Available on CMS information server

CMS CR -2007/062

The Compact Muon Solenoid Experiment

Conference Note

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

4 October 2007

The Run Control and Monitoring System of the CMS Experiment

Gerry Bauer, Vincent Boyer , James Branson , Angela Brett , Eric Cano , Andrea Carboni , Marek Ciganek , Sergio Cittolin , Vivian O dell , Samim Erhan , Dominique Gigi , Frank Glege , Robert Gomez-Reino , Michele Gulmini , Johannes Gutleber , Jin Cheol Kim , Markus Klute , Elliot Lipeles , Juan Antonio Lopez Perez , Gaetano Maron , Frans Meijers , Emilio Meschi , Roland Moser , Esteban Gutierrez Mlot , Steven Murray , Alexander Oh , Luciano Orsini , Christoph Paus , Andrea Petrucci , Marco Pieri , Lucien Pollet , Attila Racz , Hannes Sakulin , Matteo Sani , Philipp Schieferdecker , Christoph Schwick , Konstanty Sumorok , Ichiro Suzuki , Dimitrios Tsirigkas , Joao Varela

Abstract

The CMS experiment at the LHC at CERN will start taking data in 2008. To configure, control and monitor the experiment during data-taking the Run Control and Monitoring System (RCMS) was developed. This paper describes the architecture and the technology used to implement the RCMS, as well as the deployment and commissioning strategy of this important component of the online software for the CMS experiment.

The Run Control System of the CMS Experiment

Gerry Bauer [9], Vincent Boyer [5], James Branson [6], Angela Brett [5], Eric Cano [5], Andrea Carboni [5], Marek Ciganek [5], Sergio Cittolin [5], Vivian O'dell [8], Samim Erhan [5,7], Dominique Gigi [5], Frank Glege [5], Robert Gomez-Reino [5], Michele Gulmini [5], Johannes Gutleber [5], Jin Cheol Kim [4], Markus Klute [9], Elliot Lipeles [6], Juan Antonio Lopez Perez [5], Gaetano Maron [2], Frans Meijers [5], Emilio Meschi [5], Roland Moser [1,5], Esteban Gutierrez Mlot [5], Steven Murray [8], Alexander Oh [5], Luciano Orsini [5], Christoph Paus [9], Andrea Petrucci [2], Marco Pieri [6], Lucien Pollet [5], Attila Racz [5], Hannes Sakulin [5], Matteo Sani [6], Philipp Schieferdecker [5], Christoph Schwick [5], Konstanty Sumorok [9], Ichiro Suzuki [8], Dimitrios Tsirigkas [5], Joao Varela [3,5]

1. Vienna University of Technology, Vienna, Austria
2. INFN - Laboratori Nazionali di Legnaro, Legnaro, Italy
3. LIP, Lisbon, Portugal
4. Kyungpook National University, Daegu, Kyungpook, South Korea
5. CERN, Geneva, Switzerland
7. University of California, San Diego, La Jolla, California, USA
6. University of California, Los Angeles, Los Angeles, California, USA
8. FNAL, Batavia, Illinois, USA
9. Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

Abstract. The CMS experiment at the LHC at CERN will start taking data in 2008. To configure, control and monitor the experiment during data-taking the Run Control and Monitoring System (RCMS) was developed. This paper describes the architecture and the technology used to implement the RCMS, as well as the deployment and commissioning strategy of this important component of the online software for the CMS experiment.

1. Introduction

The CMS experiment [1] at the CERN Large Hadron Collider is currently being commissioned. First pp collision data are foreseen for the end of 2008. CMS has a general purpose detector to explore physics at the TeV scale. It consists of various sub-systems to measure the different types of particles produced in the collisions of protons.

The Run Control [2] controls and monitors the collection of event data from the detector elements. About 3000 PC running O(10^4) online processes are involved. The managing of configuration and control of these components is the responsibility of the Run Control system. A sketch of the read-out and control system of CMS is shown in Figure 1.

