Mu2e DAQ and slow control systems

Antonio Gioiosa[1,2,*], *Richard* Bonventre[3], *Simone* Donati[1,2], *Eric* Flumerfelt[4], *Glenn* Horton-Smith[5], *Luca* Morescalchi[2], *Vivian* O'Dell[4], *Elena* Pedreschi[2], *Gianantonio* Pezzullo[6], *Franco* Spinella[2], *Lorenzo* Uplegger[4], and *Ryan* Rivera[4]

[1]Università di Pisa, I-56127 Pisa, Italy
[2]INFN Sezione di Pisa, I-56127 Pisa, Italy
[3]Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA
[4]Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA
[5]Department of Physics, Kansas State University, Manhattan, Kansas 66506, USA
[6]Yale University, New Haven, Connecticut, 06520, USA

Abstract. The Mu2e experiment at the Fermilab Muon Campus will search for the coherent neutrinoless conversion of a muon into an electron in the field of an aluminum nucleus with a sensitivity improvement by a factor of 10,000 over existing limits. The Mu2e Trigger and Data Acquisition System (TDAQ) uses *otsdaq* as the online Data Acquisition System (DAQ) solution. Developed at Fermilab, *otsdaq* integrates both the *artdaq* DAQ and the *art* analysis frameworks for event transfer, filtering, and processing. *otsdaq* is an online DAQ software suite with a focus on flexibility and scalability and provides a multi-user, web-based, interface accessible through a web browser. The data stream from the detector subsystems is read by a software filter algorithm that selects events which are combined with the data flux coming from a Cosmic Ray Veto System. The Detector Control System (DCS) has been developed using the Experimental Physics and Industrial Control System (EPICS) open source platform for monitoring, controlling, alarming, and archiving. The DCS System has been integrated into *otsdaq*. A prototype of the TDAQ and the DCS systems has been built at Fermilab's Feynman Computing Center. In this paper, we report on the progress of the integration of this prototype in the online *otsdaq* software.

1 Introduction

Lepton Flavor Violation (LFV) has been observed in the neutral sector in neutrino oscillations but not in the charged sector. In the Standard Model (SM), the predicted branching fractions of Charged Lepton Flavor Violating (CLFV) processes are below 10^{-50}. The observation of CLFV would thus provide unambiguous evidence for the existence of New Physics beyond the SM. The Mu2e experiment at Fermilab will search for the coherent neutrinoless muon-to-electron-conversion in the field of an aluminum nucleus ($\mu^- + {}^{27}_{13}\mathrm{Al} \rightarrow e^- + {}^{27}_{13}\mathrm{Al}$) [1]. The expected Mu2e single event sensitivity (SES) is:

$$R_{\mu e} \equiv \frac{\Gamma(\mu^- N(A,Z) \rightarrow e^- N(A,Z))}{\Gamma(\mu^- N(A,Z) \rightarrow \nu_\mu N(A,Z-1)^*)} = 3 \times 10^{-17}. \quad (1)$$

[*] e-mail: antonio.gioiosa@df.unipi.it

The current world's best limit $R_{\mu e} < 7 \times 10^{-13}$ (on gold) is from the SINDRUM II experiment at Paul Scherrer Institut [2]. In addition to Mu2e, the COMET experiment in preparation at J-PARC has an expected SES of 3×10^{-15} for Phase-I and $O(10^{-17})$ for Phase-II (on aluminum) [3], while the DeeMe experiment, also in preparation at J-PARC, has an expected SES of 10^{-13} (on carbon) [4].

The Mu2e apparatus includes three superconducting solenoids:

1. The Production Solenoid, where the 8 GeV pulsed proton beam (period of 1.7 μs) hits the tungsten production target and produces mostly pions;

2. The Transport Solenoid, which serves as a decay "tunnel" for the pions and performs a charge and momentum selection, thus producing the low-momentum μ^- beam;

3. The Detector Solenoid, where muons are stopped in the aluminum stopping target, form muonic atoms, then decay to 105 MeV$/c^2$ electrons, which are detected by state-of-the-art detectors for tracking and energy reconstruction.

