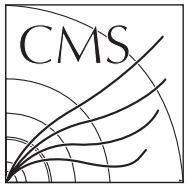
Available on CMS information server

CMS CR -2009/345



The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

17 November 2009

The CMS Tracker Data Quality Monitoring Expert GUI

Francesco Palmonari for the CMS Collaboration

Abstract

The CMS Tracker data quality monitoring (DQM) is a demanding task due the detector's high granularity. It consists of about 15148 strip and 1440 pixel detector modules. About 350,000 histograms are defined and filled accessing information from different stages of data reconstruction to check the data quality. It is impossible to manage such a large number of histograms by shift personnel and experts. A tracker specific Graphical User Interface (GUI) is developed to simplify the navigation and to spot detector problems efficiently. The GUI is web-based and implemented with Ajax technology.

We will describe the framework and the specific features of the expert GUI developed for the CMS Tracker DQM system.

Presented at *IEEE2009: 2009 IEEE Nuclear Science Symposium and Medical Imaging Conference*

The CMS Tracker Data Quality Monitoring Expert GUI

F.M.Palmonari, (*INFN Pisa*) S.Dutta, (*University of Wisconsin & CERN*)

***Abstract*—([1]) Data Quality Monitoring (DQM) is mandatory in a high energy physics experiment like the Compact Muon Solenoid (CMS) at the CERN Large Hadron Collider (LHC). For the CMS Silicon Strip Tracker, the largest detector of this kind ever built, the DQM tasks translate into monitoring of 15148 silicon detector modules for a total of 9.3 million channels. An expert level Graphical User Interface has been developed to manage this highly granular information where the statistical characteristics of more than 350000 histograms are analyzed online during data-taking in order to raise warnings or alarms. In addition, specific quantities can be recalled on demand for problems investigation or online feedback on the data quality.**

***Index Terms*—CMS, HEP, Silicon Strip Detector, Data Quality Monitoring, GUI, Ajax.**

I. Introduction

The Silicon Strip Tracker (SST) [1] of the CMS experiment is made of 15148 silicon strip detector modules which correspond to 9.3 million electronic channels for a total active area of 200 m^2. The Data Quality Monitoring (DQM) system for such a complex detector must ensure that any possible problem is identified efficiently at a very early stage of the data acquisition process so that actions can be taken promptly. In CMS, the DQM system is based on the CMS software framework CMSSW [3]. The general purpose analysis framework, ROOT [4] is used for histogramming. The monitoring task in CMS is fulfilled in three main steps,

- **DQM Producers (source) :** The producers are pluggable modules where histograms are booked and filled by retrieving necessary information from CMS event data at various levels of reconstruction. Depending on the type of information to monitor a number of such modules are used.
- **DQM Consumer (client) :** The DQM consumer application accesses the histograms already defined and filled by the producers and performs further operations on them. A number of analyses are performed, namely (a) creation of summary histograms by combining a group of logically connected histograms, (b) execution of Quality Tests (QTest) where statistical tests are performed to check if there is any deviation from the expected behaviour and accordingly the alarms are generated, (c) creation of synoptic views of the detector for better understanding of problem. The consumer level operation is especially important for the SST DQM system which is conprised of ~ 300 K histograms and is impossible to inspect individually. Finally all histograms and analysis results are archived in a standardized output format for further use.
- **DQM Graphical User Interface (GUI) :** The histograms and the analysis results created in producer and consumer applications are visualised here. We have a general purpose GUI available from the CMS DQM system that fulfill the basic need of the whole experiment and there is a dedicated one specially designed for the SST experts and shifters. Both these GUI applications in CMS are web based.

We are going to describe the details of the Tracker expert GUI in the following sections of the article.