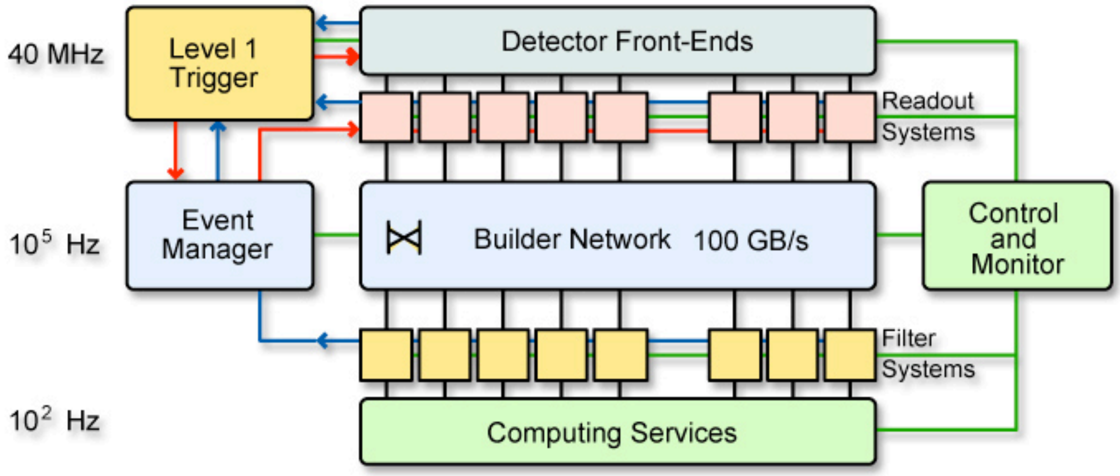

Figure 1 Read-out and control system of CMS.

The online software uses a common framework written in C++ (xdaq [3]) and runs on Scientific Linux. A centralised cluster management system ensures a coherent and reproducible installation of the software. The online processes are embedded in the hierarchical control structure of Run Control.

The Run Control is based on a set of web-applications implemented with Java Servlet technology, AJAX and JSP for user interfaces, and a data base back-end. A hierarchical control structure organizes the Run Control in Sub-systems. A set of tools have been developed to manage the flexible generation of configurations with the goal to allow fast reconfiguration of the system. A crucial test was passed with the so called "Magnet Test & Cosmic Challenge" of CMS - a small set of sub-detectors being operated to detect cosmic muons - during 2006. Towards the first run of the CMS experiment the Run Control system will be tested in another "Cosmic Challenge" exercise with the sub-detectors, DAQ and trigger components in place in the underground cavern.

2. Requirements

The Run Control and Monitor System (RCMS) is the collection of hardware and software components responsible for controlling and monitoring the CMS experiment during data taking. It provides physicists with a single point of entry to operate the experiment and to monitor detector status and data quality. The interface enables users to access and control the experiment physicists and operators can perform all programmable actions on the system.

The three main requirements are:

- Provide interactive graphical user interfaces to operate the CMS experiment.
- Manage the correct configuration of all components.
- Control and monitor of the data acquisition system and relevant sub-detector systems.

3. Technologies

Web technologies and related developments play a strong role in the implementation of the RCMS. The development process profits from a rich choice of tools and solutions based on Web technologies.

The RCMS services are implemented in Java 1.5.0 as Web Applications. The XML data format and the SOAP protocol are used for inter process communication. The interfaces are specified with the Web Service Description Language (WSDL) using the Apache Axis implementation of Web Services (WS). Tested Web Service clients include Java, LabView and Perl. The application is running in the official reference implementation of the Java Servlet technology Tomcat 5 by the Apache Software Foundation.

For persistency both Oracle 10g and MySQL 5 are supported by RCMS. The DBMS to store and retrieve configuration information is accessible through custom Java packages.

4. Design and Implementation

The RCMS is one of the principal components of the online system (see Figure 2). The Run Control structure is organized into eleven different sub-systems, a sub-system is corresponding to a sub-detectors, e.g. to the Hadron Calorimeter, central DAQ or global trigger. The RCMS framework provides a uniform API to common tasks like storage and retrieval from the process configuration DB, state-machine models for process control, and access to the monitoring system.

