Use of the ROOT framework in the KLOE experiment

W. Mei *
For the KLOE Collaboration 1

*Laboratori Nazionali di Frascati dell'INFN, Frascati, Italy.

Abstract

The data acquisition system of the KLOE experiment requires the filling of a large number of histograms and plots that are used to monitor the detector and the accelerator status as well as the quality of the acquired data.
A client-server system, based on the ROOT framework, has been developed to cope with this requirement. The server, *Kserver*, builds an online tree of histograms presenting the status of the detectors, front-end electronics and trigger system. The client, *Kbrowser*, which can be started in multiple copies in any monitoring node of the KLOE network, is used to browse and display selected histograms. *Kbrowser* is a general-purpose tool. In fact it is also used to browse Hbook histogram files generated by online processes that control the goodness of calibration constants and data quality.
The KLOE slow control system and the accelerator monitors produce ASCII files containing the history of the most significant parameters defining the status of DAΦNE and KLOE. A ROOT application, *KGeneral*, is used to display them both in online and offline mode.

Keywords: KLOE, ROOT, monitoring, online

1 THE KLOE COLLABORATION: M. Adinolfi, A. Aloisio, F. Ambrosino, A. Andryakov, A. Antonelli, M. Antonelli, F. Anulli, C. Bacci, G. Barbiellini, F. Bellini, G. Bencivenni, S. Bertolucci, C. Bini, C. Bloise, V. Bocci, F. Bossi, P. Branchini, S. A. Bulychjov, G. Cabibbo, A. Calcaterra, R. Caloi, P. Campana, G. Capon, G. Carboni, A. Cardini, M. Casarsa, V. Casavola, G. Cataldi, F. Ceradini, F. Cervelli, F. Cevenini, G. Chiefari, P. Ciambrone, S. Conetti, E. De Lucia, G. De Robertis, R. De Sangro, P. De Simone, G. De Zorzi, S. Dell'Agnello, A. Denig, A. Di Domenico, C. Di Donato, S. Di Falco, A. Doria, E. Drago, O. Erriquez, A. Farilla, G. Felici, A. Ferrari, M. L. Ferrer, G. Finocchiaro, C. Forti, A. Franceschi, P. Franzini, M. L. Gao, C. Gatti, P. Gauzzi, A. Giannasi, S. Giovannella, V. Golovatyuk, E. Gorini, F. Grancagnolo, W. Grandegger, E. Graziani, P. Guarnaccia, H. G. Han, S. W. Han, X. Huang, M. Incagli, L. Ingrosso, Y. Y. Jiang, W. Kim, W. Kluge, C. Kuo, V. Kulikov, F. Lacava, G. Lanfranchi, J. Lee-Franzini, T. Lomtadze, C. Luisi, C. S. Mao, M. Martemianov, A. Martini, M. Matsyuk, W. Mei, A. Menicucci, L. Merola, R. Messi, S. Miscetti, A. Moalem, S. Moccia, M. Moulson, S. Mueller, F. Murtas, M. Napolitano, A. Nedosekin, M. Panareo, L. Pacciani, P. Pages, M. Palutan, L. Paoluzi, E. Pasqualucci, L. Passalacqua, M. Passaseo, A. Passeri, V. Patera, E. Petrolo, G. Petrucci, D. Picca, G. Pirozzi, M. Pollack, L. Pontecorvo, M. Primavera, F. Ruggieri, P. Santangelo, E. Santovetti, G. Saracino, R. D. Schamberger, C. Schwick, B. Sciascia, A. Sciubba, F. Scuri, I. Sfiligoi, J. Shan, P. Silano, T. Spadaro, S. Spagnolo, E. Spiriti, C. Stanescu, G. L. Tong, L. Tortora, E. Valente, P. Valente, B. Valeriani, G. Venanzoni, S. Veneziano, A. Ventura, Y. Wu, Y. G. Xie, P. F. Zema, P. P. Zhao, Y. Zhou