[1] *Corresponding author: F.Palmonari, e-mail:francesco.palmonari@cern.ch*

II. The Tracker Expert GUI

The histograms created by the producer and consumer are visualised through the web based Graphical User Interface. The interface is based on a basic Asynchronous JavaScript and XML (Ajax) pattern [7] which matches desktop applications in terms of interactivity and simplicity. The advantage over a desktop application is that the interface is accessible from anywhere a modern web browser is available and no specific software installation is required. This is very important in large experiments like CMS where users spread around the world. The DQM consumer is a part of Cross-Platform DAQ Framework (XDAQ) [5], [6] and can act as a web server. The client-server communication is handled by an Ajax engine. The web interface is fully loaded only once and subsequently just the relevant parts of the interface are updated following different actions by the users. This model is in sharp contrast to the traditional approach where the browser reloads the full page on each action. Although there are relatively more communication between web server and browser, the time needed to send a request and receive the response is significantly smaller. The responsibilities of the web server and web interface are well defined and the band width is properly used. Different components of the SST Web GUI are described in Figure 1. Any user request through the web interface is intercepted by Javascript and passed to the Ajax engine after forming the proper request. The Ajax engine in turn sends the request to the web server, which is the DQM consumer in this particular case. Depending on the specific request one or a number of actions are executed in the back-end action performer of the DQM consumer. The response, which can be in the form of image, XML, html or plain text, is sent back to the Ajax engine by the web server; the response is then interpreted and relevant parts of the interface are updated. The web interface is designed using a number of widgets to fulfill our needs. The

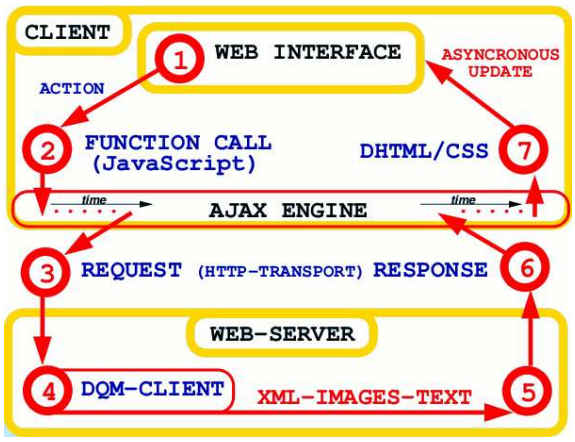


Fig. 1. Different components of the SST Web GUI is described in a block diagram here describing how the communication actually takes place.

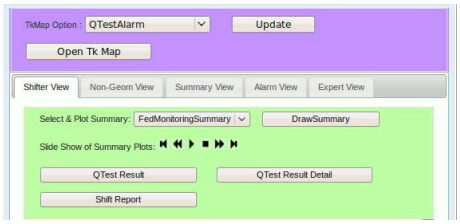


Fig. 2. The SST expert GUI client-controls widgets are shown here.

jQuery JavaScript library[8] has been used extensively. The web interface design and the corresponding development has been a continuous process aimed at integrating both the expert requests on one side and the most up to date web technology on the other.

III. Different Features of the SST expert GUI

The GUI design relies heavily on the needs of the expert and shifter usage of DQM to understand the detector performance in a fast and reliable way. The interface design is optimised to ease the problem spotting which is difficult due to the size and complexity of the SST. The dynamically generated web page containing the SST expert GUI is divided in three regions: (a) the **client-controls**, (b) the **drawing canvas**, a dedicated window where the images of the user-selected histograms are shown, and (c) a **message window** where plain text or html responses from the server appear. In Figure 2 the **client-controls** window is shown that consists of options to create two dimensional synoptic views of the tracker and five different tabs corresponding to different actions to be performed by SST shifters and experts.

A. Shifter View

This tab is intended for the SST shifters. It allows easy access of DQM histograms grouped in several categories. Shifters can select them from the drop down menu or can use the slide-show option to view them automatically. One can also access a list of detector modules that failed during a QTest and further select detailed or summarised information. In this case the server response is html and the **message window** gets updated. A snapshot of this view is reported in Figure3.

