September 19, 2007
Version 1.0

Online DQM Baseline Architecture

Summary of DQM Workshop 27 August 2007
http://indico.cern.ch/conferenceDisplay.py?confId=19909

A. Meyer

DESY, Hamburg

with contributions from:
D. Acosta, L. Agostino, W. Badgett, J. Berryhill, H. Cheung, G. Franzoni,
M. Klute, A. Korytov, K. Maeshima, E. Meschi, L. Tuura, P. Wittich, M. Zanetti

Abstract

The note is based on input provided at the DQM Workshop Meeting 27.8.07, as well as from a number of informal meetings with DQM and subsystem experts. The plan for the central CMS online Data Quality Monitoring System is summarized. The baseline goals, constraints and limitations are described.

1 Introduction

The CMS DQM software framework provides tools for the creation, transport and manipulation of monitoring elements (ME), i.e. histograms. The framework is based on CMSSW and root. ”Source” modules are used to create and fill monitoring elements with event level quantities. The monitoring elements are periodically fed into “client” modules for further processing, e.g. fitting, determination of means etc. A central function of the client modules is to perform quality tests on the incoming monitoring elements (ME). The quality test algorithms as well as facilities to attach the test results to the ME are provided by the DQM framework.

A requirement to the DQM framework is that it can be operated both in an online and offline environment and works on both input from file as well as on live online data. The portability of the code is particularly important for the purpose of developments and tuning.

The DQM infrastructure is foreseen to be used for online detector monitoring as well as for the monitoring of HLT filtering and physics reconstruction. While the developement of detector monitoring for the different subdetector components is rather advanced, the development of DQM tools for use at Tier-0 (and possibly downstream at Tier-1 and Tier-2) is just starting. It is foreseen that DQM will also be used for calibration jobs.

Online DQM

Online DQM applications receive events at two different levels:

1. **DQM Sources in the HLT Filter Units:**
   The HLT provides access to all events accepted by the L1-trigger, at a rate of up to 100 kHz. The main purpose of the HLT being the filtering of events, the CPU usage by the DQM source modules must be limited to a minimum. In addition the HLT code must be absolutely stable and robust to guarantee the uninterrupted event-filter operation. The monitoring elements (ME) are shipped out to the Storage Manager at the end of luminosity sections. The total maximum bandwidth for shipping out data (events and DQM ME) from the filter units to the Storage Manager is 1 GB/s. The SMProxyServer DQM server collates the ME collected from the FU, and provides access to the ME through the SMProxyServer. The SMProxyServer saves the DQM data to files and sends them to Tier-0 for archival.

2. **DQM Sources operating off the Storage Manager Event Server:**
   The Storage Manager Proxy Server provides an event server which delivers events at the rate of order 1-10 Hz. DQM “consumer” sources connect to the event server and retrieve events through http-request. While the event rate is rather small the main advantage of DQM off the SM event server lies in the relative flexibility of operation. Code updates are relatively simple and possible errors in DQM applications off the storage manager do not directly affect the recording of the data.

The operation of sources off the Storage Manager Event Server is the recommended mode of DQM source operation. Operation of DQM sources in HLT is generally discouraged and will be limited to those purposes which justifiably need access to all (or a very large fraction) of the L1-triggered events and possibly for purposes of the HLT filter monitoring itself.

Fig. 1 shows a sketch of the baseline design of the architecture for data quality monitoring using events served by the Storage Manager event server. Several DQM source-client processes (typically one per subsystem) retrieve events through http-request from the event server. The ME (including environmental information, e.g. the run-number, as well as reference histograms and results from quality tests) are served to one central DQM GUI for visualisation. Alarm states, based on quality test results are also served to the same central display.

Each source-client process produces root-‘file output at least once per run, at the end of a run, optionally up to once per luminosity section (LS) at the LS boundary. The output files are archived in a spool area on disk and backed-up on castor (see section 6). Optionally, summary output, e.g. means and widths of DQM distributions are written to the database (Orcon). Reference histograms are defined based on specifiable ouput-files from previous runs. The subsystem experts are expected to keep the reference histograms up-to-date. Several sets of reference histogram files may be needed to cope with different run conditions.

