CHALLENGES TO PROVIDING A SUCCESSFUL CENTRAL CONFIGURATION SERVICE TO SUPPORT CERN'S NEW CONTROLS DIAGNOSTICS AND MONITORING SYSTEM

Z. Makonnen, M. Buttner, Z. Zaharieva, CERN, Geneva, Switzerland

Abstract

The Controls Diagnostic and Monitoring service (DIAMON) provides monitoring and diagnostics tools to the operators in the CERN Control Centre. A recent reengineering presented the opportunity to restructure its data management and to integrate it with the central Controls Configuration Service (CCS). The CCS provides the Configuration Management for the Controls System for all accelerators at CERN. The new facility had to cater for the configuration management of all agents monitored by DIAMON, (>3000 computers of different types), provide deployment information, relations between metrics, and historical information. In addition, it had to be integrated into the operational CCS, while ensuring stability and data coherency. An important design decision was to largely reuse the existing infrastructure in the CCS and adapt the DIAMON data management to it e.g. by using the device/property model through a Virtual Devices framework to model the DIAMON agents. This article will show how these challenging requirements were successfully met, the problems encountered and their resolution. The new service architecture will be presented: database model, new and tailored processes and tools.

INTRODUCTION

The Diagnostic and Monitoring system (DIAMON) provides the means to monitor the Controls infrastructure (e.g. Front-End computers, servers, etc.) as well as presents tools to the operators in the accelerators Controls Centre to diagnose issues. This task is achieved through the use of agents, deployed on the different Controls equipment. There are several agents depending on the type of equipment and processes or parameters to be monitored, e.g. PING, CLIC, JMX, etc. [1]. In order to know which type of agents to deploy on each component as well as for each agent to know what parameters to monitor requires a set of configuration data to be made available. The configuration data also provides the rules necessary to interpret the data sent by agents to the DIAMON Acquisition Layer and Server.

In order to provide the configuration data for a large Diagnostics and Monitoring System such as DIAMON, used for all accelerators at CERN, a central configuration management service is a necessity. The Controls Configuration Service (CCS) provides the aforementioned functionalities to the DIAMON system.

The Controls Configuration Service (CCS) provides a centralized storage location for all configuration data (a database) together with a suite of tools and configuration processes in order to present a real time representation of the state of the configuration items as well as instance in time views of configuration state (baselines) and systematically tracks changes to all registered components as well as maintaining the consistency of the components with respect to each other to ensure coherent functioning of the Controls System [2].

The CCS provides configuration management facilities such as the unique identification of the configuration items and presents their authorized configurations, complying with predefined criteria, in addition to controlling configuration changes and status accounting of the items.

The Controls Configuration Database (CCDB) is an essential part of the CCS and the heart of the CERN Accelerators Controls System. The CCDB maintains data for all configuration items and their relationships which is required for the correct functioning of the Controls System. The configuration items are heterogeneous in nature, and represent the many different areas of the Controls System – such as 4500 Computers, 80000 software devices allowing remote control of the accelerators, and valid states of the Accelerators Timing System.

Providing the configuration capabilities for such a diverse number of sub-systems and components and representing them in a unified model is an enormous challenge. Currently the relational database model of the CCDB comprises of 960 tables, storing over 15GB of current reference data and 85GB of historical versioned data.

MOTIVATION FOR PROVIDING A CENTRAL CONFIGURATION SERVICE

A decision to upgrade the DIAMON system was taken two years ago, aiming to improve scalability, simplify the maintenance of the system with the introduction of dynamic re-configuration at run-time, and develop additional functionalities for improving the user interfaces [3].

This was the perfect moment to reengineer the Configuration Management of DIAMON and to fully integrate it with the central Controls Configuration Service due to numerous reasons - the main ones are listed below:

- ✓ Enhance the previously used configuration data model for DIAMON
- ✓ Ease the integration of the configuration data for the DIAMON service with the configuration data from others systems which use the CCS, e.g. Front-End computers configurations, Controls Devices configurations, etc.
- ✓ Eliminate the redundancy and other problems related to synchronization of data between the previous DIAMON configuration repository and

ISBN 978-3-95450-139-7

the CCDB.

