ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR1BCO01

STATUS OF THE EUROPEAN SPALLATION SOURCE CONTROLS

T. Korhonen[1], European Spallation Source ERIC, Lund, Sweden
[1]Representing the Integrated Control Systems Division

Abstract

The European Spallation Source is progressing towards completion in a few years. The control system has been through first commissioning rounds and is now in production use. While development is still going on to reach full functionality, most of the central supporting features are already in place. This paper gives an overview of the current status, the principles we have been following on the way, our use and experience with the central technologies like MTCA.4 and EPICS 7, plus an overview of the next steps. We also look at what was planned and reported in ICALEPCS 2015 and how our system of today compares with ideas of that time, and the evolution from green field project to an operating organization.

INTRODUCTION

ESS is one of the very few true green-field projects where not only the facility but also the surrounding organisation has been built from scratch. With the project, the control system has gone through a series of growing pains. This paper will touch some of them, and how have we managed (or not) to resolve the issues that have come up. Also, we look at the overarching goals that we have tried to follow and at core principles that support these goals.

Obviously, the expectation of a new project is that the system is built to take advantage of the technological advances that are available. But this is only momentary and the real goal is to build an infrastructure that can be maintained with reasonable amount of effort, and in a way that allows not only updates but also evolution to implement features that are not known at the time of development. Experience has taught that this is a common occurrence in real life, and also defines the ultimate success of a scientific facility.

Cost efficiency is also an important goal but should be considered with the whole system lifecycle in mind. Of course, project budget and time are limited, but within these limits there are several possible ways to follow.

From these considerations the following principles emerge:

- Strive for clear standards to be followed. Standards are not meant to be set in stone forever, but to create clarity and enable collaboration between units that are often remote and disconnected.
- Support the development process so that skills of people can be optimally utilized
- Try to remove barriers towards updates so that it is easy to keep systems up to date.

Let us look at what has been done to achieve these goals.

STANDARDIZATION

When embarking on a long project, one has to define the technology base such that it covers the project needs as well as possible. Also, one has to pay a lot of attention to the evolution capability of technologies. Controls systems are work-intense, and the work continues long after all hardware has been installed. In a sense, the work goes on as long as the facility is in use.

Hardware Platforms

One of the earliest decisions to be made was the main hardware platforms to be used for field I/O and control.

To be useful, the selected standards should cover the majority of the expected use cases. We ended up with a three-layer strategy which has been able to cover a vast majority of use cases we have encountered so far (see Fig. 1).

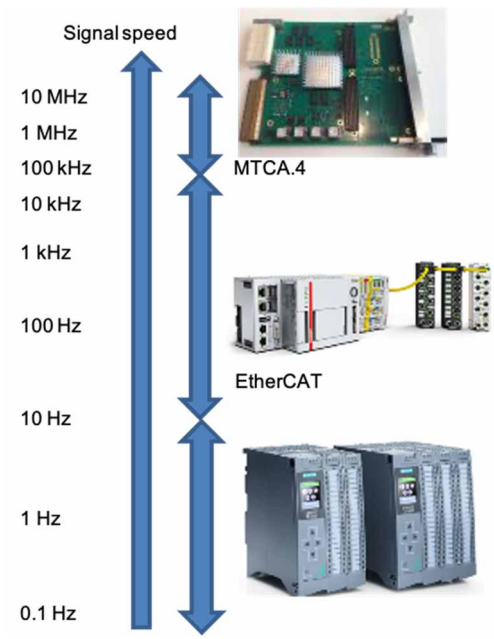

Figure 1. Hardware standards and performance levels.

Hardware costs are directly visible, unlike software and that leads often to closer scrutiny than for software projects. There is also the temptation to lower the upfront cost instead of considering the lifecycle of the decisions.

MTCA.4

At the point of ESS decision, MTCA [1] could still be considered as an emerging standard. Pioneering work had been done at DESY to develop the MTCA.4 standard to a useable level, by investing time in creating the standard, but also developing products together with industrial partners.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR1BCO01

