DETECTOR AND RUN CONTROL SYSTEMS FOR THE NA62 FIXED-TARGET EXPERIMENT AT CERN

Piotr Golonka, CERN, Geneva, Switzerland
Riccardo Fantechi, CERN, Geneva, Switzerland
Manuel Gonzalez-Berges, CERN, Geneva, Switzerland
Fernando Varela, CERN, Geneva, Switzerland
Valeri Falaleev, JINR, Dubna, Russia
Nicolas Lurkin, University of Birmingham, Birmingham, UK
Ryan Frank Page, University of Bristol, Bristol, UK

Abstract

The Detector and Run Control systems for the NA62 experiment, which started physics data-taking in the autumn of 2014, were designed, developed and deployed in collaboration between the Physics and Engineering Departments at CERN. Based on the commonly used control frameworks, they were developed with scarce manpower while meeting the challenge of extreme agility, evolving requirements, as well as integration of new types of hardware. The paper presents, for the first time, the architecture of these systems and discusses the challenges and experience in developing and maintaining them during the first months of operation.

INTRODUCTION

NA62 is a fixed-target experiment at CERN aiming at high-precision measurements of rare Kaon decays, currently taking physics data in its second run. Operation of the detector and its infrastructure is supervised with dedicated control systems, the most complex being the Detector Control System (DCS) and Run Control (RC). The two are similar in many aspects and share common infrastructure, as well as development and maintenance workflows. The Run Control has already been introduced in [1,2]. This paper focuses on the - often neglected - topic of complete architecture, including infrastructure, project organization and development life cycle.

The NA62 detector is composed of a set of sub-detectors developed and maintained by semi-independent groups. Some equipment is reused from previous experiments (such as the Liquid Krypton Calorimeter), while others must be built anew. Certain sub-detectors or their parts were not ready for integration before the first run, and they are added progressively In consequence, the development, commissioning and maintenance of control systems is a *process* that spans over years. This is unlike many turn-key control systems in industry. Assuring consistency and long term maintenance in the project, taking into account a high-turnover of people in experimental collaboration is one of the primary challenges.

REQUIREMENTS

The DCS and RC systems for NA62 resemble those of other CERN experiments. Relative to the LHC experiments they are an order of magnitude lower in number of devices, yet their complexity remains similar.

The systems have to allow for operation and monitoring of a variety of devices, react on value changes, evaluate alarm conditions and notify the operators about the alarms. The operators need to be presented with an alarm screen, as well as synoptic views that summarize the state of parts and subsystems of the experiment, and be able to drill down and quickly identify or locate the required device. They need to easily command individual devices, as well as groups of them (e.g. complete subdetectors)

The experts need to tune the set-points and alarm threshold, and bring up the trend plots of historical values. Reconfiguration should be applied to a selected device or using configuration sets (*recipes*) reflecting the run modes of the experiment. The experts must be able to detect and diagnose malfunctions and communication problems, and dynamically reconfigure the system, e.g. re-wire faulty sensors or mask alarms.

The history of acquired data (*conditions*) needs to be recorded to an Oracle database to be used in physics analysis. CERN computer security rules need to be applied. Robust infrastructure and hardware should be employed to guarantee round-the-clock operation during 8 months of data taking with minimal human supervision.

OPERATION

The independent operation of the DCS and RC is performed on dedicated consoles in the NA62 Control Room. The User Interface window requires authentication with CERN credentials, and enables access to shifter or expert operations, on various parts of detector, depending on privileges defined by administrators.

Standard shifter operation employs two tools: the alarm screen, displaying a time-sorted list of anomalous incidents ("alarms") and the hierarchical control supervision screen. The latter, also known as "FSM" (Finite State machine), abstracts out the hardware to present a tree view of the detector’s parts and subsystems. Hardware states are evaluated and represented by colour-codes and text labels ("*ERROR*", "*RAMPING*"). These states are propagated in the upstream direction of the tree using summarization logic. This allows for rapid root-cause identification by expanding the coloured tree view. High level commands ("SWITCH_ON", "CONFIGURE") are broadcast down the tree allowing control of large parts

ISBN 978-3-95450-148-9

of the hierarchy with single button-presses. It is possible to exclude devices and sub-systems from central hierarchical operation and hand them over to an expert in a standalone or shared mode of operation. Whereas the shifters may only see the current values for set points, the experts can also modify them, as well as mask the alarms or change their thresholds. The principles of hierarchical operation, and the employed tools, are identical to those used by the LHC experiments.