A tree of finite state machines implemented with RCMS controls the data taking operation of the experiment. The so-called "Function Manager" (FM) consists of a finite state machine and a set of services and is the basic element in the control tree. The state machine model has been standardized for the first level of FM's in the control tree to facilitate the integration and commissioning of sub-detectors.

A set of services are accessible to the FMs. The services comprise a security service for authentication and user account management, a resource service for storing and delivering configuration information of online processes, access to remote processes via resource proxies, error handlers, a log message application to collect, store and distribute messages, and Job Control to start, stop and monitor processes in a distributed environment.

Since the generation of configurations is complex task when many components are involved, an application has been developed to generate configurations for the DAQ applications, the so called Configurator. The Configurator generates configurations programmatically by combining information about the available hardware and knowledge of the online software DAQ components and fills those into the Resource Service.

In the following some components of RCMS will be discussed in more detail.

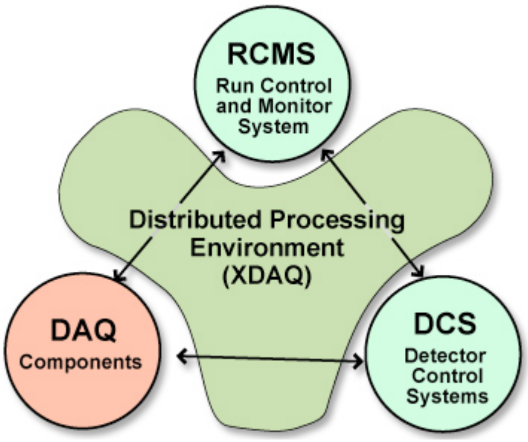

Figure 2 Components of the online system.

4.1. Function Manager

The principal components of the Function Manager (FM) are listed below and shown in Figure 3.

- Input Handler: this module handles all the inputs to the FM. Inputs can be commands from FMs or Operator via the GUI, or state notifications and error notifications from the controlled resources.
- Event Processor: Receives messages from the Input Handler. The messages are processed dependent on their message type. Typically the sequence of processing consists of a generic framework part and a custom FM part. According to an user defined behaviour, such messages can trigger a state transition and/or act on the Resource Proxy to directly control the associated resources.

- Finite State Machine (FSM) Engine: a user customizable module to define states and transitions and implement actions. The FSM Engine takes care of executing the actions and driving the state transitions according to the received inputs. The Moore FSM model has been adopted.
- Resource Proxy: Abstraction layer of the remote resource to control. The Resource Proxy hides the communication protocol and process control from the FSM Engine. It provides a common API to control all resources. The module is extensible with a plug-in mechanism and allows to easily accommodate different kinds of resources.

The FM is the basic entity of the control system. To customize the control to the needs of a sub-system it is sufficient to modify the FM. The implementation of the FM is separated from the framework of RCMS and is managed by the sub-system integrating with central Run Control.

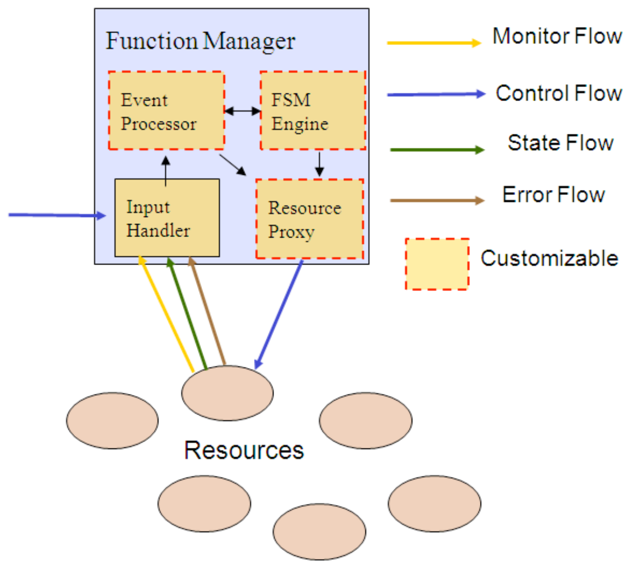

Figure 3 Principal components of the Function Manager.

