ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA044

INTEGRATION OF THE VACUUM SCADA WITH CERN'S ENTERPRISE ASSET MANAGEMENT SYSTEM

A. Rocha, S. Blanchard, J. Fraga, G. Gkioka, P. Gomes, L. Gonzalez, G. Riddone, T. Krastev, D. Widegren, CERN, Geneva, Switzerland

Abstract

With over 128Km of vacuum chambers, reaching pressures as low as in interstellar space, CERN is home to the largest vacuum system in the world. Its underlying architecture comprises approximately 15 000 pieces of control equipment, supervised and controlled by 7 Supervisory Control And Data Acquisition (SCADA) servers, and over 300 Programmable Logic Controllers (PLCs). Their configuration files are automatically generated from a set of ORACLE databases (vacDB) using a Java application (vacDB-Editor).

The maintenance management of such an amount of equipment requires the usage of an Enterprise Asset Management system (EAM), where the life cycle of every equipment is tracked from reception through decommissioning.

The equipment displayed in the vacuum SCADA is automatically integrated in its user interfaces (UIs) based on data available on vacDB. On the other hand, the equipment available in Infor-EAM for maintenance management activities (creation of work-orders, stock management, location tracking) resides in its own database. This leaves room for mismatches between what users see on the SCADA and in Infor-EAM. Although manual imports of equipment lists from vacDB to Infor-EAM are possible, the process is time consuming, error prone, and only guarantees the correctness of data while no equipment is added, deleted or modified in vacDB. Aiming to solve this issue, a web-based application called vacDM was developed to ensure continuous consistency between vacDB, Infor-EAM and CERN's dictionary database for equipment descriptions, the naming-DB.

Following the implementation of vacDM, the vacuum SCADA was updated to allow the generation of Infor-EAM work orders.

INTRODUCTION

Vacuum at CERN

Since the foundation of CERN in 1954 that its accelerator complex has been in constant growth. As new accelerators are designed and built to achieve higher energy beams, the requirements on the vacuum levels in the accelerator chambers become stricter. In order to achieve the specified beam lifetimes and also to reduce thermal conduction on cryogenic systems, the accelerator chambers at CERN operate at pressures between 10^{-6} and 10^{-12}mBar, achieved by a multitude of pumps, valves and gauges.

In order to meet the operational requirements, the vacuum control system has evolved over the years to what is now a multi-tier architecture as illustrated in Figure 1.

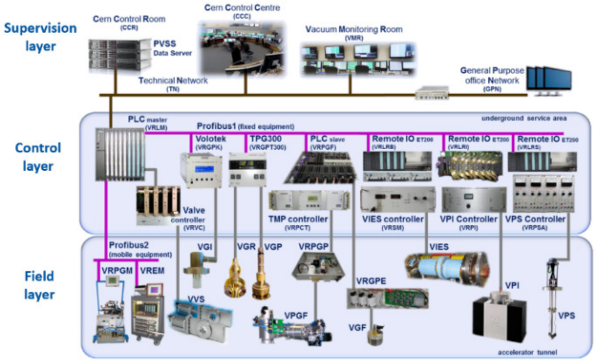


Figure 1: Vacuum controls architecture at CERN [1].

Data from field devices is acquired by the PLCs that are interconnected in CERN's technical network. The PLCs communicate with the SCADA server on the supervision layer, where the incoming data is archived, processed and displayed to the end user via dedicated consoles.

Data Engineering

The configuration of the SCADA and PLCs is performed through the vacDB-Editor, where equipment can be added, modified and deleted (Figure 2). The vacDB-editor stores persistent data in vacDB, a set of ORACLE databases that contains all of the required configuration parameters for every device. From the data in vacDB the export module of vacDB-Editor generates automatically the configuration for the SCADA and PLCs, mapping SCADA datapoints to memory locations in the PLC.

