CENTRALISED DATA ENGINEEERING FOR THE MONITORING AND CONTROL OF THE CERN ELECTRICAL NETWORK

Anargyros Kiourkos, Jean Charles Tournier, Sonia Infante, Manuel Gonzalez-Berges, Matija Zanko, Georges Burdet
CERN, Geneva, Switzerland

Abstract

The monitoring and control of the CERN electrical network consists of a great variety of devices and software: it spans from low level acquisition devices to high level data concentrators, supervision systems as well as power network simulation tools. The main issue faced nowadays for the engineering of such a large and heterogeneous system, including more than 20,000 devices and 200,000 tags, is that all devices and software have their own data engineering tool while a great part of the configuration data has to be shared between two or more devices: the same data needs to be entered manually to the different tools leading to duplication of effort and many inconsistencies. This paper presents a web-based application called ENSDM, mainly based on open technologies, aiming at centralizing all the data needed to engineer the monitoring and control infrastructure into a single database from which the configuration of the various devices is extracted automatically. Such approach allows the user to enter the information only once and guarantee the consistency of the data across the entire system. The paper also demonstrates the configuration workflow of different parts of the system, namely the remote terminal units, devices, the global supervision system (SCADA) and the power network simulation tools.

PROBLEM DESCRIPTION

The monitoring and control system for the electrical network of CERN (ENS) is based on a three-layer architecture which consists of field devices (protection relays, IEDs etc.), remote data concentrators (RTUs) and centralised SCADA servers. The majority of the equipment in every layer includes some type of software based engineering tool which allows the configuration of the respective devices and the addition, deletion or modification of monitoring parameters.

This multi-layered structure implies that almost every configuration change in one of the layers requires changes in all the others. This process is usually manual, making engineering of large scale systems like the ENS, tedious and prone to data entry errors. Especially during system migrations, upgrades or large scale projects affecting the system, the task of maintaining data integrity across all layers becomes increasingly difficult, as the volume of configuration data and information exchange between the different systems increases.

This has been proved in practise during the Long Shutdown 1 at CERN, where extensive data re-engineering of ENS is required due to the fact that a major upgrade of the LHC and the other of CERN's accelerators are underway, including the SCADA of the electrical network.

In the sections that follow the concept of the implemented solution is presented followed by the architecture of the developed system and an overview of the technologies utilised. The main problems encountered during development are followed by the initial results, the remaining works, future improvements and additions.

IMPLEMENTED SOLUTION

The proposed solution (ENSDM) involves the centralised data engineering across all layers. Any configuration change in any of part of the SCADA system is entered through a common homogenised user interface or one of the designated entry points. ENSDM is responsible for collecting, maintaining and dispatching the required configuration information across the different domains. Data is stored centrally in a single database utilising CERN's existing database infrastructure. Existing interfaces, where available, have been re-used and respective new ones have been developed to accommodate the required information flow between the different systems. Open technologies are used to provide flexibility and future proof the development [1].

ISBN 978-3-95450-139-7

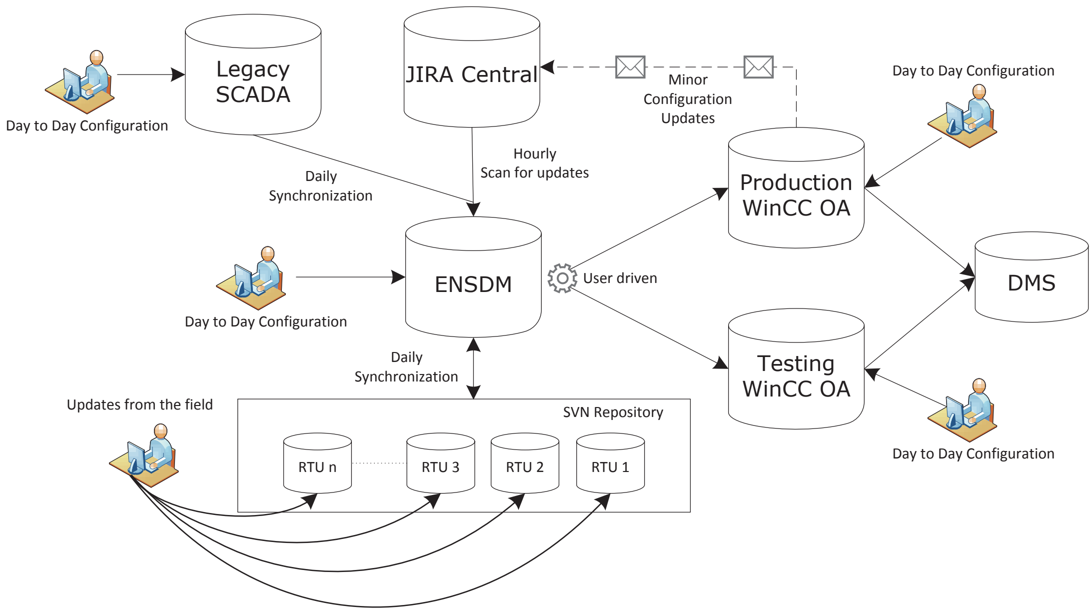

Figure 1: Engineering Data Flow ENSDM.

