Prototype Data Acquisition and Slow Control Systems for the Mu2e Experiment

A. Gioiosa, R. Bonventre, S. Donati, E. Flumerfelt, G. Horton-Smith, L. Morescalchi, V. O'Dell, E. Pedreschi, G. Pezzullo, F. Spinella, L. Uplegger, and R. A. Rivera

***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. Such a charged lepton flavor-violating reaction probes new physics at a scale unavailable with direct searches at either present or planned high energy colliders.**

The Mu2e Trigger and Data Acquisition System (TDAQ) exploits *otsdaq* as its 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 Read Out Controllers (ROC) stream out zero-suppressed data continuously from the detector subsystems to the Data Transfer Controllers (DTC). The data stream is then read over the Peripheral Component Interconnect Express (PCIe) bus to a software filter algorithm that selects events which are combined with the data flux coming from a Cosmic Ray Veto System (CRV). 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.

***Index Terms*—Mu2e, trigger, daq, slow control, muon**

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.

A. Gioiosa is with Università di Pisa and INFN Sezione di Pisa, I-56127 Pisa, Italy (email: antonio.gioiosa@df.unipi.it).

R. Bonventre is with Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA (email: rbonventre@lbl.gov).

S. Donati is with Università di Pisa and INFN Sezione di Pisa, I-56127 Pisa, Italy (email: simone.donati@unipi.it).

E. Flumerfelt is with Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA (email: eflumerf@fnal.gov).

G. Horton-Smith is with the Department of Physics, Kansas State University, Manhattan, Kansas 66506, USA (email: gahs@phys.ksu.edu).

L. Morescalchi is with INFN Sezione di Pisa, I-56127 Pisa, Italy (email: luca.morescalchi@pi.infn.it).

V. O'Dell is with Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA (email: odell@fnal.gov).

G. Pezzullo is with Yale University, New Haven, Connecticut, 06520, USA (email: g.pezzullo@yale.edu).

E. Pedreschi is with INFN Sezione di Pisa, I-56127 Pisa, Italy (email: elena.pedreschi@pi.infn.it).

F. Spinella is with INFN Sezione di Pisa, I-56127 Pisa, Italy (email: franco.spinella@pi.infn.it).

L. Uplegger is with Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA (email: uplegger@fnal.gov).

R. A. Rivera is with Fermi National Accelerator Laboratory, Batavia, Illinois 60510, USA (email: rrivera@fnal.gov).

I. Introduction

Lepton Flavor Violation (LFV) has been observed in the neutral sector in the 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 ($\mu^- + {}^{13}_{27}Al \to e^- + {}^{13}_{27}Al$) [1]. The expected Mu2e single event sensitivity (SES) is:

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

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 $\mathcal{O}(10^{-17})$ for Phase-II (on aluminum) [3], while the DeeMe experiment, also in preparation at J-PARC, has an expected SES of 10^{-14} (on carbon) [4].

The Mu2e apparatus includes three superconducting solenoids (Figure 1):

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, which houses the aluminum stopping target, where the muons get stopped and form the muonic atoms, and the detectors system (in the 1 T solenoidal magnetic field) optimized to detect the muon conversion electrons.

Mu2e will search for a signal of mono-energetic electrons with an energy just below 105 MeV (muon rest mass). This requires state-of-the-art electron track and energy reconstruction. The Mu2e detector consists of a straw-tracker and a

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

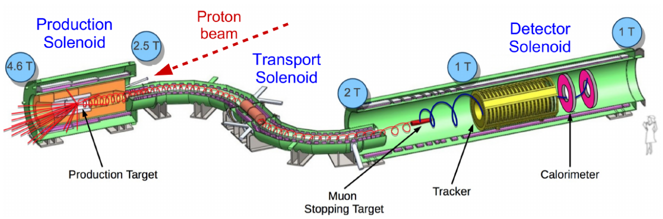

Fig. 1. The Mu2e experimental apparatus as depicted in [7]. The cosmic-ray veto system is not shown.

crystal electromagnetic calorimeter. Both the tracker and the calorimeter have cylindrical symmetry. The inner volume of both detectors has an opening that allows the free passage of non-interacting beam particles and makes the detectors blind to low-momentum ($p < 80$ MeV/c) charged particles that are largely produced in the stopping target.

