Available on CMS information server **CMS CR -2017/310**

The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

28 September 2017 (v3, 02 October 2017)

Improving the Safety and Protective Automatic Actions of the CMS Electromagnetic Calorimeter Detector Control System

Raul Jimenez Estupinan for the CMS Collaboration

Abstract

The CMS ECAL Detector Control System (DCS) features several monitoring mechanisms able to react and perform automatic actions based on pre-defined action matrices. The DCS is capable of early detection of anomalies inside the ECAL and on its off-detector support systems, triggering automatic actions to mitigate the impact of these events and preventing them from escalating to the safety system. The treatment of such events by the DCS allows for a faster recovery process, better understanding of the development of issues, and in most cases, actions with higher granularity than the safety system. This paper presents the details of the DCS automatic action mechanisms, as well as their evolution based on several years of CMS ECAL operations.

Presented at *ICALEPCS 2017 International Conference on Accelerator and Large Experimental Control Systems*

IMPROVING THE SAFETY AND PROTECTIVE AUTOMATIC ACTIONS OF THE CMS ELECTROMAGNETIC CALORIMETER DETECTOR CONTROL SYSTEM

R. Jiménez Estupiñán, D. Di Calafiori, G. Dissertori, L. Djambazov, W. Lustermann (ETH Zurich, Switzerland), P. Adzic, P. Cirkovic, D. Jovanovic, P. Milenovic (University of Belgrade, Serbia), S. Zelepoukine (ETH Zurich, Switzerland / University of Wisconsin-Madison, USA)

Abstract

The CMS ECAL Detector Control System (DCS) features several monitoring mechanisms able to react and perform automatic actions based on pre-defined action matrices. The DCS is capable of early detection of anomalies inside the ECAL and on its off-detector support systems, triggering automatic actions to mitigate the impact of these events and preventing them from escalating to the safety system. The treatment of such events by the DCS allows for a faster recovery process, better understanding of the development of issues, and in most cases, actions with higher granularity than the safety system. This paper presents the details of the DCS automatic action mechanisms, as well as their evolution based on several years of CMS ECAL operations.

INTRODUCTION

During the first LHC long shutdown, the CMS ECAL DCS experienced a major upgrade [1, 2] to improve the control system. After this upgrade, the computing hardware was optimized from fifteen standalone to only three more powerful redundant servers. The software was adapted accordingly, to monitor and control the ECAL Barrel (EB), ECAL Endcaps (EE) and Preshower (ES) partitions in a redundant and distributed environment.

The CMS ECAL DCS software was originally written by developers with distinct programming styles, not following a common strategy for several mechanisms, including the automatic protective actions. When an automatic action is executed, experts are requested to investigate the cause and, once the problem is solved, to certify the detector readiness for data taking. The traceability of the automatic actions depends on the implementation of each mechanism, requiring sometimes considerable efforts to reach a complete understanding of its cause.

Different methods to perform DCS automatic actions are compared in the following paragraphs, highlighting the strengths and weakness of each model.

What is a DCS automatic action?

A DCS automatic action is a software mechanism to judge events in the DCS context and eventually to perform control operations in an automated way. These automatic actions consist of the following stages:

- Evaluation of conditions.
- Execution of actions.
- Notification.

CMS ECAL DCS Automatic actions are safety preventive actions for switching off detector subsystems with fine granularity at the partitions level. The report of these actions also help operators and experts to analyse large amounts of data in a short time. In this aspect, the DCS automatic actions can be compared to an *expert system*. The successful execution of the automatic actions depends on the availability of the underlying technology and cannot be always guaranteed due to the complex technology stack. For this reason, they should not be considered as a substitute for the CMS ECAL Detector Safety System (DSS).

Evaluation of conditions

The CMS ECAL DCS monitors thousands of input parameters (process variables) from different hardware devices. Multiple signals are sampled at a certain rate and stored in the control system as discrete-time parameters. The signals registered by the control system are subject to a certain information loss due to the discretization and manipulation by different processes. Once the signals are stored in the control system as process variables, they can be analysed, combined and shaped into Boolean expressions. Boolean expressions can be combined to build triggering conditions, resulting in a *true* or *false* statement to execute the automatic actions.

Execution of automatic actions

The DCS automatic actions are software operations that aim to mitigate problems in a controlled way. These actions are translated into control commands to switch off one or more hardware devices, helping to guarantee the detector and related off-detector hardware integrity.

Notifications

The notification of automatic actions makes the experts aware of a problematic situation in real time. Summary messages are sent via e-mail and SMS to provide a starting point for the failure assessment.

CURRENT IMPLEMENTATIONS

Two different models of automatic actions can be found in the current implementation: the Barrel and Endcaps use Finite State Machines (FSM) while the Preshower uses control scripts.

