A SCALABLE AND HOMOGENEOUS WEB-BASED SOLUTION FOR PRESENTING CMS CONTROL SYSTEM DATA

L. Masetti, O. Chaze, J. A. Coarasa, C. Deldicque, M. Dobson, A. Dupont, D. Gigi, F. Glege, R. Gomez-Reino, C. Hartl, F. Meijers, E. Meschi, S. Morovic, C. Nunez-Barranco-Fernandez, L. Orsini, W. Ozga, A. Petrucci, G. Polese, A. Racz, H. Sakulin, C. Schwick, A. C. Spataru, C. Colin Wakefield, P. Zejd, CERN, Geneva, Switzerland
U. Behrens, DESY, Hamburg, Germany
R. K. Mommsen, V. O'Dell, FNAL, Chicago, Illinois, USA
O. Raginel, F. Stoeckl, G. Bauer, C. Paus, K. Sumorok, MIT, Cambridge, Massachusetts, USA
S. Erhan, UCLA, Los Angeles, California, USA
J. Branson, S. Cittolin, A. Holzner, M. Pieri, M. Sani, UCSD, San Diego, California, USA

Abstract

The Control System of the CMS experiment ensures the monitoring and safe operation of about 3M parameters. The high demand for access to online and historical Control System Data calls for a scalable solution combining multiple data sources. The advantage of a Web solution is that data can be accessed from everywhere with no application specific software. Moreover, a large pool of freely available components can be reused to achieve a user-friendly and effective data presentation. Access to the online information is provided with minimal impact on the running control system by using a common cache in order to be independent of the number of users. Historical data archived by the SCADA software is accessed via an Oracle Database. The web interfaces provide mostly a read-only access to data but some commands are also allowed. Moreover, developers and experts use web interfaces to deploy the control software and administer the SCADA projects in production. By using an enterprise portal, we profit from single sign-on and role-based access control. Portlets maintained by different developers are centrally integrated into dynamic pages, resulting in a consistent user experience.

ROLE AND SIZE OF THE CMS CONTROL SYSTEM

The Compact Muon Solenoid (CMS) detector is a general purpose detector operating at the Large Hadron Collider (LHC).

The Detector Control System (DCS) is the software that enables coherent and safe operation of the experiment. It displays the status of the detector at various levels of detail, allowing different interactions depending on user expertise [1].

The DCS provides a main interface that allows a single operator to monitor and control the entire detector. This interface summarizes effectively the status of the system, conveying the information from about three million parameters in an overall state. The DCS responds to high level commands and translates them into the proper sequence of low level commands submitted to the controlled hardware.

The DCS provides the supervision of the experiment 24 hours a day, allowing for the coherent combined operation of the different sub-systems. It works in synchronization with the LHC preparing automatically CMS for data taking whenever LHC is delivering stable beams and making sure that the experiment is in safe mode during non stable LHC conditions.

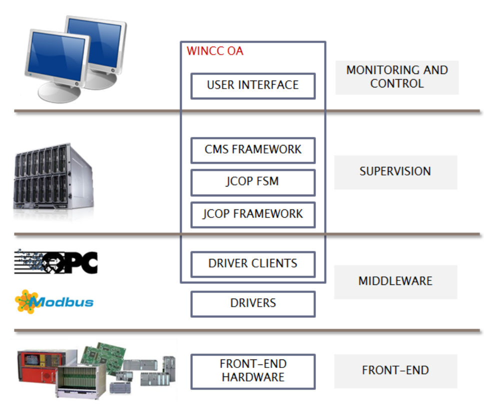

Figure 1: Schematic overview of the DCS architecture.

The DCS is implemented as a distributed system and is built on top of the WinCC OA commercial software (formerly called PVSS). The usage of a common CERN framework and the adoption of commonly applied CMS guidelines and libraries made possible the integration of all the sub-detector control systems and faciltated the software mantainance (see Fig. 1).

The experiment is modeled by a hierarchical Finite State Machine (FSM) tree where each node has a state and can accept commands. The leaves of the tree represent the hardware devices, while the internal nodes provide an increasing abstraction level, up to the root that models the overall control system status.

The DCS is in charge of archiving the changes over time in the system parameters. The values are not archived by sampling them at constant time intervals, but only when they change by more than a defined deadband.

ISBN 978-3-95450-139-7

USAGE OF CONTROL SYSTEM DATA

During the operation of CMS there is a high demand for remote access to control system data. Subdetector experts and CMS users need to quickly access both online and historical data. Access to online data allows remote monitoring of the current status of the detector or of any of its partitions. Historical data access is needed in order to perform incident analysis or to observe the trend over time of the archived parameters.