The 3.2 m long straw-tracker consists of 36 equally spaced tracking planes, made out of 6 rotated panels arranged over two faces. A panel represents the basic unit of the tracker; it consists of two staggered layers of straw tubes and has 96 straws in total. The total number of channels in the tracker is 46,080 and the estimated global data rate is 12 GB/s. A more detailed description of the tracker can be found in [5] and [6].

The crystal calorimeter is composed of two annuli with inner and outer radii of 37.4 cm and $66,0$ cm, respectively, filled with pure CsI scintillating crystals and is placed downstream of the tracker. Each annulus is composed of 674 crystals of $(34 \times 34 \times 200)$ mm^3 volume, each readout by a custom array of UV-extended Silicon PhotoMultipliers (SiPMs). The total number of calorimeter channels is 3,840 and the estimated global data rate is 8 GB/s. More details on the detector can be found in [7].

Cosmic ray muons are the dominant source of background in Mu2e. They can interact with the detector material and produce electrons that mimic the conversion electrons signal. The simulation shows this will occur at a rate of about one conversion-like event per day. To reduce this source of background, the entire Detector Solenoid and half of the Transport Solenoid are covered with a cosmic-ray veto system (CRV) made of four layers of extruded plastic scintillator bars.

Mu2e employes a beam consisting of bunches approximately 200 ns long and separated by 1.7 microseconds. The required suppression of protons outside the beam pulse is at the fractional level of 10^{-10}. The verification of this level of extinction is delegated to a special purpose spectrometer (extinction monitor) which will measure particles scattered from the production target of the experiment.

The Mu2e Trigger and Data Acquisition System (TDAQ) is a streaming system with a software-only trigger designed to satisfy the following requirements [6] [8]:

1) Provide efficiency better than 90% for the conversion electrons signal;
2) Keep the total trigger rate below a few kHz - equivalent to approximately 7 PB/year of total data rate;
3) Keep the processing time below 5 ms/event.

To achieve these goals and allow for a higher off-detector data rate, 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. This provides maximum flexibility for data processing and filtering. The DAQ architecture is further simplified by the integration of all off-detector components in a "DAQ Server" that functions as a centralized controller, data collector and processor. A single DAQ Server can be used as a complete standalone data acquisition and processing system. Multiple DAQ Servers can be connected together to form a highly scalable system.

In the following, 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 and a summary of the expected trigger and timing performance, including the results of preliminary performance tests. We also report on the integration of the DCS system into the online *otsdaq* software.

II. 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 is responsible for online data processing, event selection and storage. It also performs detector synchronization, control, monitoring, and provides operator interfaces.

The TDAQ employs 36 dual-CPU servers to handle a total rate of 192,000 micro-bunches per second and an average of 5,400 events per second per server. According to preliminary estimates, the detectors generate approximately 120 kB of zero-suppressed data per micro-bunch for a resulting average total data rate of about 20 GB/s when beam is present [6]. To reduce bandwidth requirements, front-end buffers provide temporary data storage during the spill period. Data are then transmitted to the TDAQ over the full supercycle, during the long off-spill time. Figure 2 shows the global TDAQ architecture.

The DAQ Timing System supplies a continuous low-jitter system clock to the Read Out Controllers (ROCs) and the Run Control Host. This clock is synchronized to the accelerator Delivery Ring frequency (590,018 Hz), and the leading edge of the clock is aligned with the start of each micro-bunch.

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. It sends a Readout Request control packet for each system clock. The system clock runs continuously, whether or not there is beam. This allows readout (e.g., acquisition of calibration or pedestal data) at any time in the accelerator supercycle.

Each ROC generates its own internal high-speed digitization clock, phase locked to the system clock. Each ROC also generates an internal timestamp for data within the microbunch timing window. For the ROCs outside the detector solenoid, the timing system supplies a separate clock.