1 The KLOE Online Data Acquisition System

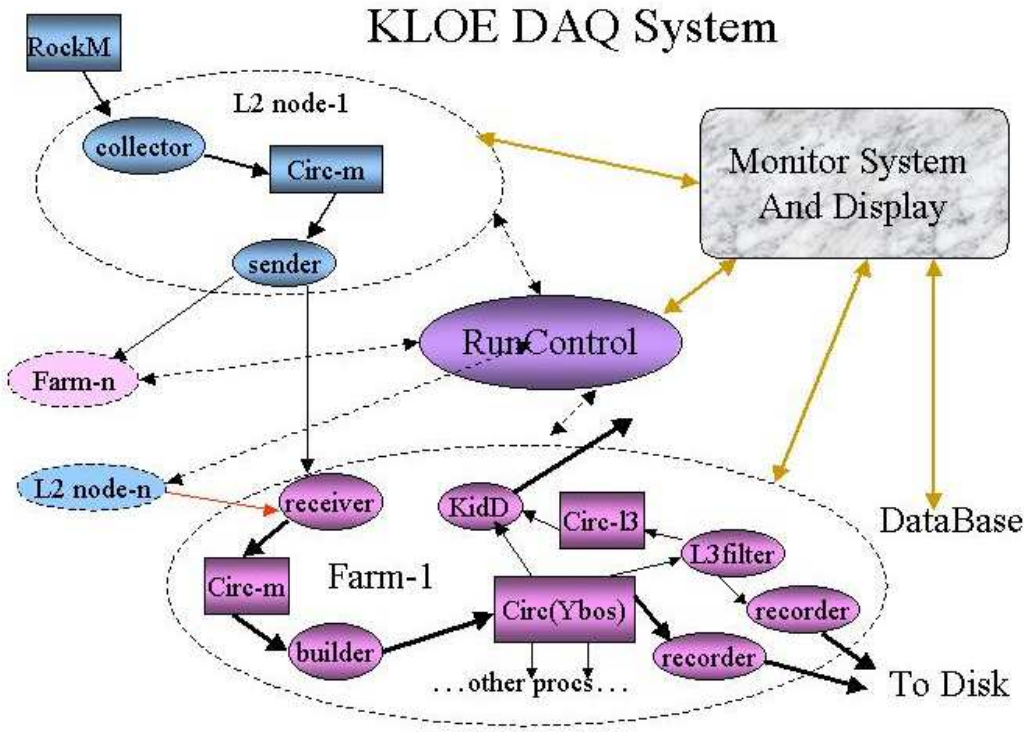

Fig. 1

The main physics goal of the KLOE experiment [1], at the DAΦNE Φ-factory in INFN Frascati laboratories, is the research for the direct CP-violation in the neutral kaon system. The KLOE data acquisition system [2] has been designed to sustain a rate of 10^4 events/s, a data throughput of 50 Mbytes/s and sufficient monitoring capabilities to keep detector and trigger functionalities under control.

A brief introduction of the KLOE DAQ system is shown in Fig. 1. The front-end electronic is distributed among 10 chains of interconnected VME crates. Data from a chain are collected in a custom board, *RockM*, that is readout by a VME CPU, *L2node*. The processes running on *L2node* are in charge of collecting sub-events from the chain and sending packets of them to the *farm* nodes using TCP/IP protocol through a switched FDDI network. Meanwhile the processes running on a *farm* node are in charge of receiving

sub-events from different *L2nodes*, validating and building them in integrated events formatted as YBOS banks, that are put in a large circular buffer (40 MB at the moment). *L3filter* peeks events from that buffer and selects different subsets of events that are written to different smaller circular buffers. Recorder processes takes the events from circular buffers and eventually save them to disk.

2. KLOE Monitoring System

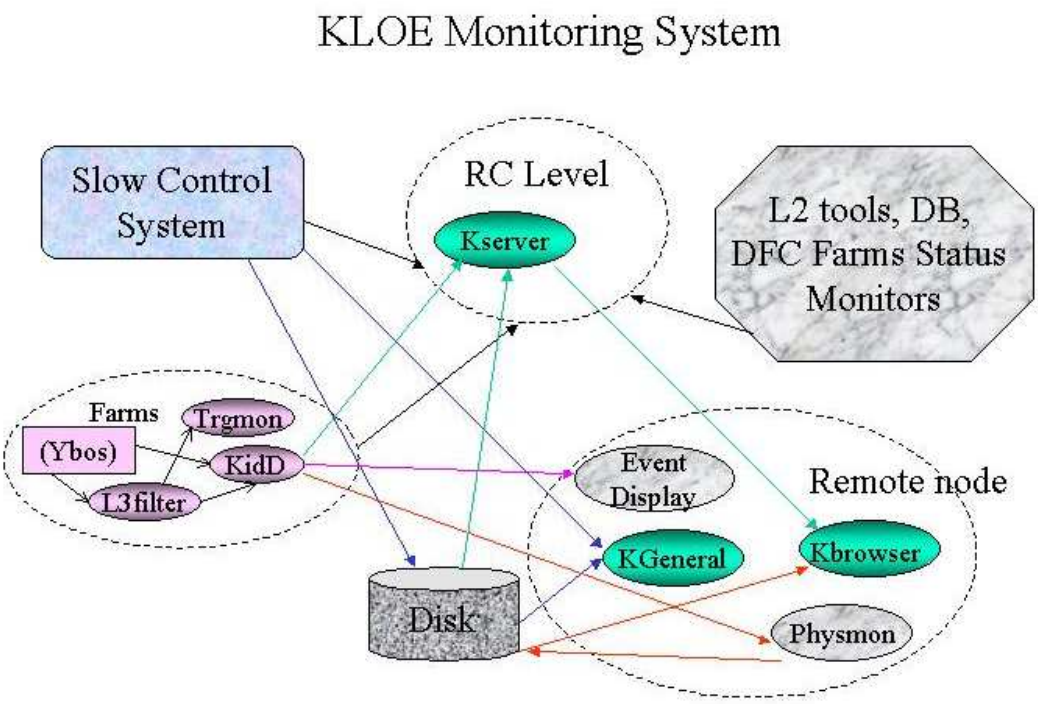

Fig. 2

Processes running in any node of the online system subscribe to a global memory a set of their most relevant variables and update frequently its values. These variables are exported by a private SNMP daemon allowing user to build centralized monitoring system of processes and run activity. Besides this, there are specialized monitoring tools running on each piece of the KLOE DAQ system. The KLOE monitoring system structure is shown in Fig. 2.

