ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA021

LARGE-SCALE UPGRADE CAMPAIGNS OF SCADA SYSTEMS AT CERN - ORGANISATION, TOOLS AND LESSONS LEARNED

R. Kulaga†, J. Arroyo Garcia, E. Genuardi, P. Golonka, M. Gonzalez Berges, J-C. Tournier, F. Varela, CERN, Geneva, Switzerland
M. Boccioli, Paul Scherrer Institute, Villigen, Switzerland

Abstract

The paper describes planning and execution of large-scale maintenance campaigns of SCADA systems for CERN's accelerator and technical infrastructure. These activities, required to keep up with the pace of development of the controlled systems and rapid evolution of software, are constrained by many factors, such as availability for operation and planned interventions on equipment. Experience gathered throughout the past ten years of maintenance campaigns for the SCADA Applications Service at CERN, covering over 230 systems distributed across almost 120 servers, is presented. Further improvements for the procedures and tools are proposed to adapt to the increasing number of applications in the service and reduce maintenance effort and required downtime.

INTRODUCTION

SCADA systems today have to accommodate many requirements that transcend basic functionality expected from them ten years ago. Apart from providing a view on the state of the controlled plant and archiving values of acquired signals, SCADA applications are becoming increasingly connected to other enterprise software, including ERP (Enterprise Resource Planning) and MES (Manufacturing Execution System) solutions and data analytics tools. Those trends increase the frequency at which industrial supervision systems should be updated in order to remain secure and compatible with other software and technologies.

On the other hand, the effort of preparing an upgrade of a supervision application can be high due to required testing, incurred downtime and re-commissioning needed in some cases. Consequences of errors or introducing regressions during the intervention can also be very severe. Because of that, upgrades are often only performed when there is a problem with the SCADA system that poses an immediate risk for operation. This reactive approach is not feasible in case of larger systems, where upgrades require extensive planning and coordination between multiple teams and users need to be notified well before the planned interventions.

The SCADA Applications Service (SAS) is provided by the Industrial Controls and Safety systems group at CERN. It is responsible for managing the full lifetime of SCADA applications for many equipment domains at CERN. This includes systems critical for the CERN accelerator complex and experiments, like machine protection equipment, cryogenics, cooling and ventilation, vacuum and magnet control. All applications in the SAS are based on WinCC OA [1], which is the most widely used solution for building SCADA applications at CERN. The scale of the service and the variety of systems pose many challenges, on both technical and management levels, especially considering the modest size of the team responsible for it.

Keeping applications in the SAS up-to-date is critical for their reliability and interoperability with other systems at CERN. This includes not only the versions of WinCC OA and components developed on top of it, but also the operating system and other software. Due to the number of applications in the service, it is important to use only supported versions of the software and to keep them as uniform as possible to facilitate support. Complex dependencies need to be taken into account, as many applications in the service exchange data with other systems. All those factors result in scarcity of time windows for upgrades, further aggravated by the fact that some applications are most intensively used during technical stops of the accelerator complex. Moreover, for certain critical application no downtime is accepted, which requires significant changes to the baseline procedure and further increases the workload on the team performing the upgrade.

In this paper we present the methodology used to plan and execute upgrades of the SCADA applications in the SCADA Applications Service. We start by introducing the scope of the service, maintained application domains and some most important statistics. Key steps in the evolution of handling of upgrades are described, together with modifications used for critical applications. The paper is concluded with a discussion of limitations of the current solution and planned improvements for future upgrade campaigns.

SCADA APPLICATIONS SERVICE

In an environment with many supervision applications for different plants, yet based on the same SCADA solution, it is important to minimize the duplication of work across different teams, both in software development and operations and maintenance.

JCOP [2] and UNICOS [3] are two frameworks developed at CERN, which provide a common base for WinCC OA applications. JCOP is mainly a set of programming interfaces, meant to be used by software engineers and it is widely used at the LHC experiments. UNICOS makes use of some components from JCOP and enables control engineers to create supervision applications without writing any code.

† rafal.kulaga@cern.ch

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA021

The SAS was established in order to provide equipment groups at CERN with a reliable, centralized service handling all issues related to operation of WinCC OA projects, including keeping them up to date. The service currently manages more than 230 applications, installed on around 120 servers. The domains handled by the service are listed in Table 1.