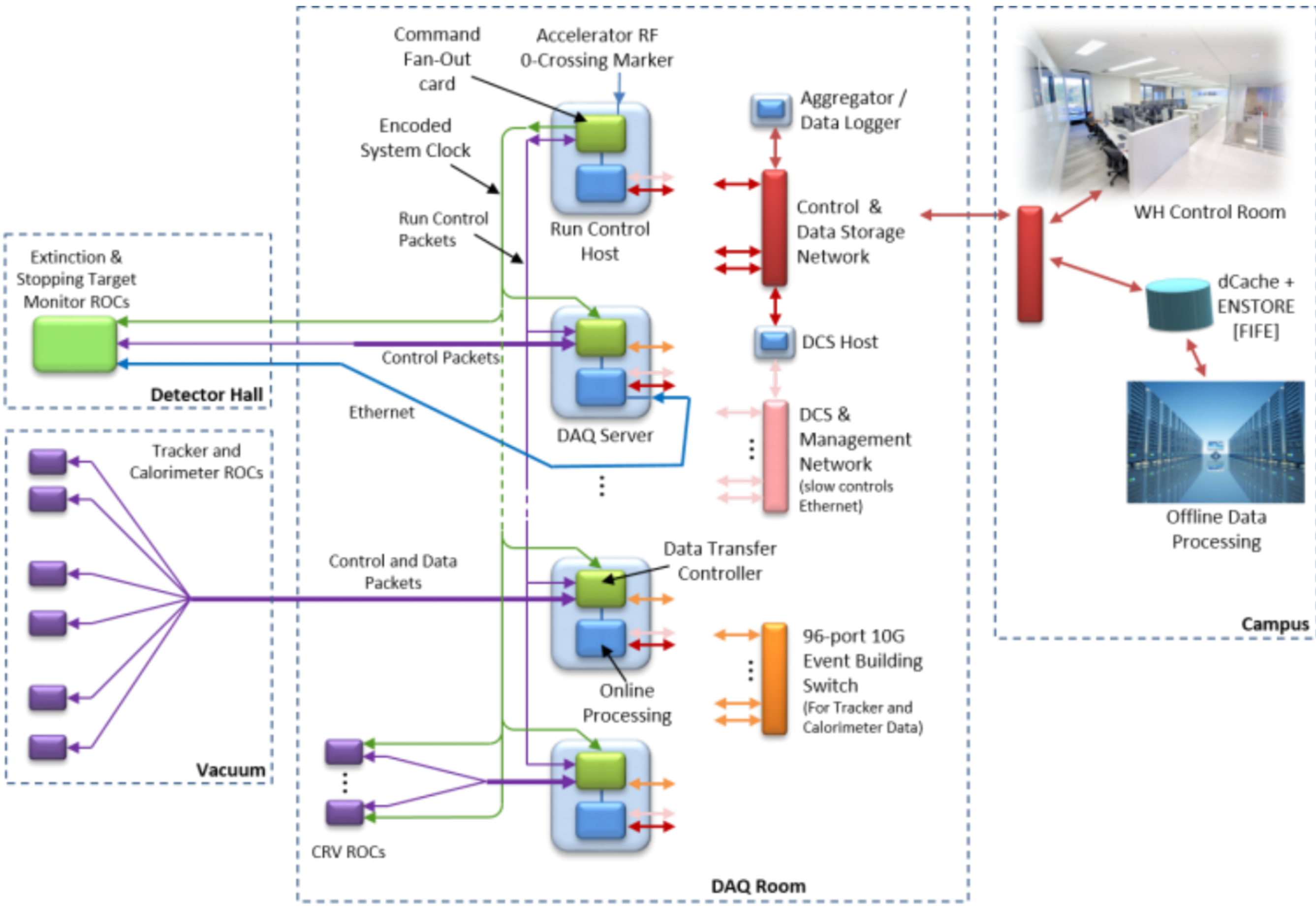

Fig. 2. Mu2e TDAQ architecture and components diagram.