Kserver & Kbrowser is a ROOT based histogram server-client system for both online and offline monitoring. *Kserver* (KLOEserver) creates thousands

of histograms in ROOT format, organized in a tree structure according to physical or logical partitions (e.g. sub-detector, calorimeter sector, readout chain), and updates the histogram contents with the incoming data. *Kserver* saves the histograms at the end of run in a ROOT formatted file. The client ROOT based task, *Kbrowser* (KLOEbrowser) is in charge of browsing and displaying histograms. Working in online mode, *Kbrowser* opens a TCP/IP connection with the *Kserver*, retrieves streams of histograms depending on the user selection and displays them in different canvas windows. In offline mode, the data source is from ROOT/HBOOK file. In order to improve the monitoring capabilities of the *Kbrowser*, the overlapping with reference histograms is implemented as an option.

Different processes (e.g. *Trgmon*, KLOE and DAΦNE *slow controls*) updates periodically daily history ASCII formatted files containing a set of relevant parameters (e.g. DAΦNE currents and luminosity, gas pressure and temperature, magnetic field, trigger rates). *KGeneral* is a ROOT based tool that presents sets of plots and graphics from those files, either in online mode (last few hours status) or offline (up to several days history).

L2 tools can be activated on *L2nodes* and are essentially TCL/TK based debugging tools, not to be used during acquisition run, that allow to read/modify any register of the VME boards in a chain. Similar graphic tools are available to monitor the activity of processes running in the *farm* and the Data flow control node, *DFC*, or to present contents of tables in the online Database.

All acquired and filtered events are put in circular buffers before going to disk or being thrown away during monitoring runs. Those buffers are used as data source for local or remote monitoring processes, using the namely KLOE Integrated Dataflow, *KID*.

Trgmon, the trigger monitor, uses the contents of trigger banks to select in particular Bhabha scattering events in order to calculate DAΦNE luminosity, or to present trigger dead time, DAQ rates and more.

Physmon is the online physics monitor of data quality. Physmon gets data from the *L3filter* buffers and stores histograms in a run number related HBOOK file, which is updated every time enough statistics is accumulated. *L3filters*, a set of event filters selects interesting events and fills the output circular buffers that can be used as data source for other processes like Physmon.

3 The KLOE Histogram Monitoring System

The KLOE online histogram monitoring system is a server-client system using the ROOT framework and libraries. The server and the client named *Kserver* and *Kbrowser* are implemented as C++ compiled programs linked with ROOT version 2.23 on IBM AIX v4.3, HP-UX 10.2 and OSF1 4.0 operating systems.

More than 40 ROOT classes are used for histogram creation and filling, visualization, file access, network communication and GUI management. At the startup the *Kserver* creates a tree structure of histograms according to physical or logical partitions (Electromagnetic Calorimeter, Quadruple Calorimeter, Drift Chamber, Trigger, Front-End electronic). After that it enters a loop where the histograms are continuously updated with the incoming data. In each iteration, *Kserver* checks for network connections, new requests or commands and eventually performs one of the following actions:

- Sends the tree structure to a client
- Sends a single histogram content
- Send the contents of a set of histograms
- Reset the contents of all histograms
- Save all current histograms into a ROOT file
- Set the value of an internal parameter (e.g. enable/disable the filling of a tree branch)
- Send the value of internal parameters

Kserver is able to create histograms both from online data (using the KID tools) and from saved raw files. Running on online mode *Kserver* updates the histograms every time a certain number of events have been analyzed. At the end of a run a ROOT file can be created depending on the Run-control configuration, and all histograms in the buffer are reset at the beginning of the next run. IN offline mode, Kserver produces and serves its histograms getting data from a raw data file. The last feature is stream useful for debugging purposes.

Kbrowser, working as a window explorer, is the client task that is able to run on any remote monitoring node; works either in online or offline mode selected by the user at the beginning of the activity and then browses and

displays the selected histograms.
Working in online mode, *Kbrowser* opens a TCP/IP connection with a remote node where a *Kserver* is running, gets the current histogram tree structure and put it in the tree window of the graphic interface. The first level branches correspond to the interested physical or logical partitions (Electromagnetic Calorimeter, Quadruple Calorimeter, Drift Chamber, Trigger, Front-End electronic). Browsing a branch step by step, the user can select histograms related to specific sectors of the sub-detectors. The histograms can be displayed in a number of pads organized in pages of canvas or one histogram per canvas mode.
If the user requires the overlap option while analyzing histograms related to a given sub-detector, a reference histogram file is used. Any new histogram is overlapped with a reference one. A set of global histograms has been built in order to help online checks of the working status of the KLOE detector.
Kbrowser is a more general tool that works in a standalone mode as a general histogram browser for KLOE. For instance the physics monitor, *Physmon* process obtains events from *L3filter*, analyzes them and writes histogram files in HBOOK format. The file is updated every time enough statistics is accumulated during the real data taking. *Kbrowser* and its overlap option allow the user to check continuously the data quality during online shifts.
In offline mode, Kbrowser can navigate over both ROOT and HBOOK formatted files. If reference histogram files exist the overlap option can always be used.
In the single histogram display mode *Kbrowser* also refits the selected histogram if it has been fitted before and display the fit parameters in the canvas. The histogram building process like *Physmon* also fits a lot of histograms for the physics check, but during the conversion from HBOOK to ROOT format the fit parameters are thrown away. *Kbrowser* contains an array of functions and parameters related to the reference histograms to refit the histograms in order to represent the parameters.

