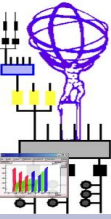
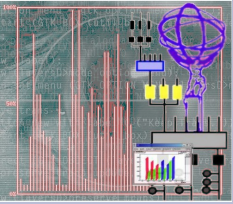


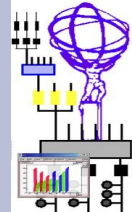
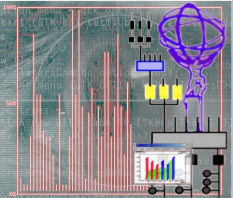
Online Histogram Presenter: an histogram presenter for the ATLAS online monitoring system

A. Dotti, P. Adragna, D. Cimino
andrea.dotti@pi.infn.it



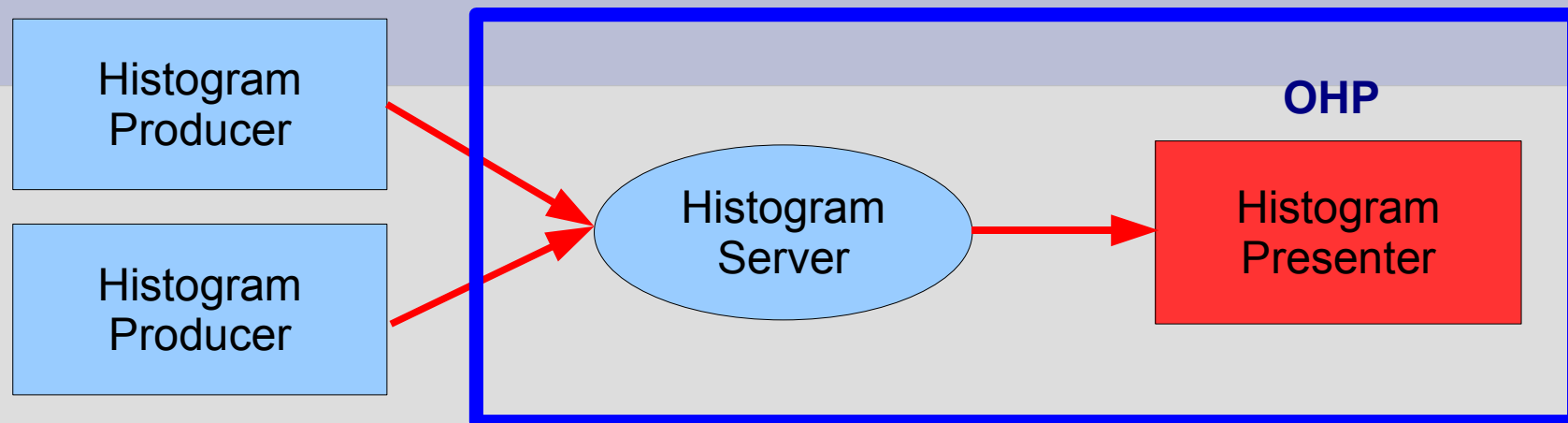
Outlook

- Introduction
- The role of the “Core”
- OHP GUI
 - Some implementation details
 - ROOT and Qt role
 - OHP peculiarities (reference histograms, MetaData)
- New development phase: what is next

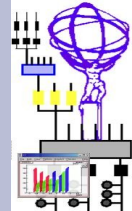
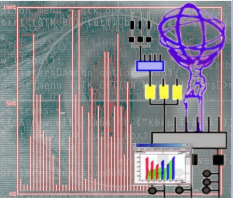


Introduction

Sw components provided by ATLAS Monitoring Working Group (inside ATLAS TDAQ/HLT sub-system)