In addition to the internal timestamp generated by each ROC, there is a System Timestamp generated by the Control Fanout Module in the Run Control Host. The System Timestamp is sent by the CFO to the Data Transfer Controllers (DTCs) at each system clock. The DTCs broadcast the timestamp to the connected ROCs in a Readout Request packet. The DTCs also send a System Timestamp as part of Data Request packets, and the ROCs return the System Timestamp in the Data Header packet. Sending the System Timestamp directly to the ROCs for each system clock (instead of relying on a timestamp generated from the clock itself) avoids loss of ROC event synchronization with the rest of the system as a result of missing or extra clocks.

The ROCs continuously stream out the zero-suppressed data collected between two proton-pulses from the detectors to the DTCs. 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, also the data from the CRV are pulled and aggregated in a single data stream. The DAQ servers filter these events (aggregator/data logger) and forward a small subset of them to the offline storage. Figure 3 shows a scheme of the Mu2e data readout topology. The total number of ROC (497) employed for the entire Mu2e experiment and the estimated data rate for each detector are listed in Table I [6].

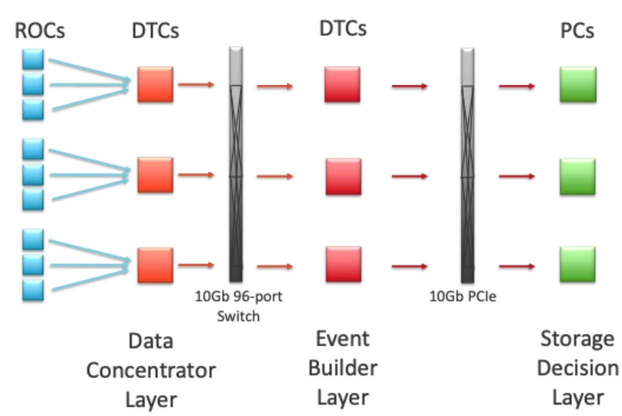

Fig. 3. Mu2e data readout topology. In the Data Concentrator layer, ROCs data are directly collected by the DTCs via optical links. Using a ethernet network switch, events are then formed and triggered in the Event Builder layer where DTCs are involved in a round-robin process. At the end, events are moved via a Peripheral Component Interconnect Express (PCIe) bus to the Storage Layer for further filtering and data logging.

A total of 240 ROCs are employed for the tracker, 240 for the calorimeter, 15 for the cosmic ray veto, 2 for the extinction and target monitor, respectively). Since each DTC is connected to six ROCs, the total number of DTCs is 83.

TABLE I
READOUT CONTROLLERS AND OPTICAL LINKS BY SUBDETECTOR.

Detector	Number of ROCs	Average Rate per ROC [MB/s]	Total Data Rate [GB/s]	Number of Optical Links	Number of DAQ Servers
Tracker	240	50	12	120	20
Calorimeter	240	30	8	72	12
Cosmic Ray Veto	15	20	1	15	3
Extinction and Target Monitors	2	low	<1	4	1

The CRV data are not used for the physics triggers mainly for two reasons:

1) The CRV veto is implemented as a 100 ns time-window w.r.t. the reconstructed muon stub. Since one Mu2e event is 1.7μs long, the presence of a muon stub in the CRV does not imply that the entire event should be discarded. If a signal candidate is detected far apart in time, the Mu2e strategy is to use it.
2) The CRV readout system does not allow to pull the detector data for all events. With the current CRV electronics, a simple logic would not match the Mu2e requirements since the time needed to pull the CRV data is too long.

The TDAQ employs *otsdaq* as a software solution. Developed at Fermilab, it uses the *artdaq* [9], [10] and *art* [12] 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 [13] implemented for the development and calibration-mode runs at CMS. *otsdaq* supports existing front-end boards and firmware modules (some with custom variants of User Datagram Protocol, for example), and also allows for user defined front-end interface plugins. Additionally, an integrated run control Graphical User Interface (GUI) provides a multi-user, web-based control and monitoring dashboard through a web browser. In Mu2e, a custom front-end interface plugin communicates with the detector ROC firmware of the tracker and calorimeter through the Data Transfer Controller (DTC). In *otsdaq*, a ROC/DTC data emulator has been developed and tested and a ROC/DTC connection integrated and successfully tested with the real DTC hardware and a tracker ROC prototype.