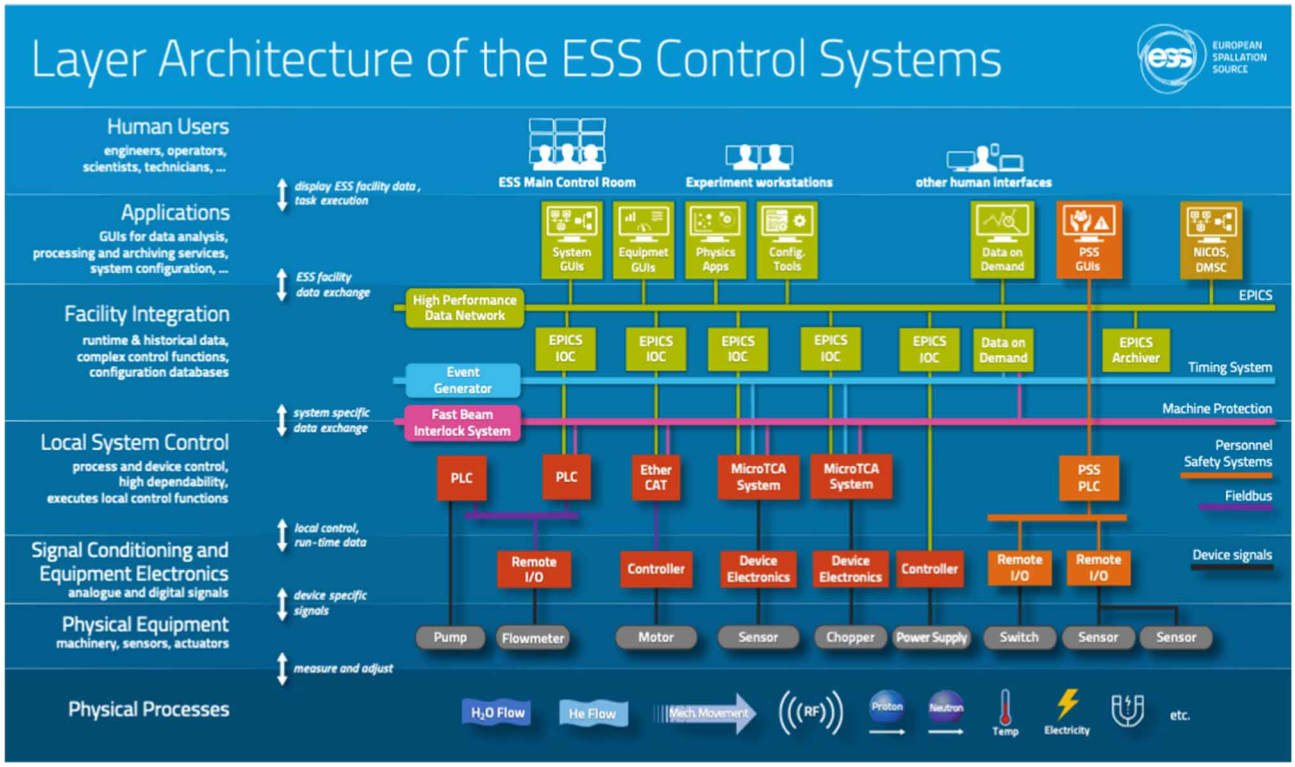

Figure 2: Architectural view of the ESS control system.

We at ESS do not have the level of in-house development resources as DESY has, so we have concentrated on adopting the standard and trying to encourage companies to invest in development of the standard. However, a lot of work has been done for creating deployment procedures to ensure consistency and maintainability [2].

MTCA.4 has been proven to work well and be stable in applications. It has been criticized for the high upfront cost, but this is offset by the modularity which allows us do to incremental upgrades and improvements during the long lifetime of the facility, as well as giving an upgrade path with the evolution of the standard. There is a new version of the MTCA base specification underway, lining up the upgrade path of the standard.

On the other hand, the MTCA is still suffering from interoperability issues: there are a couple of major "camps" and once one decides to join one camp, it is difficult to use products from the others. This is an issue for the larger community, even if a single project can cover its needs.

EtherCAT

EtherCAT [3] is a standard that has been developed by the company Beckhoff, for industrial applications. EtherCAT is based on Ethernet technology but extends it in a way that allows for high performance and deterministic responses. This makes it especially attractive for applications like distributed motion control, as well as cases when synchronisation with external components is needed. EtherCAT provides a good performance at a reasonable cost and was a natural choice to cover the range of needs where MTCA is too cost- and resource intensive.

We at ESS are using EtherCAT mainly for motion control, where the high precision synchronisation is a big advantage for coordinated multi-axis motion and in particular for on-the-fly scans which will be important for the neutron experiments.

We have so far followed two threads of development, one based on having a Beckhoff PLC as the host, with the TwinCAT [4] software, and an open-source custom-developed application module called ecmc, that covers motion applications but general I/O as well [5]. Both approaches have their respective advantages and we plan to be following both approaches at least in the foreseeable future.