The operation of the DQM online applications is automated and controlled by run-control.

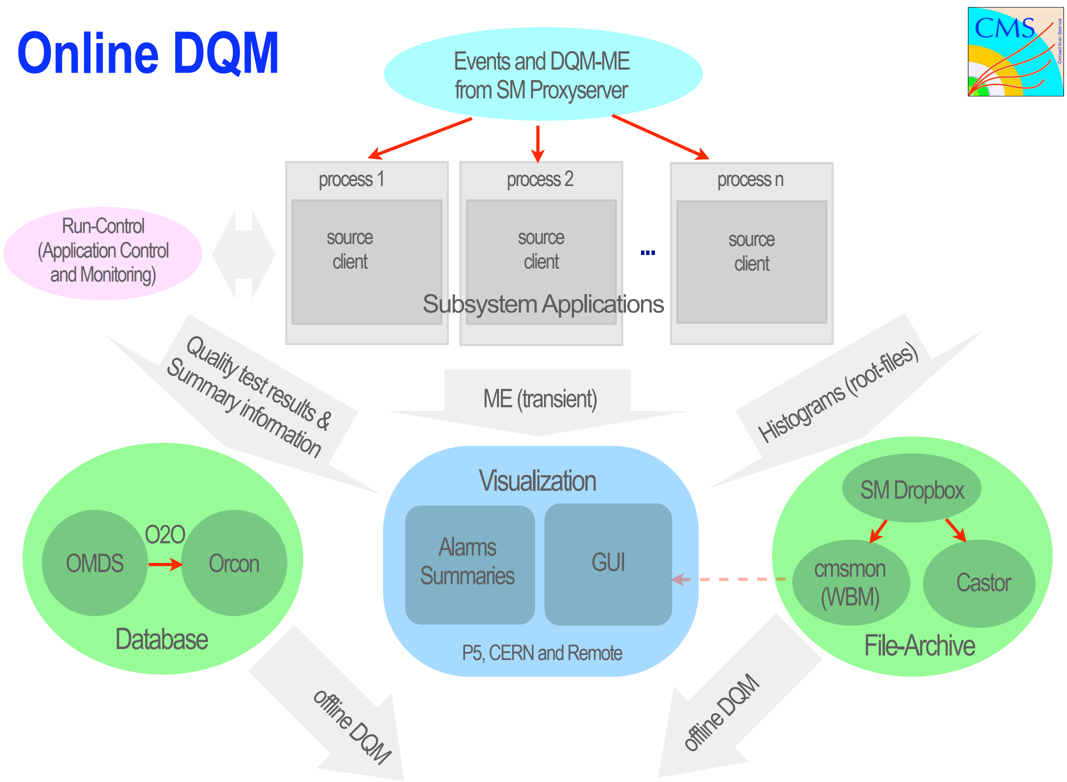

Figure 1: Sketch of the baseline online DQM system.

2 Storagemanager DQM and Event Server

The main function of the Storage Manager is to collect event and DQM data from each Filter Farm HLT process. It also acts as a server of ME and events for downstream consumers. An SMProxyServer collects the information from several Storage Managers and serves DQM-ME and events to consumers upon http-requests (fig.2).

The event server rate is configurable with a maximum of 10 Hz. Event consumer applications can select events based on triggers (like in offline cmsRun jobs). The SMProxyServer baseline design foresees no special handling or sorting of events. There will be no provision for parallel event consumers to receive always different events.

The histograms filled in the HLT filter units (DQM-ME) are served at the end of each Luminosity Section, i.e. 93 seconds. More details can be found in [2].

