Available on CMS information server

CMS CR -2017/188

The Compact Muon Solenoid Experiment

Conference Report

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

29 June 2017 (v2, 07 July 2017)

Common software for controlling and monitoring the upgraded CMS Level-1 trigger

Giuseppe Codispoti for the CMS Collaboration

Abstract

The Large Hadron Collider restarted in 2015 with a higher centre-of-mass energy of 13 TeV. The instantaneous luminosity is expected to increase significantly in the coming years. An upgraded Level-1 trigger system was deployed in the CMS experiment in order to maintain the same efficiencies for searches and precision measurements as those achieved in 2012. This system must be controlled and monitored coherently through software, with high operational efficiency.The legacy system was composed of a large number of custom data processor boards; correspondingly, only a small fraction of the software was common between the different subsystems. The upgraded system is composed of a set of general purpose boards, that follow the MicroTCA specification, and transmit data over optical links, resulting in a more homogeneous system. The associated software is based on generic components corresponding to the firmware blocks that are shared across different cards, regardless of the role that the card plays in the system. A common database schema is also used to describe the hardware composition and configuration data. Whilst providing a generic description of the upgrade hardware, this software framework must also allow each subsystem to specify different configuration sequences and monitoring data depending on its role.We present here, the design of the control software for the upgrade Level-1 Trigger, and experience from using this software to commission the upgraded system.

Presented at *TIPP2017 International Conference on Technology and Instrumentation in Particle Physics 2017*

Common software for controlling and monitoring the upgraded CMS Level-1 Trigger

Giuseppe Codispoti[1,2], Simone Bologna[3] Glenn Dirkx[2], Christos Lazaridis[4], Alessandro Thea[5], and Tom Williams[5]

[1] University of Bologna and INFN, Bologna, Italy,
[2] CERN, Geneva, Switzerland,
[3] University of Milan-Bicocca and INFN, Milano, Italy,
[4] Dept. of Physics, University of Wisconsin-Madison, U.S.A
[5] STFC Rutherford Appleton Laboratory, Harwell Oxford, U.K.

Abstract. The Large Hadron Collider restarted in 2015 with a higher centre-of-mass energy of 13 TeV. The instantaneous luminosity is expected to increase significantly in the coming years. An upgraded Level-1 Trigger (L1T) system was deployed in the CMS experiment in order to maintain the same efficiencies for searches and precision measurements as those achieved in 2012. This system must be controlled and monitored coherently through software, with high operational efficiency. The legacy system was composed of a large number of custom data processor boards; correspondingly, only a small fraction of the software was common between the different components. The upgraded system is composed of a set of general purpose boards, that follow the μTCA specification, and all transmit data over optical links, resulting in a more homogeneous system. The associated software is based on generic components corresponding to the firmware blocks that are shared across different cards, regardless of the role that the card plays in the system. A common database schema is also used to describe the hardware composition and configuration data. Whilst providing a generic description of the upgrade hardware, this software framework must also allow each subsystem to specify different configuration sequences and monitoring data depending on its role. We present here the design of the control software for the upgrade L1T, and experience from using this software to commission the upgraded system.

1 Introduction

The Compact Muon Solenoid (CMS) experiment Level-1 Trigger (L1T) selects the most interesting 100 kHz of events out of 40 MHz collisions delivered by the CERN Large Hadron Collider (LHC). The LHC restarted in 2015 with centre-of-mass energy of 13 TeV and increasing instantaneous luminosity.

In order to cope with the increasing event rate, CMS L1T underwent a major upgrade [1] in 2015 and early 2016. The VME-based system has been replaced by custom-designed processors based on the μTCA specification, containing FPGAs and larger memories for the trigger logic and high-bandwidth optical links connections. The final system, composed of 9 main components (subsystems), accounts for a total of about 100 boards and 3000 optical links and is only partially standardized: 6 different design of processor boards are used and each subsystem is composed of a different number of processors and implements different algorithms. Approximately 90% of the software has been rewritten in order to control and monitor the new system: in order to mitigate the rcode duplication, the *Online Software* was redesigned to exploit the partial standardisation and impose a common hardware abstraction layer for all the subsystems.