- ✓ Reduce the time between a user modifying a configuration item and the change being propagated to the DIAMON server and becoming visible through the different user interfaces (DIAMON viewer and DIAMON console).
- ✓ Reuse of the existing CCS infrastructure as much as possible.

Taking the decision to use a central configuration management service meant not only improving the quality of the configuration data as well as the configuration management processes, but adhering to best practices for Configuration Management formalized by different standards such as ITIL (Information Technologies Infrastructure Library) [4], IEEE standards [5], Control Objectives for Information and Related Technology (COBIT) [6], etc.

OVERVIEW OF THE ARCHITECTURE

The architecture and the data flow of the configuration data for DIAMON is depicted below in Figure 1.

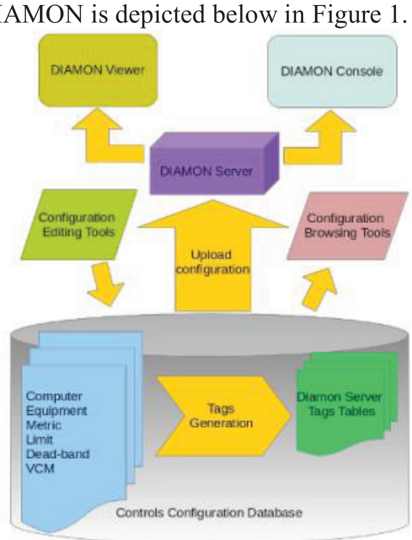

Figure 1: Configuration architecture and data flow.

The configuration items and relations for the DIAMON system are stored in relational tables in the CCDB. A dedicated process transforms the data to a standard format (configuration tag) and stores the tags into 3 dedicated interface tables. The configuration tags are then uploaded to the DIAMON server, the deployment operation is not undertaken until it has first been authorized by an expert. Once uploaded, the configurations are immediately visible through the DIAMON Viewer and Console. The process to generate the configuration tags is implemented in PL/SQL and is triggered automatically every time there is a change in the attributes of any of the configuration items.

From a configuration point of view, the main configuration items to be modelled for the use of the DIAMON system are:

- ✓ Equipment: represents software or hardware elements used for the accelerators Controls, which need to be monitored. Based on their type different agents and data acquisition processes for monitoring are used, e.g. PING, JMX, etc.
- ✓ Metric: represents the parameters, which are monitored by the agents, e.g. CPU load, memory usage, disk space usage, etc.
- ✓ Rule: represents how to decode the value of the metric in order to know if it should be treated as an error, warning or normal state.
- ✓ Command: represents the command to be executed on the particular equipment.

FUNCTIONALITIES

Automatic Generation of Configuration Tags

The generation of configuration tags, based on the type of the configuration item and its state, is a key feature of the new configuration mechanisms as it simplifies the interface between the CCDB and the DIAMON server. The configuration tags have predefined types, e.g. Create Computer, Remove Equipment, Load Limit, Update Equipment Rule, etc. The tags are queued in the Interface (Tags) tables to be applied individually to the server, as the configuration changes happen, and there is no need any more to regenerate the complete set of configuration data for the thousands of monitored computers (a process that was run semi-automatically in the past, once per week and was taking hours to fulfil), which saves time and computing resources.

Inheritance of the Configuration Items

In order to ensure that all configuration items are consistently monitored, an inheritance mechanism based on the hierarchy depicted in Figure 2 is implemented. This functionality is used not only during the generation of the configuration tags, but at run time as well, for example, if a computer ceases to be monitored, then the equipment that belongs to it is disabled automatically, which in cascade will disable the metrics, comments, etc.

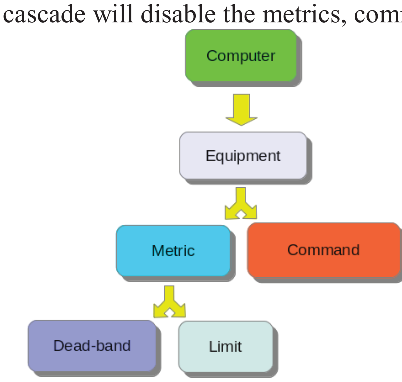

Figure 2: Hierarchy of configuration items.

