DATABASE FOUNDATION FOR THE CONFIGURATION MANAGEMENT OF THE CERN ACCELERATOR CONTROLS SYSTEMS

Z. Zaharieva, M. Martin Marquez, M. Peryt, CERN, Geneva, Switzerland

Abstract

The Controls Configuration Database (CCDB) and its interfaces have been developed over the last 25 years in order to become nowadays the basis for the Configuration Management of the Controls System for all accelerators at CERN.

The CCDB contains data for all configuration items and their relationships, required for the correct functioning of the Controls System. The configuration items are quite heterogeneous, depicting different areas of the Controls System – ranging from 3000 Front-End Computers, 75 000 software devices allowing remote control of the accelerators, to valid states of the Accelerators Timing System.

The article will describe the different areas of the CCDB, their interdependencies and the challenges to establish the data model for such a diverse configuration management database, serving a multitude of clients.

The CCDB tracks the life of the configuration items by allowing their clear identification, triggering of change management processes as well as providing status accounting and audits. This necessitated the development and implementation of a combination of tailored processes and tools.

The Controls System is a data-driven one - the data stored in the CCDB is extracted and propagated to the controls hardware in order to configure it remotely. Therefore a special attention is placed on data security and data integrity as an incorrectly configured item can have a direct impact on the operation of the accelerators.

INTRODUCTION

The idea to use a central data storage to describe the components of the Controls System for CERN's Proton Sychrotron (PS) complex was suggested in 1980. Several implementations were made using a variety of database models such as an object oriented one. In 1986 an implementation using a relational database model and in particular using the Oracle® Relational Database Management System (RDBMS) was provided. This marked the birth of the Controls Configuration Database, which has been in service ever since [1]. Throughout its existence it has been constantly evolving, growing in size and in provided functionality in order to cover the increasing complexity of the accelerators Controls System at CERN.

THE NEED FOR CONFIGURATION MANAGEMENT

Configuration Management was established as an important element of Systems Engineering during the 1950s and 1960s. Realizing its importance not only for the hardware configuration of systems, but also for software configuration, was the reason to include it as a component of best practices, formalized by Information Technologies Infrastructure Library (ITIL®) [2].

The amount of technical data, necessary for the control of the CERN accelerator complex is enormous. A common description, in a centralized storage, of all objects needed for the Controls of the accelerators is an essential prerequisite for the correct and coherent functioning of the accelerators.

The CCDB and its interfaces provide configuration management functionalities such as the unique identification and configuration of items, complying with predefined criteria, as well as controlling the configuration changes and status accounting of the configured items. Those functionalities answer the need to establish and maintain the consistency of the Controls System and its components.

SCOPE

The CCDB is the heart of the CERN Accelerators Controls System. Nowadays it covers the need for the configuration of components of the Controls System itself, e.g. the Controls Middleware (CMW), as well as accelerator components as seen by the Controls System, e.g. power converters, for all accelerators: the Large Hadron Collider (LHC), the Super Proton Synchrotron (SPS) Complex, the Proton Synchrotron (PS) Complex, and the CLIC Test Facility (CTF3). The configurations of some components used for the Control Systems for the Technical Infrastructure services at CERN are done in the CCDB too.

Providing the configuration capabilities for such a diverse number of systems and components and representing them in a unified model is quite a challenge. Currently the relational database model of the CCDB comprises of 914 tables, storing over 10GB of current reference data and 50GB of historical versioned data.

CONFIGURATION MANAGEMENT FUNCTIONALITIES, PROVIDED BY THE CCDB AND ASSOCIATED INTERFACES

Configuration of all Components of the Controls System

The data in the CCDB represents components and their properties as seen by the Controls System. It contains data for the complete Controls System topology with some of the main users of the CCDB services being the CMW, the Role Based Access (RBAC), the Accelerator Timing System, the Common Console Manager, etc. (see Fig.1).

Copyright © 2011 by the respective authors — cc Creative Commons Attribution 3.0 (CC BY 3.0)

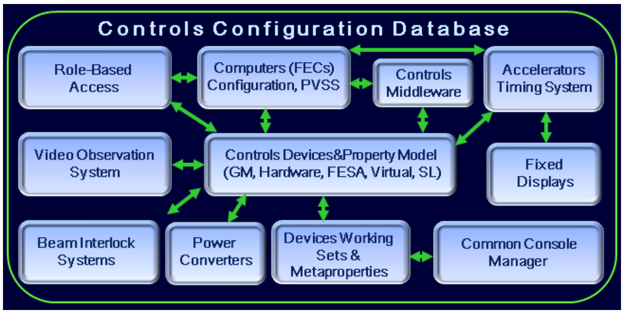

Figure 1: Overview of the different areas of the CCDB.