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 class). Supervisor classes handle a collection of related interface plugin classes. For example, Front-End Interfaces handle the control and management of front-end devices. Their type and number depend on the design and number of the front-end devices. *artdaq* plugins are strictly required and involved in event building. They include Board Readers, Event Builders, Aggregators (data logging and dispatching) in a number that depends on the system nodes number.

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 (including the appropriate JavaScript (.js) and Cascading Style Sheets (.css) files).

The Mu2e physics triggers exploit the information of the reconstructed tracks and calorimeter clusters to make the final decision [8]. The online track reconstruction is factorized into three main steps:

1) Hit preparation: the digitized signals from the tracker and calorimeter are converted into reconstructed hits. The low-momentum (a few MeV) “background” electrons hits that may be generated in the tracker by photons that undergo Compton scattering, photo-production, or pair production, and by delta-ray emission from electrons and protons, are identified and flagged using a Multi-Variate-Analysis (MVA) [6];
2) Pattern recognition: two complementary pattern recognition algorithms identify hits that form helical trajectories; the first algorithm is tracker-seeded, the second one is calorimeter-seeded [8];
3) Track fitting: the fit of helical trajectories through the hits identified by the pattern recognition algorithms is performed. The achieved momentum resolution is about 1.4 MeV/c for the conversion electrons, that is a factor of two improvement w.r.t. the resolution after pattern recognition [8].

III. TDAQ PERFORMANCE

The Mu2e trigger is implemented as a series of software filters applied after each step of track reconstruction. These filters are set up with different settings and the *artdaq* architecture allows to apply all of them w/o running the reconstruction modules multiple times in the same event. This is especially important to minimize the online processing time.

The performance of the track triggers optimized to select the conversion electrons signal and the calibration samples has been estimated w.r.t. the offline reconstruction using simulated data. The efficiency for the conversion electrons signal is 98% for the logical “OR” of the calorimeter-seeded and the tracker-seeded algorithms, while it is only 87% for the stand-alone calorimeter-seeded algorithm, or 90% for the stand-alone tracker-seeded algorithm, respectively. The total trigger rate is expected not to exceed 700 Hz [8]. The rate is dominated by low-momentum electrons from the stopping target and can be reduced by prescaling the low-momentum track triggers or by increasing the momentum threshold, with no significant impact on the conversion electrons signal efficiency.

To study the TDAQ timing performance, a test-stand equipped with the *otsdaq* web-based GUI was set up at Fermilab. The results of the 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 [8]. If this number is incremented, 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, the TDAQ processes the data from 497 ROCs and applies filters to reduce the average data volume by a factor of at least 100 before data can be transferred to offline storage, thus limiting 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 pre-processing necessary for offline analysis.

IV. 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 GUIs and archive the monitoring data to disk. The data include high-level displays of the power supplies, cryogenic liquid and gas system operational parameters, external and internal environmental temperatures (e.g. of the experimental Hall, detectors, computer crates, and control room), magnetic field intensity, and the DAQ status of every detector subsystem. (Figure 4).

Mu2e selected EPICS for the DCS implementation [14]. 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 [15]. The system is robust, supports a large number of monitoring variables and has an excellent performance in terms of monitoring and archiving rates. It can be interfaced with a large number of systems to establish a two-way communication for the control and monitor. The DCS supplies Ethernet networking and Ethernet-based controllers for the general control and monitor of power supplies, beamline equipment, and environmental sensors. The DCS does not interface to safety-related controls or monitors for purposes other than a secondary interlock system (with a relay layer driven by software), and status logging.

To provide a complete monitor and control of the detector subsystems, the DCS includes the following components:

1) ***Control systems base***: performs input and output operations and defines processing logic, scan conditions, alarm conditions, archiving rate, etc.;
2) ***Alarm server***: monitors all channels and sends alarm messages to the operators;
3) ***Data archiver***: performs automatic sampling and storage of values for history tracking;
4) ***Integrated operator interface***: provides display panels for controls and monitors.