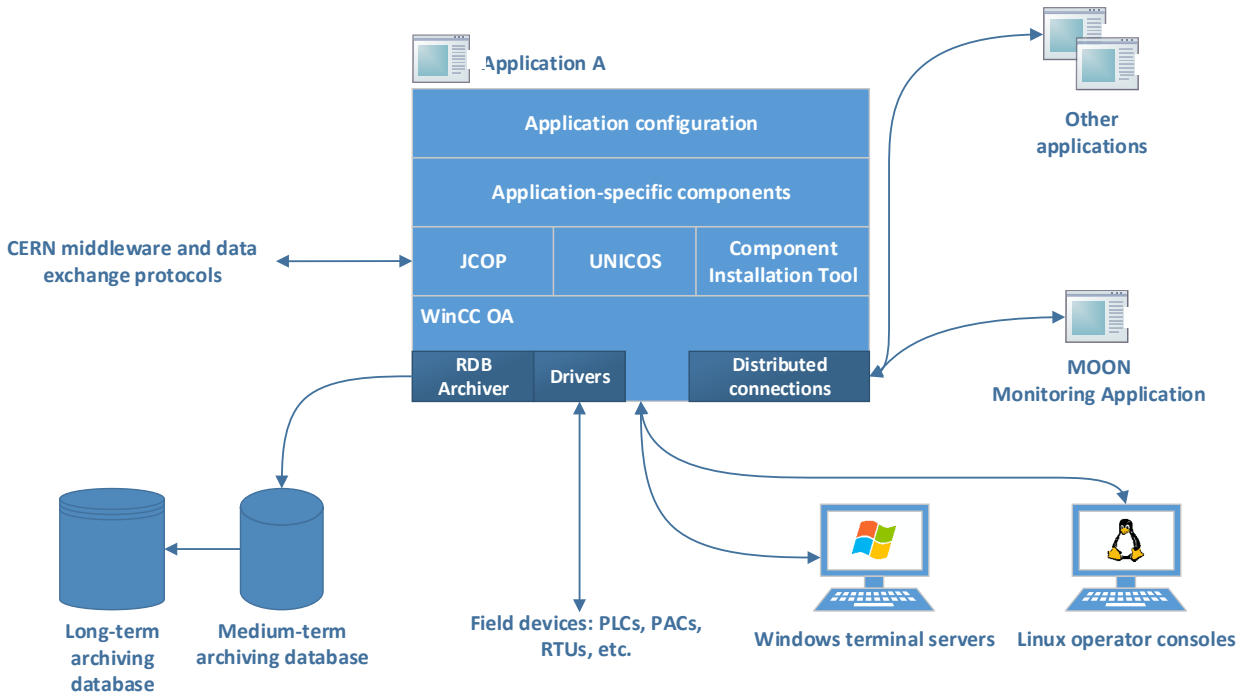

Figure 1: Architecture of a typical WinCC OA application in the SCADA Applications Service.

The architecture of a typical application in the service is presented in Figure 1. All systems in the SAS profit from running on standardized servers, both as it comes to hardware and software. The entire application lifetime is managed, from creation, through backups and upgrades, to decommissioning. Engineers from equipment groups can therefore focus on configuring the applications with devices, creating synoptic screens and developing custom extensions, while operation of applications remains the responsibility of the service.

Though the server side of the system is running on Linux machines only (CERN CentOS 7 and Scientific Linux CERN 6), users can access the applications from both Windows Terminal Servers and Linux operator consoles, which are intensively used in CERN Control Centre and other control rooms. The servers are located in the technical network, which is separated from the general purpose network for security reasons.

The state of all applications is monitored using the application MOON [4]. This includes, among others, the states of all processes, connections to field devices and archiving. Available resources on the servers are also monitored, which helps to detect abusing processes that could potentially affect their performance. Another important feature of MOON is central deployment of components on all systems connected to it.

A dedicated medium-term archiving database is available for all applications in the service, with possibility to forward selected signals to the long-term storage (LHC Logging) [5]. If additional schemas are required to store application-specific data, they can be also created. The databases used by service are provided and supported by experts in the CERN Database Group.

