PROCEEDINGS OF SCIENCE

DQM4HEP - A Generic Online Monitor for Particle Physics Experiments

Carlos Chavez-Barajas, Tom Coates* and Fabrizio Salvatore
University of Sussex (GB)
E-mail: carlos.chavez.barajas@cern.ch, tom.coates@cern.ch, p.f.salvatore@sussex.ac.uk

David Cussans
University of Bristol (GB)
E-mail: david.cussans@bristol.ac.uk

Rémi Éte
DESY (DE)
E-mail: remi.ete@desy.de

Adrián Irles-Quiles
LAL Orsay (FR)
E-mail: irles@lal.in2p3.fr

Laurent Mirabito
IPNL (FR)
E-mail: laurent.mirabito@cern.ch

Antoine Pingault
Ghent Univiersity (BE)
E-mail: antoine.pingault@ugent.be

Matthew Wing
University College London (GB)
E-mail: m.wing@ucl.ac.uk

Currently there is a lot of activity in R&D for future colliders. Multiple detector prototypes are being tested, each with different requirements for data acquisition and monitoring, which has generated different ad-hoc software solutions. We present DQM4HEP, a generic C++11 framework for online monitoring for particle physics experiments, and results obtained at several testbeams with detector prototypes using the framework as it was developed. We also present the currently ongoing work to integrate DQM4HEP and EUDAQ, which will allow these to work together as a complete and generic DAQ and monitoring system for any detector test, as part of AIDA-2020.

Topical Workshop on Electronics for Particle Physics
11 - 14 September 2017
Santa Cruz, California

*Speaker.

PoS(TWEPP-17)050

1. Introduction

There is currently a lot of activity in R&D for future collider experiments to succeed the Large Hadron Collider, such as the International Linear Collider, Compact Linear Collider, and Future Circular Collider. Each of these will require advanced, next-generation detector technologies that must be tested extensively during development to ensure that they are capable of the sensitivities necessary to meet the physics goals of these colliders. The R&D projects for these detectors and subdetector components are well underway, and many are currently in testing phases at beamlines around the world.

There is a natural tendency for each team to set their own standards, developing solutions tailored for their needs. In the past, this has generated a variety of tools for data acquisition and online data monitoring, many of which cannot be applied outside of their original intended scope. By developing tools designed for many different applications, the amount of time and effort necessary to create data acquisition and monitoring setups can be reduced significantly, simplifying and speeding up planning and deployment of physics testbeams and allowing more science to be done faster.

The AIDA-2020 project is an EU-funded research programme for developing infrastructure and technologies for particle physics detector development and testing, comprising 24 member countries and lead by CERN. The Data Quality Monitoring for High-Energy Physics framework (DQM4HEP) is part of Work Package 5 for the development of common data acquisition systems for beam tests. It aims to fulfil the needs of Task 5.4: development of data quality and slow control monitoring.

2. The DQM4HEP framework

DQM4HEP is an online monitoring and data quality monitoring tool developed for physics testbeams for high-energy and particle physics. It is designed to be used for monitoring physics testbeams in a generic way. The structure of the program allows for independent components of the framework to be used, not used, or exchanged, by isolating each function of the program into specific and independent processes. The components that are specific to particular users – the analysis and standalone modules – are written in standard C++ code, meaning they are capable of performing any data unpacking, processing or analysis that is necessary. The framework then handles packaging and transporting this information over a network, which allows it to operate without special rules for handling particular datatypes, allowing it to handle *anything* that can be packed into, decoded from, and accessed by normal C++ methods.

2.1 Prerequisites and dependencies

DQM4HEP is written in the C++11 standard and requires ROOT5 [1] for handling ROOT objects such as plots, charts and histograms. The visualisation package that contains code for the graphical user interfaces requires Qt4 [2]. DIM is used for network communication [3]. There is also an optional dependency for the usage of the LCIO event data model, which is defined as the standardised filetype within AIDA-2020. If LCIO files are being used, DQM4HEP requires the LCIO software (part of the ilcSoft package [4]) in order to compile the libraries that enable serialisation of LCIO data.