ENGINEERING WORKFLOW

The preferred input to the system is from the user interface of ENSDM. From there the operator configuring the system can add, modify or delete single data points, devices or data concentrators across the system. ENSDM user interface incorporates validation logic to check the data entered and minimise data input errors.

The configuration is object oriented, individual properties or attributes do not exist as standalone but they are always attached to objects. This modelling, originating from the specification of the new SCADA represents more accurately the situation on the field and allows the bulk engineering of objects and reuse of existing configuration.

The electrical characteristics of the devices are also entered and stored with the device so that they can be used by the power network simulation tools. This data can then be forwarded to the SCADA system interface tables which are presented in greater detail in another section. The operator has the option to pull this configuration to WinCC OA system of choice (test or production).

Operational needs though require changes also on the field level. Operatives may change configuration on the concentrator, especially in periods of testing and commissioning. ENSDM scans daily these configuration files, which are stored in an SVN repository and updates the system with the latest changes. The SCADA system will then be updated at the next pull request.

Configuration changes at the SCADA level are only supported in limited extent and this is restricted mainly to parameter change. These changes are fed back via an automated email to the CERN issue tracking system based on JIRA software by Atlassian, where a support ticket is created. ENSDM scans the specified module of the JIRA Central and updates the database accordingly.

For the interim period where the migration of the SCADA system is underway the system also synchronizes with the legacy system. Any configuration changes there are also collected and saved in the ENSDM database.

It is important to mention that the objects (tags, devices & RTUs) in ENSDM have predefined states. These signify their current status and are the following:

- NOT_COMMISSIONED: Object exists only in ENSDM (new installations not yet started or not in service)
- NOT_INTEGRATED: Object exists in ENSDM and on the field and RTU but not in the SCADA (new installations visible only at field level).
- INTEGRATED: Object has been commissioned and is operational, visible to all layers of the ENS system.
- DECOMMISSIONED: Object was in service and visible to all layers but has now been removed from SCADA but still exist on the field.
- Objects that are removed from the field are deleted from the system.

It is important to note that these states are automatically detected and assigned by the ENSDM in the majority of cases.

ARCHITECTURE & TECHNOLOGY

General

ENSDM is a multi-tier web based application. The different functions of the application are logically separated with the primary design principle being the Separation of Concerns (SoC) [2]. Each section of

ISBN 978-3-95450-139-7

ENSDM targets at a specific area and all interactions between the different areas are through well-defined interfaces. In addition the layered architecture implies that future extensions (e.g. an API or a user interface for mobile devices) can be added by re-using the existing services and adding the required components. The majority of the tools used are based on open technologies and were possible existing infrastructure has been re-used.

Application Platform

The application platform of ENSDM is JAVA and specifically J2EE. J2EE is a standardised, well-documented and reliable platform, well known for its openness (cross platform, open source), scalability and stability. The abundance of frameworks and support, significantly reduce the complexity, something that has been the weakness of J2EE in the past, and accelerate development [3].

The web architecture ensures that the application has minimal client requirements, provides high accessibility, has significant cost benefits in the deployment and maintenance and is suitable for multi-user operation.

Spring Framework

Spring framework [4] has been used extensively in ENSDM. It consists of several modules that provide a wide a range of services. The following of these modules have been used in ENSDM:

- Inversion of control container: management of application components and lifecycle of Java objects, through dependency injection
- Aspect-oriented programming: allows the implementation of cross-cutting concerns.
- Data access: strong integration with Hibernate
- Transaction management: annotation driven transaction management.
- Authentication and authorization: not intrusive security with the Spring security module, which support annotation driven access control at a function level.
- Testing: strong integration with JUnit.
- Scheduling: annotation driven job scheduler.

The framework is the key component of the application as it reduces complexity, increases productivity and provides enterprise level features to the application.

Data Access Layer and Database

CERN has extensive experience with ORACLE technology and respective infrastructure which hosts the development and production databases of ENSDM.

With the increasing complexity and the depth of the domain models of modern applications though, the use of object-relational mapping (ORM) frameworks is becoming the standard with regards to database access. ORM frameworks enable the isolation of the business logic from any relational issues that might arise in the persistence layer. Additionally the addition or removal of properties in any of the domain models does not require refactoring of the data access code [5].

Business Layer

The business layer of ENSDM is service based. This approach allows the clear segregation of functions and makes the system open to extensions and future additions. The same service can be reused by the different presentation layer and other APIs. Furthermore the service oriented design facilitates scalability and reuse of code and functionality [6].

Presentation Layer

The agile methodology that is used for the development of ENSDM dictates the use of tools and frameworks that accelerate development by hiding complexity, increasing efficiency and productivity. Especially with web applications a significant amount of the development effort goes into the Presentation Layer. The user interface is the front end of the system and plays a key *role* in user experience. *It* is not uncommon to use *many* different technologies *in order* to achieve optimum result (JavaScript, *CSS* html, JSP etc.).