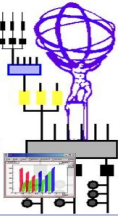
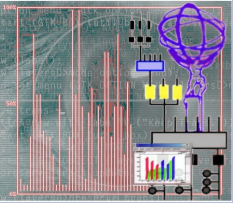
- OHP role is to present histograms produced by the ATLAS Monitoring Components in a user-friendly graphical application
- It is the evolution of various GUIs developed within the sub-detectors communities
- It unifies the approach to histogram visualization in ATLAS online environment
- For more details on ATLAS Monitoring System and OHP see presentations at CHEP06



Minimal Requirements

Red: GUI related

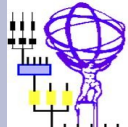
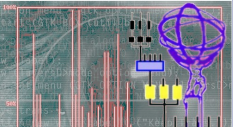
- The presenter must:
 - Be integrated in ATLAS TDAQ system
 - Be able to operate in two distinct modes:
 - Shifter (or tabbed) mode: **display a pre-configured set of histograms**
 - Expert (or browser) mode: allow to **browse the collection of produced histograms**
 - Automatically display histograms when they are available
 - Allow the shifters to **interact with the displayed histograms: zooming, fitting, changing drawing styles,...**
 - **Manage reference histograms**



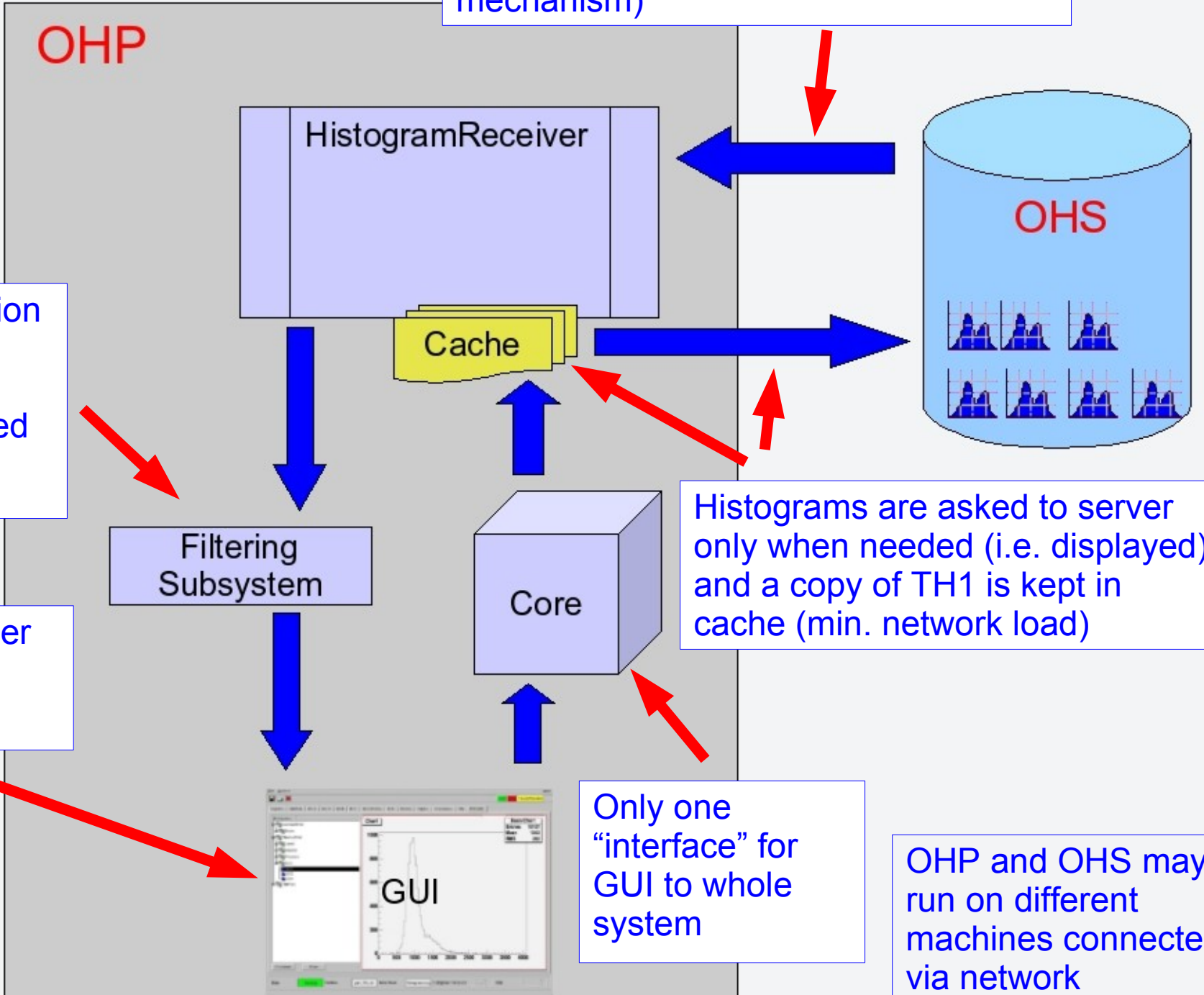
Desirable Requirements

Red: GUI related

- The presenter shall:
 - Minimize the network traffic: expected $O(100k)$ histograms
 - Be independent of the histogram producer: expected many different kinds of producers developed independently