PLCs

PLCs are the choice for applications where relatively slow processes need to be controlled with high reliability. These include for instance cryogenics, vacuum control, cooling and many protection applications.

The risk was that different vendors and in-kind partners would select different products for the same tasks, so this was one of the standards we tried to define very early in the process. After a tendering process a decision was reached and the selection was documented and, in many cases, also directly provided for the partners.

We have paid attention to the consistent and smooth integration of the PLC applications to EPICS, as described in [6]. The core idea has been to couple together the PLC program and the EPICS configuration to keep the interface is consistent when the PLC program changes. The big advantage of this approach is that the PLC-based applications have all a similar interface and it is easy for programmers to get familiar with.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR1BCO01

Another solution based on OPC-UA [7] has emerged after we had already developed and deployed several systems. However, OPC-UA is a very good solution for many use cases and we are using it for several projects, in particular for integrating protection systems to EPICS, and also for conventional facilities integration. One particularly attractive feature is that it allows discovery of the PLC tags and thus decouples the PLC development from the subsequent EPICS integration. This is excellent in cases where the PLC software is developed outside ESS and the integration happens at a later stage.

In general, the success in standardizing the hardware platforms has been rather successful; there are only a few exceptions where something else than the standard has been used, with justification but also with a risk of landing into unmaintained state.

In particular we are happy with how well our numerous in-kind partners have adhered to the standards. The worries of divergence early in the project did not materialize and there were only marginal deviations which are all on manageable level.

I/O Control – EPICS IOCs

The decision to use EPICS [8] as the control system software package across the whole facility was one of the early decisions. However, that decision only solves a small part of standardization issues. EPICS Input-Output Controllers, commonly called IOCs, can be developed in countless number of ways.

With our mix of hardware and different capabilities, one has to decide where to put the business logic. As shown in in Fig. 2, EPICS resides on the facility integration layer. Following this principle, we have in general followed the rule of keeping functionality at the lowest feasible level, and this leads to the practice of implementing system-specific functionality mostly on the PLC and FPGA layers and communication between different subsystems on the EPICS layer.

So far, we have succeeded to run all our IOCs on Linux, which makes management a lot simpler.

DEVELOPMENT

During the past years, the ICS division needed to ramp up from a small group to a major division that can support diverse use cases and groups. To accommodate this change, we have tried to follow a number of principles that make it possible to develop so that the results will be sustainable.

It is very difficult to find and hire experienced EPICS engineers on the scale that was necessary for us. This, as well as experience from earlier projects led to the desire to reduce the need of detail knowledge about EPICS internals during device integration, in other words to separate the concerns of integration and software development as much as possible.

One of our early goals was to build the system to take advantage of the emerging capability in EPICS to handle structured data. What was known then as EPICS 4 evolved into EPICS 7 [9] and the features have been developed gradually. To be able to make comprehensive use of the new features, we needed to keep the barrier to upgrades as low as possible. If we stagnate and cannot upgrade systems when the implementation is still incomplete, we would end up with an unmaintainable patchwork and cannot utilize the new features properly. This has led to the goal of "mobility", namely that systems can be easily updated with new developments. The overall system can only be kept up to recent standards if the updates are done in small but frequent steps instead of major upheavals that by nature happen only when the need becomes too pressing.

EPICS at ESS

A couple of developments have helped us to adhere to the mobility goal mentioned above. First, extensive use of virtualization made it possible to decouple the IOC from the hardware in the majority of cases. This helped us to partition the functionality in units that are easier to manage and can be developed in parallel and less independently of each other. In addition, we followed a model from PSI where instead of compiling each IOC into a standalone application, the IOC is configured at start-up time from a number of pre-compiled modules that are dynamically linked together. This approach enabled us to concentrate the management of the EPICS core software to a fairly small team that provides the modules for use by the integrators, who do not have to worry about how the modules are built and how they need to be put together. Quality control, continuous integration and provision of the system is taken care of a modest-sized group of engineers who can concentrate on this task. The integrators who have to develop the actual control functionality can concentrate on working with the devices and have less to do with intricacies of the EPICS setup and composition.

This environment called e3 [10] greatly reduces the steps that are needed for setting up an EPICS IOC. In fact, an individual IOC consists only of a couple of files; an environment definition file and a startup script. The environment file defines the EPICS base version and a few other environment variables, the startup script defines the modules need to be load, how to connect to the I/O, what EPICS records it will provide and how those records are configured.