4.2. Resource Service

The Resource Service (RS) holds the configuration information about the FMs and their controlled resources. A resource can be any hardware or software component involved in a setup. Resources can be allocated and queried and setups can only use available resources. It uses a DBMS to store the information. The main requirements for the RS are listed below.

- Store and retrieve configuration information from DBMS.
- Scale to the expected complexity of the CMS online system.
- Versioning of modified configurations for full traceability.
- Flexibility to accommodate new types of Resources.

The implementation of the RS support both Oracle 10g and MySQL 5. The instances of the RS are distinct for each sub-system. A WS interface is used to access and retrieve information.

4.3. Information and Monitor Service

Information about the status of resources and RCMS components are sent to the Information and Monitor Service (IMS) as messages, which are reported to the operator or stored in a database.

The messages are catalogued according to their type, severity level and timestamp. The IMS collects the incoming information in a database and publishes it to subscribers. Subscribers can register for specific messages categorized by a number of selection criteria, such as timestamp, information source and severity level.

This service uses a publish/subscribe system to disseminate data to the subscribers. The IMS system can publish data via a Java Message Service (JMS), distribute data via log4j Appenders, and store data in a DBMS. Clients can publish messages to IMS using the logging message packages Log4C for C++ and Log4J for Java.

4.4. Graphical User Interface

The Controller GUI is based on JSP and Ajax technologies. It is composed of a generic framework, released together with the services, containing the basic building blocks to construct a GUI. These include the interface to the Resource Service for browsing configurations and the ability to command FM applications. To fit the specific requirements of each Sub-Detector, the GUI can by providing custom JSP pages. A JSP tag library provided with the framework allows the developer to use RCMS components on their own GUIs. The custom GUI code is packaged together with the FM code and has no dependency on the framework.

4.5. Job Control

The Job Control (JC) is a utility that allows the remote execution and supervision of any software process involved in data-taking. The JC runs as a daemon process on all nodes controlled by RCMS. It has the ability to detect crashed processes and can notify FMs which in turn can start recovery actions. The JC is implemented in C++ using the xdaq framework.

4.6. Access Control Manager

The Security Service provides facilities for user authentication and authorization, including data encryption if necessary. The Access Control Manager uses the Apache Tomcat Realm. It is a authentication service based on usernames, passwords and roles. One or more roles can be associated to an username. The role is determining the access to the RCMS web application. It uses a DBMS to store the data associated with user profiles and access privileges.

5. Deployment and Commissioning

During summer and autumn of 2006 a subset of the CMS detector including the solenoid magnet with a 4 T magnetic field were operated to detect cosmic muons. The operation was called “Magnet Test and Cosmic Challenge” (MTCC). The MTCC was a milestone of the CMS experiment as it completed the commissioning of the magnet system (coil and yoke) before its lowering into the cavern.

It was also a mile stone for the read-out and Run Control system as it demonstrated the working of the full chain. Several sub-detectors (Muon Detectors, Hadron and Electromagnetic Calorimeters, Tracker) and parts of the Trigger system were participating in the MTCC. During the autumn MTCC run, a data taking time of about 300h was achieved reading out about 130 million events. A plot of the number of events taken and the magnetic field in the solenoid magnet is shown in Figure 4.