Automatic actions in the ECAL Barrel and Endcaps using the Finite State Machine

The CMS ECAL DCS implements part of the detector's control using the CERN FSM toolkit [3, 4].

The CMS ECAL FSM is organized in 40 partitions following the detector physical layout (36 supermodules [SM] and 4 half endcaps [Dee]). The FSM partitions are sub-divided in the following components: low voltage node (LV), high voltage node (HV), cooling node (C), Safety node (S) and a support mechanism known as distinguisher (D). The distinguisher mechanism allows the FSM to break down complex commands into specific sequences of hardware instructions as shown in Fig. 1. Each FSM partition computes information about several components to determine its overall status. The FSM commands are available to the operators for controlling the detector subsystems. Some of these commands are also used internally by the FSM to execute automatic actions.

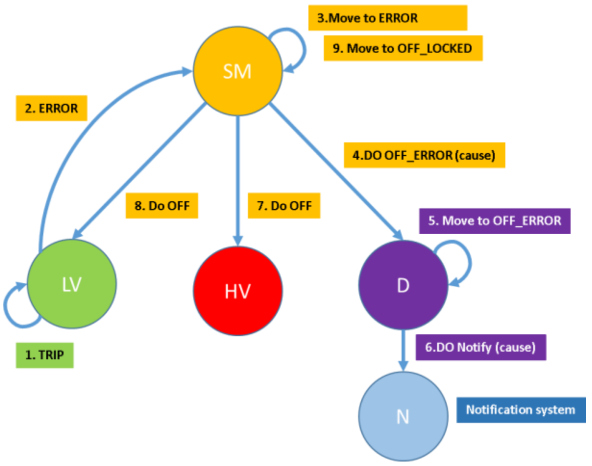

Figure 1: FSM automatic actions for EB/EE partitions

Figure 1 shows the sequence of internal FSM commands when a low voltage channel trips. The automatic action begins with the detection of a problem in one of the low voltage power supplies. When a low voltage channel trips, the LV FSM node computes an error. The LV error is transferred to its parent node SM and then it is forwarded to the distinguisher D. The SM node sends the *off* command to the HV power supplies and waits until all the high voltage elements are powered off. The distinguisher node remembers the reasons for switching off and influence the behaviour of the SM node to trigger the next command. Only after powering off the HV, the *off* command can be sent to the LV power supplies.

This implementation covers all the stages for an automatic action: the FSM is able to detect and compute a problem in the hardware, the automatic action is executed using the FSM commands and the notification is delegated to the CMS notification system.

The current tree-like organization of the FSM facilitates the tracing of problems when an automatic action is executed. On the other hand, this model presents the following disadvantages:

- Two different operational features are mixed in one place: manual control from the user interface and automatic control in form of automatic actions. Commands sent by a user can interfere with the automatic actions, and lead on to a wrong execution;
- DCS applications are divided in software components, each of them responsible for its part of the FSM. When any of these components is reinstalled, the automatic actions in the FSM are temporarily unavailable.

Automatic actions in the Preshower detector using control scripts

The ES is partitioned in Control Rings (CR). The DCS implements automatic actions, using control scripts, to power off individual CRs. Dedicated control scripts monitor the detector environment sensors and the status of the powering subsystems, reacting to failures detected by the DCS. The ES automatic actions target individual power channels, boards and converters to protect the electrical equipment against possible damages.

Control scripts have direct access to all the process variables in the WinCC OA run-time database, including some data from the distributed systems. Compared to other models, CTRL scripts are not limited to a specific scope in the WinCC OA platform. Data from various contexts and applications can be easily combined to produce automatic actions.

DATA FILTERING FOR AUTOMATIC ACTIONS

The data from multiple sensors can be used to trigger an automatic action. The sensor signals are sampled, packed and transported across electronic equipment and systems which can be temporarily unavailable. The sensors data can be affected by a loss of precision and by a temporary failure of the sensor itself (bad readings). The loss of precision is sometimes unavoidable and intrinsic to the acquisition process. However, a filtering stage is introduced to ignore bad readings and to handle other problems related to the acquisition process.

In order to illustrate the impact of data quality in the execution of the automatic actions, let's consider the following example on Fig. 3: The function *h(t)* describes the evolution of a sensor value over the time. Values above the threshold indicate a problem that should be handled by the DCS automatic actions (E.g. over temperature inside one of the CMS ECAL partitions). The sensor value crosses the threshold at t1, and goes below the threshold at t2. The sensor value crosses again the threshold at t3, and stays above it until t7. The first over-threshold event (t1) can be caused by a bad sensor readout. The second event (t3) last longer and the automatic action must be triggered after 3 time units (t6).