In contrast, using the "classic" EPICS development model, the integrator has to collect all the modules, understand their dependencies, write a Makefile and finally compile the source code together – and then start writing the startup scripts.

To be able to take advantage of the new emerging features of EPICS version 7, it has been crucial to us to be active members in the community. We have along the way funded developments in the community and also been building up our internal know-how, by participating in collaboration meetings and development events called "Codeathons" and "Documentathons", organized by the community, as well as gradually increasing our participation in the EPICS core development. This has been important in both helping the EPICS 7 to reach maturity for operation as well as securing the internal support and capability for future developments.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR1BCO01

Naturally, this development model has its pros and cons. Pros are mainly already listed above: enabling people to concentrate on the actual task at hand and making the applications easier to upgrade. As cons, the EPICS maintenance team is under a lot of pressure dealing with a myriad of modules and dependencies; a lot of thinking needs to go into how modules are composed and how do they relate to each other. Achieving consistency is not trivial.

However, I think we can say that effort has paid off and while there is still work to do, our development environment has proven to match the challenges that we have faced.

Client Applications

Equally important to the device control and integration layers are the client applications that will be used to operate the facility. The client applications need to be prepared to work with the new protocol in EPICS 7 as well, and the best way to guarantee this – apart from being a good idea in general – was to be involved in the development of the client tools. However, we originally did neither have the sufficient manpower (or desire) to develop our own tools, nor did we have the required knowledge. So, the decision was to get involved in the Control System Studio (CS-Studio) [11] collaboration. We have in between become one of the central contributors to that infrastructure. During the years we have moved from the Eclipse-based CS-Studio to the recent Phoebus, still the same concept but fully revised implementation, and were one of the early adopters of the new code base.

One of the challenges that we have been facing, more organisational than technical is the "thin client" principle that most EPICS tools follow. In this model, the business logic is mainly placed in the IOCs so that it is uniformly available for all clients. However, many users are accustomed to programming things in the user interface layer, or are unaware of the capabilities that EPICS can provide. The original intention was that applications that need post-processing that is not suitable for a thin client application would be implemented in either dedicated compiled applications or as Python scripts. We are still searching for the optimal balance between these approaches.

In the same spirit, we also have been increasing our involvement in other central tools like the Archiver Appliance (AA) [12]. In our environment, Archiver Appliance has been used in ways that it was not originally foreseen, for example handling large arrays of data. Proper configuration and capability planning are needed for AA to operate as expected, and for that, in-depth knowledge is valuable. Also, we would like to extend the capabilities of AA towards handling the data structures (aka Normative Types) that EPICS 7 provides. Work towards that goal is in progress.

MOVING FROM DEVELOPMENT TO PRODUCTION

When systems, hardware and software have been developed, they have to be deployed in the production environment to be used by the subsystem developers and machine operators. Different classes of systems face different issues.

IOC Deployment

The fundamental building blocks of our control system are the EPICS IOCs. Once an EPICS IOC has been developed and gone through testing, it has to be deployed to the production environment. To do it in a way that is uniform and sustainable, a deployment service – in reality a suite of services – has been developed (reports and presentations in preparation.) This service guarantees that

- All I/O systems are deployed in a consistent and well-known way to host computers that are properly managed.
- All central services are set up in the proper way.
- Users have an easy access to the things that the central services offer, for instance easy access to logging information (IOC message log, "caPutlog" that logs write actions, etc.)

CE-deploy is a suite of services that run behind a web application. The user interface of ce-deploy is rather simple to make it easy to use and remember. When a deployment task is launched, a background application collects the necessary pieces of information from different sources, and based on an Ansible playbook applies a number of standard actions to the host machine to install the IOC as a service process that uses Linux systemd for management.

The deployment service removes the need for the IOC engineer to manipulate the host setup, gives a unified maintenance interface and allows us to collect the statistics and bind together the various services in the infrastructure. Prior to availability of the deployment service, the installations were done in many different ways and there was no oversight about what had been installed and the management was totally dependent on the sole integrator who deployed the IOC. Now, only a minor number of custom deployments are left and they are on their way to use the standard deployment.

User Interfaces

One of our ambitions is to have a consistent look and feel for our user interfaces, also known as OPIs. Due to the close coupling with the EPICS layer, OPIs are usually initially developed by the integrators and then deployed for use in the control room and by system users and experts. For that, we have provided help in form of consultation and reviews by a user interface expert. We also provide a style guide for OPI developers.