3 Online DQM Applications

Sources and Clients

Online DQM applications implement sources and clients as EDAnalyzers. We are envisaging introducing a DQM specific EDAnalyzer class 'DQMAnalyzer' that provides infrastructure (methods and data members) that is specific to DQM and generic for all subsystems. Sources act on events (analyze-method) while clients act upon EndLuminosityBlock transitions. Sources and clients are run in the same process. A process can include several sources and several clients as well as any other CMSSW module necessary and/or interesting. No histogram subscriptions are necessary, as the DaqMonitorROOTBackEnd service provides access for the client to all ME created and filled in the source. Optionally additional modules can be included. In online DQM, there will typically be one process (DQM-application) per subsystem, possibly comprising several modules of different sources and clients. For more details see [3].

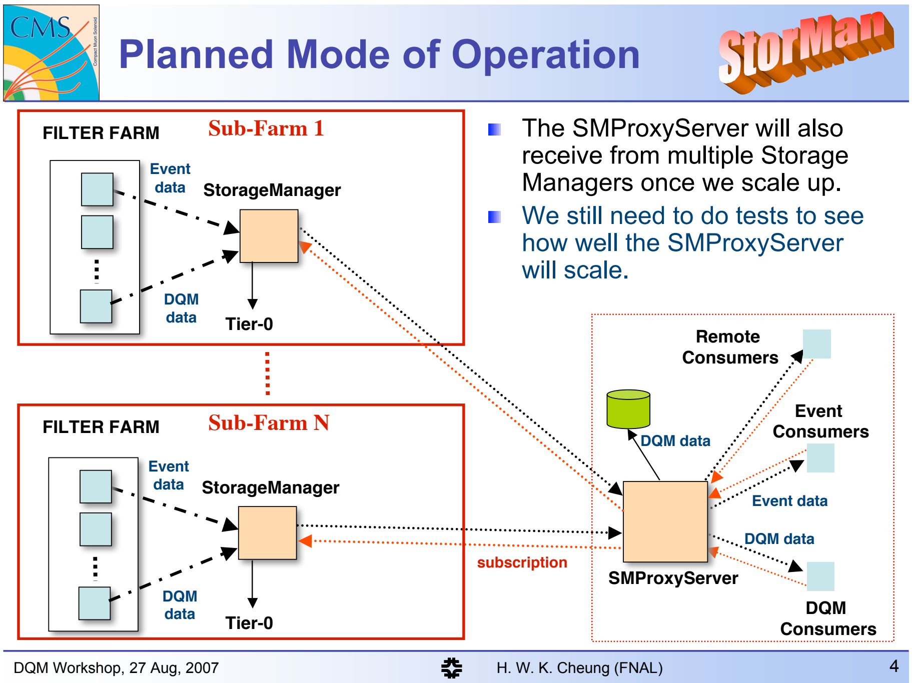

Figure 2: Storage Manager Event and DQM-ME Server (figure taken from [2]).

Database

In online operation mode access to the database is restricted to the retrieval of configuration data (from Orcon) once per run at the beginning of the run. For this purpose the clients retrieve the necessary run-dependent configuration information at the beginning of each run from the Orcon database. It is mandatory that the relevant information for each given run is transfered from OMDS to Orcon (O2O, see fig 10), before the DQM clients are configured. Access to non-event data (as provided by XMAS) from within DQM applications is being discussed and planned for the mid-term, but is not part of the baseline architecture. For the near future trend and correlation studies will be developed as part of semi-online or offline analyses, when both conditions data and event data are readily available from persistent storage, thus providing reproducible results (see also section 11).

Interactivity

Interactivity is constrained to navigating through the list of available DQM results and visualizing them. In addition it is foreseen that users can modify the sensitivity of the display to alarm states by changing specific histogram decorations and sound-levels.

However, there will be no real-time interaction between the user (e.g. subsystem expert) and DQM-client applications. DQM applications can be configured to perform any activity required at the beginning of each run. In addition stand-alone DQM applications that receive events from the same Storage Manager event server can be launched at any time.

Collector

At the level of sources and clients the DQMCollector is not needed and its use is deprecated. However, in the present implementation the MonitorDaemon service is used to ship the monitor elements to a central DQMCollector which is used as a "backend" for central Visualization tools (see fig.1).