The Mu2e Trigger and Data Acquisition System (TDAQ) is a streaming system with a software-only trigger designed to satisfy the following requirements [5][6]:

- Provide efficiency better than 90% for the conversion electron signal;
- Keep the total trigger rate below a few kHz - equivalent to approximately 7 PB/year of total data rate (to keep the collected data less than 21 PB in 3 years of running);
- Keep the processing time below 5 ms/event (that is the spill off time between two spills where the beam will be on).

To achieve these goals and allow for a higher off-detector data rate (desired with a 'software-only' trigger architecture), the Mu2e Data Acquisition System (DAQ) is based on a streaming readout. This means that all detector data are digitized, zero-suppressed in the front-end electronics and then transmitted off the detector to the DAQ. In this paper, we present the Mu2e Trigger and Data Acquisition System (TDAQ) and the Detector Control System (DCS) prototypes built at Fermilab's Feynman Computing Center. We also present the Mu2e online DAQ software suite *otsdaq* designed and developed at Fermilab. We report at end the integration of the DCS system into the online *otsdaq* software.

2 The TDAQ System

The Mu2e Trigger and Data Acquisition System (TDAQ) provides the necessary infrastructure to collect digitized data from the tracker, calorimeter, cosmic ray veto and monitor the beam status. The TDAQ employs 36 dual-CPU servers to handle a total rate of 192,000 proton pulses per second and an average of 5,400 events per second per server. According to preliminary estimates and DAQ bandwidth requirements, the detectors generate approximately 120 kB of zero-suppressed data per proton pulse for a resulting average total data rate of about 20 GB/s when beam is present [5]. Figure 1 shows the global TDAQ architecture.

The Run Control Host receives beam status and timing information from the Accelerator Control network. The Control Fanout (CFO) module in the Run Control Host is responsible for generating and synchronizing Readout Requests.

As its main function, each Read Out Controller (ROC), that has a large front end buffer to average over off-spill time, continuously streams out the zero-suppressed data collected between two proton pulses from the detectors to the DTCs (Data Transfer Controller). The data of a given time-frame are then collected in a single server using a 10 Gb/s switch. Then, the online reconstruction starts and a trigger decision is made. If an event gets triggered, the

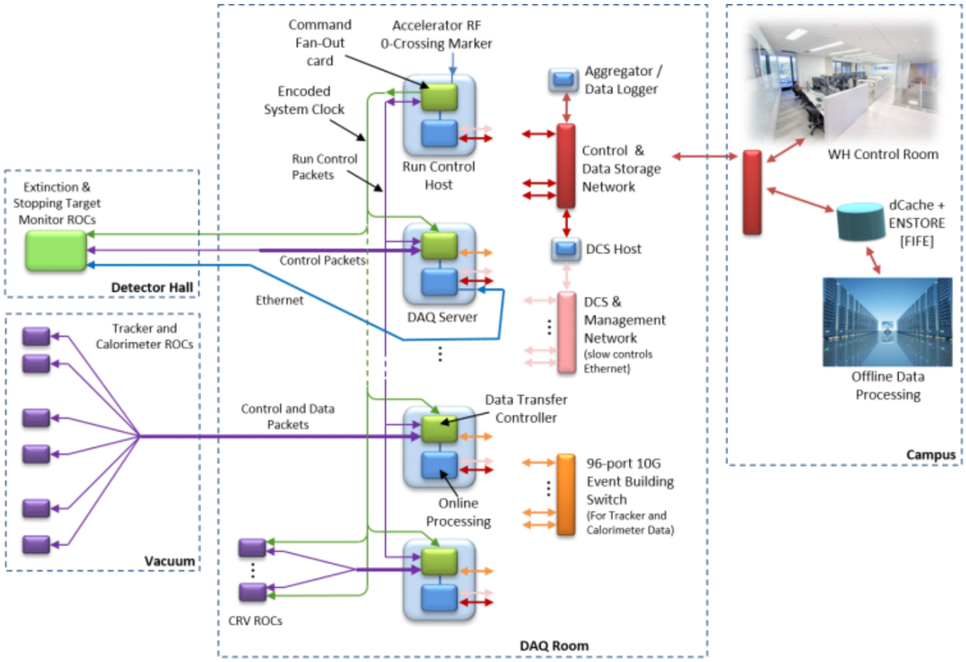

Figure 1. Mu2e TDAQ architecture and components diagram.

data from the cosmic ray veto (CRV) are pulled and aggregated into a single data stream. The DAQ servers filter these events (aggregator/data logger) and forward a small subset of them to the offline storage. A total of 497 ROCs and 83 DTCs will be used.