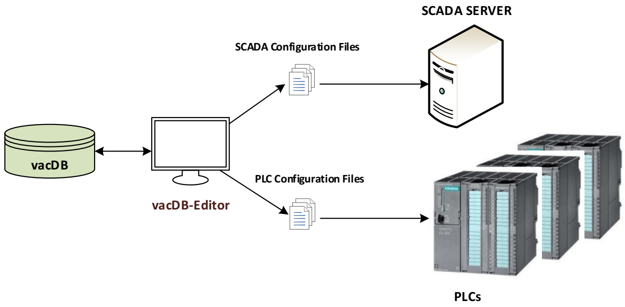


Figure 2: Control System Configuration Workflow.

The Need for an EAM System

With the increase over the last years on the number of equipment under the responsibility of the vacuum group, the task of tracking the location, stock levels and relationships between equipment became almost impossible using traditional methods such as excel sheets. Several commercial products are available in the market to aid in this task, and the official product at CERN is Infor-EAM [2].

Infor-EAM provides functionality for vacuum in three domains:

- Equipment Management – allows tracking of the equipment under the responsibility of the vacuum group for: location, works performed, and hierarchical relationships (control hierarchy and installation hierarchy).
- Work Management - creation of work orders on equipment and post-analysis – who did what and when.
- Stock Management - tracking of inventory levels and equipment locations within the stock.

In addition, equipment is distinguished into three categories:

- Assets – physical equipment, labelled with a unique barcode identifier. Assets can be installed in functional positions or associated to other assets, forming a hierarchical relationship.
- Functional Positions – placeholders for physical equipment. The names of functional positions correspond to the names of equipment seen in the SCADA.
- Systems – a logical grouping of assets or functional positions.

The following terminology is also required for the understanding of the coming chapters

- Equipment Code – identifier for the type of equipment. Example: HCVRGPT300 (TPG300 crate)
- Asset Code – identifier for a physical unique piece of equipment. Asset codes contain the equipment code, a manufacturer code and a sequence of digits. An example of an asset code is HCVRGPT300-CR000056, where HCVRGPT300 is the equipment code (in this case a TPG300 crate), CR the manufacturer (CERN), and 000056 the sequence number.
- Functional Position Code – an identifier for a functional position. The construction of functional position codes is subject to the naming conventions on where the functional positions are inserted.

Problems When Managing Assets

Prior to the usage of vacDM, when new equipment was received or manufactured, an asset code would be generated and a label printed and attached to the device for future identification. The equipment would then be manually declared in Infor-EAM so that the information about the new asset could be introduced and the device tracked for location and works performed on it.

According to registries held in excel sheets on the number of labels printed for each asset type, it was observed that the number of labels printed did not match the number of assets declared in Infor-EAM by several thousands, posing an issue for the traceability of vacuum assets. It was determined that this issue arose due to the lack of tight linking between the printing of labels and the declaration of assets.

Problems When Managing Functional Positions

While it is possible to manually import all equipment available in vacDB to Infor-EAM, this operation is time-consuming, error prone, and frequently needed, due to continuous upgrades on the vacuum system of CERN accelerators.

The Solution

In order to ensure that asset declaration in Infor-EAM is tightly linked with barcode printing, and that functional positions are synchronized with vacDB with the correct descriptions defined in CERN's naming database, a web-based application was developed. This application is called vacuum data manager, referred to as vacDM. This paper describes the functionalities implemented as well as the architecture and the technologies involved.

OVERVIEW OF FUNCTIONALITIES

Asset Management in vacDM

Via a web interface (Figure 3), users are able to request labels for new assets by choosing the equipment code. The equipment codes available for label printing are displayed in a tree structure, linked with CERN's naming database to ensure that only official, approved codes, are used.

After a user selects the equipment code and number of assets to be created, vacDM accesses the Infor-EAM database to check the latest asset sequence number printed for this equipment code. The sequences of the new asset codes are then calculated, combined with the official equipment code description for the asset available in the namingDB, and the assets are created using Infor-EAM's SOAP API. In parallel with the creation of assets, an email is sent to the user responsible for printing the labels and a ticket is created in JIRA for the printing job.