4 Cross Detector, HLT and Physics Monitoring

In parallel to subdetector monitoring online DQM comprises applications for online cross-detector monitoring, including monitoring of physics objects and physics distributions. This work is just starting. A status overview can be found at [4]

5 Graphical User Interface

Central DQM-GUI development has recently been relaunched with the aim of achieving a unified customizable DQM GUI for viewing of online and offline data from P5, from CERN and from remote. The goal is to have a tool with common look-and-feel for online and offline DQM data. The GUI should comprise all capabilities for shift and expert use. In its final form it should replace the existing expert GUIs with no effective loss of functionality.

A webserver-based technology was chosen in order to provide both online, CERN and remote access with no overhead for single users to install particular software.

The new DQM GUI comprises a webserver and a number of backends for access to online (live and from file), as well as offline DQM output (see fig. 5). The webserver reuses web-tools technologies [5]. For the backend the Qt-independent parts of the previous iguana code are being reused [6]. The webserver for online-DQM is operated on a Head-node (i.e. on the border between the .cms and cern.ch domains). It reads the histogram objects received from the backends, renders the histogram pictures as png pictures and provides navigation facilities. In the following the main requirements are listed [7, 8].

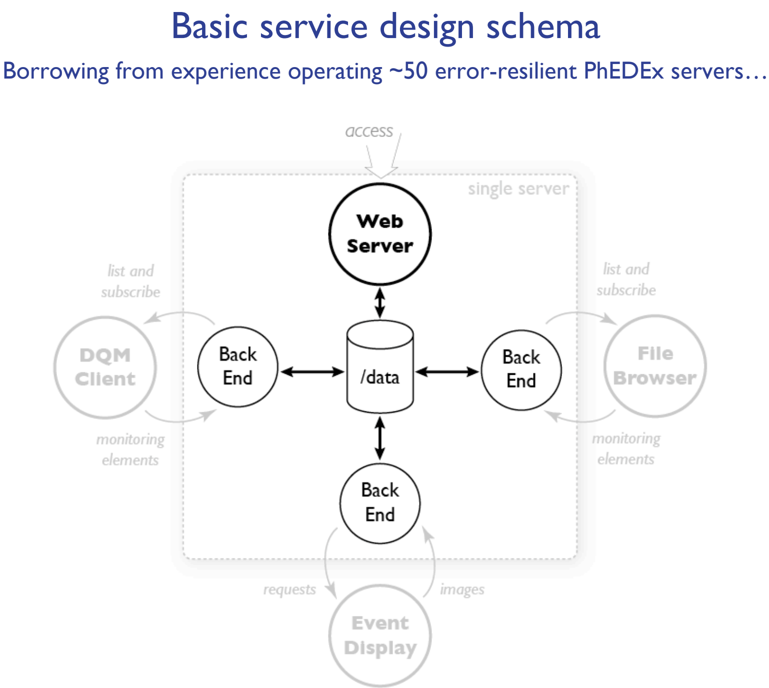

Figure 3: Webserver based DQM Visualization with different backend for access to online DQM as well as output from offline processing (figure taken from [8]).

Requirements

- Histogram collection, rendering and display
  - The GUI collects DQM information from all possible sources, possibly including subsystem-based DQM facilities as well as results from XDAQ based DAQ monitoring.
  - It is expected that up to 10-20 histograms are displayed at once at any given time. However it is desirable to create all png-pictures in advance, such that display changes are with no computational

overhead. In-advance rendering has the advantage that it scales well with the number of simultaneous users requesting different views.

  - A 'heart-beat' display will be provided that indicates available information, e.g. the current situation of subscriptions and histogram updates.

