



GO4 Version 5.0

A ROOT based online and offline analysis environment

J.Adamczewski-Musch, S.Linev

ROOT 2015, Saas-Fee



Outline



History: **“Go4 turns 16”**

Review: **Go4 features - as told at ROOT workshops**

New V5.0: **Go4 with/for ROOT THttpServer & JSROOT**

Summary



ROOT turns 20 - Go4 turns 16



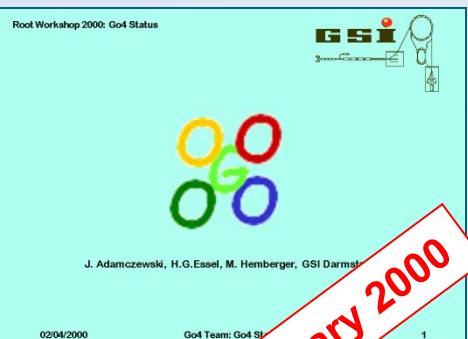
- Development start: April 1999 - GSI hires 2 postdocs for **GSI Object Oriented Online Offline (Go4) project**
aim: successor of GOOSY (VMS), analysis of MBS DAQ data
- Go4 v.1.0 May 2002 - first complete version:
analysis framework, Qt-ROOT GUI, inter-task communication
- Go4 v.2.0 November 2002 - first public production release
- Go4 v.3.0 December 2005 - analysis server, GUI redesign
- Go4 v.4.0 February 2008 - Qt4
- **Go4 v.5.0 June 2015 :**
Qt5, ROOT6, THttpServer , JSROOT/jQuery UI GUI

Go4 developers:

Jörn Adamczewski-Musch, Mohammad Al-Turany, Denis Bertini,
Hans.G.Essel, Marc Hemberger, Sergey Linev



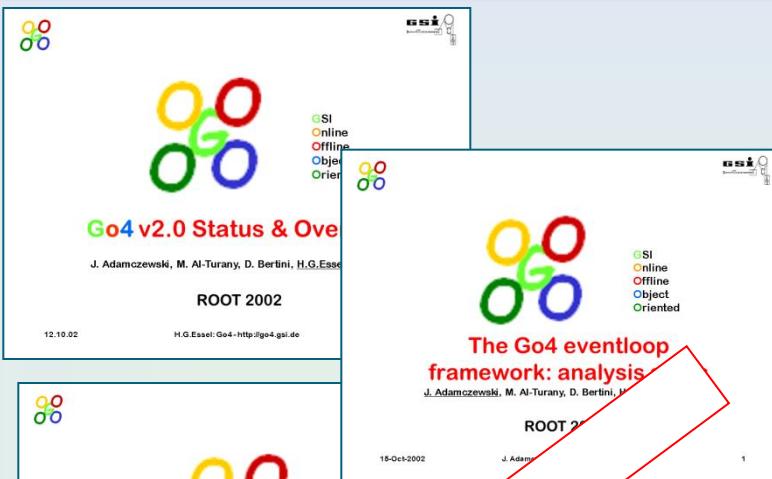
Go4 at ROOT workshops



J. Adamczewski, H.G. Essel, M. Hemberger, GSI Darmstadt

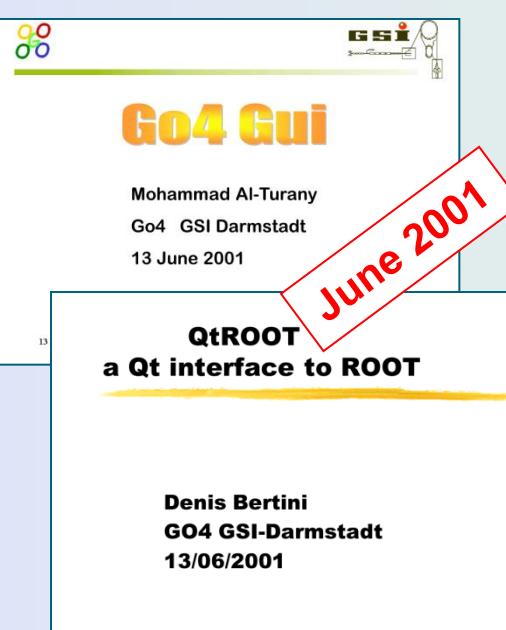
E.Sessel, M. Hemberger, GSI Darmstadt
Go4 Team: Go4 SI