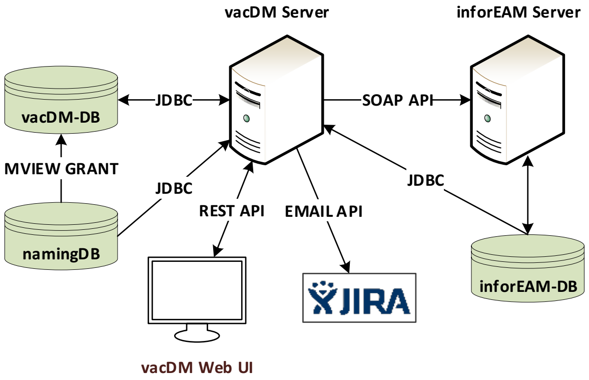


Figure 3: Asset Management sub-system architecture.

The historical records of all asset requests, along with the status of the creation attempts for each asset in Infor-EAM are kept in the vacDM-DB. For assets that failed to be created, a further import can be launched.

Functional Positions Management in vacDM

For functional positions, the main objective set for vacDM was the elimination of discrepancies between data stored in the vacuum SCADA, Infor-EAM, and the naming portal. To achieve that (Figure 4), a materialized view on the vacDM-DB joins the functional positions declared in vacDB and Infor-EAM-DB, with the equipment code descriptions in the naming-DB. This materialized view provides a transversal view on functional positions data and is used by vacDM to check for inconsistencies between the three databases. This materialized view is refreshed daily or on-demand, and a scheduled job updates Infor-EAM with the differences detected with respect to vacDB or naming portal.

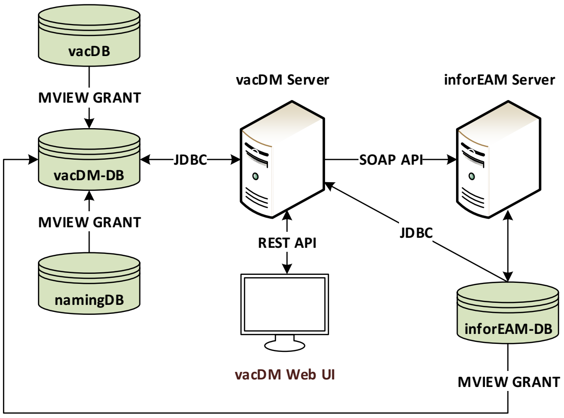


Figure 4: Functional positions sub-system architecture.

Work Order Generation from the SCADA

Having vacDB functional positions automatically synchronized with Infor-EAM allowed for the development of a Work-Order generation tool in the vacuum SCADA (Figure 5). Accessible with a right-click on every equipment, the work order generation tool provides a subset of vacuum relevant work order options (Figure 6) in relation to all available fields in Infor-EAM.

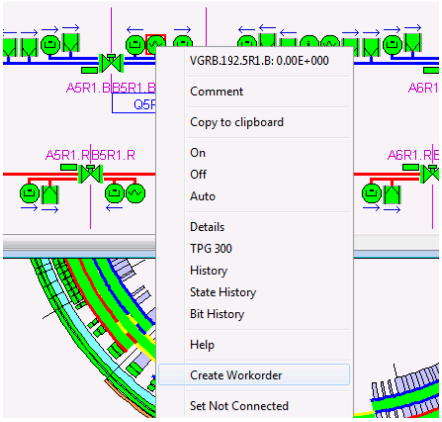
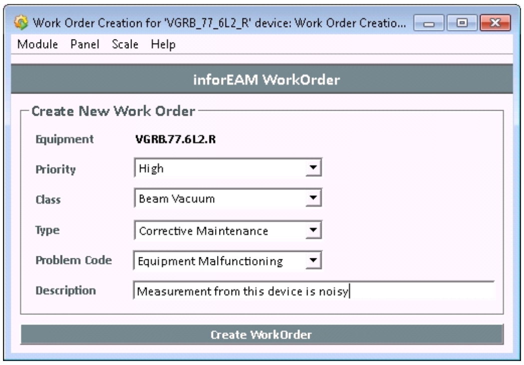


Figure 5: Work Order generation in the vacuum SCADA.


Work Order Creation for 'VGRB_77_6L2_R' device: Work Order Creatio...