The EPICS-based software can be interfaced with external systems (e.g. the cryogenics control system) and databases (e.g. the beam database) to export data into the slow control process variables (or channels), named pieces of data associated with a machine (e.g. status, readback, and setpoint parameter), for archiving and status displays. This allows integrating displays and warnings into one system and provides archiving capability for sampled data in the archive database.

The archiver software allows data storage, with adjustable rates and thresholds, in a SQL database such that it is easy to retrieve data for any channel by using channel name and time range. The alarm server software keeps track of the channels state (e.g. alarm type and severity), supports an arbitrary number of clients, and provides logic for delayed alarms and acknowledgement alarms. As part of the software, a standard naming convention for channels is followed to aid dealing with the large number of channels and subsystems.

An Input Output Controller (IOC), running for each subsystem on a central DAQ server, will provide “soft” channels for all data [16].

The total number of slow control quantities (Observe Process Variables) 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, customisable;
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 software data writing in EPICS.

From *otsdaq* it is so possible to record all PVs related to TDAQ hardware, software and *artdaq* metric into EPICS. It is also 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. The interface uses the EPICS Channel Access Client Library functions and Postgres database connections to read/write data.

Figure 5 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.

For data monitoring and alarm handling the EPICS Interface uses Channel Access (CA) subscription to EPICS to retrieve Process Variable (PV) data: Value, Alarm (Status, Severity) and Settings of all PVs registered into EPICS IOC and