- Histogram Navigation and Decoration
  - The same webserver GUI provides switches between different backends (e.g. live-online, online from file-archive, offline etc).
  - Operation modes, such as expert and shift, remote and P5, online and offline and corresponding authorizations are configurable.
  - For online DQM a detector component based directory structure is foreseen, thus reflecting the several subsystem-dependent sources.
  - A number of histogram views are foreseen, based on different aspects such as synoptic views (e.g. for shift operation), as well as detailed views based on geometrical position, frontend and cabling map, alarm state etc. In addition a search functionality (based on the histogram name is foreseen).
  - Individual layouts simplify retrieval of user-specified views making lengthy repetitive navigation obsolete. A slide show mode will be useful for the purpose of shift-browsing.
  - Subsystems can provide display plug-ins which specify the way histograms are rendered (colour-palettes and decorations etc.)
  - A facility for overlaying reference histograms is foreseen.
  - Customizable menus for histogram decoration (e.g. depending on quality test results) and labeling (including instructions to non-expert users (shift) about possible features that are significant of error states) will be provided.
  - In order to display distributions of single LS a subtraction facility is foreseen, in which accumulated data from two subsequent LS are subtracted to retrieve the distributions of one single LS.
  - Facilities for histogram zooming and pointering are considered important, while drag-and-drop operation is not essential.

- Implementation issues:
  - Apart from the implementation of the various navigation and decoration tools a crucial aspect of the GUI development is to design a clean, extendable interface between DQM clients and GUI. This requirement might include the need to rewrite the DQMCollector.
  - Furthermore it may be advantageous to remove the present subscription mechanism, replacing it by choosing views.
  - For navigation and interactivity aspects a webserver public interface is being considered. This way javascript components which conform with this interface can be collected in a library and attached to the GUI on demand.

- Timeline:
  - The baseline goal is to provide a full replacement for iguana-based DQM viewing by October 2007.
  - In parallel, the implementation of a backend for offline DQM is started.
  - Online DQM functionality beyond that implemented in iguana, e.g. components of the well-advanced implementation of DQM visualization tools based on XDAQ by the silicon and strip tracker groups is going to be realized somewhat later. It is however important to plan for the migration steps well in advance.

6 DQM Output Archiving and Retrieval

At least once per run, maximally once per luminosity section each subsystem stores histograms in a root-file. Key plots (a small number of histograms per subsystem) are stored in a snapshot each luminosity section (LS). DQM plots are accumulated over all luminosity sections of a run. The root-files are saved in a dropbox on the storage-manager disk (presently /data1/dropbox). The filename contains the name of the subsystem, the runnumber and

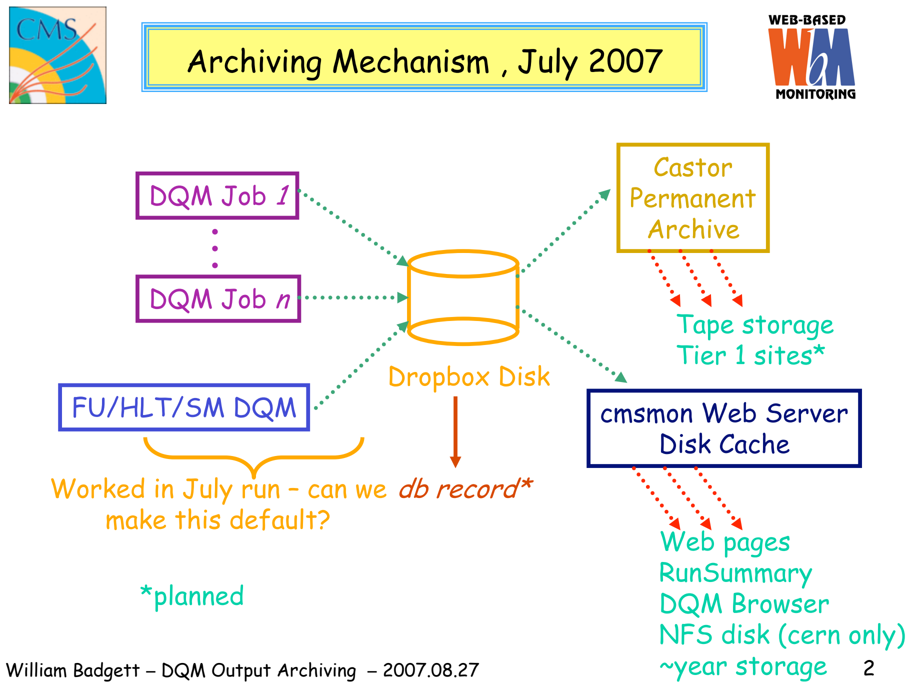

Figure 4: DQM output file archiving mechanism (figure taken from [9]).