February 2000

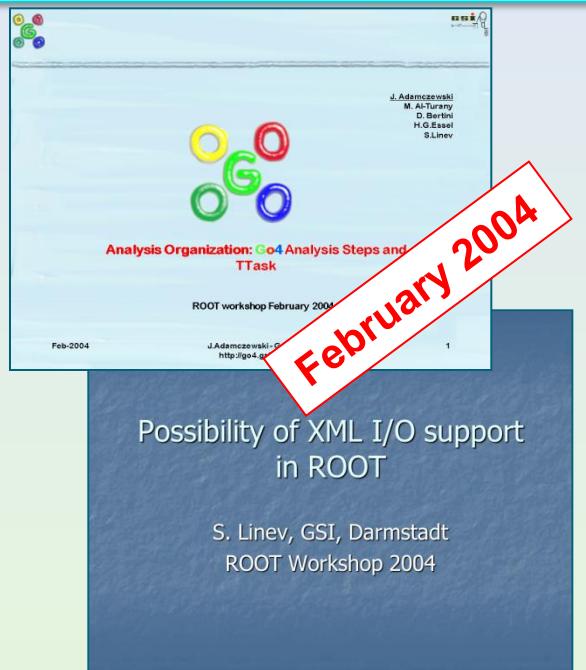
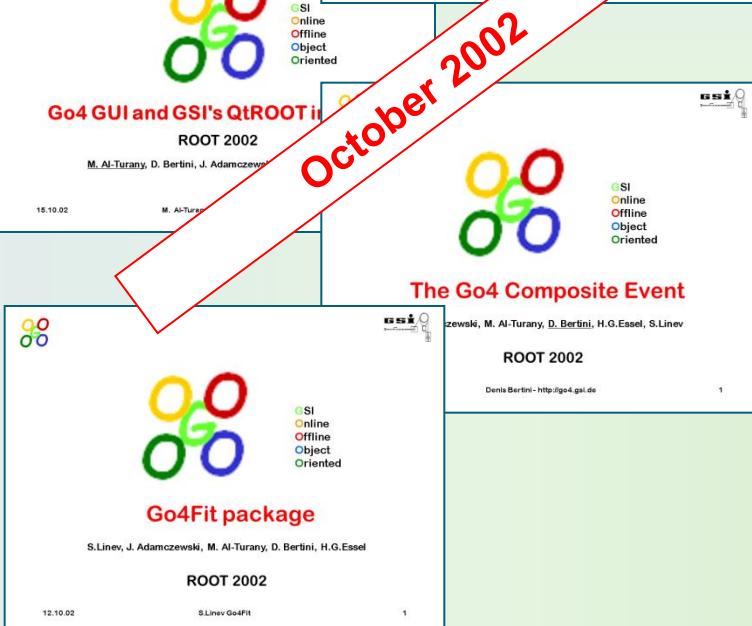


Go4 v2.0 Status & Overview

201/200



June 2001



The image shows a blue background with a large, diagonal watermark in red text. The text reads "February 2004" in a bold, sans-serif font. The watermark is oriented diagonally from the bottom-left towards the top-right. In the top-left corner, there is a small portion of a blue circle, likely from a previous slide. The overall composition suggests a presentation slide.



September 2005

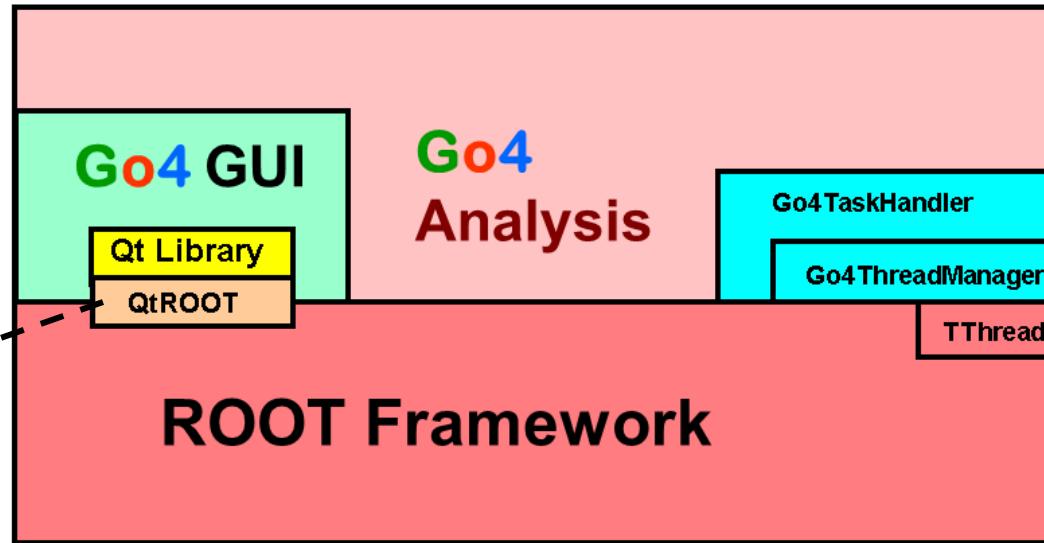


The image shows the five interlocking Olympic rings, which are colored blue, yellow, black, green, and red from left to right.