Figure 2 shows a scheme of the Mu2e data readout topology. In the Data Concentrator layer, ROCs data are directly collected by the DTCs via optical links. Using the 10 Gb/s ethernet network switch, events are then formed and triggered in the Event Builder layer where DTCs are involved in a round-robin process. At the end, events are moved via a Peripheral Component Interconnect Express (PCIe) bus to the Storage Layer for further filtering and data logging.

The TDAQ employes *otsdaq* as a software solution. Developed at Fermilab, it uses the *artdaq* [7, 8] and *art* [9] software as event filtering (data transfer, event building and event reconstruction) and processing frameworks. *otsdaq* includes a run control system using the data acquisition software XDAQ [10] implemented for the development and calibration-mode runs at CMS. The *otsdaq* development for Mu2e follows two main directions: server side and web side. The server side is developed in C++. The specific code for Mu2e is added through plugins (C++ classes inheriting from the appropriate base class). The web side (directly accessible to the end user through a web browser) is developed in HTML and JavaScript. Custom code for Mu2e is added in the form of web-apps through html files.

The Mu2e physics triggers identify signal event candidates [6]. It is implemented as a series of software filters applied after each step of track reconstruction. The total trigger rate is expected not to exceed 700 Hz [6]. With the *artdaq* framework, it is possible to limit the offline data storage to less than 7 PB/year with a reduction factor of about 100 at the event building level [11].

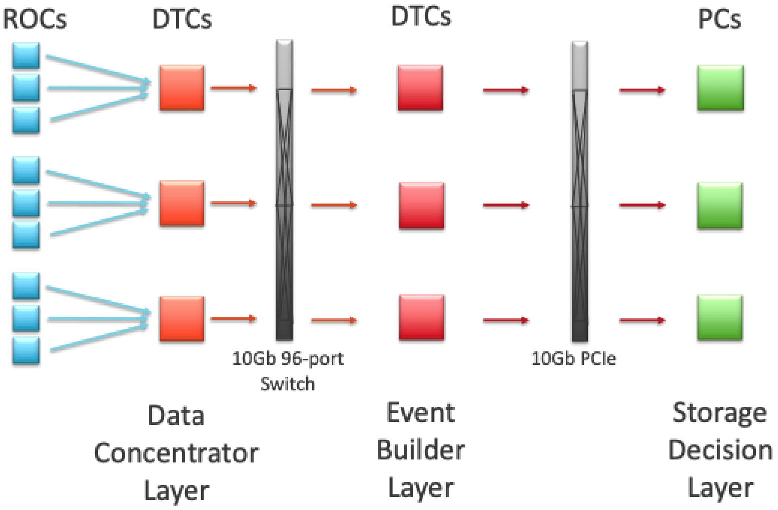

Figure 2. Mu2e data readout topology.

3 The Detector Control System

The Detector Control System (DCS) function is to monitor the detectors status and operational conditions. For each subsystem, the DCS allows to display real-time event and detector data via Graphical User Interfaces (GUIs) and archive the monitoring data to disk.

Mu2e selected EPICS (Experimental Physics and Industrial Control System) for the DCS slow control and monitoring software[12]. EPICS, with the Control System Studio (CSS) GUI software, is an open source framework originally developed at Argonne and Fermilab and now used in numerous experiments [13]. An Input Output Controller (IOC), running for each subsystem on a central DAQ server, will provide channels for all data [14]. The total number of slow control quantities is expected to be of the order of thirty thousand. On average, these quantities will be updated approximately twice per minute, for a resulting generated data rate of 10 kB/s.

