A Monitoring System for the LHCb Dataflow

João Barbosa, Clara Gaspar, Beat Jost, Markus Frank, Luis G. Cardoso

***Abstract*—This document presents a new system used to monitor the data acquisition of the LHCb experiment. The monitoring takes place from the time the data is triggered in the Front-End Boards all the way to the time it is temporarily stored in the online storage system, thus providing the operators of the experiment with a detailed overview of the status of the data acquisition system. This system was thought of as a complement to the already existing monitoring and control tools available in the Experiment Control System in order to facilitate the operation of LHCb. We will present the various features of the monitoring tool and how it is being used by the ones involved in the operation of the experiment.**

I. Introduction

THE LHCb [1] experiment is one of four experiments that use the beam provided by the LHC particle accelerator, and is now undergoing Run 2 of physics data taking. The operation of LHCb is usually overseen by 2 shifters who, among other responsibilities, have to make sure all the data is being acquired and stored according to requirements. To do this, the operators make use of the monitoring and controls implemented by Run Control [2], which is based on a SCADA system (WinCC-OA) [3] and the JCOP framework [8]. The framework's guidelines include the use of a Finite State Machine Tree [4] which propagates commands downstream and states upstream, where leaf nodes of this tree represent device controllers responsible for both hardware and software devices as illustrated in Fig. 1. The state of these devices is summarized by the Finite State Machine tree in a generic way, as this state machine has to accommodate several different types of devices. WinCC-OA is an industrial controls software that allows for the control and monitoring of large distributed systems. LHCb's Run Control is spread across many virtual machines and controls, monitors and configures many aspects of the operation by also making use of a direct connection to the Finite State Machine Tree. The monitoring system presented in this document complements the Run Control by providing the operators and other users with a broader overview of the data flow through the Data Acquisition system [5], [6].

II. The DAQ System

A. Architecture

LHCb's Data acquisition system is composed of a three stage trigger system, the first performed in hardware and the two last ones in software. LHCb operates with a bunch crossing rate of 40 MHz and the resultant measurements are passed through a first hardware trigger, with a fixed latency of

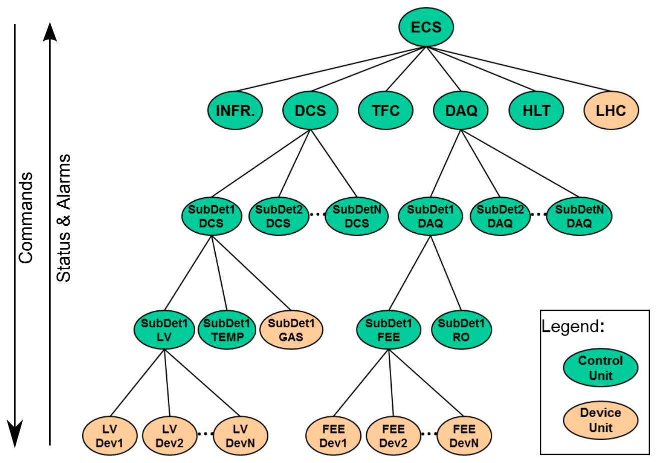


Fig. 1. Finite State Machine Tree.

4 μs that will filter these measurements and accept a maximum sustained rate of 1MHz of events. These events are pushed into FPGA-based readout cards, denominated "TELL1" cards, which perform zero suppressing and event packaging. Each one of these cards is associated with a sub detector, and so each one only holds an event fragment. The second stage trigger, denominated the High Level Trigger (HLT) is a pure software trigger that operates on a PC farm composed of 62 subfarms of approximately 25 servers each. The Timing and Fast Control (TFC) subsystem distributes a common clock and commands through the TELL1 boards and the sub-detectors, letting these know to which server the packets need to be sent via a LAN network through a credit system. Part of the data is redirected to the Monitoring, Reconstruction and Calibration farms. Each server will then process several events in parallel, assembling the events from their composing fragments and filtering out the less interesting events, thus performing the software trigger on the data. These accepted events are stored locally on the disk of each server, awaiting the CPU availability for the third stage of processing. When there is CPU available, tasks will be deployed to pick up these events, which will then go through a third stage of triggering and further processing. Lastly the events are classified and divided in streams and sent to a temporary storage system in the experiment's facilities before they are sent to the more permanent storage that handles the data gathered by most of the experiments at CERN.