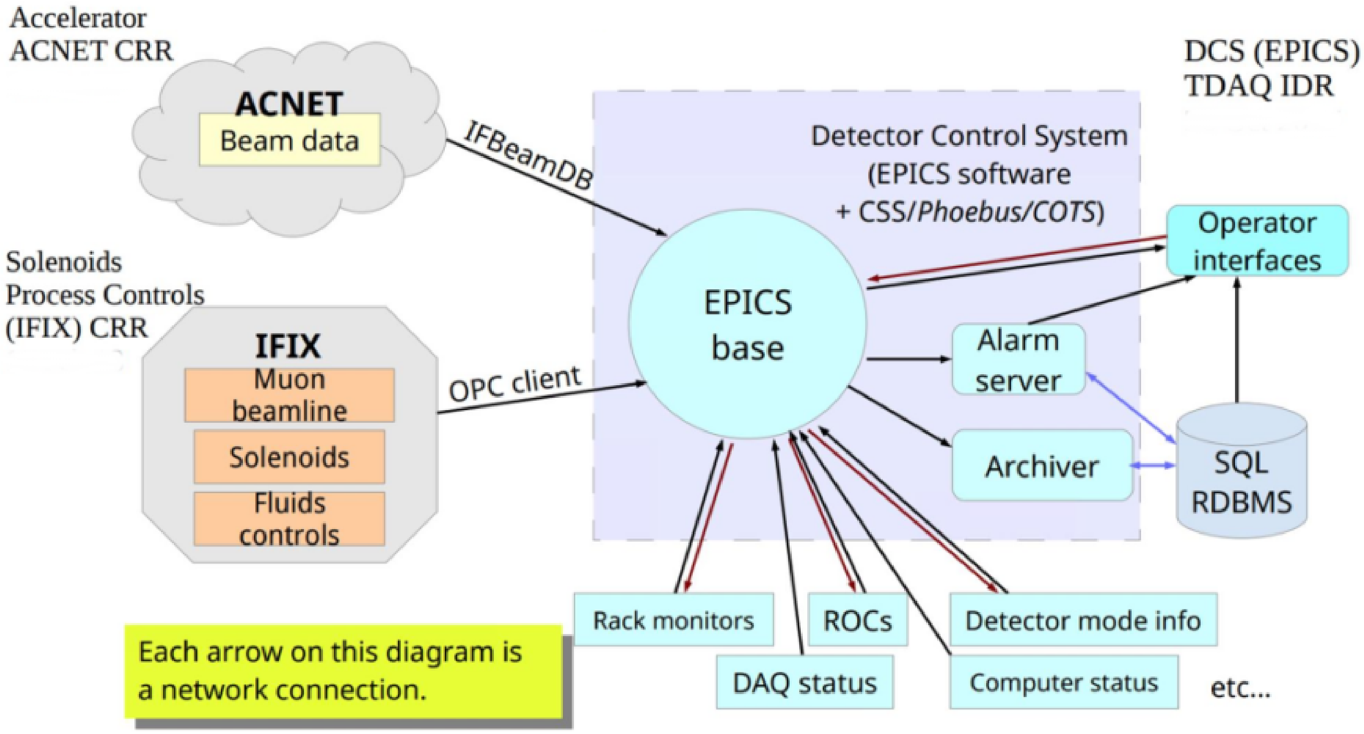

Fig. 4. Mu2e DCS layout. The slow control system EPICS is connected through the network to the Mu2e components: ACNET is the Accelerator Control NETwork, a system of computers that monitors and controls the accelerator complex; the IFIX system monitors physics variables like temperature and pressure and uses OLE for Process Control (OPC) protocol to communicate with EPICS; the ROCs are the Read Out Controllers and the SQL RDBS is the database management system where the slow control channels settings and history are stored.

databases. For data writing (DAQ hardware data and metrics) into EPICS *otsdaq* uses an *artdaq* Interface specifically developed: the *artdaq* EPICS metric plugin that use the CA library presented before.

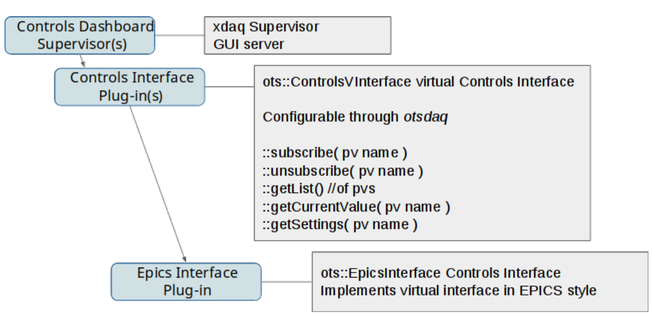

Fig. 5. *otsdaq* C++ classes designed for EPICS interfacing.

Figure 6 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.

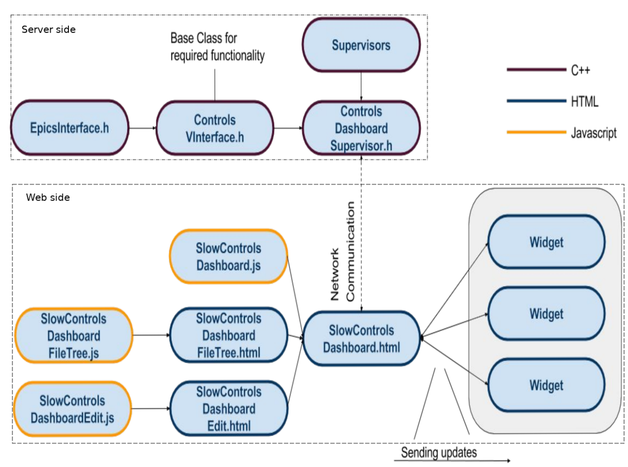

Fig. 6. *otsdaq* slow control web GUI C++ classes diagram and web-app connections scheme. General information on Interface plugin classes used in *otsdaq* is reported in section II.

Figure 7 shows the GUI graphical aspect. It includes a slow control dashboard with the following features: a library of widgets, like thermometer, gauge, chart, text, label, 2D stop-light matrix, etc., template widgets for customization, quick-snap pages for easy viewing, UI Scalability for high pixel screens or TVs, variable polling rates, Drag and Drop