As part of the DCS, *otsdaq* delivers slow control data from the DTCs and ROCs to EPICS. The *otsdaq* allows the user to monitor and interact with the DAQ hardware and the other devices managed by EPICS to:

1. Observe Process Variables (PVs) such as settings, alarms, warnings, readouts, timestamps, status;

2. Interact through a web interface that is lightweight, user-friendly, ready to use, customizable;

3. Implement custom handling of PV alarms integrated with the TDAQ state machine transitions.

The design of *otsdaq* involved C++ and web-app applications to include the Mu2e slow control monitoring, alarm handling, and TDAQ hardware and online daq slow control entities writing in EPICS. From *otsdaq* it is possible to handle alarm propagation from EPICS IOC to automated plug-in threads or to the web user interface and, ultimately, the users.

To connect *otsdaq* to EPICS, a C++ interface has been developed and it uses the EPICS Channel Access Client Library functions and Postgres database connections to read/write data. This interface allows for monitoring and alarm handling for the following PV data: Value, Alarm (Status, Severity) and Settings.

Figure 3 shows the C++ classes and hierarchy design of the *otsdaq* EPICS interfacing. The EPICS Interface is a child of the ots::ControlsVInterface Class that inherits Configurable and VStateMachine Classes (not shown). It allows to handle the State Machine transitions halt, configure, start, stop, pause and resume during State Machine traversal. Moreover, it is "Configurable", so it is possible to set PV names and alarms handling, using the *otsdaq* configuration tools. All these classes stem from the underlying XDAQ (used in *otsdaq*) framework that defines the actions on the controlled nodes by such state machine transitions.

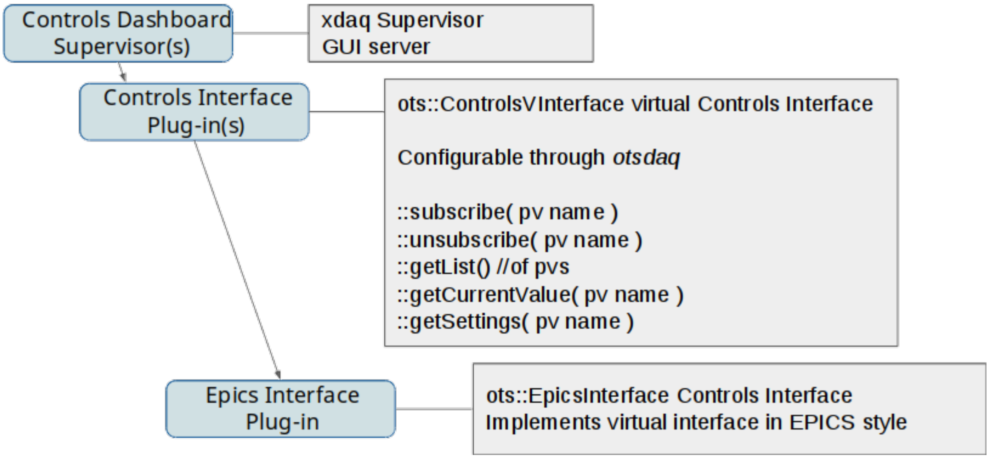
Controls Dashboard Supervisor(s)
xdaq Supervisor GUI server
Controls Interface Plug-in(s)
ots::ControlsVInterface virtual Controls Interface
Configurable through *otsdaq*
::subscribe(pv name)
::unsubscribe(pv name)
::getList() //of pvs
::getCurrentValue(pv name)
::getSettings(pv name)
Epics Interface Plug-in
ots::EpicsInterface Controls Interface
Implements virtual interface in EPICS style

Figure 3. *otsdaq* C++ classes designed for EPICS interfacing. ots::ControlsVInterface Class inherits Configurable and VStateMachine Classes (not shown).

otsdaq uses an *artdaq* interface to EPICS CA in order to write EPICS PVs for DAQ hardware and metrics.