Package Layers Go4 v2.0



ROOT 2002



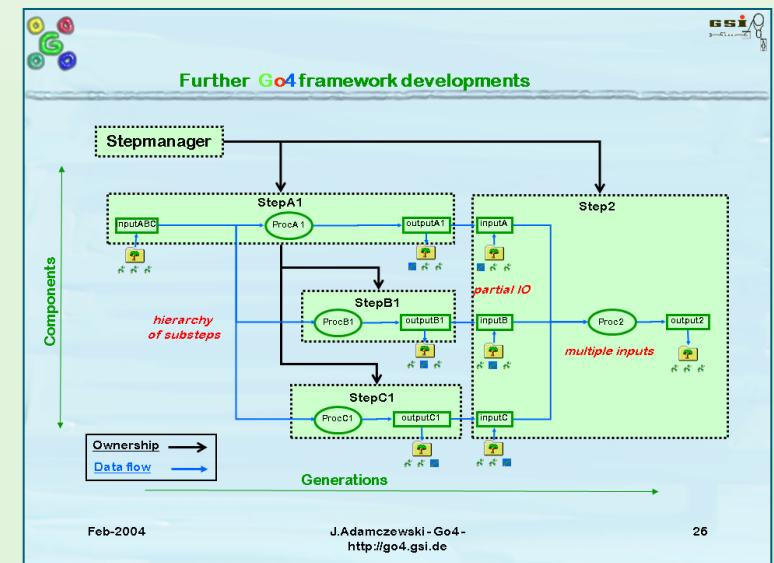
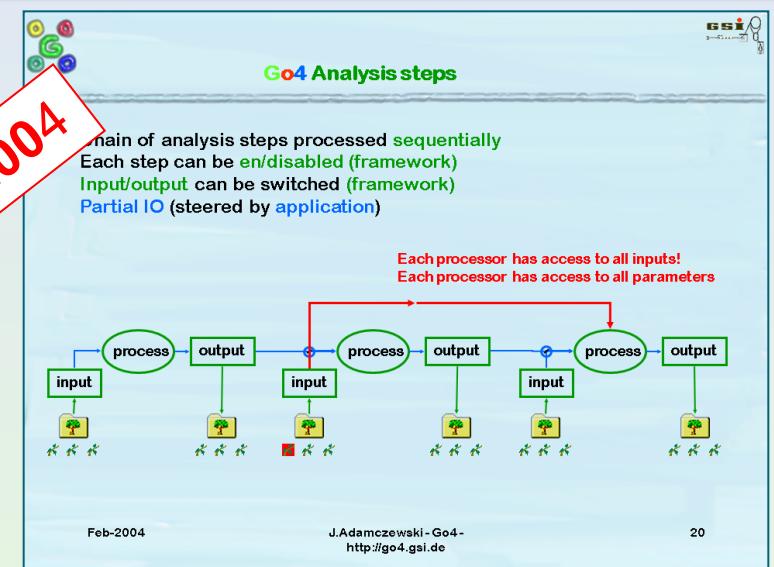
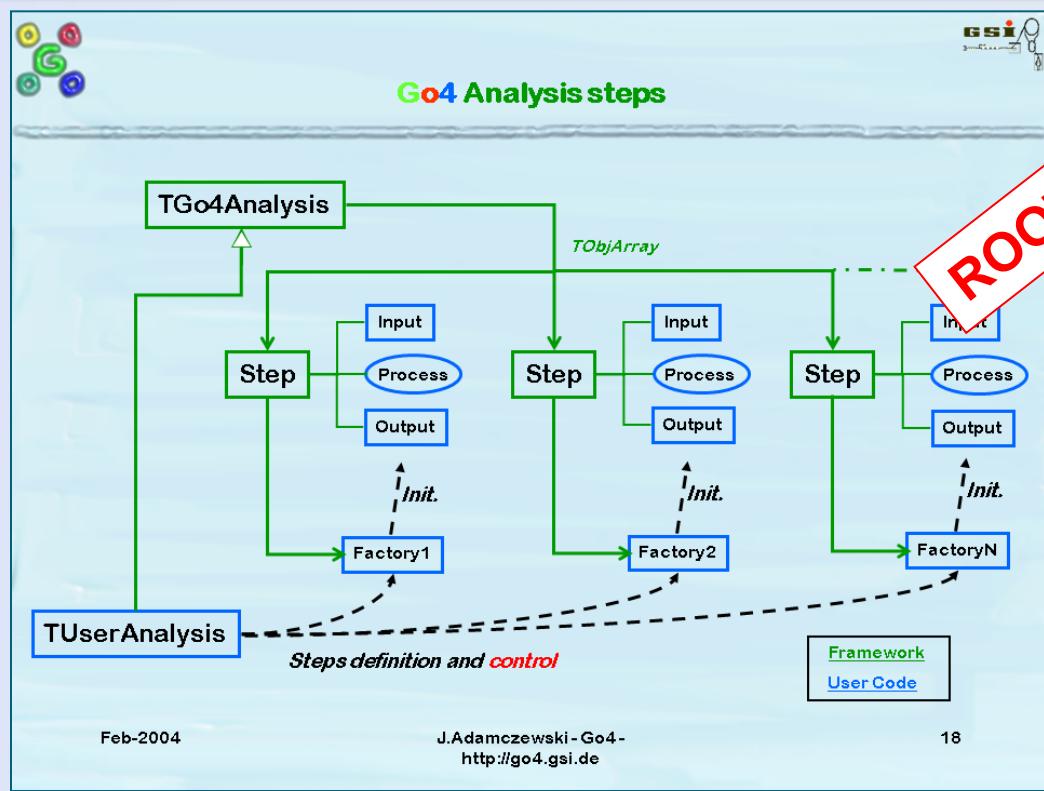
QtROOT GSI

libThread

12.10.02

H.G.Essel: Go4 - <http://go4.gsi.de>

2



User analysis configuration:

- generic batchmode arguments
 - interactive set-up from generic GUI



Go4 GUI

The image shows two versions of the ROOT application side-by-side, separated by a large red diagonal watermark reading "ROOT 2000" above and "ROOT 2002" below.