2.2 Programming paradigms and structure

DQM4HEP is designed with genericness as its core paradigm, using processes and algorithms that are independent of data type. The ability to run multiple instances of each process of the framework is also key to its flexibility, allowing users to, for example, separate sub-detector data from data that has undergone event building, operate in online or offline modes, or distribute the computational load of the analysis over several networked computers.

The generic nature of the framework lies in two core features:

- The **Event Data Model abstraction** allows the user to define the type and structure of an event and how serialisation should be handled.
- The **plugin system** allows the inclusion of any user-defined classes via external libraries, such as to select the serialisation process, online analysis, etc.

The online architecture is shown in Figure 1. The colour key on the figure refers to the different roles within a team, and who is responsible for each aspect of the framework. The orange boxes are the "DQM modules" that must be created by the user, specific to their use-case. Green boxes are interfaces for interacting with the framework or viewing data, and are used by shifters.

There are two varieties of DQM modules:

- **Analysis modules** receive events from the DAQ system and process the data into a form useful for monitoring, then encapsulate these as monitor elements to be sent to the monitor element collector. Analysis modules must be written for each use-case but are produced from templates in which ordinary C++ code is used.
- **Standalone modules** are almost identical to analysis modules, except that they receive data from somewhere other than the DAQ system. These are mainly used for monitoring environmental conditions.

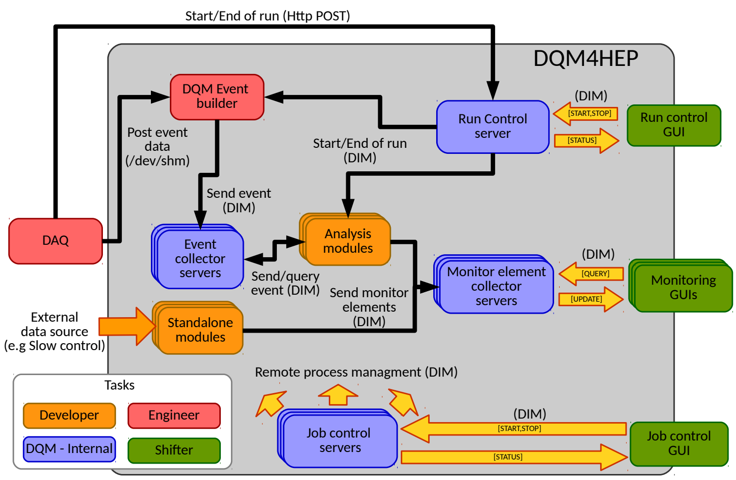


Figure 1: The global online architecture of DQM4HEP.

3. GUI and visualisation

DQM4HEP's user interface is based on the Qt4 framework and is divided into three separate windows, which can be seen as the green boxes in Figure 1. The most important of these is the monitoring GUI.

The monitoring GUI accesses and views monitor elements, using a highly flexible and customisable interface featuring multiple canvases. The individual monitor elements are arranged in a tree-like structure, and are also editable from within the UI, allowing manipulation such as zooming, scaling and fitting. Combinations of monitor elements can be predefined using an XML steering file, allowing complicated setups or large number of plots to be shown immediately upon startup.

4. Data quality testing

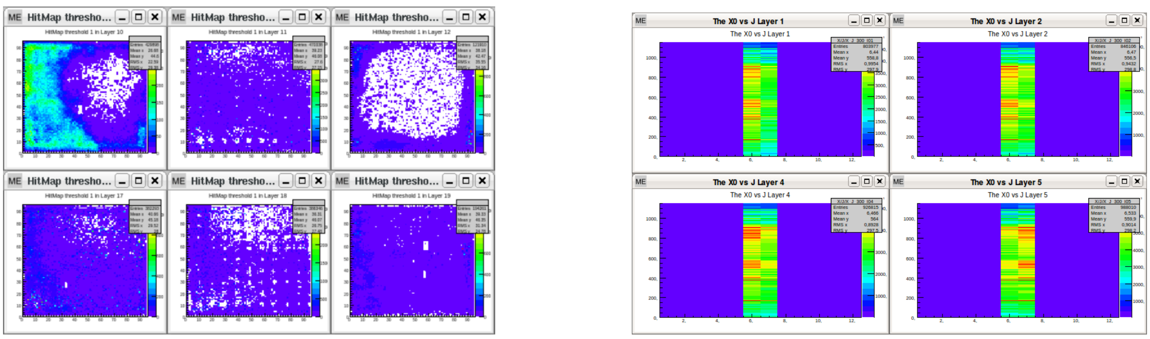
An additional level of online monitoring is data *quality* monitoring, which assesses data being received in real-time to allow operators and shifters without detailed knowledge of the hardware to determine the performance of the device under test. Data quality monitoring (DQM) uses a variety of statistical and comparative methods to assess the quality or "goodness" of data received.