Data filtering using Control Script (CTRL)

A CTRL script can be programmed to monitor hardware parameters and to execute actions when certain conditions are reached. In the WinCC OA platform, the data is received by a certain driver (D), processed by the WinCC OA event manager (EV) and stored in the runtime database (DB). The CTRL script can retrieve new data only after the event manager process it and the two processes operate at different rates. For certain driver configurations, the CTRL scripts could be under-sampling and missing values that could have been relevant for executing an action.

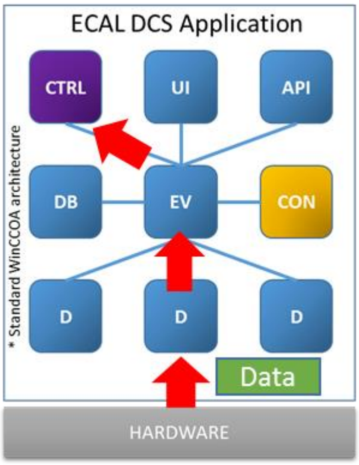

Figure 2: Standard WinCC Open Architecture (OA).

In the Figure 3, a CTRL script polls data every 3 time units. If the polling process begins at t1, then the values at t2 and t3 are ignored. A second value is retrieved at t4. At this point, two consecutive values are registered above the threshold and the conditions for triggering the automatic action apply without considering the intermediate values.

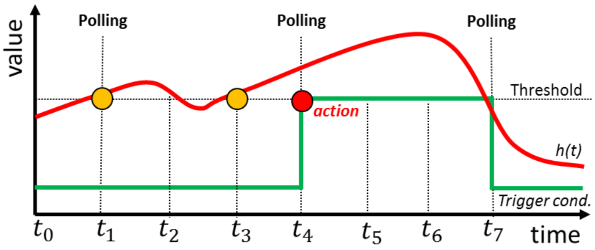

Figure 3: Under-sampling issue in CTRL scripts

Data filtering using statistical functions

Statistical functions can be configured directly in the data structures of the WinCC OA applications. This type of function is computed by the EV and automatically evaluated when new data is available.

The function *f(t)* [Eq. (1)] calculates if a set of consecutive values of *h(t)* are above or below the threshold for a given time interval, as shown in Fig. 4.

The sensor values are flatted into a binary signal using the function *g(t)* [Eq. (2)]. The function *g(t)* calculates if the sensor value is above or below a threshold. The area described by the function *g(t)* represents the duration of the sensor value above the threshold.

$$f(t) = \begin{cases} 1 \; if \; \sum_i^{freq\,\Delta t} g(t_i) = freq \times \Delta t \\ 0 \; if \; \sum_i^{freq\,\Delta t} g(t_i) \neq freq \times \Delta t \end{cases} \quad (1)$$

$$g(t) = \begin{cases} 1 \; if \; h(t) \geq VThreshold \\ 0 \; if \; h(t) < VThreshold \end{cases} \quad (2)$$

$$t_i = T.i \qquad freq = \frac{1}{T} \quad (3)$$

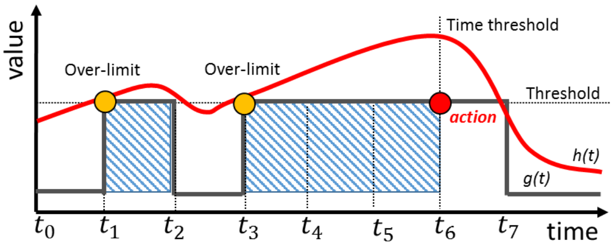

Figure 4: Data filtering using statistical function.

Using this method, the triggering conditions are updated at the same rate as the event manager. The usage of this mechanism increases the accuracy of the triggering conditions and simplifies the design of the automatic actions.

AUTOMATIC ACTIONS ON REMOTE SYSTEMS

The CMS ECAL DCS software runs on three machines which are part of a larger distributed system (CMS DCS). Some of the resources of the CMS ECAL detector are monitored and controlled by external applications in the CMS DCS environment. The automatic actions on external systems require a communication channel between the triggering system and the system hosting the remote resources. The CMS ECAL DCS team is currently evaluating a new type of automatic actions across remote systems.

Remote automatic actions using the WinCC OA distributed connection

The CMS DCS is based on a centralized network of servers using the WinCC OA distribution mechanism. The distribution mechanism is a standard WinCC OA feature that enables the access to the internal data structures across WinCC OA systems. Distributed connections are very reliable and widely used in large control systems. On the other hand, the usage of distributed connections has a strong impact on the software design. Distributed connections forces the applications to be aware of each other internal structures, making the systems highly interdependent and less easy to adapt to changes.