In order to operate different parts of the detector independently, LHCb uses a partitioning scheme that creates independent DAQ slices which are able to operate concurrently. The DAQ system is thus divided in partitions, each one having its own allocated readout cards, subfarms and storage slices. Moreover, it is also possible to exclude devices and subtrees

J. Barbosa, C. Gaspar, B. Jost, M. Frank and L. G. Cardoso are with CERN, Switzerland
978-1-5090-2014-0/16/$31.00 ©2016 IEEE

from each partition's operation when there is a malfunction or maintenance undergoing, thus avoiding commands and status to be taken into account by them.

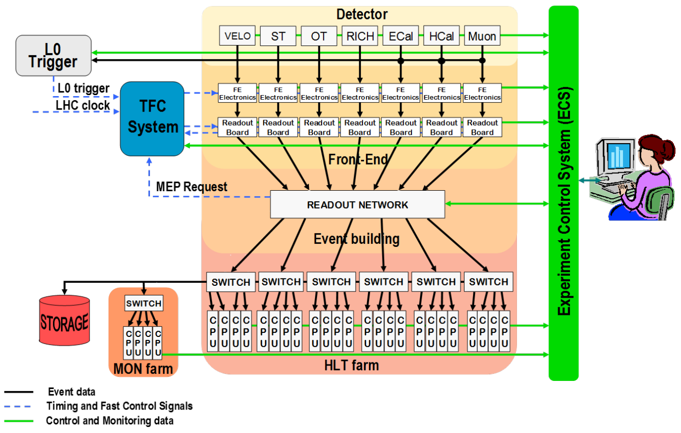


Fig. 2. Data Acquisition architecture.

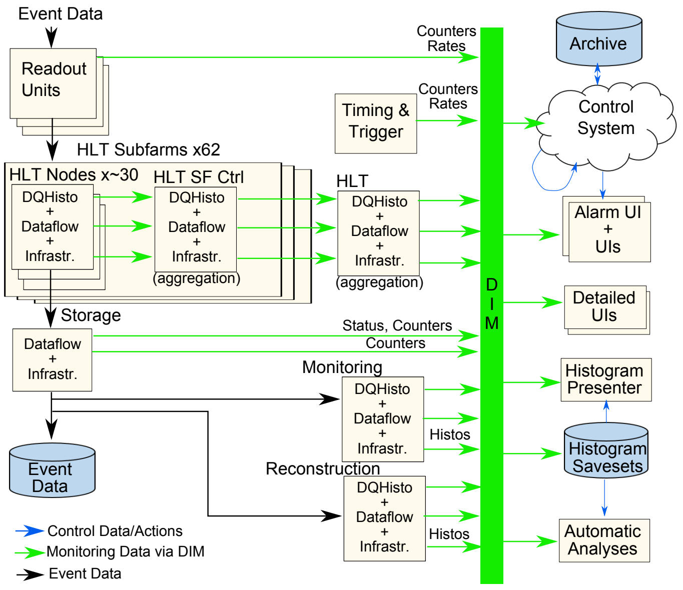


Fig. 3. Monitoring Data gathering scheme.

B. Monitoring Data

In each one of the hardware or software devices present in the data acquisition stages presented in Section II-A, there are counters and histograms being produced for monitoring purposes. These counters are accessible via DIM [7] servers that either communicate with hardware or are embedded in the software devices. These counters are then passed to Run Control and other systems present in the control system through the DIM protocol, either directly, or passing through a series of DIM servers that perform the aggregation of the monitoring data. Aggregation is particularly necessary in the case of the High Level Trigger farm due to the number of servers and tasks running in each of them. All the counters, histograms and other forms of monitoring data are then available to the histogram presenter, detailed user interfaces and the control system that can generate alarms, archive data and perform automatic analyses. This scheme is evidenced in Fig. 3.