the luminosity-section number. As the number of events provided by the event server is limited a post-processing of the DQM-data based on all events is considered. Its realization will depend on possible CPU limitations and disk-access, and might only happen in Tier-0.

The root-files are archived in a disk-based spool-area cmsmon.cern.ch and made accessible on the web through a RunSummary-DB web-interface DQMBrowser [10]. In parallel, several root-files are concatenated into larger files and stored on castor. More details can be found at [9]. Specific files in the spool-area can be tagged for the purpose of providing reference histograms. Note that the DQMBrowser is not a replacement for the planned access to the DQM-files through the DQM-GUI.

7 Online Operation and Control

Online DQM processes (as well as the online eventdisplay) are controlled by run-control using a separate level-one function manager. The DQM applications (containing sources and clients) are XDAQ applications (FUEventProcessors) with standard message and error report lines. The 'macro-state' of applications (i.e. running, error, crashed etc.) should be put back into the xdaq monitoring. XDAQ monitoring should also be used to monitor the actual 'micro-state' of the applications, i.e. rates of events received etc.) [11].

The hardware to operate DQM processes is provided by the DAQ team. The DAQ team also provides OS, XDAQ and CMSSW installations as well as disk space for DQM output.

Subsystems maintain their own job-configuration, initially from cfg-file, later from HLTConfDB. These configurations can be developed and tested in a cmsRun (offline) environment. As the frequency of release updates for DAQ and HLT purposes foreseeably decreases in the future, it is considered to have regular DQM releases which define the base for online DQM processing.

8 Online Summary Views

The purpose of online DQM being prompt and efficient identification of problems, the most prominant goal of online Monitoring must be to distribute all relevant information quickly to whereever it is needed.

Standardized displays that summarize the subdetector status are crucial to detect problems as soon as they happen. At shift level the number of displays from DQM should be limited to 10 per subsystem, such that they fit nicely on one (or less than one) screen. Various views as presently implemented by the different subsystems have been presented in [12].

The amount of time for routine browsing of all histograms relevant for fault-detection should not exceed 20 minutes. A slide show mode will be useful for the purpose of shift-browsing.

Apart from detector specific distributions common views, such as ϕ-distributions of detector occupancies will be useful, as they allow to correlate the distributions from different detectors 'by eye'. For shift crew operation overlayed reference histograms as well as histogram annotations (instructions what to look for in a given histogram) will be crucial.

The DQM displays are just one source of problem detection. Other display, e.g. from DAQ monitoring and DCS will also be available. For the baseline, the coupling of the two informations at the display level is not foreseen, as the coupling requires that full reliability of the coupled informations is ensured first. However, the DQM applications can make use of configuration information (as read from the database at run-start) in order to configure the DQM clients for different detector and data-taking setups.

An error collector provides central information about actual and history errors. Persistent errors can be masked to prevent known errors to hide newly appearing problems. The error collector should be connected to a problem database, in which shift comments as well as expert acknowledgments and remarks are stored.

9 Data Certification

The DQM quality test results form the basis of persistently stored data quality flags (see section 9). This can be a set of states (Good, Bad, unknown), or a number characterizing the percentage of the system correctly running.

Ultimately, the data quality is evaluated for the purpose of physics analyses. The physics working groups will centrally produce 'good run lists' of data samples suitable for specific physics analyses.

The final data certification decision will be produced during full event reconstruction and it will rely on a number of sources, i.e. online DQM, calibration and alignment results as well as offline DQM.