Table 1: Domains of Applications in the SAS with Corresponding Numbers of Servers and Applications

Domain	Applications	Servers
Machine Protection	58	20
Cooling & Ventilation	34	9
LHC Cryogenics	33	29
Experiment Cryogenics	18	15
Magnet Control	13	9
Vacuum Control	12	7
NA62 Experiment	11	3
Experiment Cooling & Ventilation	10	4
Gas Control	8	5
Power Converters	4	3
Electrical Network	1 (redundant)	2
Other	31	11
Total	233	117

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA021

PLANNING UPGRADE CAMPAIGNS

Due to the number of applications in the SCADA Applications Service, their diversity and numerous operational constraints, all large-scale upgrades need to be planned well in advance. The majority of interventions are performed during maintenance stops of the accelerator complex:

- Technical Stops (TS) are the shortest (up to 5 days) and most frequent (typically 3 times per year). For many domains and applications, finding upgrade slots in TS is hard due to interventions on hardware during which access to supervision is required.
- Year-End Technical Stops (YETS) are scheduled yearly and typically last for 2.5 months; in 2016/2017, the Extended YETS (EYETS) was 4.5 months long. During such time window, major interventions on all applications in the service can be performed, including upgrading the version of WinCC OA.
- Long Shutdowns (LS) are planned once every 3 to 4 years and are up to 2 years long. Since the start of the LHC, only one LS took place between 2013 and 2015; LS2 is planned for 2019 – 2021.

For reliable operation of the service, it is essential to align the upgrades with the release cycle of WinCC OA, as a new version is released yearly and remains supported by the company for 3 years. During the lifetime of each release, several patches are published, but their deployment is much simpler and boils down to updating RPM packages on the servers, without the need to run application upgrade tools.

Before upgrade campaigns that involve major changes, like deployment of a new version of WinCC OA, user representatives responsible for the applications are interviewed a few months in advance to check when there will be an opportunity to perform the interventions and what downtime is acceptable. Additional requirements (like installation of new framework components) are also gathered in the process.

When initial feedback from representatives of equipment groups is collected, scheduling of the campaign begins. The availability of experts for each of the applications is taken into account, as their participation is crucial in case of problems. People responsible for the applications also need to be available to perform the last verification steps and to give the final confirmation that applications function properly after the intervention.

Once a first version of the schedule is prepared, committees and working groups responsible for coordination of activities during technical stops are informed about planned activities. Representatives from the equipment groups can consult the schedule and inform the person responsible for the campaign if there are any constraints that were not captured during interviews.

To keep track of the upgrades and to facilitate reporting progress, a project tracking tool Jira [6] is used. Tasks for upgrades of all applications are automatically created before the beginning of the campaign from the schedule spreadsheet. The cases are also used by people performing upgrades to report progress. Calendar entries for upgrades of all applications are automatically generated and sent to all participants, detailing their responsibilities.

Notifications about planned upgrades and their progress are sent to all or only selected users, depending on the policy defined by the person responsible for the application. This approach is planned to be changed in the next campaign – notifications will be sent automatically to all users to reduce the workload on the person coordinating the upgrades and avoid cases in which some users remain unaware of planned downtime.

UPGRADE PROCEDURE AND TOOLS AND THEIR EVOLUTION

Despite the evolution of the methodology used for performing upgrades, its outline has remained mostly unchanged since the first campaign:

1. A few days before the upgrade, the entire filesystem of the server is backed up.
2. On the day of the upgrade, users are notified about the start of the intervention, the application is stopped and the previously created backup is updated using rsync.
3. OS on the server is updated or new OS is installed.
4. Basic upgrade prerequisites are checked, like presence of the application backup and WinCC OA license.
5. Framework and application-specific components are uninstalled from the application. Configuration stored in application's internal database is preserved.
6. The upgrade tool provided with the new version of WinCC OA is executed. The tool converts the application's internal database to conform to requirements of the new version.
7. The new versions of framework and application-specific components are installed.
8. Application integrity reported by System Integrity component is checked.
9. The upgraded application is backed up and handed over to the responsible person for final verification.
10. Depending on the result of verification, upgrade is considered successfully completed or rolled back.
11. The outcome of the upgrade is sent to the Jira case.

In the early days of the service, when it contained only around 30 applications, all steps were performed manually. A detailed upgrade procedure was prepared by the person coordinating the campaign and distributed among the people selected to perform upgrades; it usually took a full working day for one person to upgrade one application and the process was very error-prone.

As the service expanded and included more applications, key steps of the procedure were automated by WinCC OA

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA021

and Python scripts. A separate tool was developed to facilitate reporting progress.