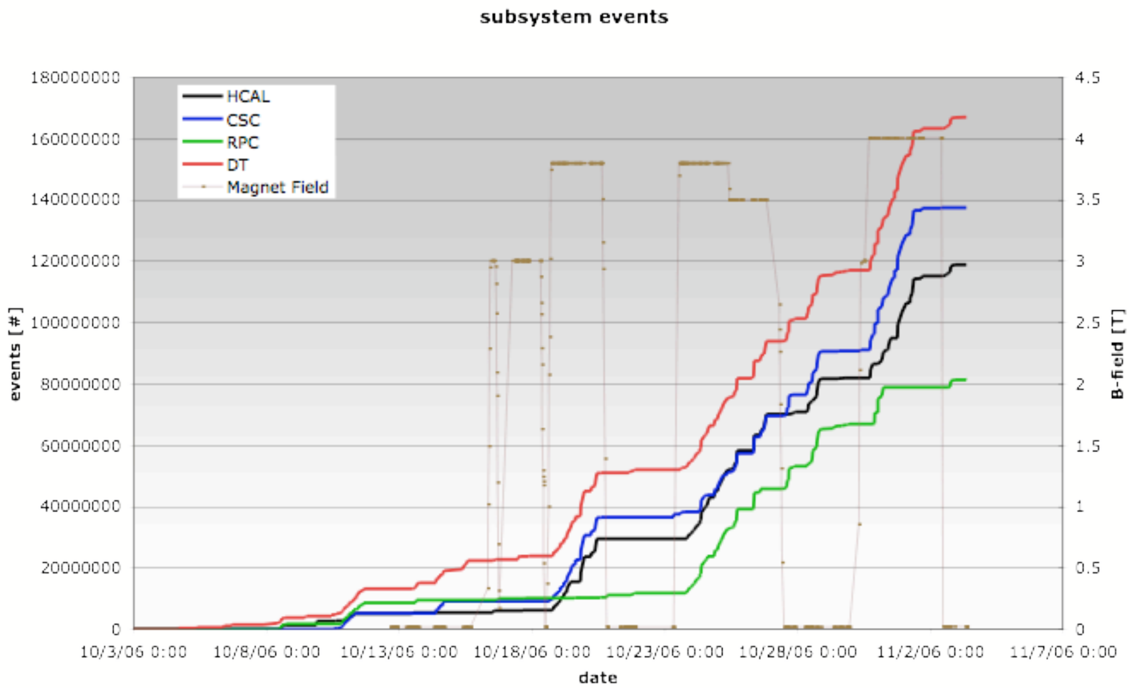

Figure 4 Number of events recorded by different sub-detectors and the B-field in the solenoid during MTCC in autumn 2006.

The MTCC used a provisional set-up of Run Control and the read-out systems. With the ongoing commissioning of CMS the final hardware for many components has arrived and is being installed. The PCs to run the final RCMS are installed and are being used in commissioning runs with a set of sub-detectors. Ten PCs running Scientific Linux are sufficient to control the experiment. One common database (Oracle 10g) is shared by all online processes and RCMS installations. The RCMS installations of each sub-detector use separate accounts for the Resource Service. Configuration management across sub-systems is achieved using global configuration keys.

6. Summary

The Run Control system has been discussed. The Run Control system is designed as a tree of finite state machines. Web technologies are used to implement the Run Control services. The hardware for Run Control has been installed and the system is used in commissioning runs of CMS.

With the successful running of a set of sub-detectors with Trigger and DAQ using RCMS an important milestone was achieved. The commissioning plan foresees regular “global runs” with a gradually increasing number of sub-systems and number of nodes participating. These “global runs” will provide valuable input to improve and refine the Run Control system of CMS to achieve the highest possible data taking efficiency.

Acknowledgements
This work was supported in part by the GRIDCC Project (EC Framework Programme 6 IST-511382), DOE and NSF (USA), KRF (Korea) and the Marie Curie Program.

References

[1] The CMS Collaboration 1994 *CMS Technical Proposal* CERN LHCC 94-38

[2] The CMS Collaboration 2002, CMS, The TriDAS Project, Technical Design Report, Volume 2: *Data Acquisition and High-Level Trigger*, CERN LHCC 2002-36

[3] J. Gutleber and L. Orsini 2002 *Software architecture for processing clusters based on i2o* (Cluster Computing 5(1)) pp 55-64