Module Panel Scale Help

inforEAM WorkOrder

Create New Work Order

Equipment: VGRB.77.6L2.R

Priority: High

Class: Beam Vacuum

Type: Corrective Maintenance

Problem Code: Equipment Malfunctioning

Description: Measurement from this device is noisy

Create WorkOrder

Figure 6: Work Order Generation Tool.

After the user pushes the button *Create WorkOrder*, the SCADA application opens a web browser with a pre-filled WorkOrder.

ARCHITECTURE AND TECHNOLOGIES

The vacDM application is built around the Spring Framework (Figure 7), and therefore it inherits many of its architectural traits. Spring is a Java EE open source framework created to address the complexity of enterprise applications development [3], and relies on the principle of Inversion of Control (IoC) by automatically managing bean creation and dependency injection [4].

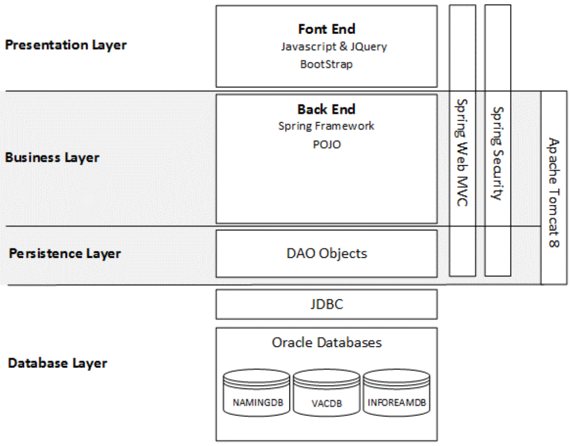


Figure 7: vacDM architecture.

This approach enables the development of java classes in a loosely coupled fashion, promoting single-responsibility and separation of concerns. Several spring modules were bundled together to address the requirements of vacDM:

- Spring core – provides the basic functionalities of spring such as bean dependency injection, lifecycle management, auto-wiring, etc…
- Spring web-mvc – provides web functionalities on top of the spring core using a model-view-controller architecture.
- Spring security – provides authentication and authorization management.
- Spring LDAP – provides wrappers that simplify LDAP operations, used to authenticate users against CERN's active directory database.

- Spring Data – provides functionality for database access and support for transactions.

Taking into consideration Spring design principles, vacDM was developed into a four-tier architecture, composed of the presentation, business, persistence, and database layers.

Presentation Layer

The presentation layer is responsible for the visual part of the application. It runs partially on the web-server and partially on the client's browser. Once the static content of the webpage is rendered by the server and sent over to the client, the client then issues web-service requests to get, update and delete data, according to the user inputs.

In the presentation layer, vacDM makes use of the Apache Tiles view resolver on the server-side and of Bootstrap, CSS, Jquery and Javascript on the client-side to handle the aspect, animations, and API requests to the backend server.

Business Layer

The business layer implements spring controllers for handling web resources, restful web-services, and general business logic. After receiving an HTTP request, the business layer controllers might access data-access object (DAO) functions in the persistence layer to get database data. After the data is processed according to the business logic and, depending on the requested URL, the business layer controllers return a view, later processed by a view resolver on the presentation layer, or JSON objects that will trigger a renderization of the concerned module on the client's browser.

Persistence Layer

The persistence layer implements the DAO objects that provide abstraction to the business layer of the database details. The classes in this layer follow the DAO pattern, providing a Java interface to the business layer, and allowing an easy swap of the implementation classes in case needed.

Database Layer

The database layer is composed of Spring JDBC template beans and CERN hosted ORACLE databases. DAO classes access database parameters through the Java Naming Directory Interface (JNDI).

Security

The application is secured using Spring Security, a Spring module that provides the backbone for managing authorization and authentication, being customizable to address specific user needs. The vacDM spring security module was customized to obtain user authentication from CERN's active directory database. This was done to avoid the duplication of accounts between CERN's infrastructure and the vacDM service.

For authorization, vacDM implements a concept of user roles. A set of roles is associated with each user, being the roles what defines which pages or web services the user will have access to.

Out of the box and with minimal configuration required, spring security provides a login/logout mechanism, and protection for cross-site request forgery (CSRF), session fixation, and the ability to secure the application at the method level.