Left (2000 Version):

- Spectrum Window:** Displays a spectrum plot with a peak at approximately 500 Hz. A status bar at the bottom shows "File Options".
- Object Status Window:** Shows objects like "spectrum1", "exp01", "sample01", "usercamera", and "mag333".
- Message Window:** Lists several reports (Report 221, 222, 223, 225) with event counts (e.g., 737058, 239779, 243153, 748073, 751867).

Right (2002 Version):

- Analysis Started and Running:** A green banner at the top.
- Go4 Window:** Shows a histogram titled "Monitored histogram" under "Type TH1D".
- Analysis Terminal:** Displays log messages related to analysis tasks and histograms.
- Event List:** Shows a table with columns "Date", "Time", and "Description", listing events from 12.10.02 to 15.11.02.
- Bottom Status Bar:** Shows "12.10.02", "10 s", "Event 1", and "field...".

Screenshot of Go4 GUI

The screenshot shows the Go4 software interface with multiple windows open:

- Analysis Configuration**: Shows the analysis tree with nodes like `/Unpack`, `/Analysis`, and `/Step Control`. It includes settings for `EventSource` (set to `MG5 Random`), `Name` (`GS1/lea/gauss`), and `Tagfile`.
- Panel 1: Set conditions**: A histogram titled "Condition histogram 14:51:48" showing a single peak. The x-axis ranges from 2000 to 3100, and the y-axis ranges from 0 to 20000. A red box highlights the peak area.
- File Edit Options**: A menu bar at the top of the main window.
- Apply to all**: A button in the top right corner.
- Workspace**: A tree view showing the project structure. It includes nodes for `Analysis0.root`, `Analysis0Free`, `Analysis0Event`, `Analysis0Event`, `Analysis0Event`, `Analysis0Event`, `Analysis0Event`, and `Analysis0Event`.
- Analysis**: A panel showing histograms and event displays. It includes:
 - Histograms**: A 3D plot of "Crash 1 channel 1x2 14:08:16" showing a single point-like event.
 - Condition histogram 14:51:48**: A histogram with a red shaded region representing selected events.
 - Marker 1**: A text box with values $X = 1.09208 \times 10^3$ and $C = 11.9\%$.
 - Marker Modes**: A set of buttons for marker selection.
- Analysis Configuration File**: A file browser showing `Go4AnalysisPrefs.root`.
- Condition editor**: A panel for setting analysis conditions. It shows a histogram of `ch1is1` with limits set to `0.02-0.05` and `0.129-0.131`.
- Date**, **Time**, **Description**, **Type**, **Info**, and **Log** panels at the bottom.

29.09.05

S.Linev: Go4 - <http://go4.gsi.de>

3

- ROOT object monitoring
 - interactive control and configuration
=> elaborate GUI required!

decided for **Qt graphics library (with GUI *designer* tool),
embed ROOT graphics with **QtROOT interface****

ROOT 2015

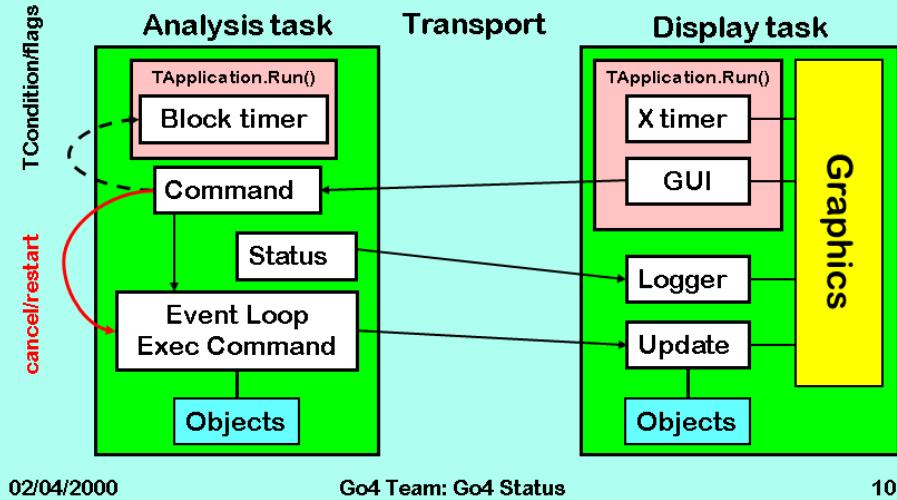
Go4 v5 - <http://go4.gsi.de>

7



Project 4 Functional Prototype

Mixture of tasks, threads and communications



02/04/2000

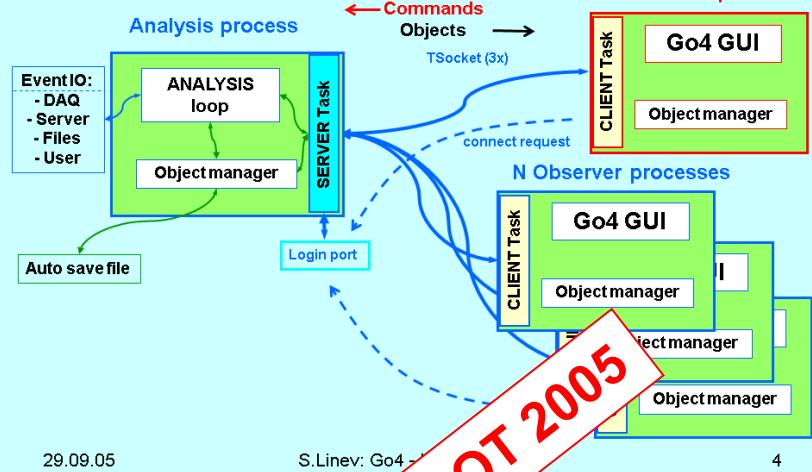
Go4 Team: Go4 Status

10

The logo consists of four overlapping circles in orange, green, blue, and red, arranged in a square pattern.