ADVANTAGES OF A WEB SOLUTION

The currently used SCADA software does not provide a remote access tool fullfilling all CMS requirements. In fact, remote WinCC OA user interfaces require installation of specialized software on the client side and may put a non negligible stress on the control software. This solution does not scale because every new client causes a significant overhead on the running system and risks to affect the operation of the experiment.

A Web-based solution to access the control system data offers several advantages. First, it does not require any specific software to be installed on the client side but only needs a web browser. Web interfaces can be accessed not only from PCs but also from mobile devices, such as tablets or smart phones. Dedicated presentations of the data can be used to target mobile devices. A well designed web application is scalable (see next section) and can serve a large number of users without significant impact on the SCADA system.

Moreover, the web thin client approach is very convenient because it minimizes the deployment effort when a new software version is released and centralizes access control with a positive impact on security.

The web solution also improves the reaction time of the experts given the ease to find a web browser to respond to a call and diagnose the problem.

The adoption of standard web technologies gives access to a large pool of freely available components that can be reused in the custom software for data visualization and processing (both on the client and on the server side).

STRATEGY FOR DATA ACCESS

Our solution does not provide an automatic translation of existing DCS user interfaces into web pages. Rather, a common web framework for accessing DCS data was developed and used in several applications (portlets) integrated in an enterprise portal environment.

Data displayed by web applications mainly comes from two types of sources: the condition database, where data is archived by the DCS, and online data.

Archived data is queried from the Oracle database and is used to generate plots and summary tables. The database schema has a separate table that holds the last archived value for each parameter. This table is significantly smaller than the table containing history data hence it allows for much faster queries. The last values are guaranteed to be equal to the current reading in the SCADA software ($\pm$ the archiving deadband) and can be used in the web display in place of the online values.

Regarding online data, we put special effort in ensuring that web access does not affect the performance of the control system. Hence no direct connection from the web server to WinCC OA core software is used, rather data is accessed using other lightweight access protocols.

FSM states of the hierarchical tree modeling the detector can be displayed on the web. The user can browse the hierarchy to reach the desired level of detail. The web interface uses both static and dynamic data. Static data (i.e. data that changes rarely, only in case of redesign of the FSM tree) includes node names, hierarchical relations between the nodes, node types and colors associated to each state. Dynamic data (that changes in response to the hardware state and to the operation of the experiment) includes the state of each node and the inclusion of each node in the command chain of the hierarchical tree.

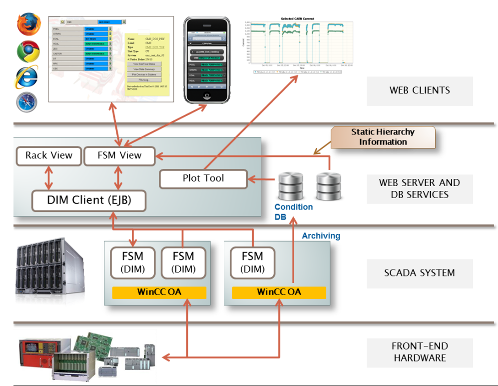

Figure 2: Data Access Paths to online and archived data.

Static data is dumped to a custom database and queried by the web application, while dynamic data is accessed via Distributed Information Management System (DIM) [2], a CERN-developed distributed protocol that is used natively by the FSM software to publish the state of each node (see Fig. 2).

DIM is based on the concept of service. A DIM server (for example an FSM process) publishes a number of services by registering them with a name server. Clients subscribe to services by searching a service name in the name server and then contacting the DIM server directly. After the subscription, the server notifies the client when the value changes or with a predefined rate.

The web server uses a single DIM client, implemented in an Enterprise JavaBeans (EJB) container. When the state of one node is requested for the first time, the client subscribes to the corresponding service. When the state is requested again, the value is returned from the local cache of the client running in the EJB. With this solution repetitive accesses are served by the DIM client local cache, independently of the number of users accessing the FSM states from the web, and the impact on the running online system

ISBN 978-3-95450-139-7

is minimal.

The same strategy is used to access other relevant information (other than FSM states) published explicitly to DIM by the DCS, for example specific data used to display the online version of the central DCS panel and PC monitoring information (such as memory used, running processes, etc.) that is published by a dedicated process running in each WinCC OA project.