Figure 4 shows the *otsdaq* slow control web GUI C++ classes diagram and web-app connections that explicit the GUI design and functioning. The Supervisor C++ Class ControlsDashboardSupervisor manages the *otsdaq* web clients requests, dispatching backward the EPICS Interface instance data collections. The HTML object SlowControlsDashboard forms the dashboard window directly handled by the end user which may have tools such as widgets editor and file pages access to retrieve/save slow control pages from/to the web server.
We developed the dashboard GUI to read/write slow control pages in the XML format compatible with CSS Phoebus version of EPICS GUI software.

4 Conclusions

In this article, we have presented the Trigger and Data Acquisition System (TDAQ) and Detector Control System (DCS) currently being developed for the Mu2e experiment at Fermilab.

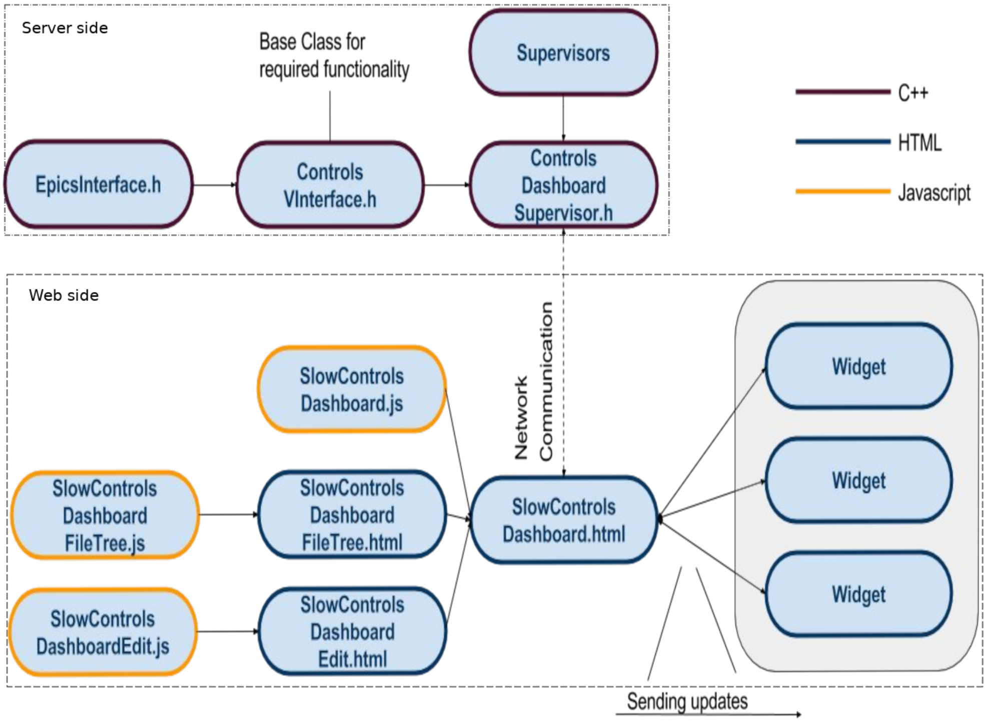
Server side
Base Class for required functionality
Supervisors
C++
HTML
Javascript
EpicsInterface.h
Controls VInterface.h
Controls Dashboard Supervisor.h
Web side
Network Communication
SlowControls Dashboard.js
SlowControls Dashboard FileTree.js
SlowControls Dashboard FileTree.html
SlowControls Dashboard.html
SlowControls DashboardEdit.js
SlowControls Dashboard Edit.html
Widget
Widget
Widget
Sending updates

Figure 4. *otsdaq* slow control web GUI C++ classes diagram and web-app connections scheme. General information on Interface plugin classes used in *otsdaq* is reported in section 2.

The TDAQ system uses the online DAQ software suite *otsdaq* developed at Fermilab to provide a high level of flexibility and scalability. We have reported on the preliminary results of the system performance. The Detector Control System (DCS) system uses the open source framework EPICS developed at Argonne and Fermilab and widely employed in a number of experiments, including CMS. The *otsdaq* system includes a part of DCS that communicates with EPICS. A run control GUI has been developed and integrated in *otsdaq* to provide a multi-user, web-based control and monitoring dashboard.