An important input to tuning the validation and certification criteria will come from the evaluation of the entries made by shift and experts in the online DQM error collector as described in section 8.

Online DQM information should be stored in a coherent and homogeneous format, such as the format suggested in [13]. The details of fault-condition storage and the standardization of errors among different subsystems still have to be worked out.

10 DQM and Database

The online databases consist of the 'online master data storage' (OMDS) and the 'offline reconstruction conditions DB Online' (Orcon). OMDS hosts most detector configurations. In addition, some subsystems presently use OMDS as conditions database. Orcon contains configurations necessary to configure the reconstruction algorithms performed in the HLT, as well as information relevant for DQM applications. As mentioned in section 3 the synchronization between OMDS and Orcon is Orcon can also be used to receive conditions data that are produced in CMSSW applications (such as DQM monitoring). uni-directional, i.e. new configurations residing in OMDS are copied to Orcon by means of a O2O process.

A number of subsystems has substantial experience with the use of conditions databases (see also next section 11). A unified scheme of database usage for DQM purposes is being explored in cooperation between the subsystems and the database group.

11 Client output archival, history and trend plotting

Quantities available in DQM client applications, such as summary quantities of histograms (e.g. means and widths of distributions) as well as the results from quality tests can be stored in the conditions database. Once in the database these conditions data produced by DQM clients can be easily correlated with non-event conditions data

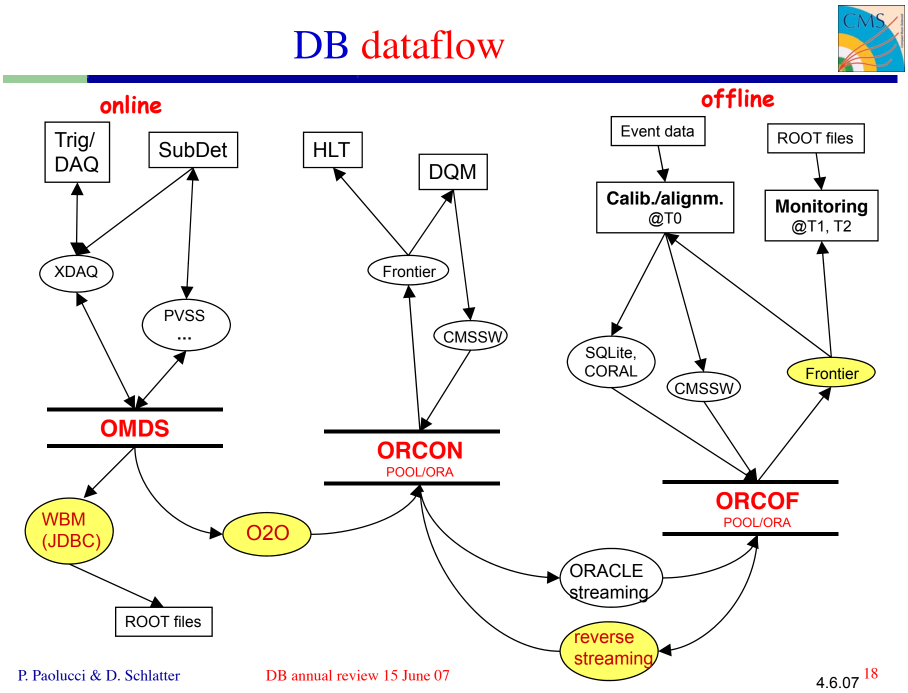

Figure 5: Online Databases. Figure taken from [14].

such as temperatures or DAQ monitoring information (trend-plots). Similarly history plots can be produced by plotting a given database-resident quantity versus time or runnumber.

A possible implementation of this functionality have been presented for the case of ECAL in [15]. Previously the use of the conditions database for the silicon strip tracker was presented [16]. The integration of the existing browsing tools with web-based-monitoring should be considered.

