ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA151

MARS: EASING MAINTENANCE AND INTERVENTIONS FOR CERN CONTROLS

F. Varela, U. Epting, M. Gonzalez, E. Mandilara, S. Podgorski, CERN, Geneva, Switzerland

Abstract

Industrial control systems for the CERN technical infrastructure and accelerator complex consist of a myriad of devices and components geographically distributed around the CERN facilities. In the event of an intervention in such systems, the on-call engineer or the system expert needs detailed information about the nature of the problem, e.g. what device, what problem, intervention procedures, and contextual data like the location of the device, current access conditions to this place, the list of access rights required and whether he/she is granted with these rights. This is of special relevance when the person responsible for the intervention has only limited knowledge of the control system as it is the case for some on-call services. At CERN, this information is scattered over a number of data sources. This paper presents MARS (Maintenance and Assets for contRolS), a web-based tool designed to federate data from heterogeneous sources with the aim of providing support for interventions and maintenance activities. The information can be displayed in a single web page or be accessed through a REST API.

INTRODUCTION

The Industrial Controls and Safety systems group of the Beams Department (BE-ICS) at CERN provides turn-key solutions for controls, as well as safety and access controls systems for the experiments, accelerator complex and technical infrastructure. In the domain of industrial controls, BE-ICS develops full controls systems ranging from PLCs or front-end computers executing real time tasks, to supervisory applications using the commercial SCADA package WinCC Open Architecture (OA) [1] from Siemens/ETM. All these controls applications are made using common building blocks from the two industrial controls frameworks developed on top of WinCC OA, JCOP (Joint COntrols Project) [2] and UNICOS [3].

Although all these applications are built from standard components, they can be very different depending on the functionality required and the domain where they are deployed. This also applies to the technologies chosen from the BE-ICS catalogue to implement the applications, e.g. Siemens vs. Schneider PLCs, Modbus vs. OPC Unified Architecture, Oracle Archiving vs. file archive, etc. Typically, specialists in different technologies or layers of the control system, e.g. PLC and SCADA developers, work together in the development of these applications.

Most of these applications have a very long lifetime, typically the LHC lifetime, which given the large turnover of people at CERN, in some cases, it may represent a challenge to ensure that the detailed knowledge about the application is preserved.

BE-ICS is currently responsible for the development and maintenance of more than 200 SCADA applications running on around 100 Linux servers. These applications comprise more than 400 PLCs from both Siemens and Schneider and around 40 front-end computers. These controls devices are geographically distributed around the CERN premises.

The large number of applications, the variety of technologies used, as well their geographical distribution represent a major challenge for on-call services that have to provide coverage to all these applications and control devices. The industrial controls standby service is formed by members of BE-ICS who are specialized in a particular set of technologies and who participate in the development of a number of applications. Having the knowledge about every single application provided by BE-ICS and all technologies involved in their development would be impossible. Moreover, besides the technical knowledge, a lot of contextual information is also required in order to perform interventions in the controls applications. The following are some of the most common questions to be addressed prior to an intervention:

- What is the device affected and what problem is?
- What are the procedures to follow? Where to find them?
- Where is the device located? How can I access this location? Do I have all necessary access rights?
- What are the dependencies with other systems?
- What software runs on this device? What version of the program? Where do I get a copy from?
- What is the device made of? Where do I get spare parts from?

However, the information required to reply to these questions is scattered over multiple sources with no clear relations between them as it will be described in the next section.

ASSET INFORMATION MANAGEMENT

CERN's technical infrastructure is maintained by many specialists and experts in different domains like cooling and ventilation, electricity, cryogenics and computing infrastructure. Each expert group is responsible for the documentation, configuration and handling of their equipment. Different systems and databases are used for specific tasks, thus leading to have information about assets, their functionality and use scattered in different data sources. Each data source is used to describe a specific aspect of an asset in detail, but only the combination of all data will give a complete picture for an equipment.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA151

The following list gives a brief overview of the most important data sources and the information they contain:

- LANDB - database containing all equipment that can be connected to the CERN network.
- InforEAM - asset management and maintenance database (functional positions, assets, spare parts, maintenance plans, etc.)
- EDMS - electronic document management system (documentation, pictures, technical drawings, etc.)
- ICESAS - industrial controls equipment configuration database (Applications layout, software configuration, etc.)
- Layout - database containing data about physical composition of equipment (Physical installation data, location hierarchy, etc.)
- GIS - geographic information system (geographical coordinates, maps).
- IMPACT - Intervention Management Planning & Activity Coordination Tool (work planning and authorizations)
- ADAMS - Access Distribution and Management System (access authorizations, safety training requirements, etc/)
- LASER/PHOENIX - alarm console service for the CERN Control Centre (live and historical alarm data).
- HelpAlarm - additional information and documentation for alarms (Intervention procedures)
- MOON - monitoring of industrial control systems (detailed alarms for control systems equipment).

These data sources are independent instances and only few of them use common key fields to link data between them. In addition, naming conventions and implementation constraints prevent from direct linking between data sources. However, with some experience and data mining, it is possible to find correlations which can be used to find useful data for display in MARS.

MARS

The BE-ICS group has developed a tool called MARS to assist members of the standby services and equipment responsible groups during interventions. MARS accesses all data sources mentioned in the previous section to present users with a comprehensive view of the problem with a device, its location, current access conditions, as well as all information necessary to replace a device, for instance, the current PLC composition, where to get spare parts and pointers to the software repository hosting the version of the program required by the device.

MARS allows querying the different data sources by using the functional position of a device (name bound to a physical location and purpose) or by the asset code (ID bound to an actual physical device). Functional position remains unchanged even if the asset is replaced. In most cases, queries will use the functional position, however, there are special cases where users will only have access to the asset code, e.g. in many cases, only the asset name is printed on the physical devices and encoded in a barcode. For this reason, a proof-of-concept mobile application also allows scanning device barcodes to access the information gathered by MARS. This functionality is important for users to verify the identity of the equipment prior to starting the intervention.

In the following sections, the architecture and technologies used for the implementation of MARS, as well as its main functionality are described.

Architecture

MARS is designed as a modern application compliant with Backend-As-A-Service approach, where the frontend and the user interface are separated from the business logic part. The business logic can be accessed as if it was an external service, via REST API. This enables multiple independent frontends to connect to the backend and third-party tools to use MARS as a data source. Figure 1 shows the architecture of MARS as well as the different data sources accessed.

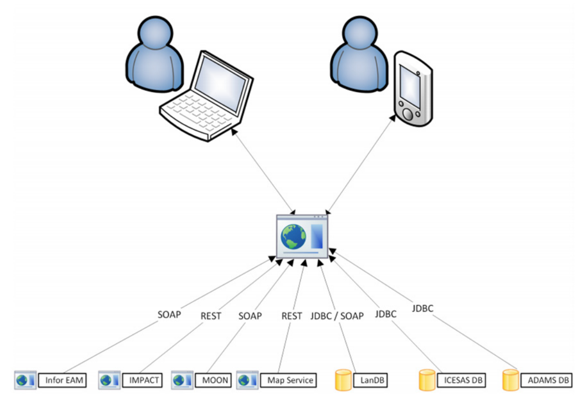

Figure 1: MARS architecture showing the main data sources accessed by the tool.

The backend federates all data coming from the different sources and acts as a proxy to these systems. MARS does not store any data resulting from the queries to the data sources. Moreover, MARS does not communicate directly with any operational equipment. The downside of this approach is a strong dependency on the availability of the external data sources.

The front-end of MARS is based on modern Web technologies like Angular JS [4] and Bootstrap [5].

Implementation

The application is being developed as a multi-module project with Maven [6] as underlying build technology to provide a clear separation of different functions and to simplify concurrent development.

The separation of business logic into modules is based on the data sources - each data source has its own dedicated module. Modules are self-contained and their structure follows the same pattern:

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA151

- Data Transfer Objects definitions - POJO classes which are mapped to JSON as outputs from MARS backend, and input data definitions received from external systems.
- Domain logic - all domain data processing, correlating information and collecting data from external sources.
- REST endpoint definition - implementation of Spring Model View Controller REST Controllers, matching URLs with appropriate service calls.

Although currently all modules are deployed within a single web container, the design follows a *microservices* approach, where each module can be deployed separately. Although there is a certain level of dependence between the modules, these dependencies are tracked and resolved by Maven when packaging the module to include all necessary libraries.

