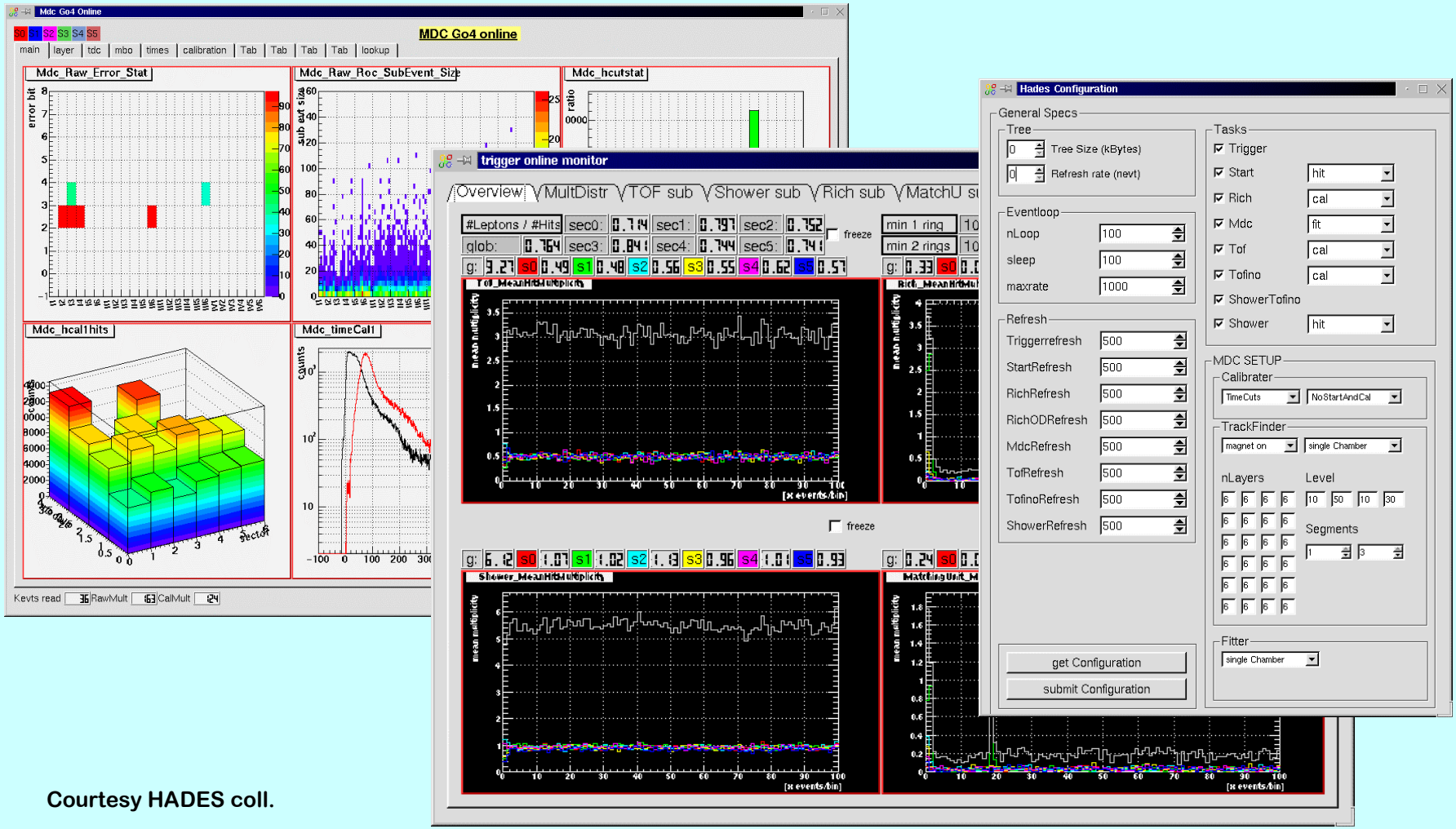


Dynamic list editor



User GUI (Qt)

On-line monitoring of HADES processing data objects sent periodically from analysis



Courtesy HADES coll.

Summary

- **Use Go4 GUI as ROOT-file browser / viewer**
  - Histograms
  - Trees
  - Fitter
- **Connect your ROOT analysis to GUI**
  - little efford to adapt
  - dynamic tree draw
  - GUI and analysis in two tasks
- **Develop analysis with Go4 framework**
  - Event classes (composite)
  - Event IO (any input, trees, branches)
  - Analysis steps
  - Parameters, conditions + editors
  - Dynamic histogramming
  - GUI and analysis in two tasks

The End

GSI
Online
Offline
Object
Oriented

Go4 v2.2 free download

http://go4.gsi.de

tested with
Debian 3.0, RedHat 8.0, Suse 8.1
gcc 2.95-x, gcc 2.96-x, gcc 3.2, icc 7.0

J. Adamczewski, M. Al-Turany, D. Bertini, H.G.Essel, S.Linev