Improved communication mechanism

651



A red diagonal banner with the text "ROOT 2005" in white, overlaid on a map of the San Francisco Bay Area. The map shows various landmarks and water bodies. A dashed blue line starts from the top left and curves down towards the bottom right, passing through the city of San Francisco.

29.09.05

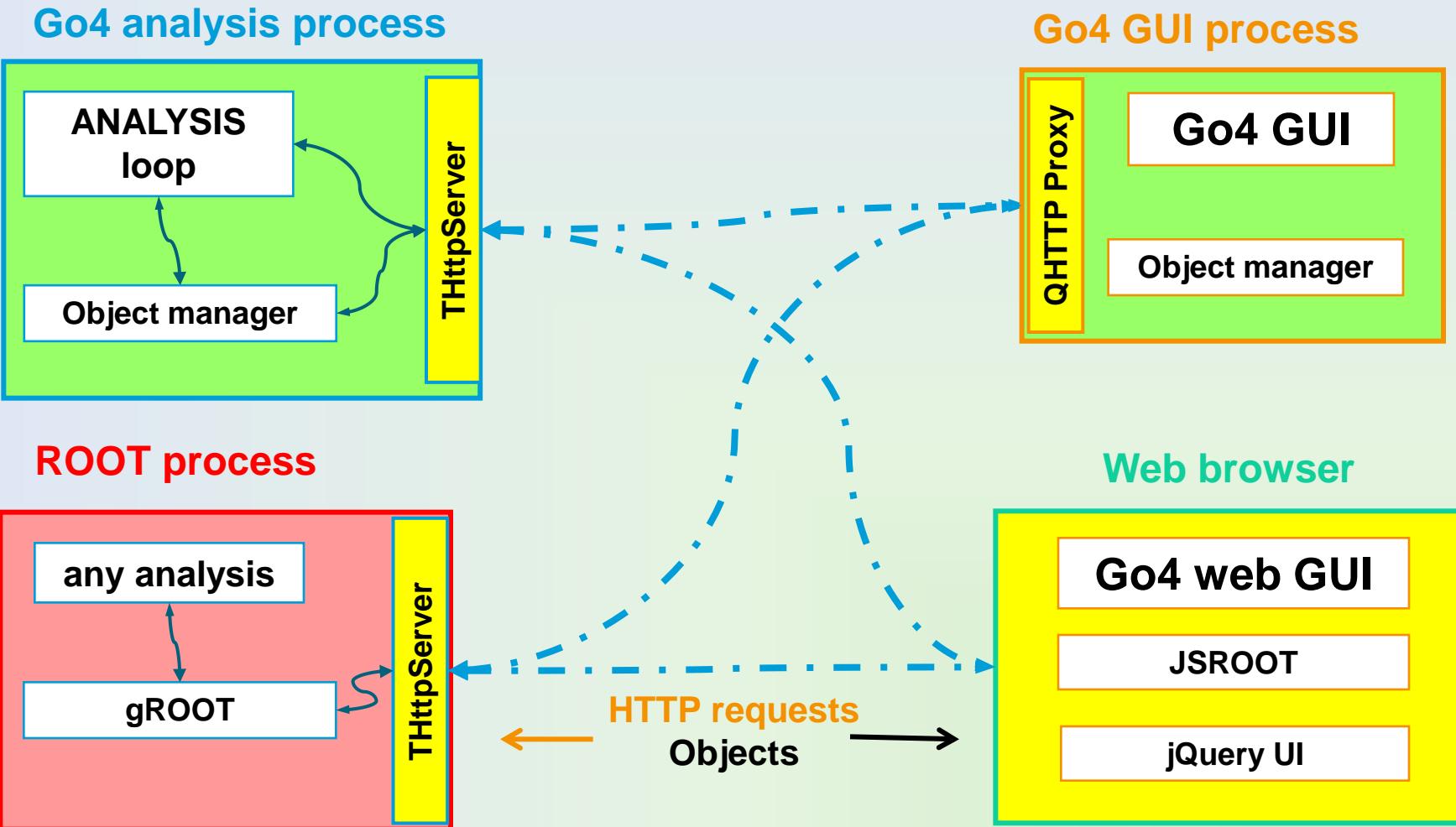
S. Line

4

GUI must not be blocked by analysis, may run on remote node

- Inter-task communication by „transport“ layer:
 - Go4 taskhandler (TThread, TSocket, ROOT streamer)
 - other? -> **ROOT THttpServer (Go4 V5)**

Go4 V5 with ROOT THttpServer and client



[please see THttpServer and JSROOT presentations by B. Bellenot and S. Linev](#)



Go4 V5 with ROOT http server and client

optionally use [generic ROOT THttpServer](#) in Go4 analysis process

(please see THttpServer presentation by S. Linev)