III. Dataflow Monitoring

Run Control, with its Finite State Machine, is the main tool used to operate and monitor LHCb as a whole. The Dataflow Monitoring system envisioned to complement all the tools already present by providing the operators with metrics and alarms for every subsystem in a compact, detailed and versatile way. This system is based in WinCC-OA and the JCOP framework. WinCC-OA uses a database with a datapoint structure and provides tools to archive data, generate alarms, build user interfaces and scripting. The JCOP framework consists on a set of guidelines and software tools to be used by developers in specific control systems. The project can be thought of as being composed of 2 essential parts: the back-end, where the data is gathered, archived and alarms are generated, and the front-end that comprises all the graphical user interfaces.

A. Data gathering and archiving

The monitoring data provided by each device of the experiment is made available to this system in a variety of formats, although they can be summarized by two sources:

- DIM services with structured names so they are easily discovered.
- Datapoints already present in WinCC-OA.

Even though the sources are basically two, the format and structure of the monitoring data provided by each stage of the DAQ system can be significantly different. Because of this, the backend of the Dataflow Monitoring project consists of several processes that aggregate the data in different ways, but act essentially on the same philosophy. Although this project does not implement a Finite State Machine of its own, it connects to the states of each DAQ partition to guide the discovery and archiving of monitoring data. If a partition enters an idle state it will be disregarded and no further work is done, otherwise the backend processes will discover which resources are allocated to that partition. This means accessing Run Control or searching DIM services in search for any stream, board or farm that is currently allocated in that specific partition. For each of these devices, a data point with a common format. Currently the gathered monitoring data comprises:

- Event rates.
- Byte rates.
- IP packet rates.
- A listing of the allocated devices in each partition.

Archiving is optimized in order to reduce the amount of samples stored while still providing enough information for historic debugging. Samples are taken with a minimum period of 30 s and only when the value varies significantly within a percentage of the total value, thus performing a form of smoothing. This ensures no samples are being taken in periods of inactivity or when the rate values are stable, which accounts for a very good percentage of time.

B. Alarms

The generation of alarms is performed on the data available, and since these are mainly event rates or raw data rates, alarms are triggered mainly when a certain device is reporting a rate lower than is to be expected. These are reported in the Graphical User Interfaces of the project and, in some cases, they are relayed to the central Alarm Screen of the experiment, which reports all the error and warning alarms that the operators need to attend to as soon as possible.

C. Graphical User Interface

The simplest graphical interface to access the Dataflow Monitoring, exposed in Fig. 4, is comprised of a summary of each stage of the DAQ system as well as a control panel where several aspects of the interface can be manipulated.