Remote access and expert operation is also possible through dedicated Terminal Servers and secondary consoles which run the same "User Interface" application as the main control room console. Conflicting operations from two operators are avoided thanks to the concept of partitioned operation and sub-tree ownership.

PROJECT STRATEGY

Following the good experience with the LHC Detector Control Systems, it was proposed to develop and maintain the NA62 controls projects by a Central Team with assistance of experts from the CERN EN/ICE group. The Central Team, recruited from NA62 members, would work closely with sub-detector experts to implement their requirements; the experts would provide the necessary expertise, training as well as technical supervision of the projects. To maximize the use of scarce development manpower allocated for the projects, main developer(s) would work in proximity and under direct supervision of EN/ICE experts especially at the initial stage of project development.

A general policy was adopted to apply standard, centrally supported and recommended technologies and make use of existing services available at CERN. In addition, in order to enhance the stability of the running system, simplify long-term support and assure evolution of the project, guidelines and a development process with elements of quality-assurance and testing were agreed on.

ARCHITECTURE AND TECHNOLOGIES

The systems are developed with standard CERN set of technologies, and a layered architecture. Figure 1 presents the geographically scattered layout of the system, with the Supervision, Process/Frontend and Field/Equipment parts.

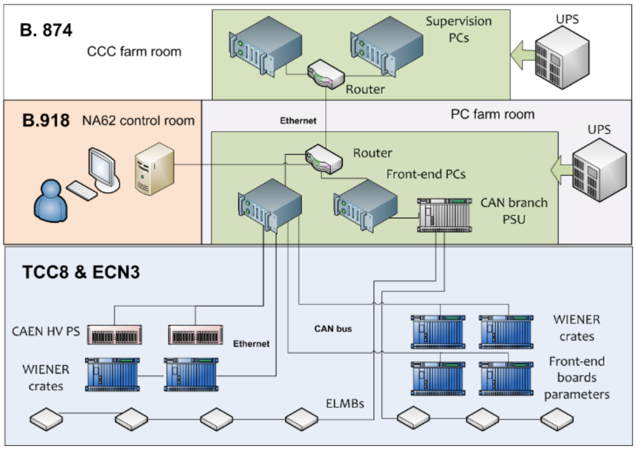

Figure 1: Architecture of the NA62 Control Systems with equipment location.

In the Supervision Layer there is a set of 10 supervision applications: one for the Run Control, one per sub-detector (or a set of smaller ones), and one Central DCS application that orchestrates the execution of sub-detector applications. The cores of SCADA applications are running on dedicated control servers (located in the server room which hosts other CERN control systems) and the User Interfaces (UIs) programs running on a set of consoles located in the the control room. The server room and the experiment facilities are connected by a direct optical fiber link.

The Frontend layer is composed of dedicated computers, hosting hardware-interface cards (for CAN bus). They aggregate the traffic from hardware, translate the necessary middleware technologies eliminating platform-dependent factors and interface with the Supervision layer over a dedicated experiment network. A few PLCs are used to interface with relays and sensors.

The Field and Equipment layer is populated by the high and low voltage systems (over 4000 channels), electronics crates, and I/O boards with sensors. A significant number of custom frontend-electronics boards are configured and monitored.

Technologies

The supervision layer is built with the standard CERN controls software stack based on the commercial *WinCC Open Architecture* SCADA [3], and the *JCOP* and *UNICOS* frameworks [4,5,6].

Communication with the hardware layer is mediated by CERN's standard DIP, DIM [7] and OPC-DA middleware protocols; Ethernet and CAN bus media are used to reach the field layer.

Standard hardware used by other CERN Detector Control Systems integrated within the JCOP Framework [4] are used for high voltage, electronic crates and analog and digital I/O.

The readout for the unique, legacy Liquid Krypton Calorimeter is interfaced through a genuine solution based on the mixture of control frameworks and technologies. A PLC powered by UNICOS is used to control relays, ELMB I/O boards integrated through JCOP Framework are used for voltage readouts, while the electronics monitoring is provided by a dedicated low-level software interfaced through DIM.