The Controls devices are the software representations of the accelerator components, controlled by the Controls System, e.g. power converters, radio frequency (RF), beam instrumentation (BI) components, etc. The CCDB currently supports 5 device-property models – General Module (GM), Front-End Software Architecture (FESA), SPS&LEP (SL), Hardware and Virtual. There are more than 1,400 device classes, more than 79,000 devices and more than 2,000,000 properties declared in the CCDB.

The GM and SL legacy frameworks are in the process of being phased-out; they were developed respectively for the PS and SPS complexes.

The FESA framework was introduced 8 years ago. The original data management solution was to store directly XML configuration files in the CCDB, produced by the FESA development tools. This solution presented several shortcomings mainly with the integration of the FESA-related data to the rest of the configuration data in the CCDB. Another problem was tracking of devices and properties changes, contained within the XML files. A major effort to restructure the data management part of the new FESA framework version (3.0) was initiated in 2009 [3]. The main challenge was to implement a structured handling of XML files, which should be used only for data exchange between the FESA end-user tools and the CCDB. The XML files are no longer stored in the CCDB - their contents are extracted on the fly (using a PL/SQL API) and the FESA configuration data is stored into the relational tables. This solution has allowed a complete integration of FESA data into the CCDB relational model and facilitates the handling of common data management tasks.

The Hardware devices framework was developed in 2007 and originally dedicated to the Power Converter Controls [4] devices and PVSS devices for the Machine Protection of LHC (Quench Protection System, Warm Interlock System, Power Interlock System, etc.). Nowadays the scope of this framework has been broadened to a wide variety of physical equipment.

The Virtual devices framework was the last Controls devices framework to be introduced in the CCDB (only 1 year ago) and represents abstract elements that need to be controlled, e.g. signals for the Software Interlock System.

The challenge before the CCDB was to implement as much as possible a unified database model, which suites those 5 diverse Controls devices frameworks [5]. It is the database that provides a layer of abstraction and provides the device-property data in one and the same way to the multiple db clients (e.g. CMW, RBAC).

The CCDB also provides data for the hardware and software configuration (e.g. start-up sequences) of all Front-End computers used by the Controls System. There are more than 3,200 computers declared in the CCDB. Special functionality is available to configure the drivers for the modules, used in the Front-End computers.

The Accelerators Timing System configuration is another important area. At the beginning of 2011 a major renovation of the Timing System was undertaken by the timing experts and this had an impact on the CCDB. The database model was completely reworked in order to introduce high-level timing events and to allow full configuration of low-level timing events. The new model ensures the completeness of the configuration of the Timing System. It allows better integration of the timing data with the rest of the configuration items in CCDB.

It is important to have a complete description of the components, needed for the accelerators Control, stored in a single location such as the CCDB. This provides the possibility to validate configurations against business rules and to establish a coherent picture of all components and their relationships (dependencies between the components).

Extraction of Configurations – Data-Driven Controls System

The CCDB is not merely a storage repository for the configuration of the different components, related to the Controls of the accelerator. The CCDB actively serves to provide real-time configuration data, which is extracted from the database whenever a component needs to be configured. In order to provide the on-line data for the configuration items and their relationships, there are a number of APIs and scripts which have been developed.

Configuration Change Management

For the data-driven Controls System the data in the CCDB describes the Controls components and their properties. Those components need to be upgraded regularly, e.g. Front-End Computers, Device Classes, etc, therefore there is a need for regular data modifications and maintenance in the CCDB.

Each configuration item in CCDB is uniquely identified, which allows the database to trace the history of all operations (data modifications) performed on that item. The users are provided with reports on the current configuration and its evolution over time.

A solution to review potential configuration changes and accept or reject them is under development. The change acceptance process is mainly guided by the status of a given component (operational or not) and whether a change is considered to be backward compatible [6].

On-line Feedback of Deployed Configurations

A recently implemented new feature of the Controls Configuration is the availability of status accounting of the different configuration items. The status data comes in

Copyright © 2011 by the respective authors — cc Creative Commons Attribution 3.0 (CC BY 3.0)

the form of a feedback from the configured components identifying the currently loaded configuration.