 - Allow to interact with the histogram producers (to change scales, to reset histogram content, to turn on/off histogram production,...)
 - **Allow to set graphic visualization attributes (log scales, draw format)**



A notification (i.e. histogram name) is send from OHS to OHP (subscription mechanism)



Only notification relative to on focus canvas are propagated to GUI

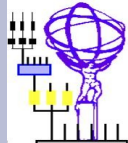
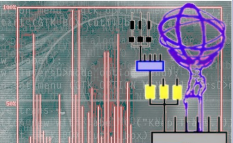
Graphical User Interface subsystem

Histograms are asked to server only when needed (i.e. displayed) and a copy of TH1 is kept in cache (min. network load)

Only one "interface" for GUI to whole system

OHP and OHS may run on different machines connected via network





Online Monitor Presenter

File Actions Help

Start Stop Pause/Resume

CryoSci MuWall BC-2 BC-1 BC0 BC1 BC2-Profile TOF MuSci TrigSci Cheren Tile Browser

Histograms

- CommonProv
- Beam
- TileAuxProv
 - Laser
 - MuWall
 - Phantom
 - Scin
 - SC1
 - SC2
 - Sum
- TileProv

Command Draw

Cher1

Beam/Cher1	
Entries	10107
Mean	1043
RMS	282

Status bar

State: Running Particles: part_Tile_cb Server Name: Histogramming Histograms Received: 840 Rate: 0

- Actions:
- Save histograms in a ROOT file for offline analysis
 - Print window content
 - Pause/Resume application

Browser (or expert) mode:

histograms organized in folders



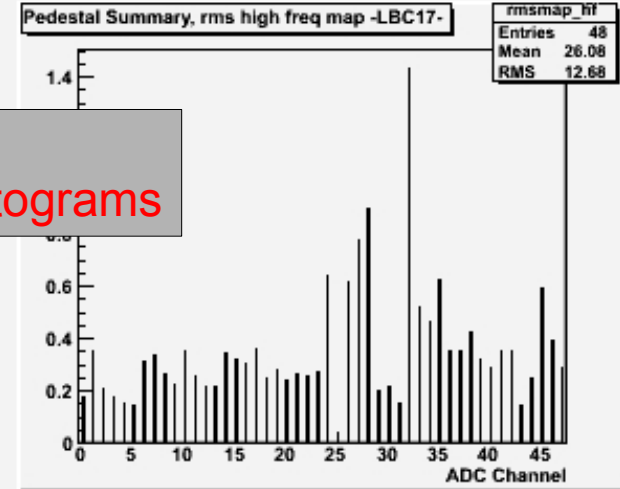
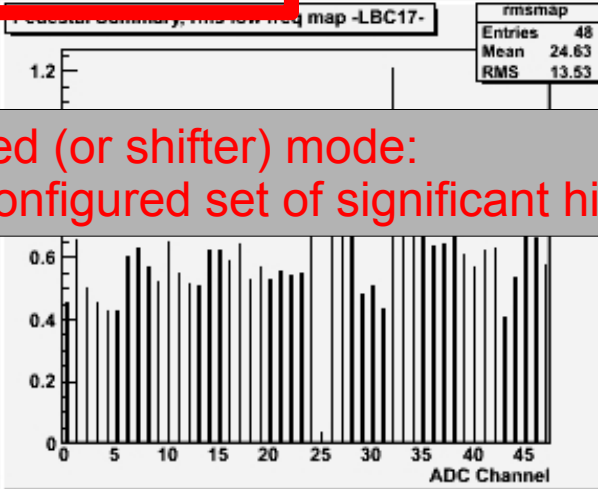
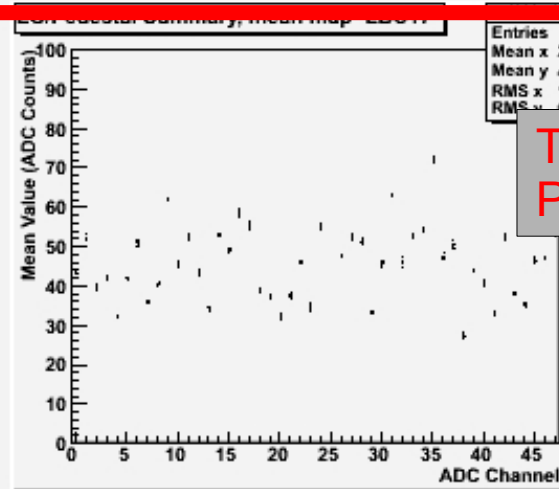
Ask Update

Start

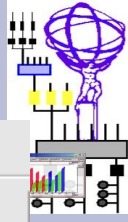
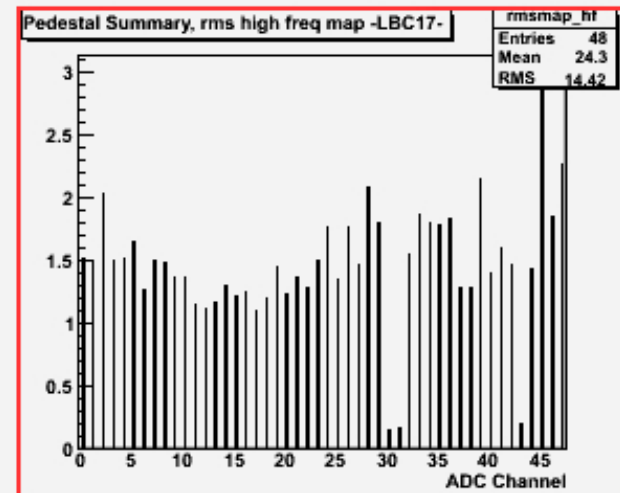
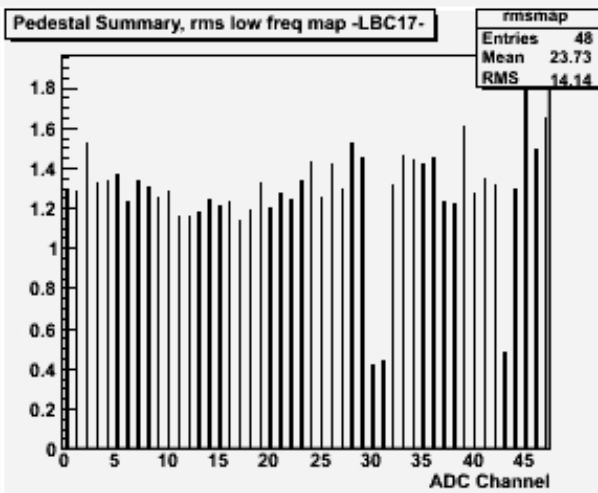
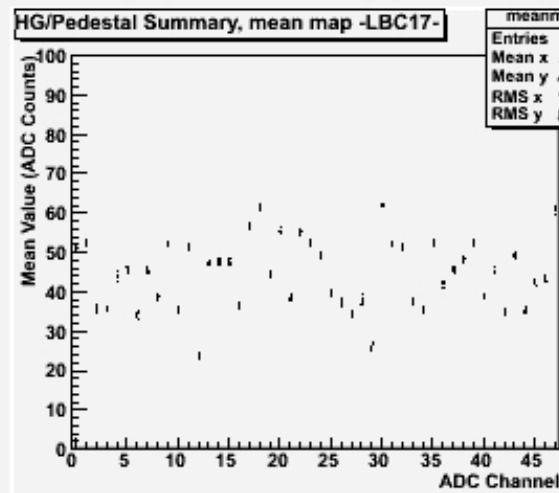
Stop