A particularly challenging task was to interface the control for the thresholds frontend electronic boards. Initially interfaced using the CanOpen technology (used also by ELMB boards), they suffered from stability and performance issues in large setups over long CAN bus lines for the LAV detector. Efforts by the developers of the custom firmware as well as low-level protocol debugging brought only a moderate improvement. An alternative control channel was then exploited: a set of RaspberryPi mini computers connected over USB to the frontend boards was programmed to expose a control and monitoring interface over DIM, which allowed for integration into the Run Control. DIM was also used to interface non-standard high- and low-voltage system for the STRAW subdetector.

A notable achievement is the part of DCS for the KTAG subdetector that performs a "pressure scan": it commands the NA62 Gas Control System, through the DIP protocol (uni-directional version of DIM) to set the gas parameters in the subdetector as necessary, and then automates the task of sequencing the parameters which otherwise would need to be executed manually by the operator using the Gas Control System console.

Infrastructure and Services

To reduce the maintenance effort, NA62 control systems share infrastructure and use available CERN services. Even though additional coordination is needed, technical expertise of others, for a wide range of technologies, is employed.

Numerous services provided by the CERN IT Department were employed to construct the infrastructure. A dedicated NA62 experimental network was set up, isolating the traffic for controls and data acquisition from the CERN General Purpose Network. Inter-network communication (such as the Technical Network or GPN) was configured to make the CERN central services (disk servers, domain controllers, DNS, DHCP or databases) available as necessary while securing the access to the experiment's equipment. Remote access to this internal network is provided by the Windows Terminal Server service, run on the CERN virtualization infrastructure, fully maintained by the service providers. The same virtualization infrastructure is used to form the integration test setup. The Oracle database service provides performant and highly-scalable storage for historical process data as well as for configurations. All the services come with the possibility of adaptation or upscaling and offer test and validation setups as well as expert's help.

Infrastructure maintenance for the supervision layer (hardware, operating system software installations, security, monitoring and backups) is in the hands of the experts from the CERN BE/CO group. The NA62 servers and consoles are maintained together with around 200 other production control servers and over 500 operator consoles by a team of experts.

The software and hardware for the Frontend Layer are specific to the equipment it controls, and hence their maintenance need to be under the responsibility of NA62 experts. Nevertheless, the computers were purchased with the help of the BE and IT Departments, to profit from established contracts for the long-term availability of spare parts and on-site support. Support for the PLCs, as well as the WinCC OA service, middleware software and standard protocols (OPC Server software) are provided by the EN department and guarded by maintenance and support contracts between CERN and hardware vendors.

The frontend computers, consoles and data servers are powered through UPS units providing up to an hour of autonomy in case of a power-cut.

SYSTEM DEVELOPMENT LIFE CYCLE

Control systems for physics experiments evolve constantly: subsystems are upgraded, replaced or added, hardware and software faults need to be resolved, infrastructures change, new versions of software packages installed. To optimize new operation modes new features are requested throughout the lifetime of the experiment. On the other hand, uncertainties in hardware prototyping often lead to late delivery of hardware and drastic changes in specifications and requirements. Requirements for operation may only be specified on very high level, and they evolve according to the phase of the experiment and operation experience. Planning and development of control systems, in particular with highly constrained manpower became complex, and the classical system development life cycle (SDLC) [8] workflows were therefore not applicable to NA62 controls projects. The boundaries between the classical planning, analysis, design and implementation phases are blurred and overlap. However, we tried the best practices from these models and came up with a method that tries to match the resources, constraints and requirements.

At the heart of the methodology is the requirement of maintaining a running system, while being able to constantly evolve and enhance it and fix issues. For that, we apply a policy of packaging all development into so called components, which are released with version numbers - the concept underpinning the JCOP Framework [4]. Engineering tasks are done on development machines, and changes are saved in the Subversion software repository. Then new features sets and bug fixes are released as a new version of the component, and installed in the integration setup to perform acceptance tests. Finally, they may be applied to a production system, with a possibility of a rollback.

This workflow goes against the need for rapid debugging and bug-fixing directly on the production system, often desired for operation. The changes prototyped in production need to be transferred to the code repository and go through the whole component release and acceptance process. While working under high time pressure, it is tempting to abandon the workflow, and proceed with development only on the production system. This results in software regression bugs whenever a new version is deployed. Potential time saving needs to be over-compensated by increased efforts in debugging.

The timing and phases of the development cycle are driven by the physics data-taking schedule. The system is only partially used during experiment shutdown time (winter/spring) which makes it the perfect time for scheduling development and maintenance tasks. Prior to the startup of the run, new requirements and new tasks for integration typically start to appear, as new hardware is becoming available. During the data-taking, feedback is given by operators and experts, who request new ad-hoc features to be necessarily added.