Configuration Editors

Three editors, developed with Oracle APEX technology [7], have been provided to the DIAMON users in order to manipulate the configuration data.

The ‘Controls Devices Editor’ is a generic tool that is used for data manipulations related to configuration items used by the different Device-Property frameworks (e.g.

ISBN 978-3-95450-139-7

FESA, Hardware, etc.). This editor is widely used by equipment and controls experts and operators. The editor was extended to provide additional features for the Virtual devices framework, which is used for the modelling of the DIAMON equipment.

The 'Controls Configuration DIAMON Editor' is dedicated to editing the configurations for the metrics' limits, dead-bands, etc. It also provides tailored views on the Virtual devices framework, customized for the DIAMON users (Figure 3).

The 'DIAMON TagId Configuration Editor' has been designed for the upload of the configuration data to the DIAMON server.

Special attention is 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 misconfiguration 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.

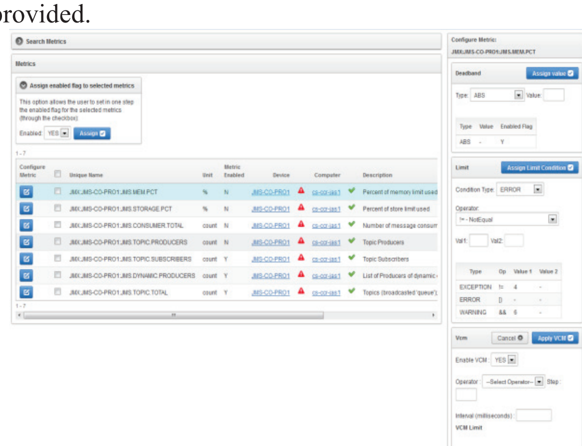

Figure 3: Controls Configuration DIAMON Editor.

Configuration Browsing Tools

Reporting facilities, based on Oracle APEX, have been provided alongside the data editing tools.

The reports are integrated into the Controls Configuration Data Browser, which gives the possibility to browse through all of the configuration data passing seamlessly through different CCS logical domains such as DIAMON, FECs, CMW, Controls Devices, etc. One of the most critical reporting functionalities for the DIAMON experts is the report which provides an audit of the discrepancies between the state of the configuration items in the CCDB and the data loaded into the DIAMON server.

Historical changes to the configuration data over time are also available through the 'History Log Browser' as part of the standard CCS functionalities.

CHALLENGES

Data Model Complexity

The modelling of the DIAMON data into the CCDB was a real challenge, due to the wide variety of equipment that the system supports, which translates into a proliferation of configuration items and their relations with a diverse set of attributes and a wide range of configuration processes. Several iterations of the database model were implemented in close collaboration with the DIAMON team in order to best understand the needs of the system and translate those into database requirements.

Relational tables, views, configuration processes PL/SQL triggers and packages for the configuration processes have been reviewed in the process.

Complete Integration with Configuration Models

Taking into account the fact that most of the systems with which DIAMON needs to interact, e.g. the Controls Middleware (CMW), Role-based Access (RBAC), etc. provide functionalities based on the Device-Property model, an important design decision was to present part of the DIAMON-related configuration data in such a way as to fit that model too. This was done mainly through the use of the Virtual Devices framework and some database model extensions.

Integration was done while respecting an important constraint - to guaranty no impact on other clients using the CCS, i.e. not to perturb the configurations of the other Controls Device frameworks. In order to achieve this, special isolation layers of views and PL/SQL packages were provided between the physical database tables and the database objects seen by the different clients.

Implementing Coherent Data Flows and Configuration Management Processes

The DIAMON configuration data covers many components, some of which are not under the responsibility of the DIAMON team at all. Therefore establishing the configuration management processes for all configuration items and clients of the CCS was a very important point, which required knowledge of all affected systems.

Some of the implemented processes are related to the timing and the exact sequence of the generation of the tag configurations. The complete data flow was established from the moment a decision to monitor a given item is taken to the moment when the item is seen on the DIAMON console.

There are strict naming rules, which are established for all configuration items. In order to keep all of the items in a consistent state, processes have been put in place, which are automatically triggered when a rename of a DIAMON monitored computer is performed. All of the actions in the renaming mechanism are implemented inside the CCDB and they are completely transparent to the users.