Quality tests are currently being implemented in the framework but when complete will be able to be applied to any monitor element(s) from within analysis modules. It will also be possible to run quality tests in both online and offline modes.

5. Implementation examples

DQM4HEP has already been used in a number of testbeams for multiple detector prototypes, including combined testbeams. So far these have been with the AHCAL+beam telescope, and SDHCAL+SiWECAL. More information on these detectors and testbeams can be found in the references [5][6] and some examples of the framework in use can be seen in Figures 2a and 2b.

During AHCAL testbeams, DQM4HEP is a useful tool to identify issues with the detectors such as new scintillator tile layers that were being tested for the first time. Hitmaps allowed quick and simple visual identification of noisy or dead tiles, plainly visible as erroneous hits or gaps,



(a) Several hitmaps of the SDHCAL in use.

(b) AHCAL+beam telescope correlation plots.

Figure 2: Two plots produced by DQM4HEP during SDHCAL and AHCAL testbeams

respectively. In SDHCAL testbeams, hitmaps were used to identify that some detector elements showed a lack of hits in their centres due to an overflow of incoming gas. Using hitmaps made in DQM4HEP, the problem could be identified and corrected, restarting a new run with a more stable detector.

6. Conclusion

With its generic and flexible programming, DQM4HEP forms a powerful framework for online monitoring and data quality monitoring for physics testbeams that will allow physicists to focus on the physics goals instead of engineering and software issues. Planned future work to allow DQM4HEP to interface directly with EUDAQ [7], a generic data acquisition system within AIDA-2020, will allow the two software frameworks to work together to form a generic data acquisition, monitoring and quality monitoring system that is capable of being used for nearly any type of detector model.

The DQM4HEP framework can be found on Github at `https://github.com/DQM4HEP`. User and technical documentation is currently in progress but any enquiries about the framework can be directed to `dqm4hep@gmail.com`.

References

[1] R. Brun and F. Rademakers, *ROOT – An Object Oriented Data Analysis Framework*, Nucl. Inst. & Meth. in *Phys. Res. A 389*, 1997, pp. 81-86. Available at: http://root.cern.ch

[2] Qt Company, http://www.qt.io, v4.7, 2016.

[3] C. Gaspar et al., *DIM a Portable Lightweight Package for Information Publishing Data Transfer and Inter-process Communication* presented at the Int. Conf. Computing in High Energy and Nuclear Physics, Padova, Italy, 2000)

[4] F. Gaede and J. Engels, 2007. *Marlin et al-A Software Framework for ILC detector R&D*. EUDET-Report-2007-11. Available at: http://ilcsoft.desy.de/portal

[5] CALICE collaboration, 2016. *First results of the CALICE SDHCAL technological prototype*. *Journal of Instrumentation*, 11(04), p.P04001. [hep-ex/160202276]

[6] C. Adloff, Y. Karyotakis, J. Repond, A. Brandt, H. Brown, K. De, C. Medina, J. Smith, J. Li, M. Sosebee and A. White, 2010. *Construction and commissioning of the CALICE analog hadron calorimeter prototype*. *Journal of Instrumentation*, 5(05), p.P05004. [hep-ex/10032662]

[7] Y. Liu. *EUDAQ2 User Manual*. Retrieved October 13, 2017, from https://github.com/eudaq/eudaq

Acknowledgments

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 654168.