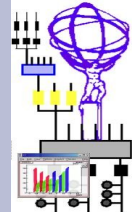
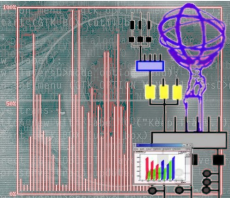
Pause/Resume

Dr5 PED Dr4 PED Dr3 PED Dr2 PED Dr1 PED Dr0 PED Browser



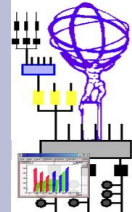
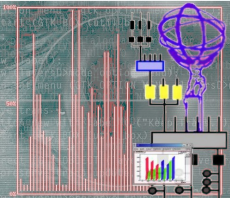
Tabbed (or shifter) mode:
Pre-configured set of significant histograms





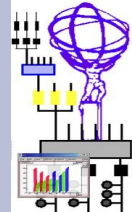
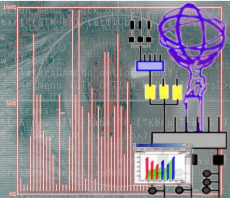
OHP GUI: details

- Histograms are received from OHS as TH1* (automatic conversion if they are produced in different format)
- We have chosen Qt to design GUI for two main reasons:
 - Qt can be interfaced easily to ROOT
 - Qt provides the “designer” tool that helped a lot in the GUI development phase (we are not GUI experts)
- QtROOT from GSI used to glue ROOT and Qt



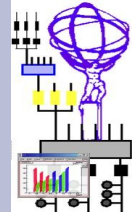
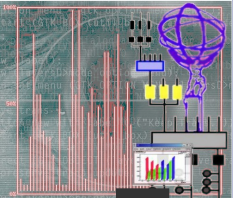
OHP GUI: details

- The GUI is a pure Qt application with embedded QtCanvas objects (wrappers to TCanvas)
- Histograms are drawn using `TH1::Draw()`, that's (almost) all our ROOT code in the GUI
- A double-click creates a pure TCanvas object and the histogram is drawn with `TH1::DrawCopy()`, the user can interact with it in a familiar context
- Middle mouse button has a special function in OHP: it opens a pop-up window to send cmds to OHS
- We plan to provide a context menu extending ROOT's one