The latest version of the upgrade tool, used during EYETS 2016/2017 campaign, integrates execution of all steps with progress reporting, lowering the time required to upgrade an application to around 20 minutes (excluding OS installation time). The upgrade progress is automatically recorded upon completion of each of the steps and sent to the Jira case. No expert knowledge of WinCC OA or the SAS production environment is required to use the tool and it can be effectively applied by one person to multiple applications simultaneously.

During EYETS 2016/2017 campaign, all upgraded machines in the SAS were migrated from Scientific Linux CERN 6 to CERN CentOS 7, which required considerably more downtime than typical OS updates (up to 90 minutes for some systems). All hard disks were repartitioned as LVM logical volumes in order to profit from the snapshotting feature to perform backups in the future. Because of the longer OS installation time, for some critical systems the OS was installed on a spare machine and the backup created in the first point of the procedure was restored to it. On the day of the upgrade, after stopping the application, the disk was placed in the production server and some final configuration adjustments were performed. The backup on the prepared disk was then updated and upgrade continued from step 4 of the procedure. This modification of the procedure resulted in total upgrade time lower than 90 minutes for some systems.

Two rollback scenarios were prepared in case of problems: rollback of an application upgrade only and a full rollback to Scientific Linux CERN 6. The former was used only once, while the latter never had to be performed thanks to very good compatibility of the previously used version of WinCC OA and the components developed on top of it with CERN CentOS 7.

HANDLING OF CRITICAL APPLICATIONS

Downtime of less than 4 hours is more than satisfactory for most applications. However, certain critical systems require changes to the baseline procedure to satisfy their strict availability constraints. In the past, those procedures had to be performed on many applications, mainly for cryogenic systems, but now their usage is limited, thanks to short downtime achieved by using the current version of the upgrade tool.

Cooling and Ventilation PLC Alarm Application

The application is acquiring alarms evaluated by PLCs in many CV installations and transferring them to the LHC Alarm SERvice (LASER) [7]. It is critical for operation and provides basic supervision of plants supervised by other applications when they are upgraded.

As no downtime is allowed for this system, its copy was started on another server on the day of the upgrade. Once this is done, there are effectively two applications retrieving the alarms from PLCs and sending them to LASER. The copy remained active while the standard upgrade procedure was performed on the main instance. After it was confirmed that it received and sent alarms correctly after migration to the new version of WinCC OA, the second instance was stopped and the upgrade was finished.

Having two copies of the application running simultaneously is possible because only polling PLC communication is used and sending the same alarm state updates twice to LASER does not cause any problems.

CERN Electrical Network Supervision (PSEN)

Upgrade of the application for CERN Electrical Network had to be performed in a way that resulted in no loss of supervision and a very short gap in archived data. PSEN is the only system in the SAS using WinCC OA redundancy mechanism. However, this does not make its upgrade easier, as it is not possible for peers in the redundant system to safely run with different versions of WinCC OA.

A completely different upgrade procedure was therefore developed. A copy of the application, running under the new version of WinCC OA, was set up on an additional pair of servers to enable extensive testing and provide a system to which users would connect during the upgrade of the main instance. It archived historical data into a test schema and was connected to all data concentrator RTUs (Remote Terminal Units), but without possibility to send controls. On the day of the upgrade, all users were disconnected from the main system and directed to the replica. The new OS was installed on the second peer of the main redundant system and the upgraded version of the application was then copied to it from the replica. After some configuration adjustments, the OS was also reinstalled on the first peer of the main system and the application was copied to it from the second peer and started. Users were then disconnected from the replica system and connected to the main instance, which concluded the upgrade.

FUTURE WORK

The key to improving all aspects of upgrades lies in reducing the required downtime and better testing of new versions of WinCC OA and component before deployment.

The most obvious benefit of lowering downtime is reduced inconvenience for users. Scheduling of interventions also becomes easier, as shorter time windows for upgrades are more frequent. Upgrade campaigns for entire application domains could take place during shorter technical stops. Even though the current version of the upgrade tool reduces the time required to perform an upgrade, somebody is still required to apply it. This increases the manpower cost of the current methodology.

A system capable of performing the upgrades on backups of applications on demand, with an option to deploy them on testing or production servers, would enable a single person to perform many upgrades per day. Availability of application experts on standby would still be required, but the entire process would become more lightweight and even less error-prone than today.