We present here the design of the control software for the upgraded L1T and the tools developed for controlling and monitoring the subsystems.

2 The hardware abstraction layer: SWATCH

A CMS L1T subsystem is composed of one or more processor boards hosted in μTCA crates, each equipped with a module providing clock, controls and data acquisition services. Across different board designs and subsystems, the processor boards have the same logical blocks: input/output ports; an algorithmic block for data processing; a readout block sending data from the input/output buffers to the CMS Data Acquisition system; a trigger Timing and Control block receiving clock and zero-latency commands. This common abstract description of the subsystems and the processor boards is the founding concept of SWATCH[2], SoftWare for Automating the conTrol of Common Hardware, a common software framework for controlling and monitoring the L1T upgrade.

SWATCH models the generic structure of a subsystem through a set of abstract C++ classes. Subsystem-specific functionalities are implemented in classes that inherit from them. The objects that represent the subsystem are built using factory patterns and an XML-formatted description of the system. Hardware access for both test and control purposes is generalized through the concept of *Command*: a stateless action, represented by an abstract base class, customised through subtype polymorphism. *Commands* can be chained in *Sequences*. A *Finite State Machine* (FSM) defines the possible state of the subsystem and is connected with the high-level CMS run control states: *Commands* and *Sequences* are the building blocks for the transitions between the FSM states. The parameters used by the commands are provided through the *Gatekeeper*, a generic interface providing uniform access to the SWATCH application both from files and database. Similarly, items of monitoring data are represented through the *Metric* class that provides a standard interface for accessing their values, registering error and warning conditions, and persistency. *Metrics* can be organized in tree-like structures within *Monitorable Objects*, whose status is the cumulative status of all constituent *Metrics*. Values stored in *Metrics* are automatically published to an external service and then stored in an Oracle database through a dedicated process. *Metrics* considered crucial for the data taking are stored directly through a separate monitoring thread.

Common monitoring and control interfaces are provided through web pages based on *Google-Polymer* based on modern web technologies (ES6-javascript,

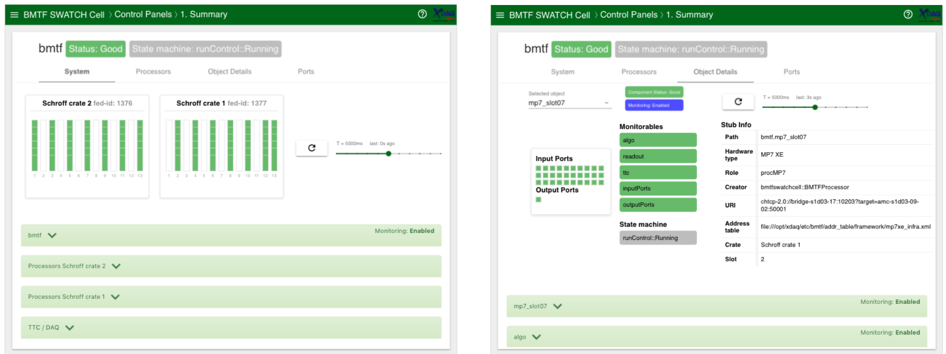

Fig. 1. System (left) and Processor (right) views of a subsystem in the SWATCH web interface. Copyright 2017 CERN for the benefit of the CMS Collaboration. CC-BY-4.0 license.

SCSS, HTML5). The XDAQ[3] framework, the CMS platform for distributed data acquisition systems, provides the networking layer to drive the FSM from central CMS Run Control and publish monitoring information.