ISBN 978-3-95450-148-9

EXPERIENCE

The operation of the subsystems that are well advanced in their development, notably the Run Control and DCS for KTAG, GTK and LKr subdetectors was smooth. The availability of the infrastructure and services was very high, with smooth interventions and good coordination. Integrated operation of the detector through the Central DCS (rather than ad-hoc hardware control) has become regular, despite minor instabilities and rough edges. The priority areas where development efforts need to be intensified are the homogeneity and consistency of the hierarchical control (FSM tree) and consistent use of recipes stored in a configuration database. Experts demand more diagnostic tools for hardware communication and system integrity problems. Integrated with the supervision layer these should allow for better feed-back to operators. Overall stability of the, sometimes unstable, frontend software needs to be improved.

The component-based development method and integration tests worked well, even though it required significant initial effort from developers. Not only did it facilitate the integration of developments without affecting system stability, but also allowed for smooth migration to a newer version of SCADA software. Similarly, it makes the complete reinstallation of the system (in case of unlikely failure of hardware or infrastructure) possible within a short time.

A layer of NA62 common software components (e.g. widget libraries) allowed for new features to be easily deployed across all parts of the system. The use of standard hardware components supported by the JCOP Framework allowed for the rapid integration of significant parts of the system. For non-standard hardware, the use of DIM proved to be particularly effective, where applicable.

The development of certain systems benefited from having hardware that was at least partially installed before the physics run. This allowed the development chain to be completed in full before deploying on the production system and having the system operational in 2014.

We noticed that despite the fact that the core features and tools were planned and developed during the inter-run period, a significant amount of work was required during data taking. Many of the sub-system specific expert and diagnostic tools were requested as the need arose with the availability of new custom hardware. These needed to be implemented quickly to pursue the commissioning of the experiment.

Additional burden on development was induced by the non-standard, undocumented behaviour of hardware. Debugging and integration-testing needed to be interleaved with the development. Software that was initially modelled according to official documentation needed to be adapted (e.g. state transitions for the HV system).

We also stress the importance of the user requirement formulation process. Negligence, inaccuracies and "last-minute" changes led to important increase in development efforts or even the complete redesign of certain subsystems. Combined with lack of developers, this delayed the delivery and commissioning tasks.

The complexity of the systems makes the learning curve for new contributors very steep and requires an education phase of a few months. A number of developers were trained only to leave the project shortly after they delivered initial (often not yet complete) results. In effect, it was not possible until now to put in place the collaboration model of the strong central team made of experienced users.

Drawing proper balance between (often very ambitious) requirements and available developer resources was among the main challenges in the project management.

CONCLUSION

We presented the architecture, project management and development life cycle for the NA62 control systems. By maximizing the reuse of existing components, applying standard solutions and best practices and using the services made available by CERN groups we were able to deliver working control systems, and perform their maintenance with minimal effort and very scarce resources. The need to enhance and maintain control systems throughout their life time was stressed by gathered experience. Constant effort and resources need to be assured in order to maintain the robustness of the system and to implement requests coming from the operators. High agility in project management is required to accomplish this task.

REFERENCES

[1] F. Varela, *et al,* "Reusing the Knowledge from the LHC Experiments to Implement the NA62 Run Control", ICALEPCS 2013, San Francisco, USA

[2] N. Lurkin, "The NA62 Run Control", proceedings NSS/MIC Conference IEEE 2013, Seoul, Korea

[3] SIMATIC WinCC Open Architecture (previously PVSS) SCADA software from ETM (Siemens subsidiary), http://www.etm.at

[4] M. Gonzalez-Berges *et al*, "The Joint COntrols Project Framework", CHEP 2003, La Jolla, USA

[5] H. Milcent *et al*, "UNICOS: An open framework", ICALEPCS 2009, Kobe, Japan, 2009

[6] J. Arroyo Garcia *et al*, "Integrating Controls Frameworks: Control System for NA62 LAV Detector Test Beams", ICALEPCS 2011, Grenoble, France

[7] C. Gaspar *et al*, "DIM, a portable, light weight package for information publishing, data transfer and inter-process communication", Computer Physics Communications 140 (2001) 102-109

[8] "System Development Life Cycle" wikipedia article, https://en.wikipedia.org/wiki/Systems_development_life_cycle