A test-stand equipped with the *otsdaq* web-based GUI was set up at Fermilab to test TDAQ and its timing performance. First level results of tests show that the requirement to achieve a total processing time/event below 5 ms is matched only if one event-builder/DAQ server is implemented [6]. With two or more event-builders/DAQ servers implemented, the competition for the shared resources and the increased architecture complexity degrades the system performance. The system is still being developed and a number of possible alternative solutions to mitigate these effects are currently being explored.

Concerning data reduction performances, the TDAQ system is able to limit the offline data storage to less than 7 PB/year. With the *artdaq* framework, it is possible to satisfy the requirement with a reduction factor of about 100 at the event building level [11]. Moreover, by default, Mu2e data are saved in *art* ROOT format to further reduce the amount of preprocessing necessary for offline analysis.

Acknowledgements

We are grateful for the vital contributions of the Fermilab staff and the technical staff of the participating institutions. This work was supported by the US Department of Energy; the Istituto Nazionale di Fisica Nucleare, Italy; the Science and Technology Facilities Council, UK; the Ministry of Education and Science, Russian Federation; the National Science Foundation, USA; the Thousand Talents Plan, China; the Helmholtz Association, Germany; and the EU Horizon 2020 Research and Innovation Program under the Marie Sklodowska-Curie Grant Agreement No. 690835, 734303, 822185, 858199, 101003460. This document was prepared by members of the Mu2e Collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359.

References

[1] R.H. Bernstein *et al*, “Charged Lepton Flavor Violation: An Experimenter’s Guide,” Phys. Rep., 532, 2, 27-64, 2013.

[2] W.H. Bertl *et al*, “A Search for muon to electron conversion in muonic gold,” Eur. Phys. J., C47 p. 337-346, 2006. doi:10.1140/epjc/s2006-02582-x.

[3] M. Lee, “Comet Muon conversion experiment in J-PARC,” Frontiers in Physics, v. 6 p. 133, 2018. doi:10.3389/fphy.2018.00133.

[4] N. Teshima, “Status of the DeeMe Experiment, an Experimental Search for μ-e Conversion at J-PARC MLF,” PoS, v. NuFact2019 p. 082, 2020. doi:10.22323/1.369.0082.

[5] L. Bartoszek *et al*, “Mu2e Technical Design Report”, 1501.05241, arXiv, physics.ins-det, 2015.

[6] D. Brown, G. Pezzullo, P. Murat, “The Track finder algorithm for the Trigger system of the Mu2e experiment at Fermilab,” Connecting The Dots, Princeton, 2020. *https://doi.org/10.5281/zenodo.4088480*

[7] K. Biery, E. Flumerfelt, J. Freeman, W. Ketchum, G. Lukhanin, A. Lyon, R. Rechenmacher, R. Rivera, L. Uplegger and M. Votava, “Flexible and Scalable Data-Acquisition Using the artdaq Toolkit,” [arXiv:1806.07250 [physics.ins-det]], 2018.

[8] K. Biery, C. Green, J. Kowalkowski, M. Paterno and R. Rechenmacher, “artdaq: An event filtering framework for Fermilab experiments,” doi:10.1109/RTC.2012.6418358, 2012.

[9] C. Green, J. Kowalkowski, M. Paterno, M. Fischler, L. Garren and Q. Lu, “The Art Framework,” J. Phys. Conf. Ser. 396, 022020 (2012) doi:10.1088/1742-6596/396/2/022020.

[10] G. Bauer et al., “The CMS data acquisition system software,” J. Phys. Conf. Ser., vol. 219, p. 022011, 2010.

[11] Kurt Biery et al 2017 J. Phys.: Conf. Ser. 898 032013.

[12] https://epics-controls.org/, 2020.

[13] https://epics-controls.org/epics-users/projects/, 2020.

[14] “EPICS Record Reference Manual” http://www.ing.iac.es/ docs/external/epics/recordref.pdf.