3 Definition and persistency of hardware configuration

The system description and the input parameters for *Commands* and FSM transitions are stored in an *Oracle* database, ensuring that the hardware configuration that was used for any given test or data taking run can be identified, and re-used if required. The database schema is based on a tree-like structure of foreign *keys* that mimics the structure of the L1T. Subsystems' configurations are then split in logical blocks and stored in form of XML chunks.

To simplify the process of editing of the L1T configuration a *custom database Configuration Editor*(*L1CE*) has been developed. The *L1CE* was designed as a client-server application to enable multi-user access whilst keeping safe, atomic database sessions attached to the user session. Both client and server are developed using web technologies: the server runs a *Node.js* application, while the client is a web page based on *Google-Polymer*. This choice minimises the number of technologies involved, thus reducing development and maintenance effort and keeping a native, efficient communication between the two parts.

The *L1CE* implements unambiguous bookkeeping of the configurations, by imposing naming conventions and versioning at all levels, author identification through the CERN Single-Sign-On mechanism and in general explicit and automatic metadata insertion. It also implements in-browser XML editing and comparison between different configurations at all levels.

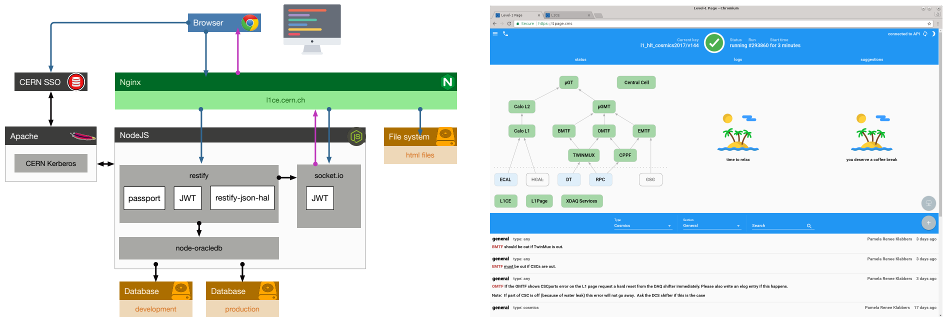

Fig. 2. The architecture of the L1CE (left) and a view of the Level-1 Page (right). Copyright 2017 CERN for the benefit of the CMS Collaboration. CC-BY-4.0 license.

4 The Level-1 page

A web application called *Level-1 page* provides a single access point for shifters and experts to all of the L1T monitoring information, including the running conditions and the warnings and alarms from all trigger subsystems. Through a concise visual representation of the system, including the interconnections among trigger subsystems and with CMS detectors, it allows non-experts (e.g. control room shift personnel) to identify the source of problems more easily. It also provides access and control for the trigger processes, links to the documentation and relevant contacts, and a list of reminders that the user can filter based on the type of the ongoing run (e.g. collisions, cosmics, tests, etc.). The *Level-1 page* is based on the same architecture as the *L1CE* in which the server is responsible for aggregating the information from etherogeneous sources.

5 Conclusions

The CMS L1T has been completely replaced in 2015 and early 2016 and commissioned in a very small time window. The *Online Software* has been rewritten to accomodate its new structure, exploiting hardware standardization and imposing a common abstraction model. In this way, the design of the new *Online Software* increased the fraction of common code and reduced the development time and maintenance effort, playing a vital role in the commissioning of the new system and enhancing the data taking reliability. Moreover the design choices will simplify implementation of new features, continuously adapting to the needs of the next years of data taking.

References

1. The CMS collaboration: CMS Technical Design Report for the Level-1 Trigger Upgrade, CERN, Geneva 2013. CMS-TDR-12

2. Bologna, S. et al.: SWATCH: Common software for controlling and monitoring the upgraded level-1 Trigger of the CMS experiment , Proceedings of 20th IEEE-NPSS Real Time Conference (RT2016), doi:10.1109/RTC.2016.7543077
3. XDAQ, https://svnweb.cern.ch/trac/cmsos, Accessed 25th June 2017