Testing new versions of the deployed software is currently largely performed manually. Certain functionalities,

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA021

like Oracle Archiver, have dedicated testing tools developed at CERN, but their coverage is far from satisfactory. They also fail at detecting issues manifesting themselves only when processes run for prolonged time, like resource leaks and memory corruption. All bugs discovered in WinCC OA and components during EYETS 2016/2017 were non-trivial: some required specific configurations of field devices, extended project runtime or were platform-dependent, which calls for an automated testing methodology capable of detecting such issues.

A test bench with real or simulated hardware, a dedicated database instance and a test setup of CERN middleware would help to find bugs before they affect production systems. Backup of any application in the SAS, including the ones upgraded by the on-demand upgrade tool described above, should be deployable in this environment without manual configuration adjustments. Depending on the setup, different scripts generating workload and simulating user actions would be started in the context of the application to sufficiently "exercise" different functionalities of WinCC OA and components over long periods of time. Resources used by all processes, as well as output written to log files should be monitored for anomalies during the tests.

One of the challenges in building the described test bench environment is being able to quickly rebuild it in a different configuration with no human intervention. So far only parts of such solution exist in the form of a Python API automating the most important project management activities and the unit testing framework for WinCC OA code.

Some fundamental assumptions about the infrastructure on which the WinCC OA applications are deployed should also be re-evaluated in the future, to make sure that OS update or reinstallation can be performed in acceptable time. The described disk swap procedure requires manual intervention of a system administrator and will likely be hindered by more heterogeneity in the infrastructure, caused by new tendering rules for servers introduced at CERN. Using virtualization would facilitate many tasks, performed not only during upgrades, including making backups and rolling back in case of problems. WinCC OA applications are currently successfully deployed on virtual machines in the LHCb experiment at CERN, which serves as a proof that such solution can be reliable enough for deployment of critical systems.

Software containers could also be useful in deployment of new versions of WinCC OA and applications. Their introduction requires less changes to the infrastructure than migrating to a fully virtualized solution. Docker is a de facto standard of software containers and is used widely at CERN, including a few successful applications in the Industrial Controls and Safety group. From the perspective of the SAS, one of the major advantages of using containers is the ability to package application as completely independent images that can be deployed on any server in the service. Their application could also help to solve some of the problems encountered when multiple applications are running on the same host through isolation and resource partitioning they provide.

CONCLUSION

The currently used methodology of upgrades, both in terms of tools and procedures, is the result of almost 10 years of evolution in order to keep up with steadily increasing number of applications in the SCADA Applications Service. It enabled three major upgrade campaigns to be carried out successfully within time and resource constraints.

To make the process even more scalable, reduce incurred downtime and required manpower, all aspects of the upgrades have to be further automated, from testing and deploying new versions of software to notifying users about the interventions.

ACKNOWLEDGEMENT

We thank Axel Voitier and Lyuba Petrova for their important contributions to the development of the tools described in this paper. The upgrades and reliable operation of the SCADA Applications Service would also not be possible without the crucial support of system administrators from the BE-CO-IN section and database experts from the IT-DB group. We would also like to express our gratitude to the members of the BE-ICS-SDS section who performed bulk of the work described here.

REFERENCES

[1] Simatic WinCC Open Architecture (previously PVSS) SCADA Software from ETM (Siemens subsidiary), http://www.etm.at

[2] JCOP Framework, http://jcop.web.cern.ch

[3] UNICOS Framework, http://unicos.web.cern.ch

[4] F.Bernard, M. Gonzalez, H. Milcent, L.B. Petrova, F. Varela, "Monitoring Control Applications at CERN", Conference Proceedings of ICALEPCS2011, Grenoble, France.

[5] P. Golonka, M. Gonzalez Berges, J. Hofer, A. Voitier, "Database Archiving System for Supervision Systems at CERN: A Successful Upgrade Story", Conference Proceedings ICALEPCS2015, Melbourne, Australia.

[6] Jira issue and project tracking software from Atlassian, https://www.atlassian.com/software/jira

[7] F. Calderini, B. Pawlowski, N. Stapley, M.W. Tyrrell, "Moving Towards a Common Alarm Service for the LHC Era", Conference Proceedings of ICALEPCS2003, Gyeongju, Korea.