4. An Example of the Use of the KLOE Histogram Monitoring System during DAQ

1) “start_kserver” command activates a shell script starting *Kserver* on the run control node.
2) “start_kbrowser” command activates a shell script starting

Kbrowser on the monitoring node. The user will get a window like Fig. 3

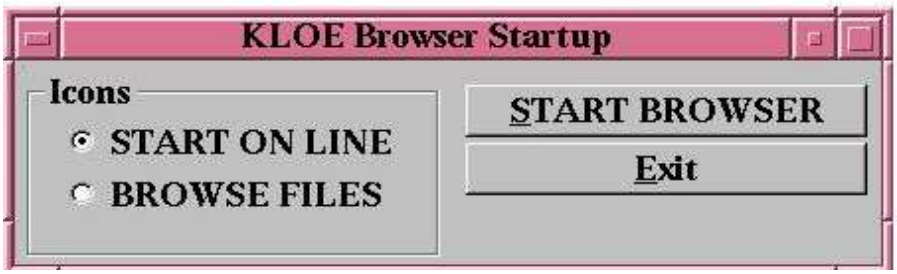

Fig. 3

3) After the mode selection, the user will get the main window of Kbrowser shown in Fig. 4

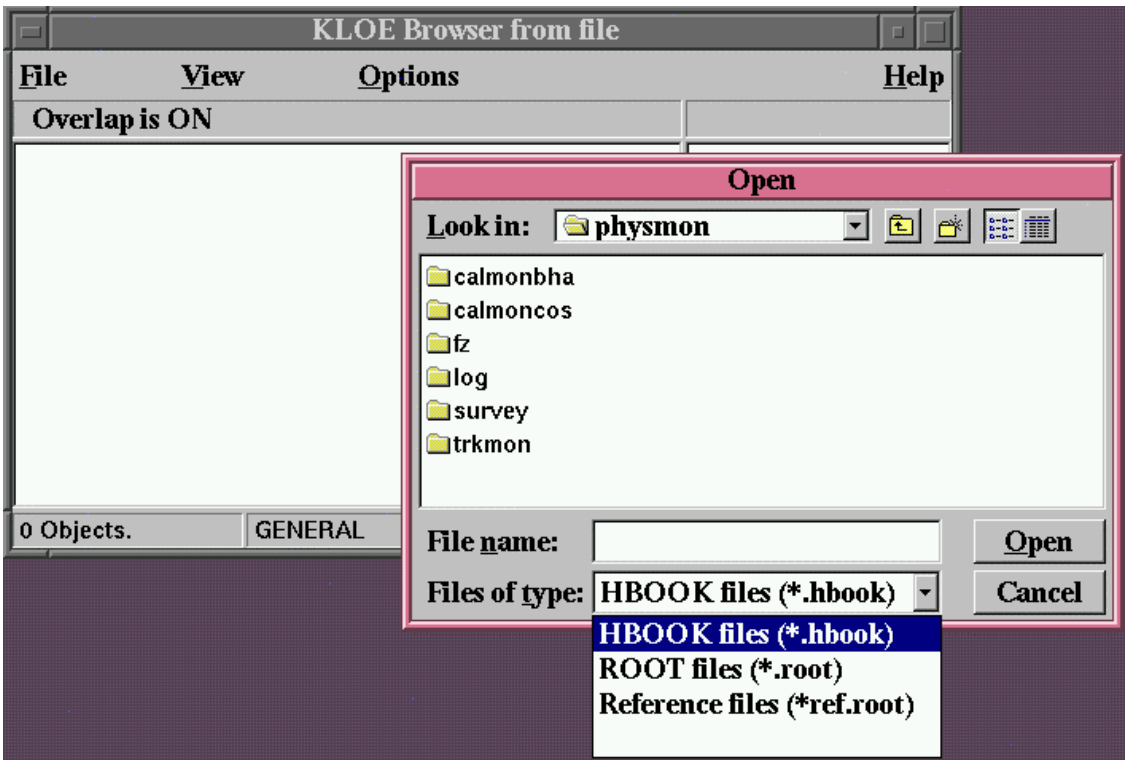

Fig. 4

From the "Options" item of the menu bar the user has a choice to switch on/off the overlap mode. After that, in the "File" item of the menu bar, the user can activate a file dialog interface where a file

either in ROOT f or HBOOK format will be opened to be browsen.