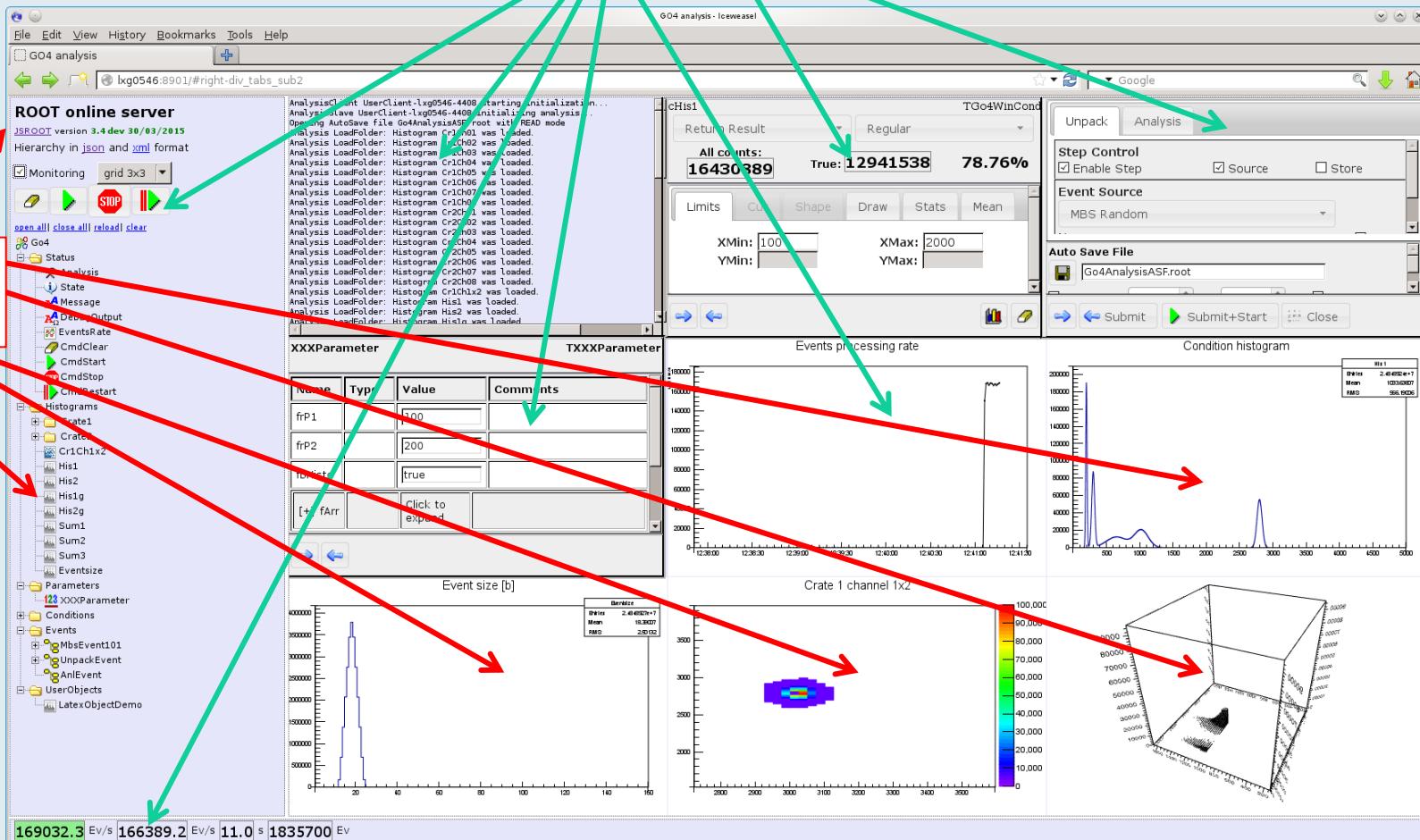
- any web browser may display and control Go4 objects
 - JavaScript ROOT GUI extended by Go4 specific GUI (jQuery UI)
 - Go4 Qt GUI can communicate with Go4 analysis via http server with (almost) full functionality
 - Faster start-up of http server connection than Go4 taskhandler, less resource consumption!
 - Go4 Qt GUI can also display objects from any ROOT THttpServer
-> better performance than browser with JavaScript ROOT display



any web browser!

Go4 specific elements

JSROOT environment





Go4 specific GUI elements (1)



Analysis configuration editor

- enable/disable „analysis steps“
 - select event input (MBS online, MBS file, ROOT TTree, user plug-in)
 - select event stores (ROOT TTree)
 - specifiy histogram/object store (ROOT TFile)
 - specify default setup (ROOT TFile)



Analysis configuration editor (Qt4)



Step selection

Analysis Configuration

Raw xxx ✓ Profile xoo

Step Control

Enable Step Source Store

Event source

MBS Stream Server

Name: depc418

Port: dfilt Tmout: 1 s Retr: never

0 all 1

Event store

Go4FileStore (1 tree/step) (*.root)

Name: /data/test/unpack5.root

99 32 kB 5 1000 Overwrite

Auto Save File

Go4AnalysisASF.root

Enabled 500 s 5 Overwrite

Analysis Configuration File

Go4AnalysisPrefs.root

Submit Submit+Start Close

Apply and Refresh

Event input

Event output

Object persistency

Load/save config



Analysis configuration editor (jQuery UI)



Step Control

Enable Step Source Store

Event Source

MBS Stream Server

Name: more...