For security and operational reasons, the access to DCS data from the web is mainly read-only. In fact, according to CMS operational rules, commands to the hardware can be given only by the shifters in the control room or by experts from the standard DCS interface. However there are some exceptions. For example, the VME crates used for data acquisition and controlled by the DCS can be switched on and off from the web interface. In this case an access control check is performed in order to guarantee that the command can be given only by users with proper privileges.

USER INTERFACES

The main web applications developed for CMS are:

- *FSM Monitoring*: interactive application to browse the entire FSM tree and display color-coded summary diagrams.
- *Plotting Tool*: provides access to archived data to produce predefined or customized plots.
- *Alarm Screen*: the online version of the DCS alarm screen providing access to present and historical alarms.
- *Rack and Crate Monitoring*: dedicated interface to access the environmental and electrical status of all experiment racks and VME crates.
- *Main Screen for DCS Operation*: summarized view of CMS and LHC status, including a log window showing the latest events and operator actions.
- *Automation*: interactive display of the automatic actions programmed in response to LHC state changes.
- *DIP Browser*: Interactive browser for data published via Data Interchange Protocol (DIP), a middleware lightweight infrastructure based on DIM.
- *Infrastructure Montitoring and Component Installation*: administrative web tools used by DCS developers to monitor and control the DCS infrastructure and deploy software in production (see next section).
- *System Overview*: powerful administrative tool, only available to central DCS experts, offering the same functionalities of the infrastructure monitoring but with possibility of sending mass commands to projects running on multiple servers.

The web server also hosts other applications not directly related to DCS data, such as a database-based *electronic logbook* used by shifters and experts and a *shiftlist tool* for scheduling all CMS shifts and send reminders to the involved people.

Web interfaces developed for CMS DCS can combine transparently data coming from various sources. This feature greatly increases the usability of the web tools. For example when displaying the FSM status of a device, the last values of the related archived parameters are also presented and it is possible to access directly the trend of each parameter over time. It is also possible to open summary plots for all the devices of a given type in a certain subtree (e.g. “measured currents of all the HV channels in the Tracker Outer Barrel”).

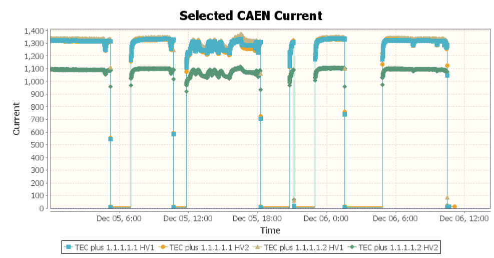

Figure 3: Example of plot of the currents of 4 HV channels produced by the web application. The current drops to 0 when the channels are switched off in response to the LHC state.

Existing visualization libraries are extensively used to achieve user-friendly and effective data presentation. The plotting tool used to display trends of the archived parameters uses the open source javascript library *jqplot* to create interactive plots on the client side and the java library *JFreeChart* to produce professional plots to be exported to pictures. An example of plot is shown in Fig. 3.

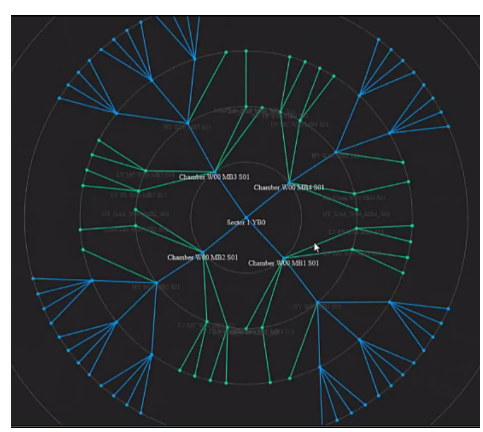

Figure 4: Example of radial graph showing color-coded states of the nodes in an FSM Tree. This kind of visualization is very effective to display the state of a large subtree.

The states of large FSM trees are effectively displayed using javascript visualization libraries such as *Javascript InfoVis Toolkit* and *sigmajs* (as shown in Fig. 4).

Asynchronous web applications are programmed using *Ajax* techniques mainly with the support of the *jquery* javascript framework.

ADMINISTRATIVE TOOLS FOR EXPERTS AND DEVELOPERS

Web interfaces are used in CMS not only to access DCS data but also for administrative purposes. CMS DCS used to be distributed over about 80 machines, reduced with the migration to the new redundant blade architecture [3] to 64 blade servers running 32 redundant projects. Direct access to the production servers is restricted to the members of the central DCS team, hence a web-based approach is used for deployment and monitoring.