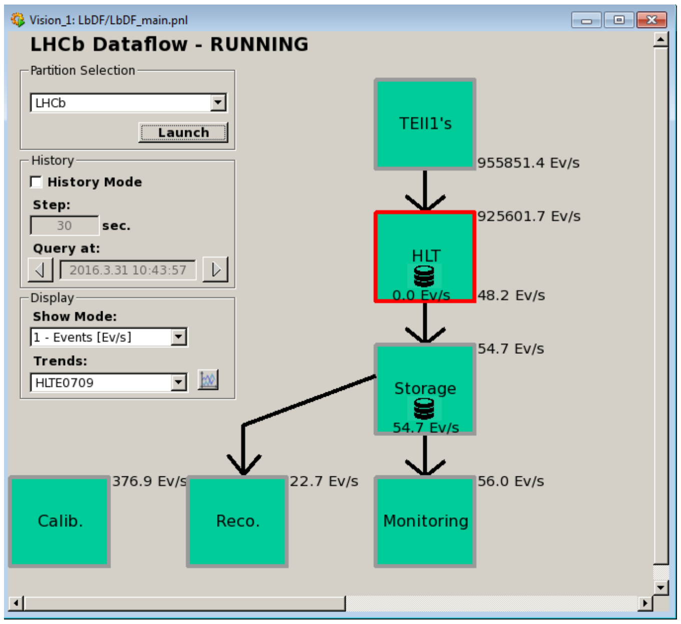
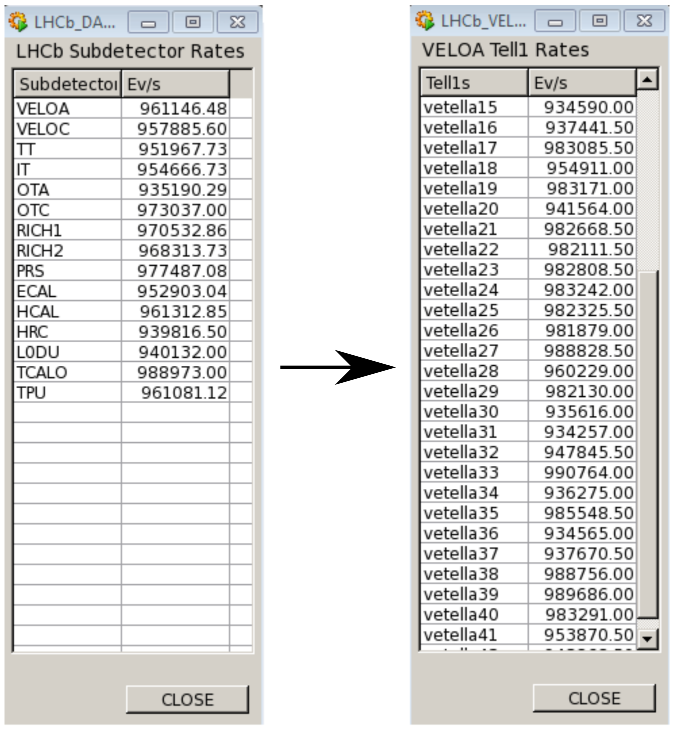


Fig. 4. Main Graphical Interface of LHCb Dataflow Monitoring.

From this initial panel the user is able to access a detailed description of each of the stages of DAQ, leading to tables that can show as much detail as the event rate in the input and/or output of each device unit, as shown in Fig. 5.

The project also includes a trending tool that requires the user simply to select any metrics showing either in detailed tables or in the main user interface and issue the plot command with a click. The trending tool also allows for templates to be saved, in which the devices names are stored and can be accessed at any time to produce the same plot(s) without the need of re-selecting devices in the interface. These templates are accessible to anyone on the network through the control panel in the main interface. Example of a plot generated with this tool is shown in Fig. 6.

With the help of the control panel shown on the left of Fig. 4, it is also possible to access the history mode, which makes use of the archive described in Section III-A. With the history mode, one can make the whole interface and tables show the data correspondent to a past period in time, providing