4) Browsing the file by following branches the user can find all the interested histograms. (Fig. 5)

5) Choosing the "Draw Selected" option in the "View" item, the histograms can be displayed either in the mode of Fig. 6 or in the mode of Fig. 7 if only one histogram is selected. The displayed histograms are normalized with their references. The histogram of Fig. 7 has also been refitted and the fit parameters have been displayed in the status box, as well as the same fit results of its' reference histogram.

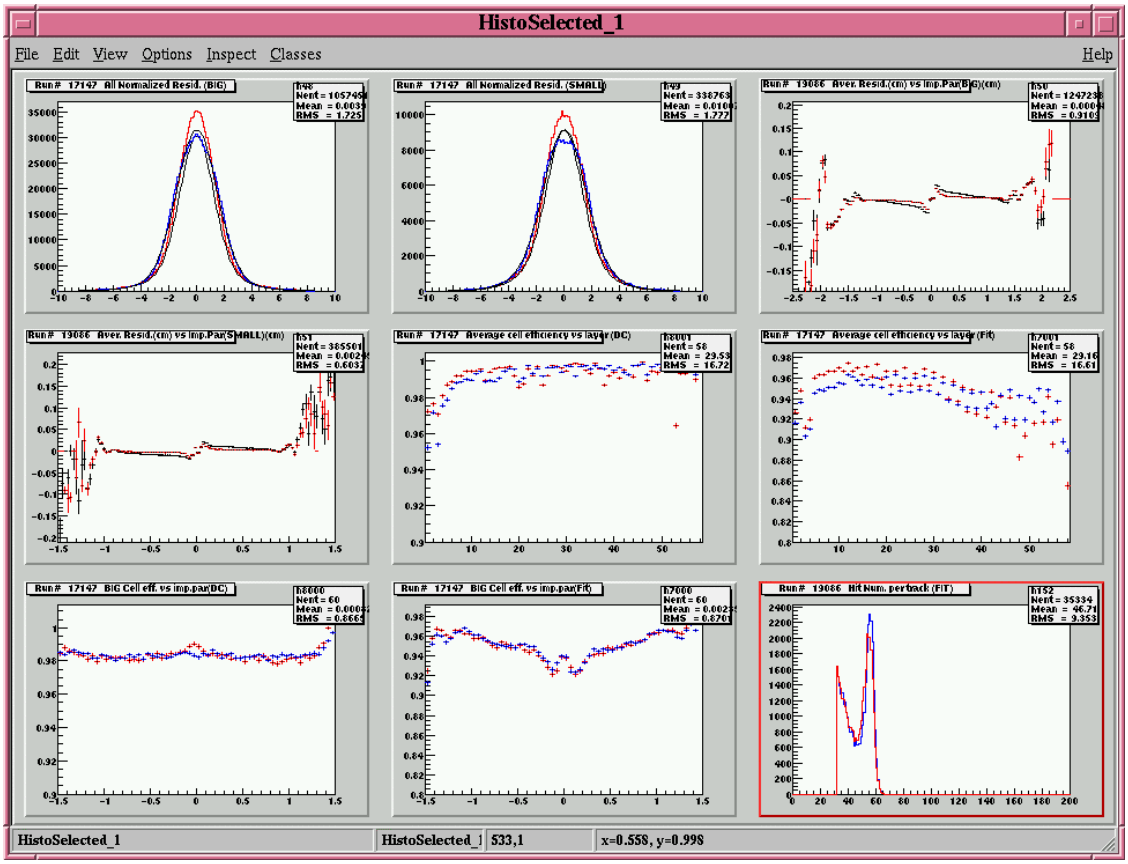

Fig. 6

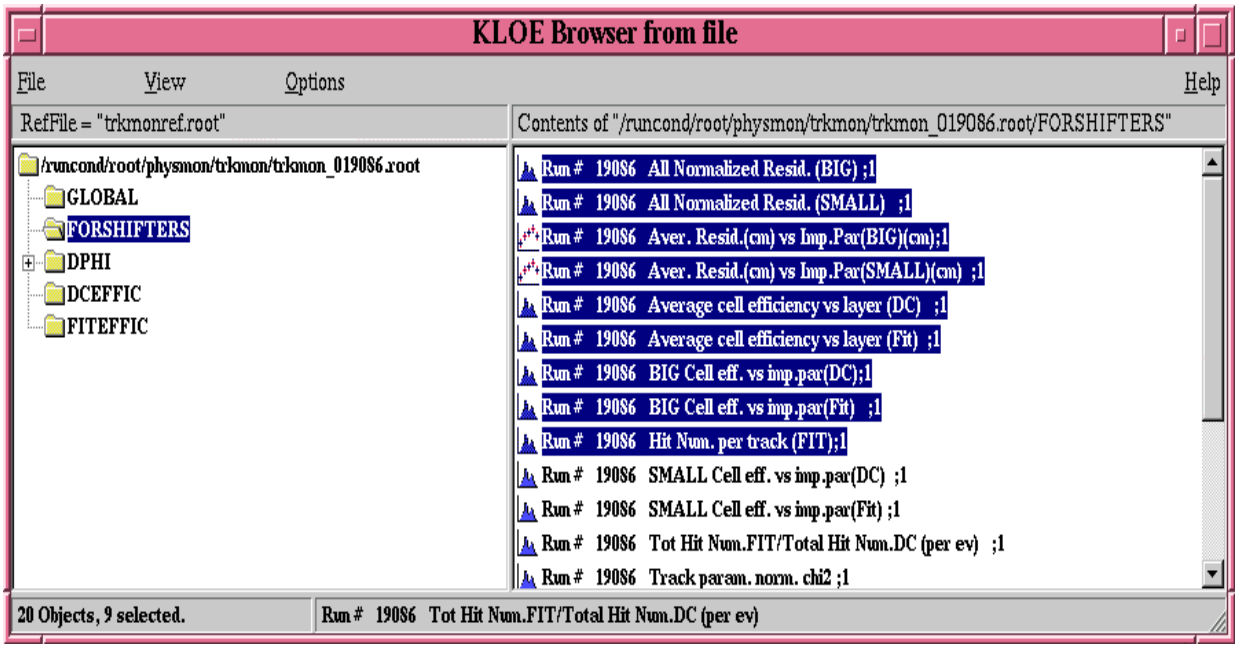
KLOE Browser from file
File
View
Options
Help
RefFile = "trkmonref.root"
Contents of "/runcond/root/physmon/trkmon/trkmon_019086.root/FORSHIFTERS"
/runcond/root/physmon/trkmon/trkmon_019086.root
GLOBAL
FORSHIFTERS
DPHI
DCEFFIC
FITEFFIC
Run # 19086 All Normalized Resid. (BIG) ;1
Run # 19086 All Normalized Resid. (SMALL) ;1
Run # 19086 Aver. Resid.(cm) vs Imp.Par(BIG)(cm);1
Run # 19086 Aver. Resid.(cm) vs Imp.Par(SMALL)(cm) ;1
Run # 19086 Average cell efficiency vs layer (DC) ;1