The first configuration items to start sending feedback data are the drivers for the Front-End computer modules and some components from the CMW configurations. This feature of the Controls Configuration Management is extremely important as it solves the years-old problems of discrepancies between the loaded Front-End configurations and the current running configurations, resulting in an unexpected behaviour after a reboot.

It is previewed that this functionality will be extended to cover other areas of the Controls Configuration in the near future, e.g. the software devices configurations deployed on the Front-Ends, etc.

OVERVIEW OF THE CONTROLS CONFIGURATION ENVIRONMENT

Database Complexity

The Controls Configuration database is quite a complex one as it models the Controls System into a relational database. The main challenge for the database model is to maintain data consistency and to enforce the predefined business rules. This is achieved by numerous constraints as well as thousands of lines of PL/SQL code in database triggers or packages. Table 1 provides the exact figures showing the complexity of the CCDB data model.

Table 1: CCDB Statistics

Tables	914
Constraints	2,388
Lines PL/SQL code	42,100
Volume	60GB

Database High Availability

The Controls Configuration services need to be continuously available 24 hours a day, 365 days a year to assure the ability to configure components at any moment in order to operate the various accelerators at CERN as well as to allow configuration changes to be introduced during scheduled accelerator “technical stops”. This has been achieved through the use of Oracle cluster technology (RAC) which guarantees not only the hardware but also database software redundancy. This solution ensures no down time even during routine software patching of the database.

Configuration Data Responsibility

There is a diverse user community for the Controls Configuration services, which makes the job of providing tailored user applications challenging. The responsibility to maintain the correctness of the configuration data lies with the relevant equipment experts (BI, RF, etc.), as well as controls experts and accelerator operators.

Data Editing Interfaces

The CCDB Data Editing Interfaces are a suite of web-deployed applications used to modify the stored configuration data. There are currently 12 Configuration Editors, which implement the business logic and processes in the various areas of the CCDB. These applications are based on Oracle’s J2EE ADF technology and are used on a daily basis by more than 250 users.

Special attention must be paid to the access rights given to the users of these applications, which need to be tightly controlled due to the sensitive nature of the data and the associated risk of mis-configuration of accelerator components. Strict access control rules are therefore implemented via a custom authorization module. Through the use of the virtual private database feature of Oracle, additional fine grained access control mechanisms are provided within each application.

Data Browsing Interfaces

The CCDB Data Browsing Interfaces comprise of read-only applications (reporting tools), for situations where the configuration data only needs to be consulted. Some 160 reports covering all areas of the CCDB are provided to the user community of roughly 300 people and are built using Oracle APEX technology [7].

APIs and Scripts

Various APIs, written in several languages such as Java (e.g. Java Directory Service, Beam Interlock Systems API), PL/SQL (e.g. Front-Ends Drivers Generation APIs, FESA data extraction API) and some legacy Pro*C scripts (e.g. Front-End configurations), are implemented to extract the configuration data or to generate files to be used by the different components of the Controls System.

Diverse output formats of the components configurations are produced from the CCDB, ranging from text files for drivers generation and hardware and software configuration of computers to XML files and binaries for Controls devices configuration.

DATA SECURITY

Data security is paramount in the CCDB. The CCDB provides specially developed functionality to audit every session opened in the CCDB, allowing monitoring of who changed what data and when.

In order to trace data changes, a custom history framework was developed. All data modifications are recorded and kept on-line since 2005. There is a special web-deployed application – the CCDB History Browser, which gives access to the history logs. This application is used heavily by the Controls Exploitation team.

The history log also serves as a basis for versioning of the configuration data for each configuration item.

QUALITY ASSURANCE – TESTING AND DEVELOPMENT ENVIRONMENTS

The Control of CERN’s accelerators is a very dynamic environment with new and changing user requirements coming regularly, directly impacting the database model as well as the different applications and APIs. The best software development style to suit such an environment is

Copyright © 2011 by the respective authors — cc Creative Commons Attribution 3.0 (CC BY 3.0)

agile programming, facilitating fast prototyping and short time to production to achieve a feature complete solution.

Four environments have been provided for the CCDB, its interfaces and APIs – *Development*, *Test* – used for unit and functional testing, *Next* – used for integration testing and *Production* environments.

The Next environment of the Controls Configuration is part of the Controls TestBed since 2010 [8]. Some changes in the Controls components are so fundamental, e.g. the renovation of the accelerators Timing System in 2011, that it could take more than 1 year before deploying the new functionality to production. The TestBed provides unique opportunities to perform integration and system testing of the new functionalities with all major components of the Controls System being available.