The start-up of the application is handled by Spring Boot, which allows creating a standalone build, requiring only Java 8 to run, without any additional application server or external web container required. The Spring IoC container handles the initial application setup and via Java CDI annotations, it sets up all of the server side services. REST endpoints are created using Spring MVC for simplicity and to avoid introducing dependencies with other packages.

Backend-Frontend Interplay

The response time of MARS, as well as its availability and provided functionality depends on the availability of the external data sources. This is especially relevant when the connection from MARS onto the remote source gets stuck with no response at all, e.g. if the remote service is under heavy load and will only respond after some delay. This issue is mitigated both on backend and on frontend side. The backend calls to external services are performed with a predefined timeout and the connection is automatically broken if no data is received after the timeout is reached. However, this is still not enough to provide a good user experience, because the service response might be too long if several timeouts are reached. Shortening the timeouts could limit the amount of collected data if the services take longer than expected to respond. To avoid this, data is accessed separately for each of the sources. In this way, the frontend is loaded instantly while the queries are sent via AJAX calls to each of the endpoints, and once responses come the data is placed in respective placeholders. In this way, the data is incrementally presented to the user as soon as received.

Communication Protocols

The data sources and services accessed by MARS were developed at different times, by different teams and using different technologies. This imposes the utilization of multiple protocols to communicate with them, as shown in Figure 1. In some cases, there is no API to access the data sources or the API is quite limited and does not expose the data relevant for interventions. In these cases, the information is directly queried from the underlying databases using JDBC. For those data sources exposing data via a REST API, Spring REST Template is used for the data exchange. Finally, some sources expose data using SOAP web services but even here, they use different versions of the SOAP standard and therefore the clients use both Apache Axis and Apache CXF. As output the MARS backend provides standard REST API endpoints with cross-origin access enabled, so that any authenticated user can access the data.

User Interface

Since MARS can also be accessed mobile devices (i.e. in case of interventions, to scan PLC barcodes), the user interface was designed to be responsive, as well as simple as possible, providing the important information in a thrifty way. Some fragments of the MARS user interface are in Figures 2-5. Opening MARS causes the dispatch of multiple AJAX requests to the respective REST endpoints in the backend.

The main web User Interface of MARS follows the same organization of the information as backend services. If information for the queried equipment is found in a data source, its data is placed inside the respective modules. Each module corresponds to an AngularJS directive.

In the following, we will assume the case of user having to intervene on a PLC to describe the information displayed in the user interface of MARS.

Network information Every piece of equipment connected to the CERN network is registered in the Network Database. These database holds information like the device type, manufacturer and model, geographical location of IT outlet as well as person responsible for the equipment. Figure 2 shows the information gathered from the CERN network database by MARS.

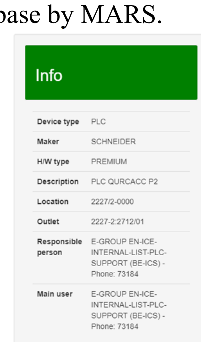

Figure 2: Fragment of the MARS UI showing device information from the CERN network database.

Structure of the controls applications PLCs are typically connected to a data server hosting the SCADA application. The structure of these applications is stored in the System Configuration Database of the JCOP Framework [7]. From this data source, MARS retrieves the physical links of the PLC, as well as the name of the supervisory application and the versions of the components used to generate the application. Figure 3 shows the data server as well as the WinCC OA application a PLC is connected to as well as the different versions of the packages used to engineer the application.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA151

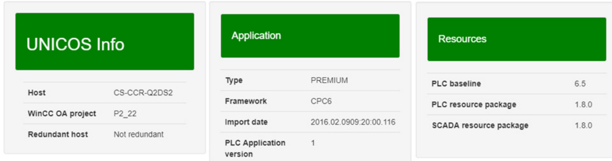

Figure 3: Fragment of the MARS UI showing the links between a PLC and the Supervisory application, as well as the different version of the packaged used for its automatic generation.

Equipment composition The list of hardware elements composing a device and its hierarchical dependencies are retrieved from InforEAM. For PLCs, the MARS web interface displays I/O, memory and communication cards and its power supplies.