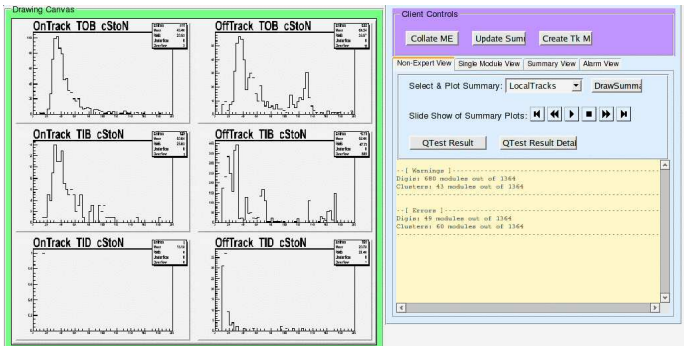


Fig. 3. The shifter view panel with different action widgets provided for shifter usage.

B. Summary View

The summary histograms created by the DQM consumer can be accessed in this tab. Different types of summary histograms are created by the DQM consumer, namely, mean values of certain quantities from individual detectors combined side by side, distributions added together to create global summary, etc. These summary histograms are created at different levels of the geometrical structure. One can select a specific layer or wheel of the SST from a drop down menu and request the summary histograms corresponding to that part of the detector. The server response is interpreted as a tree like structure and displayed in the **message window**. This eases navigation through the DQM folder structure to access summary histograms at each level of the geometrical structure to follow a problem if it is found at the level of layer or wheel. At a given level all histograms are received as images through a single request and plotted as small icons which can be zoomed in for better inspection.

C. Alarm View

This view provides a better understanding of the data quality where the QTests results are presented following the SST geometrical structure and allows to trace back all the faulty modules easily. As described in the Summary View III-B, one can also select a specific layer or wheel of SST and the full alarm tree is returned by the server and displayed in the **message window**. The QTest results like “Ok”, “Warning” and “Error” are represented in “Red”, “Yellow” and “Green” colors respectively at each level of the tree after combining the results from all histograms in that layer and considering the results from the lower levels. This is extremely useful to spot faulty detector modules from this tree as one can decide whether to scan down the folder structure just by checking the color code of the top most level. Once a faulty detector is spotted one can select it to plot the relevant histograms where QTest was applied and can also get the message containing the detailed result of the QTest as simple text. A snapshot of a *Alarm view* is reported in Figure 4.

Fig. 4. The alarm view panel showing the alarm tree for a specific layer of SST. In the **drawing canvas** two histograms are shown where one of them passed the QTest but the other one failed. The result of the rest is displayed in the **message window**.

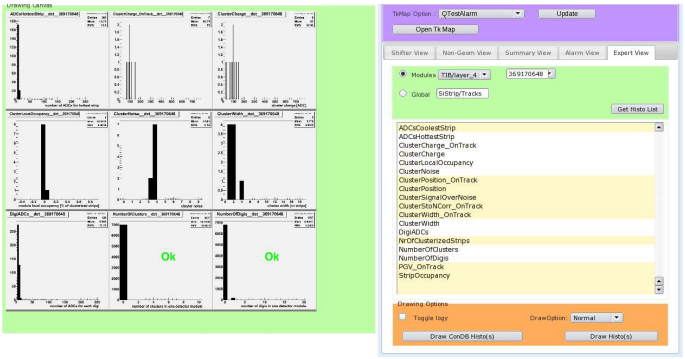


Fig. 5. The expert view panel.

D. Expert View

Any problem identified in the Summary or Alarm Views III-B,III-C can be further investigated in this tab. All histograms defined for the detector modules can be accessed here. On request, the list of detector modules for a given layer or wheel of SST, where a problem has been detected, can be obtained. The server also sends the list of histograms associated to detector modules and the **message window** gets updated accordingly. One can then select the faulty module and corresponding histograms are visualized on the **drawing canvas** on a subsequent request. Different components of this tab is illustrated in Figure 5.