Run # 19086 Average cell efficiency vs layer (Fit) ;1
Run # 19086 BIG Cell eff. vs imp.par(DC);1
Run # 19086 BIG Cell eff. vs imp.par(Fit) ;1
Run # 19086 Hit Num. per track (FIT);1
Run # 19086 SMALL Cell eff. vs imp.par(DC) ;1
Run # 19086 SMALL Cell eff. vs imp.par(Fit) ;1
Run # 19086 Tot Hit Num.FIT/Total Hit Num.DC (per ev) ;1
Run # 19086 Track param. norm. chi2 ;1
20 Objects, 9 selected.
Run # 19086 Tot Hit Num.FIT/Total Hit Num.DC (per ev)

Fig. 5

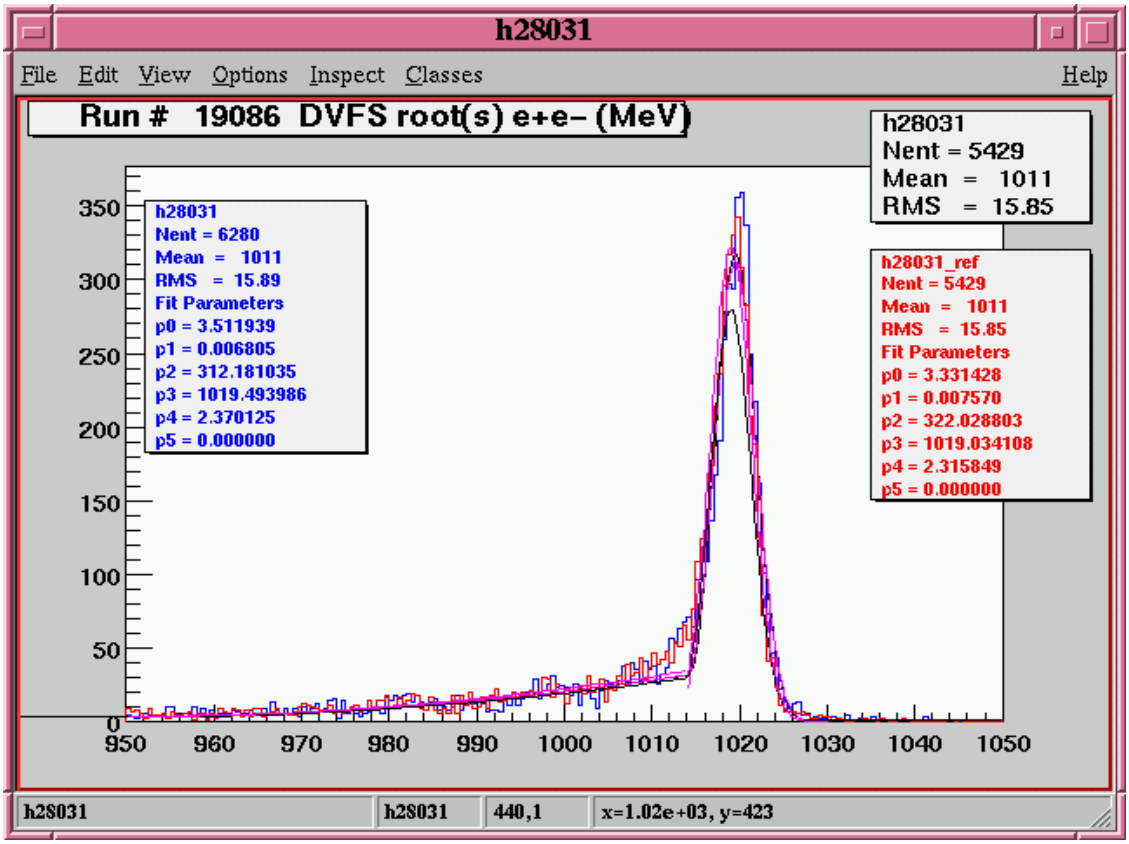
h28031
File
Edit
View
Options
Inspect
Classes
Help
Run # 19086 DVFS root(s) e+e– (MeV)
h28031
Nent = 5429
Mean = 1011
RMS = 15.85
h28031
Nent = 6280
Mean = 1011
RMS = 15.89
Fit Parameters
p0 = 3.511939
p1 = 0.006805
p2 = 312.181035
p3 = 1019.493986
p4 = 2.370125
p5 = 0.000000
h28031_ref
Nent = 5429
Mean = 1011
RMS = 15.85
Fit Parameters
p0 = 3.331428
p1 = 0.007570
p2 = 322.028803
p3 = 1019.034108
p4 = 2.315849
p5 = 0.000000
350
300
250
200
150
100
50
0
950
960
970
980
990
1000
1010
1020
1030
1040
1050
h28031
h28031
440,1
x=1.02e+03, y=423