Port: Tmout (s): Retr (s):

First: Last: Step:

Event store

Go4FileStore (*.root)

Name:

Auto Save File

Enabled Overwrite

Analysis Configuration File

Submit Submit+Start Close

Analysis configuration editor

- enable/disable „analysis steps“
 - select event input (MBS online, MBS file, ROOT TTree, user plug-in)
 - select event stores (ROOT TTree)
 - specifiy histogram/object store (ROOT TFile)
 - specify default setup (ROOT TFile)

Condition display and editor

TGo4Condition: checks if value x / point (x,y) is inside a 1D/2D region

- condition types: window, polygon (TCutG), ellipse, circle, box
 - manipulate condition limits and operation modes
 - draw condition together with corresponding TH1/TH2



The figure shows two windows side-by-side. The left window is titled 'Condition editor' and displays analysis results for 'Analysis/Conditions/ellipsecond'. It includes tabs for 'Limits', 'Cut', 'Shape' (selected), 'Draw', 'Stats', and 'Mean'. The 'Shape' tab contains settings for an ellipse: Center (X: 3000,00000, Y: 3000,00000), Half axes (A: 300,00000, B: 800,00000), and an angle Theta (35). It also has an 'Npoints' slider set to 180 and an 'Autorefresh' checkbox. Below the tabs are icons for refresh, apply, save, and test modes. The right window is titled 'Panel1: [Cr1Ch1x2]' and shows a histogram for 'Crate 1 channel 1x2 16:15:43 2015-05-19 Analysis/Histograms/Cr1Ch1x2'. The plot area shows a red elliptical boundary and a central purple pixelated region. The x-axis ranges from 2400 to 3400, and the y-axis ranges from 2000 to 3800. A status bar at the top of the right window shows various parameters: N=118102, Max=3684, Mean=2688, RMS X=21.78, RMS Y=36.17, Sigma X=5.877e-05, Sigma Y=0.00025, Skewness X=-0.4, Skewness Y=0.801687, and Kurtosis X=4.50177, Y=3.0. An orange arrow points from the 'Shape' tab in the Condition editor to the red ellipse in the histogram.

Condition editor

Panel1: [Cr1Ch1x2]

Analysis/Conditions/ellipsecond

Ellipse

Returns Result

All counts: 829112 True: 35317 4.26%

Limits Cut Shape Draw Stats Mean

Shape

• Ellipse ○ Circle
○ Box ○ Free shape

Center

X: 3000,00000
Y: 3000,00000

Theta

Half axes

A: 300,00000
B: 800,00000

Npoints

180

Autorefresh

refresh, apply, save

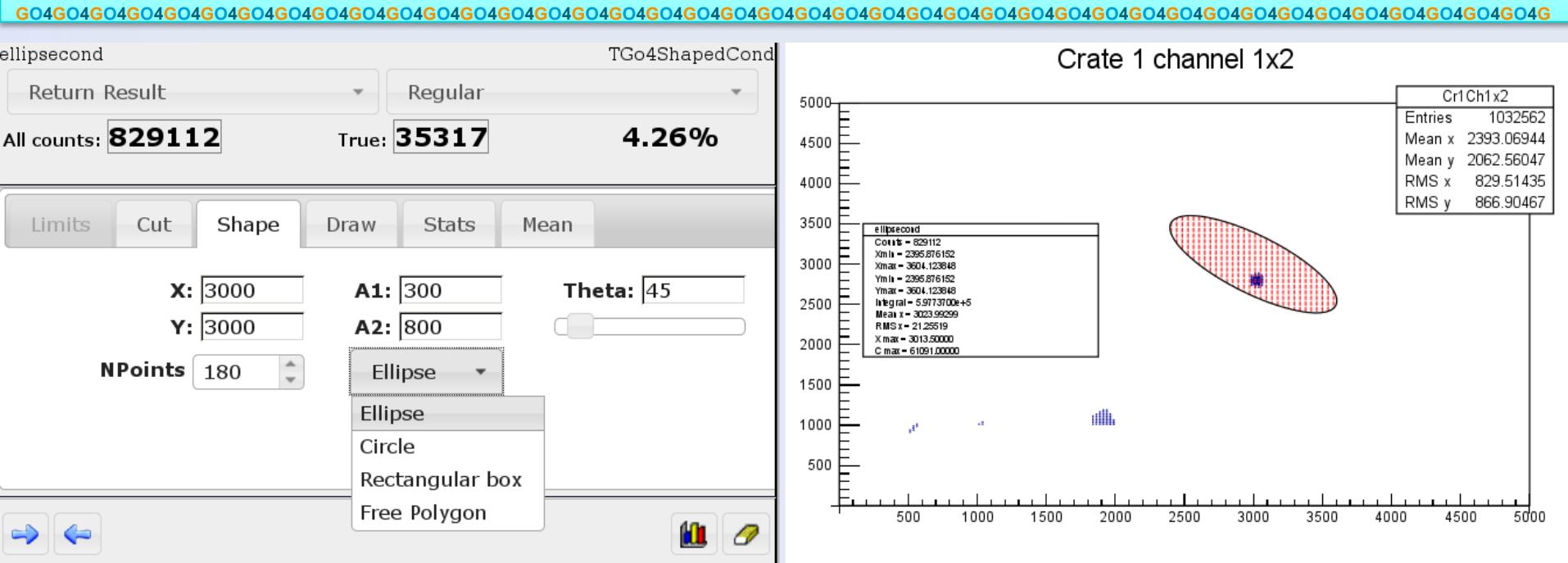
display, clear

boundary shape parameters

condition display in TH2



Condition editor (jQuery UI)