E. TrackerMap

A simple and very useful tracker specific tool called the "Tracker Map" has been developed [9] within the SST where the whole tracker is represented in a two dimensional synoptic view as shown in Figure 6. In the "Tracker Map", the SST barrel layers are presented in flattened rectangular shapes and end-cap wheels are presented as disks in Scalable Vector Graphics (svg) [10] image. Detector modules in each layer or wheel are presented as small pixels in these rectangular or disk areas. Monitoring information of each detector module can be attached to the image which is active and appropriate events are attached to send request to the server to get histogram images. The tracker map is painted with QTest result or other

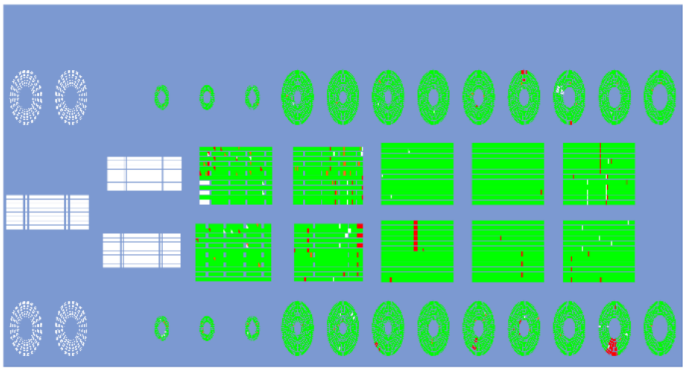


Fig. 6. Example of the Tracker Map image representing the barrel layers and the end-cap discs in a 2-D view where detector modules are represented by pixels. The image is created with QTest results where detector modules failed in QTest are presented in Red and the ones passed the test are presented in Green.

information and the options can be selected dynamically by the respective widget buttons in the GUI.

IV. Conclusion

The SST DQM application is being used regularly during extensive cosmic data taking schedules in CMS. It is providing useful information to the SST community to commission the detector efficiently. The SST Expert GUI is being used by the shifters and the experts to pin point problems which is the main purpose of the tool. We are gearing up for LHC collision period and we are sure that this toll will be very useful as it is now.

Acknowledgment

The authors would like to thanks Subir Sakar (now working for INFN, Pisa) for his precious advices and help during the SST expert GUI software engineering and development. In addition we would like to thanks Dario Menasce (now working for the University of Milan, Bicocca) that provided useful suggestions and working examples during the development of the Web Interface.

References

[1] The Tracker Project Technical Design Report, CERN/LHCC 1998-6, CMS TDR 5, Addendum CERN/LHCC 2000-016
[2] S. Dutta, "The Data Quality Monitoring of the CMS Experiment: the Tracker Case", presented at the IPRD-08, 12th Topical Seminar on Innovative Particle and Radiation Detectors, Siena, Italy, to be published in Nuclear Physics B Proceedings Supplement
[3] CMS Physics Technical Design Report, Vol.1, "Detector Performances and Software", CERN/LHCC/2006-001, CMS TDR 8.1,2 February 2006
[4] "ROOT, An Object Oriented Data Analysis Framework", *http://root.cern.ch*
[5] The Trigger and Data Acquisition project, Volume II, CERN/LHCC/2002-26, CMS TDR 6.2, 15 December 2002
[6] http://svnweb.cern.ch/trac/cmsos
[7] "Ajax:A new Approach to web Applications", James Garrett, 2005, http://www.adaptivepath.com/ideas/essays/archives/000385.php
[8] http://jQuery.com
[9] Use of Interactive SVG Maps and Web Services to Monitor CMS Tracker, M. S. Mennea, G. Zito, Proceedings of IEEE-NPSS Real Time Conference, 2005 June 4-10, Stockholm
[10] *http://www.w3.org/Graphics/SVG*