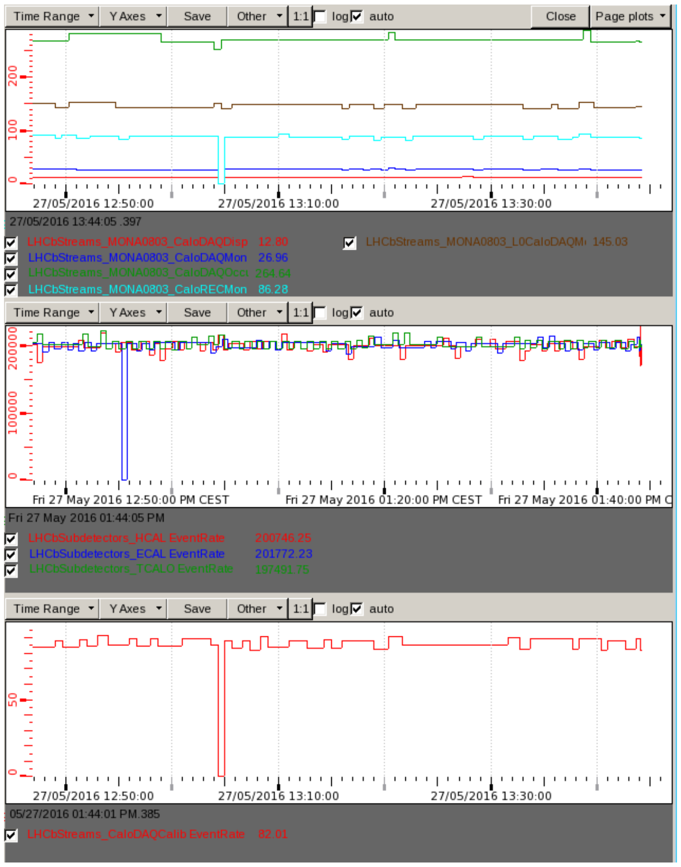
LHCb Subdetector Rates

Subdetecto	Ev/s
VELOA	961146.48
VELOC	957885.60
TT	951967.73
IT	954666.73
OTA	935190.29
OTC	973037.00
RICH1	970532.86
RICH2	968313.73
PRS	977487.08
ECAL	952903.04
HCAL	961312.85
HRC	939816.50
L0DU	940132.00
TCALO	988973.00
TPU	961081.12

VELOA Tell1 Rates

Tell1s	Ev/s
vetella15	934590.00
vetella16	937441.50
vetella17	983085.50
vetella18	954911.00
vetella19	983171.00
vetella20	941564.00
vetella21	982668.50
vetella22	982111.50
vetella23	982808.50
vetella24	983242.00
vetella25	982325.50
vetella26	981879.00
vetella27	988828.50
vetella28	960229.00
vetella29	982130.00
vetella30	935616.00
vetella31	934257.00
vetella32	947845.50
vetella33	990764.00
vetella34	936275.00
vetella35	985548.50
vetella36	934565.00
vetella37	937670.50
vetella38	988756.00
vetella39	989686.00
vetella40	983291.00
vetella41	953870.50

Fig. 5. Detailed view of the accept events rate in the readout cards, by card (right) and summarized by subdetector or type of trigger (left).



27/05/2016 13:44:05 .397

☑ LHCbStreams_MONA0803_CaloDAQDisp 12.80
☑ LHCbStreams_MONA0803_CaloDAQMon 26.96
☑ LHCbStreams_MONA0803_CaloDAQOccı 264.64
☑ LHCbStreams_MONA0803_CaloRECMon 86.28
☑ LHCbStreams_MONA0803_L0CaloDAQM‹ 145.03