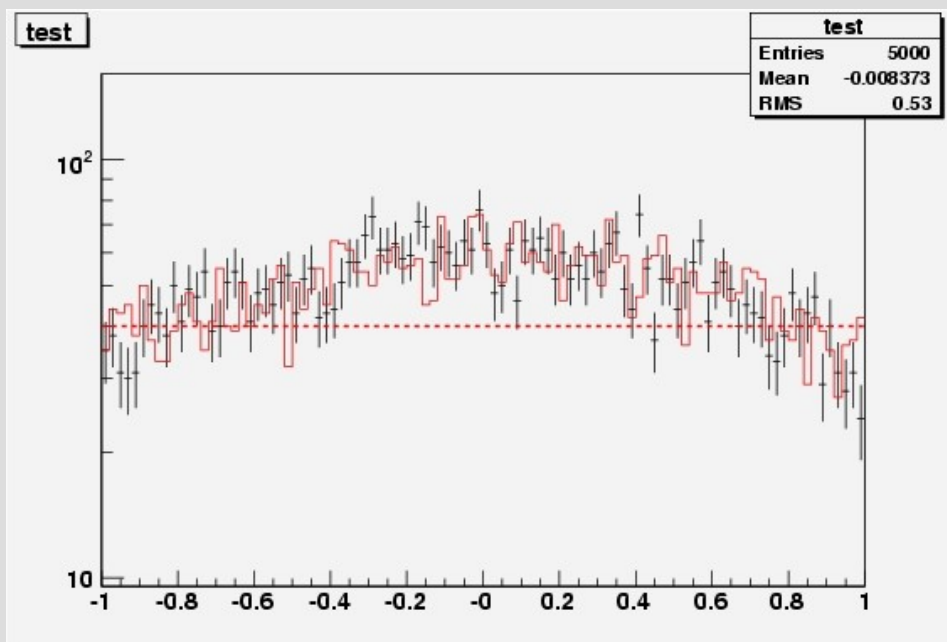
OHP peculiarities

- OHP can be configured through a configuration (ASCII, in the future XML) file
- Some histogram options may be set:
 - Draw options: argument to TH1::Draw(...)
 - Styles: a TStyle object is created and subsequently a TStyle::Execute(...) call is performed. Example: to set log scale on X: in conf file: SetLogx(1) causes: TStyle::Execute("SetLogx", "1")
- OHP can manage reference histograms (from TFile or OHS): reference histograms can be drawn on top of received histograms (through use of TH1::Draw("same") statements)

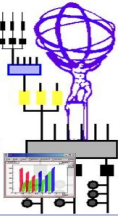
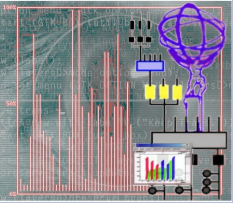


Features: Reference Histograms

- User can associate a reference histogram to an histogram in OHS that is displayed in the same window
- Reference histograms can be provided, locally, in a ROOT file or can be retrieved from the OHS

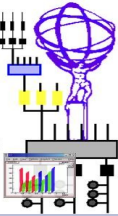
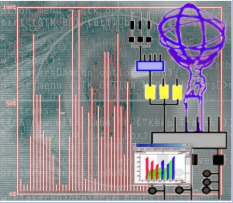


An Example of histogram with the reference histogram superimposed (in red). Graphic attributes (colours, lines, error bars) can be specified in the configuration



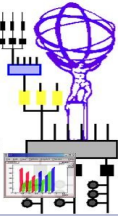
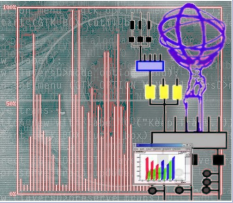
Future work

- OHP will support plugins to execute code on histograms (example: execute a fit). They'll be provided by users as shared libraries loaded at run time
- We are planning to change the GUI to support user developed “windows” (also provided as plugins) and to interact with non-graphical plugins



New GUI ideas

- We are still in a very preliminary phase of the design of the new GUI, many aspect will be probably changed in the future
- We are thinking to have a MDI application: a “desktop” with floating window, each window can be seen as a “plugin” displaying specific information



New GUI ideas

- We will provide 5 basic window (general usage):
 - Browser window (to browse OHS content)
 - Configurable set of histograms to be displayed as soon as they published on OHS
 - Same as before but for all histograms matching a regular expression
 - A Logging window to display messages/errors
 - A “status” window displaying information on the active setup (i.e. number of histogram producers, rates, ...)