Fig. 7

5. The KLOE and DAΦNE General Status Presenter

The overall status of the KLOE detector is periodically updated by monitoring tools running in different nodes of the KLOE network and in particular in the KLOE slow control system. ASCII formatted files are used to maintain such a history. The KLOE slow control also integrates the relevant machine parameters coming from the DAΦNE control system. All these quantities can be displayed by the presenter, *KGeneral,* as time chart histograms using the ROOT libraries.

The presenter can run on any node of the KLOE monitoring system. The main program is *KGeneral*. It is a unique application written using the C++ language and the ROOT v2.23 analysis environment.
The main functions of the KGeneral program are:
-- to read the appropriate data files and to create an ntuple structure;
-- to create different time charts of the monitored parameters;
-- to handle the user interface for plot requests;
-- to display the user requested plots.

The user interface of the *KGeneral* is quite friendly and easy to learn. Following is an example. (Fig. 8 and Fig. 9)

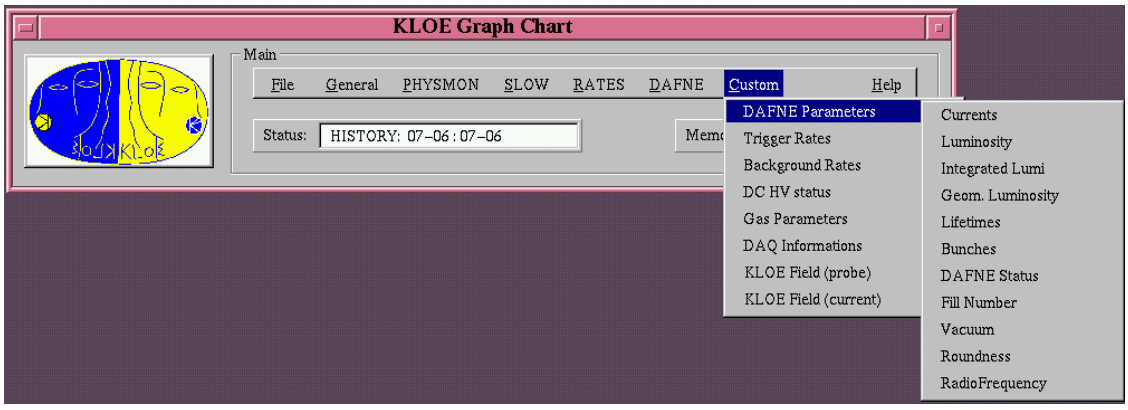

Fig. 8

1) "pres" command activates a shell script starting *Kgeneral* on a remote monitoring node and the user will get a window like Fig 8.

2) From “File” item of the menu bar the user can choose to monitor the current online parameters or to review the offline histories. After that the user can choose the interesting parameters from the rest items of the menu bar as indicated in Fig. 8 and the related time charts are displayed in a canvas as shown in Fig. 9.