Servlet Container

The servlet container chosen for the application was Apache Tomcat 8, an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Websocket technologies. It is used in several large-scale and mission-critical web applications across a diverse range of industries and organizations [5]. Tomcat is as of 2017 the most widely used application server for Java [6], having a large developer community and plenty of online support resources.

DEVELOPMENT STRATEGY

In order to minimize the risk of propagation of software faults to the production server, a continuous integration pipeline was set up using a Continuous Integration (CI) server (replica of the production server) and the gitlab platform (Figure 8). After every *git push*, the vacDM CI server pulls the affected branch, runs units tests, builds the application using maven, and deploys it locally to http://vacdm-dev/{branchName}. This allows developers to test their feature branches in an environment similar to the production server.

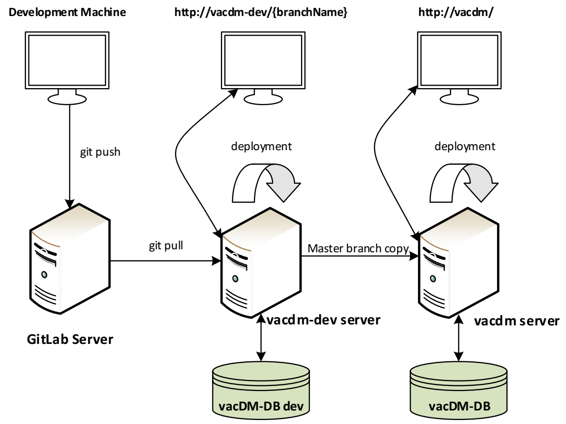


Figure 8: Continuous Integration architecture.

All *git push* commands on the master branch are automatically deployed to http://vacdm-dev/, allowing developers to test the next release of the application. From Gitlab's user interface, developers can trigger an automatic deployment to production. Upon a deployment to production, the CI server stops the tomcat service on the production server, copies the web application and restarts it. This enables an automatic application deployment in less than a few minutes.

CONCLUSION

The first version of vacDM was successfully put in production and is now managing over 9000 pieces of vacuum equipment in the LHC. With no further developments required, the SPS and CPS machines will also be integrated in vacDM, after the completion of an on-going database consolidation campaign. This will enable the vast majority of vacuum equipment to be managed in Infor-EAM, improving the tracking of equipment throughout its lifecycle. After 6 months of usage of vacDM, the number of ghost assets (with label printed, but not declared in Infor-EAM) has been reduced to zero. Similarly, 100% of the SCADA functional positions targeted to be managed in Infor-EAM have been successfully imported.

FUTURE WORK

To ensure further consistency with CERN's layout database, vacDM needs to be updated to also import layoutDB names as aliases on the equipment it creates in Infor-EAM, along with the database links between Infor-EAM and layoutDB, to allow the navigability between the layoutDB portal and Infor-EAM.

On the SCADA side, several new features will be developed to improve the user experience:

- Direct links from the equipment menu in the SCADA to the layoutDB webpage
- Direct links from the equipment menu in the SCADA to the Infor-EAM webpage
- Automatic WorkOrder creation upon detection of abnormal conditions, such as equipment failure or overpressure.

REFERENCES

[1] P. Gomes *et al.*,"The control system of cern accelerators vacuum [current status & recent improvements]", in Proc. *International Conference on Accelerator and Large Experimental Physics Control Systems 2011, (ICALECPS'11)*, Grenoble, France, Oct 2011, pp. 354-357.

[2] D. Widegren, "Asset & Maintenance Management at CERN with Infor EAM", private communication.

[3] A. Mukherjee, Z. Tari, P. Bertok, "A Spring Based Framework for Verification of Service Composition", *in 2011 IEEE International Conference on Services Computing (SCC).*

[4] Spring Docs, https://docs.spring.io/spring/docs/current/spring-framework-reference/html/beans.html

[5] Tomcat, https://tomcat.apache.org/index.html

[6] Plumbr, https://plumbr.eu/blog/java/most-popular-java-application-servers-2017-edition