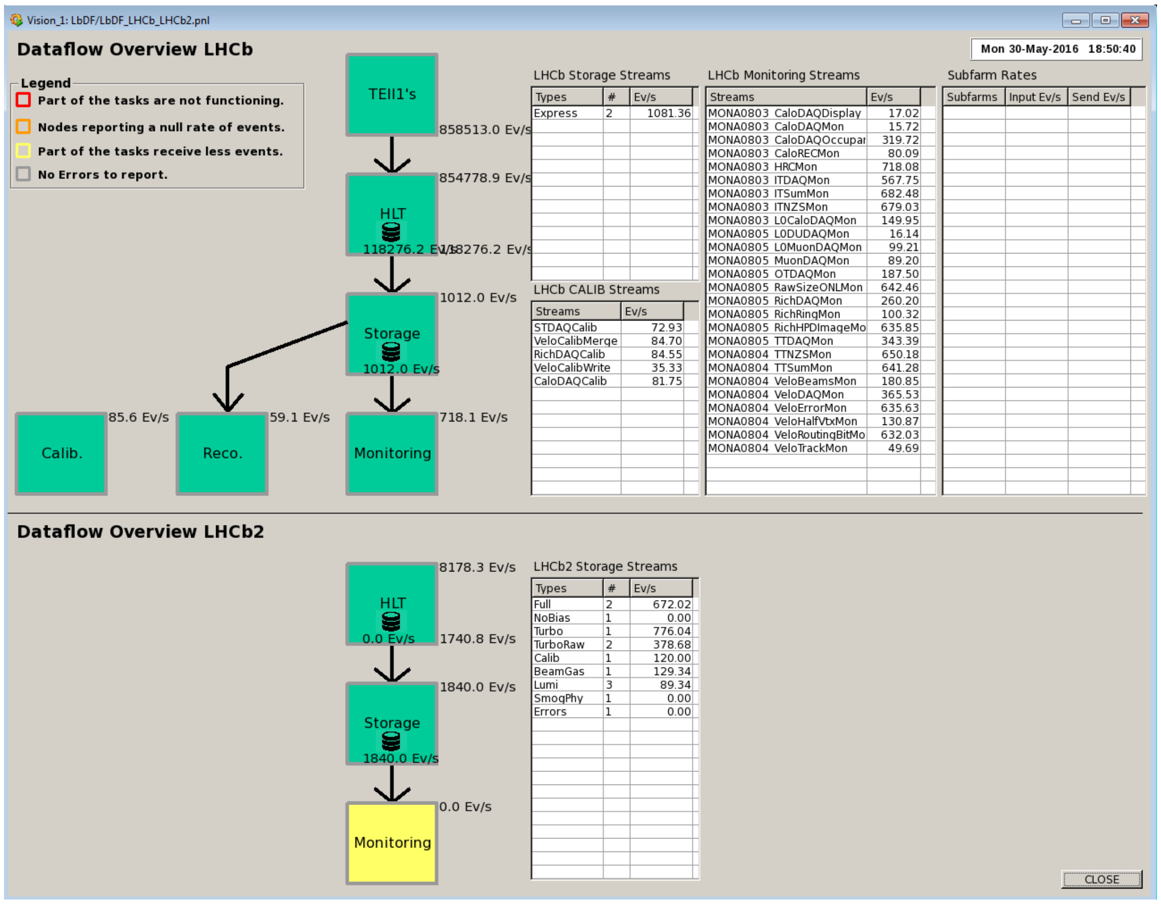
Fri 27 May 2016 01:44:05 PM

☑ LHCbSubdetectors_HCAL EventRate 200746.25
☑ LHCbSubdetectors_ECAL EventRate 201772.23
☑ LHCbSubdetectors_TCALO EventRate 197491.75

05/27/2016 01:44:01 PM.385

☑ LHCbStreams_CaloDAQCalib EventRate 82.01

Fig. 6. Example of plot generated by selecting elements of three different tables.



Dataflow Overview LHCb

Mon 30-May-2016 18:50:40

Legend

- 🟥 Part of the tasks are not functioning.
- 🟧 Nodes reporting a null rate of events.
- 🟨 Part of the tasks receive less events.
- ⬜ No Errors to report.

LHCb Storage Streams

Types	#	Ev/s
Express	2	1081.36

LHCb CALIB Streams

Streams	Ev/s
STDAQCalib	72.93
VeloCalibMerge	84.70
RichDAQCalib	84.55
VeloCalibWrite	35.33
CaloDAQCalib	81.75

LHCb Monitoring Streams

Streams	Ev/s
MONA0803 CaloDAQDisplay	17.02
MONA0803 CaloDAQMon	15.72
MONA0803 CaloDAQOccupar	319.72
MONA0803 CaloRECMon	80.09
MONA0803 HRCMon	718.08
MONA0803 ITDAQMon	567.75
MONA0803 ITSumMon	682.48
MONA0803 ITNZSMon	679.03
MONA0803 L0CaloDAQMon	149.95
MONA0805 L0DUDAQMon	16.14
MONA0805 L0MuonDAQMon	99.21
MONA0805 MuonDAQMon	89.20
MONA0805 OTDAQMon	187.50
MONA0805 RawSizeONLMon	642.46
MONA0805 RichDAQMon	260.20
MONA0805 RichRingMon	100.32
MONA0805 RichHPDImageMo	635.85
MONA0805 TTDAQMon	343.39
MONA0804 TTNZSMon	650.18
MONA0804 TTSumMon	641.28
MONA0804 VeloBeamsMon	180.85
MONA0804 VeloDAQMon	365.53
MONA0804 VeloErrorMon	635.63
MONA0804 VeloHalfVtxMon	130.87
MONA0804 VeloRoutingBitMo	632.03
MONA0804 VeloTrackMon	49.69