Remote automatic actions using FSM commands

The CMS ECAL is operated though a FSM. Multiple sub-detectors FSM trees are connected, forming a complete CMS tree structure with top nodes at the CMS DCS level. FSM commands are offered at various levels of the FSM and transferred down to the CMS ECAL FSM. In this sense, the FSM mechanism is used as an abstraction layer to communicate between sub-systems. Commands across multiple FSM trees can be transferred in a descendant direction to execute remote automatic actions. Commands in ascendant direction are not possible. The FSM toolkit permits the creation of links between scattered FSM trees (see Fig. 5). The usage of this feature impacts the spanning tree design of the FSM. The introduction of cycles in the tree can alter to the original behaviour FSM, resulting in indefinite loops in the status calculation and a possible misleading execution of commands [5].

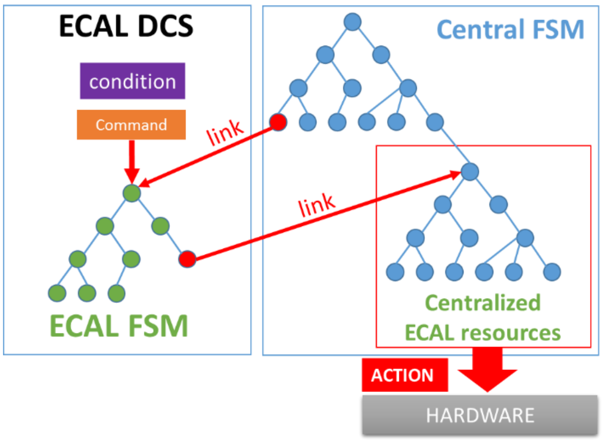

Figure 5: FSM commands across scattered FSM trees

Remote procedure calls using web services

The DCS automatic actions performed out of the subdetector's context reveal the need for a service-oriented architecture in the DCS applications. Some of the CMS DCS applications were designed as standalone control applications and are not fully ready to share functionality. Since 2016, a new DCS communication mechanism known as the CMS XML-RPC system was introduced. The CMS XML-RPC system is an implementation of the standard Remote Procedure Call (RPC) protocol using the Extensible Mark-up Language (XML) to model, encode and expose a certain set of operations using HTTP as transport protocol (Fig. 6). The CMS XML-RPC permits to organize the DCS software as services and to expose its functionality to other applications using the XML-RPC protocol.

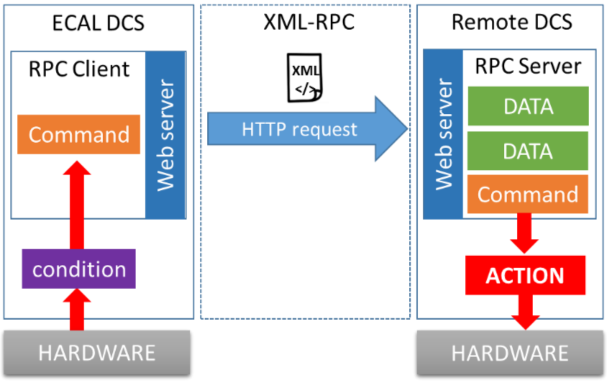

Figure 6: DCS actions using XML-RPC communication

The CMS ECAL DCS team is currently evaluating the usage of this protocol to implement remote automatic actions on some of the CMS DCS centralized resources.

CONCLUSION

The experience gained during previous years of operation has been a valuable resource to improve the CMS ECAL DCS software. Motivations for new automatic actions are eventually identified and implemented to ease operations and improve the detector safety. In addition, the extension of automatic actions presents an opportunity to define common strategies and consequently, allows for further integration between applications. Therefore, the implementation of a common framework for automatic actions can make the CMS DCS and the CMS ECAL DCS more powerful, flexible, and easier to maintain.

ACKNOWLEDGEMENT

The authors would like to thank the Swiss National Science Foundation for the financial support.

We would like to also thank our colleague O. Holme who provided insight and expertise in the topics discussed in this publication.

REFERENCES

[1] O. Holme et al., “Re-integration and consolidation of the detector control system for the Compact Muon Solenoid Electromagnetic Calorimeter”, Proceedings of the International Conference on Accelerator & Large Experimental Physics Control Systems 2013, San Francisco, California, United States.

[2] D. Di Calafiori et al., “Maintaining and improving the control and safety systems for the Electromagnetic Calorimeter of the CMS experiment”, Proceedings of the International Conference on Computing in High Energy and Nuclear Physics 2012, New York, United States.

[3] B. Franek et al., “SMI++ Object Oriented Framework for Designing and Implementing Distributed Control Systems”, Presented at: IEEE Rome 2004 and published in IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL 52, NO. 4, AUGUST 2005.

[4] “Joint Controls Project (JCOP) Framework sub-project guidelines and conventions”, JCOP Framework team, 20th October 2010.

[5] Yi Ling Hwong et al., “An Analysis of the Control Hierarchy Modeling of the CMS Detector Control System”, Presented at CHEP2010: International Conference on Computing in High Energy and Nuclear Physics 2010, 18-22 Oct 2010, Taipei (Taiwan).