Monitoring and diagnostics the BE-ICS group provides a service to monitor controls applications using MOON [ref]. In particular, for PLCs, the state of the different alarms configured, as well as the contents of diagnostic buffers of the device are shown. This information gives a first indication of the type of problem experienced by the PLC. Figure 4 shows the diagnostic information in MARS for a PLC.

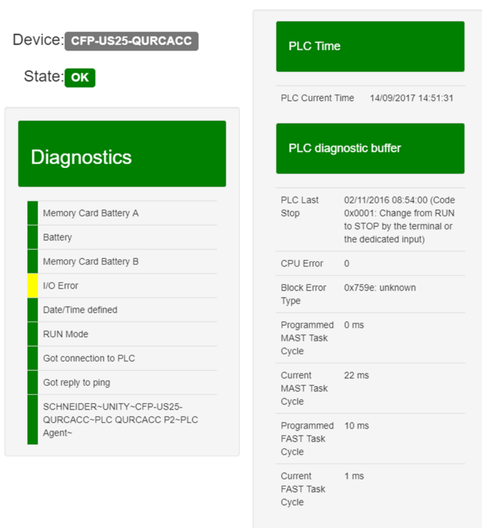

Figure 4: Fragment of the MARS UI showing the diagnostics information for a PLC.

Work orders Past interventions on equipment are traced in InforEAM by means of work orders. MARS displays the list of last 10 work orders for the device. This permits users to understand the nature of the last interventions on the devices, as well as it serves to spot recurring failures.

Equipment location The CERN layout database is used to resolve the rack where the equipment is hosted. Using the CERN GIS maps, the user interface of MARS shows the exact location of this rack. When the rack information is not available in the layout database, e.g. this information has not yet been entered in the database, the GIS map will show the location of the network outlet of the device. Figure 5 shows the exact rack location in a GIS map where a PLC is located.

Figure 5: Fragment of the MARS UI showing the rack location housing a PLC in a GIS map.

Access control Most of the equipment is located in restricted areas. areas that are not accessible without the right permissions. MARS uses the access control database at CERN, ADAMS, to display the different access points to the location of the equipment, as well as the current access conditions and if access granted to the user. In case, access is not granted, MARS also tries to resolve the reasons, e.g. a special access request is still pending to be approved.

OUTLOOK

Today the MARS tool can be used to display information about a well-defined equipment types, such as PLCs and their hardware modules. The existing data from the different data sources would also allow to extend this information to show the status of related systems or equipment located higher in the installation hierarchy, e.g. the cooling systems affected in the event of a PLC crash.

Further plans comprise establishing links to the CERN Electronics Document Management System [8] and software repositories like Gitlab [9] or VersionDog [10], a PLC software repository run by BE-ICS. This would enable to present users with links to the technical documentation for a particular device, as well as with direct pointers to download the exact version of a program running on a faulty device.

Another extension envisaged is the integration of the help alarm procedures used by the operators in the CERN Control Centre. These procedures are associated describe the steps to take, e.g. whom to call, possible cause of the problem, how to troubleshoot the problem when an alarm is raised. It is also foreseen to interface MARS to the catalogue of spare parts to inform users about the availa-

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA151

bility and location of replacement modules and devices. In addition, MARS will also profit from the ongoing improvements to the CERN maps, e.g. by adding photos of the access points and the underground facilities. Finally, as many different services add online functionality to their systems, direct links with prefilled data fields could be provided to enable data correction and to document interventions directly from MARS. An example of this could be the upload of photos taken during an intervention to EDMS directly using the MARS mobile application.

REFERENCES

[1] WinCC Open Architecture, http://www.etm.at/
[2] O. Holme, M. Gonzalez Berges, P. Golonka, S. Schmeling, "The JCOP Framework", ICALEPCS 2005, Geneva, Switzerland.
[3] H. Milcent et al., "UNICOS: An open Framework", ICALEPCS 2009, Kobe, Japan
[4] Angular JS, https://angularjs.org/
[5] Bootstrap, http://getbootstrap.com/
[6] Maven, https://maven.apache.org/
[7] F. Varela, "Software management of the LHC Detector Control Systems", ICALEPCS, 2007, Knoxville, Tennessee, USA.
[8] CERN Electronics Document Management System, http://www.cern.ch/edms/
[9] Gitlab, https://about.gitlab.com/
[10] VersionDog, https://www.versiondog.com/home.html