To achieve a flexible architecture and simplify the maintenance, DCS developers release their code and functionalities in packages called components. Using an installation tool included in the CERN framework, the components can be targeted to specific WinCC OA projects (and if needed moved across different machines). The installation tool is able to install automatically a DCS project from scratch and to upgrade a single component when a new version is released. Additionally, CMS introduced the concept of hardware configurations related to a component that can also be moved across different projects. For example a general HV component can be installed in more than one project with different configurations containing the description of distinct hardware parts (e.g. one configuration for each of the 5 wheels of the muon DT detector).

The web interface to the installation tool allows developers to import the component code directly from the SVN repository and deploy a new version into the production projects. The hardware configurations associated to a component are also configurable from the web.

Production DCS projects can be monitored on the web and users with proper access rights can start and stop the WinCC OA processes. Moreover, the web interface displays the configuration (in terms of components and hardware configurations) and the connectivity within the distributed system of each project.

The information displayed in the administration interfaces comes mainly from the dedicated installation database, integrated with online PC monitoring information published via DIM and with direct queries to WinCC OA Process Monitoring tool (PMON).

ADVANTAGES OF USING AN ENTERPRISE PORTAL

All web applications are implemented in J2EE Java portlets and deployed to an enterprise portal. The portal provides single sign-on, role-based access control and integration of portlets into homogeneous pages. Portlets maintained by different developers can be deployed independently and combined to create rich dynamic pages with consistent look and feel and standard navigation features.

The access control is integrated with the campus credentials so that users can log in to the portal with the same username and password of their PC account.

Using role based access control, pages and functionalities can be easily restricted to a certain group of users. This is an important feature for all the operations that imply commands to the hardware or to the DCS infrastructure.

Users and groups defined in the portal are periodically synchronized with the SCADA system, so the tool for managing authorization in the portal is also used for managing privileges in the DCS software.

CONCLUSIONS

Web interfaces are used daily by hundreds of CMS users for multiple purposes, such as checking the current status of the experiment, produce plots of archived parameters, investigate present and past error conditions.

The number of users having direct access to DCS data has increased while the usage of the DCS interface in the terminal server has been limited to the few cases when an expert needs to take very specific actions from remote.

The web tools for administration are now an essential part of the deployment and monitor tasks and helped to achieve the flexible and maintainable architecture of the DCS.

The web interfaces have now become indispensable for CMS running.

REFERENCES

[1] G. Bauer, U. Behrens, M. Bowen, J. Branson, S. Bukowiec, S. Cittolin, J. A. Coarasa, C. Deldicque, M. Dobson, A. Dupont, S. Erhan, A. Flossdorf, D. Gigi, F. Glege, R. Gomez-Reino, C. Hartl, J. Hegeman, A. Holzner, Y. L. Hwong, L. Masetti, F. Meijers, E. Meschi, R. K Mommsen, V. O'Dell, L. Orsini, C. Paus, , A. Petrucci, M. Pieri, G. Polese, A. Racz, O. Raginel, H. Sakulin, M. Sani, C. Schwick, D. Shpakov, M. Simon, A. C. Spataru, K. Sumorok " Status of the CMS Detector Control System ," Journal of Physics: Conference Series Volume 396 Part 1 (2012).

[2] C. Gaspar, M. Dönszelmann and Ph. Charpentier, "DIM, a Portable, Light Weight Package for Information Publishing, Data Transfer and Inter-process Communication," International Conference on Computing in High Energy and Nuclear Physics (Padova, Italy, 1-11 February 2000)

[3] G. Bauer, U. Behrens, M. Bowen, J. Branson, S. Bukowiec, S. Cittolin, J. A. Coarasa, C. Deldicque, M. Dobson, A. Dupont, S. Erhan, A. Flossdorf, D. Gigi, F. Glege, R. Gomez-Reino, C. Hartl, J. Hegeman, A. Holzner, Y. L. Hwong, L. Masetti, F. Meijers, E. Meschi, R. K Mommsen, V. O'Dell, L. Orsini, C. Paus, , A. Petrucci, M. Pieri, G. Polese, A. Racz, O. Raginel, H. Sakulin, M. Sani, C. Schwick, D. Shpakov, M. Simon, A. C. Spataru, K. Sumorok "High availability through full redundancy of the CMS detector controls system," Journal of Physics: Conference Series Volume 396 Part 1 (2012).

ISBN 978-3-95450-139-7