Subfarm Rates

Subfarms	Input Ev/s	Send Ev/s

Dataflow Overview LHCb2

LHCb2 Storage Streams

Types	#	Ev/s
Full	2	672.02
NoBias	1	0.00
Turbo	1	776.04
TurboRaw	2	378.68
Calib	1	120.00
BeamGas	1	129.34
Lumi	3	89.34
SmogPhy	1	0.00
Errors	1	0.00

Fig. 7. Dataflow Monitoring in the Control Room of LHCb.

for a complete replay of the full Data Acquisition Dataflow at 30 s intervals. This, in conjunction with the trending, can facilitate tracking issues in the Data Acquisition system.

Because the user interface was built in a modular manner, the production of new flavours of the main interface become facilitated. Given that the High Level Trigger in LHCb is split into two stages that can run concurrently, we found it would be useful to have a special flavour of the interface operate in the Control Room of the experiment so that both stages would be visible within one screen. This flavour simply uses two similar modules and applies them to two different partitions of the experiment, as seen in Fig. 7.

IV. Conclusions and Future Plans

The operation of LHCb is mostly done through Run Control and its associated user interfaces, which provide a complete control over the experiment and also summarized monitoring data. It did not, however, contain a detailed overview of what is happening which the data acquisition devices.

Dataflow Monitoring was created so that the operators and shifters would have a better view of what is happening at each step of the data acquisition process. Its operation is real-time, showing the data rates at each device as well as a summary per subfarm or subdetector readout group. It also implements a trending tool and an historic view mode in order to analyse problems that have already passed for a better diagnose.

This system is now being used daily both inside and outside the control room and is evolving according to user feedback and new needs of the operators. Plans are in place to integrate more data into this monitoring system and, consequently, generate alarms on new problematic situations.

References

[1] LHCb Collaboration, "LHCb - the Large Hadron Collider beauty experiment, reoptimised detector design and performance", *CERN/LHCC* 2003-030

[2] F. Alessio *et al.*, "The LHCb Run Control System," *Real Time Conference (RT), 2010 17th IEEE-NPSS*, Lisbon, 2010, pp. 1-6.

[3] WinCC-OA http://w3.siemens.com/mcms/human-machine-interface/en/visualization-software/simatic-wincc-open-architecture/Pages/Default.aspx

[4] B. Franek and C. Gaspar, "SMI++ - an object oriented Framework for designing distributed control systems", *IEEE Trans. Nucl. Sci.* **45** 4 1946-50, 1998.

[5] B. Jost, "The LHCb DAQ system," *Nuclear Science Symposium Conference Record, 2000 IEEE*, Lyon, 2000, pp. 26/1-26/6 vol.3.

[6] D. Campora, N. Neufeld and R. Schwemmer, "Improvements in the LHCb DAQ," *Real Time Conference (RT), 2014 19th IEEE-NPSS*, Nara, 2014, pp. 1-2.

[7] C. Gaspar, M. Dönszelmann and Ph. Charpentier, "DIM, a portable, light weight package for information publishing, data transfer and interprocess communication", *Computer Physics Communications* **140** 1+2 102-9, 2001.

[8] D. R. Myers et al, "The LHC experiments Joint COntrols Project, JCOP", *Proc. Int. Conf. on Accelerator and Large Experimental Physics Control Systems* (Trieste) Italy, 1999