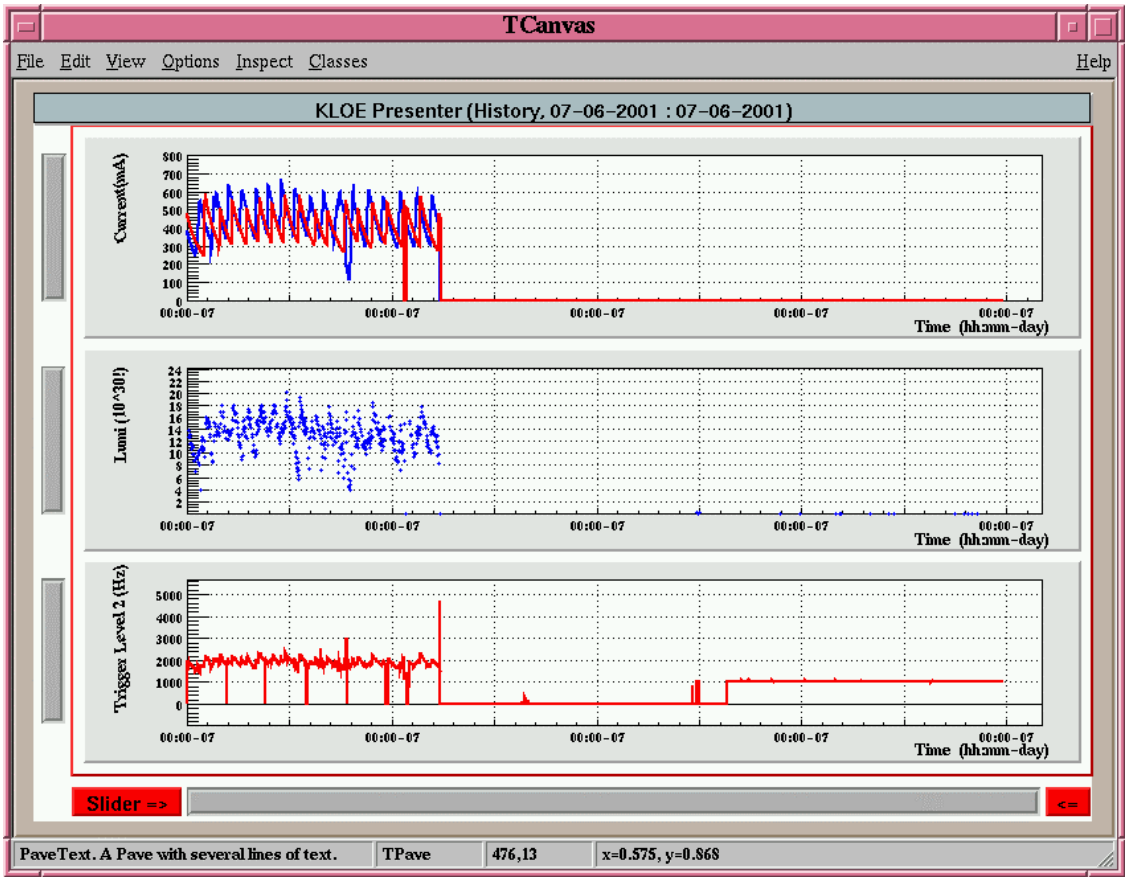

Fig. 9

6. Problems Related with ROOT

Two considerations are described in the following:

1) HBOOK to ROOT conversion loses message of fit parameters

As we introduced above the browsing of the HBOOK formatted files created by the physics data quality monitoring process is one of the most important goal of the *Kbrowser*. H2root function is called by *Kbrowser* to convert HBOOK formatted files to ROOT formatted files. But the lose of the fit parameters during the H2root conversion make the ROOT histogram less significant. In order to represent the parameters, *Kbrowser* contains an array of functions and parameters related to the reference histograms to refit the histograms. But refit hundreds of histograms in ROOT environment with Kbrowser is a big job.

2) Hard to follow up the frequently updated ROOT version

Kserver, *Kbrowser* and *KGeneral* tools are very complex and have been developed by many experts of the KLOE collaboration since 1996. Each expert might work on only some pieces of the software and could have left after his/her work finished. ROOT version frequently changed brought on a question that different processes are created with different versions of ROOT. For instance *Kserver* was released earlier with ROOT version 2.20, but *Kbrowser* and *KGeneral* were released with ROOT v2.23. We tried to compile *Kserver* with ROOT v2.23 but it did not work, in particular due to changes on memory allocation functions. So we have to keep two default versions of ROOT. We will try to use for any KLOE application the new ROOT Version 3.00 and we hope this will remain stable.

References

[1] The KLOE Collaboration, KLOE, A General Purpose Detector for DANE, LNF-92/019(IR), April 1, 1992;

[2] The KLOE Collaboration, The KLOE Data Acquisition System, Addendum to the Technical Proposal, LNF-95/014(IR), March 29, 1995;

[3] A. Doria for KLOE Coll., Online Monitoring System at KLOE, Proceedings of CHEP 2000, Padova;

[4] A. Doria for KLOE Coll., A ROOT based monitoring system for the KLOE Experiment, Proceedings of AIHENP 1999, Crete;

[5] E. Pasqualucci for KLOE Coll., Process and Data Flow Control in KLOE, Proceedings of CHEP 2000, Padova;

[6] I. Sfiligoi for KLOE Coll., Data Handling in KLOE, Proceedings of CHEP 2000, Padova;

[7] J. Snow etc. al., Use of ROOT in the D0 Online Event Monitoring System, Proceedings of CHEP 2000, Padova;

[8] W. Carena etc. al., the Usage of ROOT for Online Monitoring in the ALICE DATE System, Proceedings of CHEP 2000, Padova;

[9] M. Martemianov etc. al., The KLOE online presenter, KLOE memo #236, April 2001