ISBN 978-3-95450-139-7

Integration between the CCDB and the DIAMON System

The DIAMON server can be dynamically reconfigured at run-time by sending to it a new configuration tag. The main objectives for the configuration data propagation to the Server were to ensure the correct order of the generation of the tags, this ordering must also be maintained when applied to the server and that the propagation is done within the time required by DIAMON. A process for expert approval to load the generated tags is put in place, which includes a notification system to alert if new tags are available as well as reporting capabilities.

Scalability

Scalability and time performance were major issues when designing and implementing the configuration management processes due to the high number of configuration items. At the moment there are around 3000 components that are monitored by DIAMON, with more than 6000 equipment, 96000 metrics, 24000 commands and 54000 rules and this number will continue growing. Major effort has been put into the design focusing on the normalization, creation of indexes and other technique to optimize as much as possible the performances.

Configuration Data for the New DIAMON System

In order to put in place the new configuration model for DIAMON and to feed it with data, it was necessary to migrate the old DIAMON configuration data. This was done in two phases. The first phase comprised of retrieving and reorganizing into only relational tables the old data which was scattered between tables and XML files. During the second phase the main objectives were to optimize, enhance and develop new structures following the requirements of the new DIAMON system.

Parallel Operation of the Old and New DIAMON Systems

For more than six months the old and the new DIAMON systems were running in parallel during a switch-over phase in order to verify the new system in production environment. Managing to keep the configuration data synchronized and to avoid conflict or missing data was one of the main concerns during this period. In order to achieve this, custom processes were implemented in CCDB. Those processes were removed, once the old DIAMON system was switched off.

QUALITY ASSURANCE AND TESTING

Following the QA procedures set-up in the CCS, design reviews and code reviews are undertaken as part of the development process, in order to ensure the high quality of each new functionality release.

Testing is another important aspect of the development process and dedicated environments are used by the CCS for each phase: Development, Test to run unit and functional test, Next to run continuous integration tests [8] and Production. The DIAMON Test server is set-up against the CCS Test environment, which provides the possibility to thoroughly test all new functionalities before they are deployed to production environment.

CONCLUSION

The Controls Configuration Service provides the complete set of configuration management functionalities needed for the new Diagnostics and Monitoring System despite the important constraints that needed to be respected while transitioning from the old to the new DIAMON System.

The main goals, to centralize the DIAMON configurations, to improve the scalability and the robustness of the configuration processes as well as the responsiveness to changes as configuration data can be loaded into the DIAMON server in real-time have been achieved. Other important advantages are the possibility to integrate the DIAMON data with the configuration data for other systems (e.g. FECs configurations, CMW, RBAC, etc.) in order to obtain a global and consistent configuration across the entire Controls System, to eliminate the redundancy of the data between the old DIAMON DB schema and the CCDB and thus improve the quality of the data thereby minimizing the errors and down time due to erroneous configurations.

From the users perspective new and improved tools have been made available to provide the configuration data, a notification system has been put in place, processes, auditing and reporting capabilities have been set up in order to ensure a coherent data flow from the CCS to the DIAMON system and to check the quality of the data and its consistency.

REFERENCES

[1] W. Buczak *et al.,* “DIAMON2-Impoved monitoring of CERN’s Accelerator control infrastructure”, ICALEPCS’13, San Francisco, USA, Oct-2013.

[2] Z. Zaharieva *et al.,* “Database Foundation for the configuration management of the CERN Accelerator Control System”, ICALEPCS’11, Grenoble, France, Oct-2011.

[3] M. Buttner *et al.,* “Diagnostic and Monitoring CERN Accelerator infrastructure”, ICALEPCS’09, Kobe, Japan, Oct- 2009.

[4] ITIL: www.itil-officialsite.com

[5] IEEE Std. 828-2012: standards.ieee.org

[6] COBIT: www.isaca.org/COBIT

[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.

[8] J. Nguyen Xuan, V. Baggiolini, “A Testbed for validating the LHC controls system core before deployment”, ICALEPCS’11, Grenoble, France, Oct-2011.

ISBN 978-3-95450-139-7