Analysis configuration editor

- enable/disable „analysis steps“
 - select event input (MBS online, MBS file, ROOT TTree, user plug-in)
 - select event stores (ROOT TTree)
 - specifiy histogram/object store (ROOT TFile)
 - specify default setup (ROOT TFile)

Condition display and editor

TGo4Condition: checks if value x / point (x,y) is inside a 1D/2D region

- condition types: window, polygon (TCutG), ellipse, circle, box
 - manipulate condition limits and operation modes
 - draw condition together with corresponding TH1/TH2

Parameter editor

TGo4Parameter: user subclass (plug-in) keeps parameter variables

- change values of fundamental ROOT data type members
 - support C arrays of those (1D,2D,3D)
 - show data member comments/explanations



Parameter class example



```
#include "TGo4Parameter.h"
#include "TArrayI.h"

class TXXXParameter : public TGo4Parameter {
public:
    TXXXParameter(const char* name = 0);
    virtual ~TXXXParameter() {}

    Float_t frP1; // Offset for calibration
    Float_t frP2; // Factor for Calibration
    Bool_t fbHisto; // Enable Histogramming
    TArrayI fArr; // example of usage of TArray in the parameter
    Int_t fArr2[3][4]; // example of 2d array usage
    Int_t fArr3[3][4][5]; // example of 3d array usage (works only
with browser)

ClassDef(TXXXParameter,2)
};
```



Parameter editor (Qt 4)

Parameter Editor

Parameter
localhost:8091/Parameters/XXXParameter - TXXXParameter

Object Members

| Name | Type | Value | Comments |
|-------------|---------|------------|---|
| frP1 | Float_t | 100.000000 | Offset for calibration |
| frP2 | Float_t | 200.000000 | Factor for Calibration |
| fbHisto | Bool_t | 1 | Enable Histogramming |
| fArr[0] | Int_t | 1234 | example of usage of TArray in the parameter |
| fArr2[0][0] | Int_t | 0 | example of 2d array usage |
| fArr2[0][1] | Int_t | 1 | example of 2d array usage |
| fArr2[0][2] | Int_t | 2 | example of 2d array usage |
| fArr2[0][3] | Int_t | 3 | example of 2d array usage |
| fArr2[1][0] | Int_t | 1 | example of 2d array usage |
| fArr2[1][1] | Int_t | 2 | example of 2d array usage |
| fArr2[1][2] | Int_t | 3 | example of 2d array usage |
| fArr2[1][3] | Int_t | 4 | example of 2d array usage |
| fArr2[2][0] | Int_t | 2 | example of 2d array usage |
| fArr2[2][1] | Int_t | 3 | example of 2d array usage |
| fArr2[2][2] | Int_t | 4 | example of 2d array usage |
| fArr2[2][3] | Int_t | 5 | example of 2d array usage |

refresh, apply, save

class member name

value to edit

user-defined explanation

array compressed

array expanded



| Name | Type | Value | Comments |
|-------------|---------------|-----------------|---------------------------|
| frP1 | Float_t | 100 | Offset for calibration |
| frP2 | Float_t | 200 | Factor for Calibration |
| fbHisto | Bool_t | true | Enable Histogramming |
| [+] fArr | TArrayI | Click to expand | Array |
| [+] fArr2 | Int_t [3,4] | Click to shrink | Array |
| fArr2[0][0] | Int_t | 0 | example of 2d array usage |
| fArr2[0][1] | Int_t | 1 | example of 2d array usage |
| fArr2[0][2] | Int_t | 2 | example of 2d array usage |
| fArr2[0][3] | Int_t | 3 | example of 2d array usage |
| fArr2[1][0] | Int_t | 1 | example of 2d array usage |
| fArr2[1][1] | Int_t | 2 | example of 2d array usage |
| fArr2[1][2] | Int_t | 3 | example of 2d array usage |
| fArr2[1][3] | Int_t | 4 | example of 2d array usage |
| fArr2[2][0] | Int_t | 2 | example of 2d array usage |
| fArr2[2][1] | Int_t | 3 | example of 2d array usage |
| fArr2[2][2] | Int_t | 4 | example of 2d array usage |
| fArr2[2][3] | Int_t | 5 | example of 2d array usage |
| [+] fArr3 | Int_t [3,4,5] | Click to expand | Array |





Summary

Go4 “turns 16”: has continuously been improved together with ROOT

Go4 is still alive and in action (~50 various users):

- atomic/nuclear/plasma physics analysis
- detector tests monitoring
- frontend electronics characterization

New Go4 version 5 introduces THttpServer and jQuery UI GUI:

- Go4 analysis can be controlled by any (?) web browser
- Go4 Qt GUI can visualize Go4 analysis and any THttpServer

Custom GUI elements can be added to JSROOT default GUI

(take Go4 configuration/condition/parameter editors as example)

Go4 v5.0 is available under GPL at <http://go4.gsi.de>