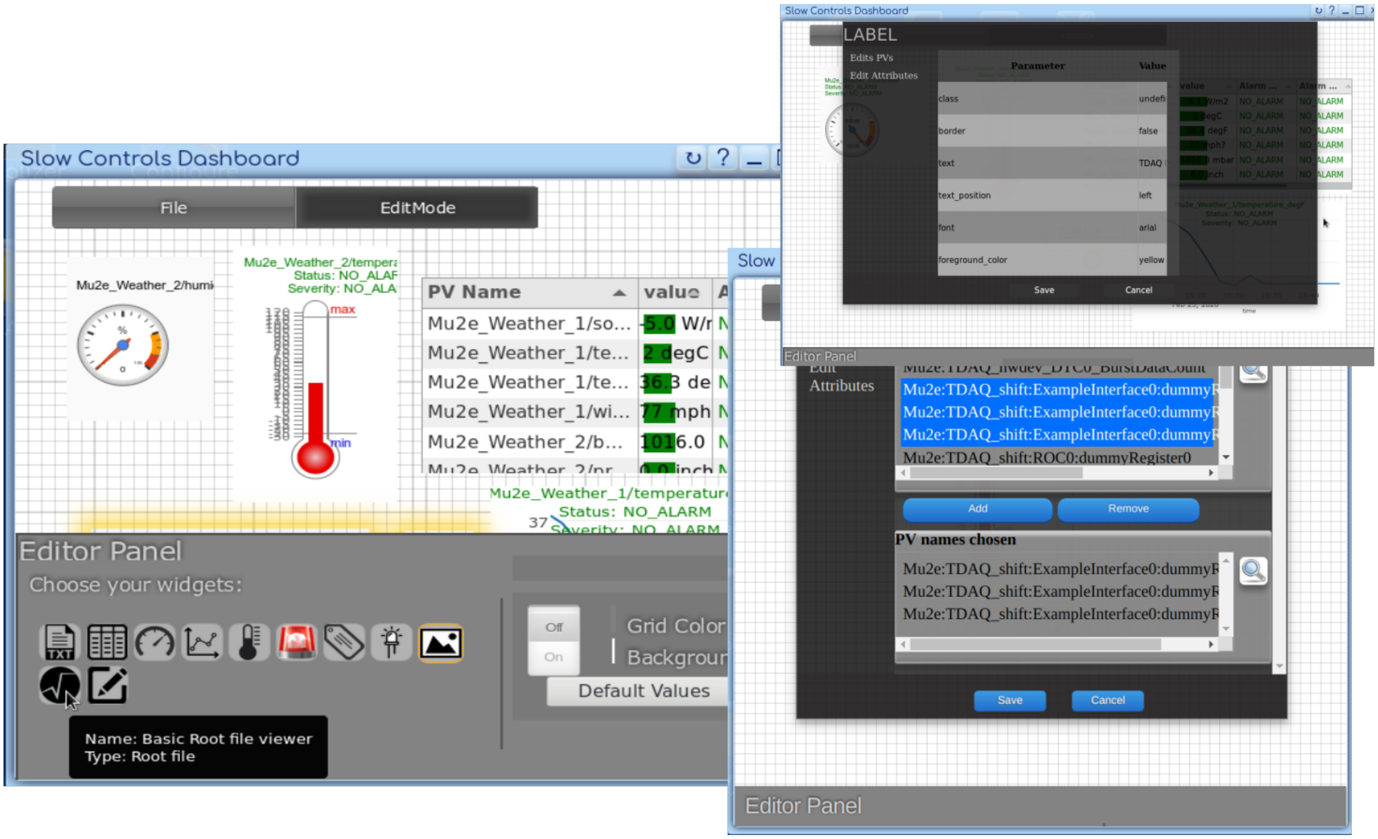

Fig. 7. Examples of DCS web GUI developed within the *otsdaq* framework. A number of widgets and the editor are shown.

widgets, snap grid, infinite UI color customization, simple notes and names incorporated into the settings of each widget, page saving and loading. It also includes configurable system message alarm notifications that use the web and mail services to broadcast alerts to users.

V. 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. The TDAQ system exploits 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 study.

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.

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] G. Pezzullo, "The Mu2e Tracker 39th International Conference on High Energy Physics," FERMILAB-CONF-18-660-E,11/2018.

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

[7] N. Atanov *et al*, "The Mu2e Calorimeter Final Technical Design Report," 1802.06341, arXiv, physics.ins-det, FERMILAB-DESIGN-2018-01, 2018.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

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

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

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

[16] "EPICS Record Reference Manual" http://www.ing.iac.es/ docs/external/epics/recordref.pdf.