At present dedicated analyses of test beam data are on-going with the aim to investigate trend distributions, possibly uncovering unexpected correlations between seemingly independent quantities. For these analyses, the use of the database has the distinct advantage that information from long periods of time is quickly available. This is contrast to history and trend plots that would be produced in real time from live-online DQM and conditions data correlations. For the latter, continuous data taking operation over longer periods of time and an already good understanding of the detector behaviour are necessary.

12 Beyond the Baseline Goals

A number of planned features are not part of the baseline goals. Their descoping is only in part due to their having less priority, it is also because their implementation require an established base system to exist, thus providing a test platform for extensions.

- Refinement of the selection of histograms relevant for online DQM.
- Integration of validated source modules for operation in HLT.
- Automated DQM post-processing of all event data.
- Optimization of GUI navigation tools and features.
- Visualization of live history plotting.
- Assembly of static run-wise summary webpages ('page-1 of DQM')
- Standardization and refinement of DQM quality criteria.

- Full development, deployment of the error collector and integration with offline data certification.
- Transparent access to live and persistent non-event (conditions) data
  Online access to conditions data will be provided by Live Access Services (LAS) provided by the XMAS framework. A number of interface issues will have to be solved such that DQM applications can transparently receive conditions data from both in a live online environment and in playback mode (i.e. from the database).

13 Summary

The planned architecture of CMS online data quality monitoring has been described. The aim is to realize the described baseline scenario by early 2008. Note that the full set of described features can only be realized if the number of active developers can be increased to the appropriate level. At present a total of 2 FTE are missing in the area of DQM core development.

References

[1] A.Meyer, talk at DQM Workshop 27 August 2007, http://indico.cern.ch/conferenceDisplay.py?confId=19909

[2] H.Cheung, talk at DQM Workshop 27 August 2007, http://indico.cern.ch/conferenceDisplay.py?confId=19909

[3] M.Zanetti, talk at DQM Workshop 27 August 2007, http://indico.cern.ch/conferenceDisplay.py?confId=19909

[4] P.Wittich, talk at DQM Workshop 27 August 2007, http://indico.cern.ch/conferenceDisplay.py?confId=19909

[5] CMS Data Management Webtools https://twiki.cern.ch/twiki/bin/view/CMS/DMWebtools

[6] G.Eulisse, https://twiki.cern.ch/twiki/bin/view/CMS/CMSSW_dqmIGUANAGUI

[7] D.Menasce, L.Tuura, S.Dutta, P.Merkel, Private communication

[8] L.Tuura, talk at weekly EvF/DQM meeting, 7 June 2007 http://indico.cern.ch/getFile.py/access?contribId=29&sessionId=6&resId=0&materialId=slides&confId=16991

[9] W.Badgett, talk at DQM Workshop 27 August 2007, http://indico.cern.ch/conferenceDisplay.py?confId=19909

[10] http://cmsmon.cern.ch/cmsdb/servlet/DQMBrowser?RUN=<runnumber>

[11] E.Meschi, talk at DQM Workshop 27 August 2007, http://indico.cern.ch/conferenceDisplay.py?confId=19909

[12] D.Acosta, talk at DQM Workshop 27 August 2007, http://indico.cern.ch/conferenceDisplay.py?confId=19909

[13] S.Khalathian Talk at Run organization meeting 10 August 2007, http://indico.cern.ch/getFile.py/access?contribId=9&resId=0&materialId=slides&confId=19853

[14] P.Paolucci, talk at CMS annual review 15 June 07, http://indico.cern.ch/getFile.py/access?contribId=28&resId=1&materialId=slides&confId=16736

[15] G.Franzoni, talk at DQM Workshop 27 August 2007, http://indico.cern.ch/conferenceDisplay.py?confId=19909

[16] D.Kcira talk at weekly EvF/DQM meeting 31 May 2007, http://indico.cern.ch/conferenceDisplay.py?confId=15919#6

[17] DQM Global Run Wiki Page: https://twiki.cern.ch/twiki/bin/view/CMS/DQMGlobalRun0907