DATA PROPAGATION

The data management for the Control and Operation of CERN's accelerators is implemented as a distributed database environment [9]. Part of the data stored in the CCDB is propagated to other database systems for the needs of the Operation of the accelerators (see Fig.2).

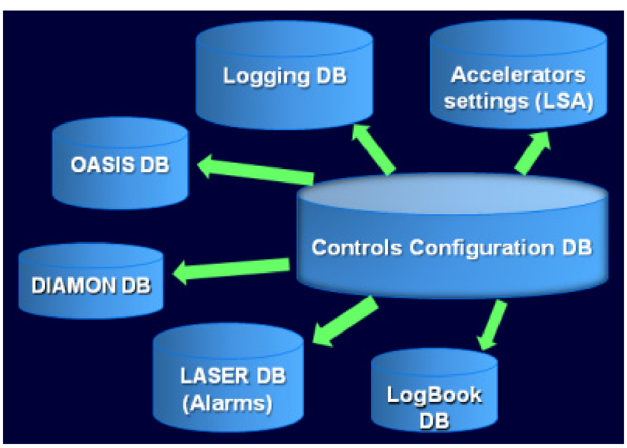

Figure 2: Propagation of data from the CCDB to other databases, used for the Operation of the accelerators.

The alarm definitions (>120,000) for all accelerators devices, are generated and propagated to the Alarms Database [10]. Part of the devices' data is propagated towards the Accelerators Settings (LSA) database, where it is used as a basis for the Operation of the accelerators. The CCDB provides the data for the computers (Front-Ends, consoles, servers, etc.), which need to be monitored by the Diagnostics and Monitoring System (DIAMON).

Safe Propagation of Data Changes

Changes to configuration data in CCDB could have an impact on the related databases. A strategy for smooth data upgrades of the data-driven Controls System has been established with the objective to ensure a coherent set of data throughout all distributed databases. Database procedures are developed in the CCDB to safely propagate data changes, based on knowing the data dependencies between the different systems. The impact of the configuration data changes is analyzed and appropriate actions taken – this process is part of the configuration Change Management functionality provided by the CCDB.

CONCLUSION

The Configuration Management has proven to be an indispensable part of ensuring the correct functioning of any large system.

The Controls Configuration DB, its interfaces and the specific processes implemented around those, are providing the basis for the Configuration Management of the Controls System for all accelerators at CERN. The CCDB ensures conceptual unification and centralization of the diverse configurations of the different items and their relations, thus describing the different components of the Controls System and their dependencies.

Continuous effort is being put into rationalizing, improving, federating and developing new functionality in the existing database and its interfaces.

REFERENCES

[1] J. Cuperus, R. Billen and M. Lelaizant, "The Configuration Database for the CERN Accelerator Control System", ICALEPCS'03, Gyeongju, Korea, Oct-2003, WE114.

[2] ITIL®: www.itil-officialsite.com

[3] M. Arruat *et al.*, "FESA 3.0", ICALEPCS'11, Grenoble, France, Oct-2011, WEPMN002.

[4] Q. King, S.T. Page and Z. Zaharieva, "Automatic Inventory and Configuration Management Tools for the LHC Power Converter Controls", ICALEPCS'09, Kobe, Japan, Oct-2009, TUA004.

[5] M.Peryt *et al.*, "Database and interface modifications: change management without affecting the clients", ICALEPCS'11, Grenoble, France, Oct-2011, MOPKN010.

[6] V.Baggiolini *et al.*, "Backward Compatibility as a Key Measure for Smooth Upgrades to the LHC control system", ICALEPCS'11, Grenoble, France, Oct-2011, WEPMS007.

[7] Z. Zaharieva, R. Billen, "Rapid Development of Database Interfaces with Oracle APEX, used for the Controls Systems at CERN", ICALEPCS'09, Kobe, Japan, Oct-2009, THP108.

[8] J.Nguyen Xuan, V.Baggiolini, "Testbed for validating the LHC controls system core before deployment", ICALEPCS'11, Grenoble, France, Oct-2011, WEPMS003

[9] R. Billen *et al.*, "Accelerator Data Foundation: How It All Fits Together", ICALEPCS'09, Kobe, Japan, Oct-2009, TUB001.

[10] Z. Zaharieva, M. Buttner, "CERN Alarms data management: state & improvements", ICALEPCS'11, Grenoble, France, Oct-2011, MOPKN011

Copyright © 2011 by the respective authors — cc Creative Commons Attribution 3.0 (CC BY 3.0)