The UI of ENSDM is developed following the MVC pattern by utilising JSF 2.0 and the PrimeFaces JSF component suite. JSF is a request driven MVC framework based on a component-driven UI design model and is intended to simplify development of web-based user interfaces with built in AJAX support and other features. JSF aims to minimize the amount of JavaScript and CSS required for achieving an optimum result.

Interface with SCADA and Other Systems

The SCADA is engineered through its own input database. The interface between ENSDM and the SCADA is implemented by a PL/SQL script which transforms the data to the required format by the SCADA. The reason for having an intermediate set of tables between ENSDM and the SCADA is to enforce the separation of concerns between the systems, thus ENSDM does not to be aware of the specifities of the SCADA (e.g. the internal name and ID of the different datapoint names) and vice versa. The supervision system is actually built with two systems: the SCADA itself based on Siemens WinCC OA and a calculation engine for power system (e.g. for state estimation and contingency analysis) based on PowerFactory from DigSilent. All the data needed by both systems is stored in ENSDM, but the ENSDM only interfaces with WinCC-OA which in turn configures PowerFactory: such approach enforces data consistency between the SCADA and the calculation engine as any data import to SCADA is automatically propagated to the calculation engine. While the above described process is unidirectional (from ENSDM to SCADA), it envisions to be able to propagate engineering data from SCADA to ENSDM (e.g. alarm, archive or communication settings) through the automatic creation of a JIRA request. It will allow performing minor engineering changes directly on

ISBN 978-3-95450-139-7

the SCADA, while still requiring the approval of the responsible to be stored in ENSDM.

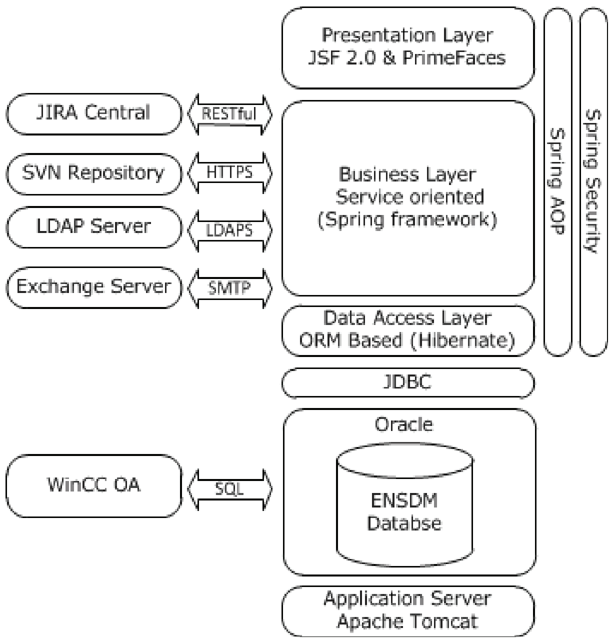

Figure 2: ENSDM architecture.

PROBLEMS ENCOUNTERED

The biggest challenge during the development of ENSDM is the actual problem the application is built to address. That is the data consistency across all domains in an environment of constant change. Configuration changes occur daily in all layers due to changing needs of the projects that affect the control and monitoring system of the electrical network. Numerous engineers reconfigure the ENS system and unless this data is fed back through one of the predefined paths, changes are not visible to the system.

Furthermore since the migration of the SCADA is in progress the system needs to evolve to meet the changing needs of each stage. E.g. during start-up the focus is on getting the existing data into the system but as the migration progresses the focus is into configuring new equipment.

Another issue has been the lack of homogenisation in the configuration data. Rules cannot be universally applied and there are cases where data has to be manually rationalised.

PRELIMINARY RESULTS

As the commissioning of both the SCADA and ENSDM is at its early stages there are only preliminary results available. These have been positive as the data checked is consistent across all layers and data flow is efficient and smooth from and to all relevant systems (SCADA, SVN, ENSDM etc.).

CONCLUSION & FUTURE WORK

Data engineering in multi-layered environments can be prone to errors and can lead to data inconsistency. Further testing and full time operation will provide reliable data in order to access the full benefits of the centralised data engineering approach realised by ENSDM.

The focus of development is now on finalising the RTU engineering part of the application and ability to bulk engineer new equipment based on existing installations.

REFERENCES

[1] Cerri, D. & Fuggetta, A. 2007, "Open standards, open formats, and open source", Journal of Systems and Software, vol. 80, no. 11, p. 1930-1937.
[2] W. L. Hursch & C. V. Lopes 1995, "Separation of Concerns", College of Computer Science, North-eastern University Boston, MA 02115, USA.
[3] Boranbayev, A.S. 2009, "Defining methodologies for developing J2EE web-based information systems", Nonlinear Analysis: Theory, Methods & Applications, vol. 71, no. 12, p. e1633-e1637.
[4] Walls, C. 2011, Spring in Action, 3rd Edition, Manning Publications, Greenwich, USA.
[5] Seddighi, A.R. 2009, Spring Persistence with Hibernate, 1st Edition, Packt Publishing, Birmingham
[6] Draheim, Dir 2010, "Business Process Technology", Service-Oriented Architecture Springer Berlin /Heidelberg, p. 221-241