However, OPIs are often best developed by the people who actually use them. For this reason, we have been giving the tools to the operations group who are also engaged in the OPI development, one example of the results is reported in [13]. This is a win-win for both sides; the end users can develop and adjust the OPIs and applications to best match the day-to-day use, and integrators have less to worry about. In some cases, the developments have led to sophisticated applications beyond what has been originally foreseen [14].

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR1BCO01

In all cases, good communication between the control system and OPI developers is crucial to avoid mistakes.

Managing Configurations

As the facility grows, more and more subsystems, often similar to each other have to be deployed. Configuring tens or hundreds of similar IOCs by using hand-written text files is not going to be feasible. In addition, this way the configuration can only be managed by the expert user. Earlier in the development we were planning to use a suite of SQL-based tools, called CCDB (for Controls Configuration Database) [15].

CCDB development had started early in the project, and during the progress many surrounding things changed. The original idea of serving as an inventory became obsolete as other ESS-wide tools were introduced. This and many other changes resulted in most of the foreseen features being obsolete. While the system served us well for a long time, finally it did not match the development process and user expectations anymore, so finally it was decided to deprecate it, pick up the essential features that are still relevant and desired, and embark on a new development. The new development is based on extensive use of templates which is a more familiar model for most of our developers. The templating service, while still under development, has already gained widespread adoption and is used to manage deployment of systems, especially cases where the same functionality is instantiated numerous times; RF systems are a good example.

One of the final goals is again separation of concerns: several systems require configuration that is out of the work scope of the original integrator. Examples could be limits for beam optics settings that come from physics calculations, or beam loss thresholds along the accelerator. These should be accessible to the relevant expert without having to know the details of the low-level implementation. Work towards that goal is in progress. With the new templating service, we expect to fulfil the original goals, but be better adopted to the changed surrounding situation.

PROGRESS WITH SUBSYSTEMS

Accelerator

In the spring 2023 commissioning period, and partly even earlier, most types of subsystems of accelerator equipment have been taken into operation, albeit obviously with lower than the final quantities. A notable major exception are the superconducting cavities and cryomodules; however, even they are being successively tested and commissioned as single units, without beam but otherwise in full configuration. Full integration and commissioning will follow when they are installed in their final locations.

One major contribution of the ICS division is the machine protection system (MPS) that has been in operation during the spring commissioning campaign. Results from that period, along with several other aspects are reported in [16]. Design and implementation as well as operational experience is reported in [17].

As anticipated, we have encountered issues with handling the big data sizes that some of the components produce, mainly RF and beam instrumentation systems. Long 2.86 ms pulses mean that there is beam in the entire accelerator. A single PV, while necessary, will not characterize the pulse sufficiently. Further work is needed to find out best ways to manage the data volumes so that the network and storage infrastructure can handle it, user displays show intelligible data that is useful and data with sufficient resolution is available for off-line analysis.

Target

Mainly process-control type of equipment, using PLCs as the low-level control and EPICS as the interface layer, and as the construction proceeds, layer for integrating multiple systems. Effort is underway to define the details of how the target will be operated as a single unit.

The central point of the ESS target is the rotating target wheel. The rotation has to be synchronised with the accelerator operation, using the timing system. This has been planned since the beginning, but some operational aspects remain still to be verified before the target can be operated. One of these is the verification of the method to communicate between the wheel control and the master timing.

Neutron Instruments

The principles and technologies to be used in the neutron instruments have been developed in the YMIR test facility. Timing integration is one of the central components and methods have been already verified in tests at partner institutes. Integration of different methods for different needs, especially in motion control is reported in [18]. Substantial progress has been made in several other areas as well, including neutron choppers and detector development.

Timing

ESS timing system provides timing and synchronisation for the whole facility. It has been in operation for a while already, and has performed according to expectations. We use the 300-series on MTCA as well as on PCI express cards that can be installed in e.g., industrial PCs. We make use of the new features in that series, most notably the delay compensation that can be used to eliminate the effect of temperature variations in long distance optical cables, as well as the dual-purpose EVM that can be used as an event generator as well as a fanout and concentrator device. Timing system has been reported in previous conferences, for example in [19]. Enhancements are being done to the integration in machine operation [20], as well as integration with LLRF [21].

Conventional Facilities

Conventional facilities, or site infrastructure has also been integrated into the control system for the relevant parts like electrical power distribution and water cooling.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR1BCO01

FUTURE WORK

While we have been through the commissioning of a substantial part of the ESS accelerator and many systems have been proven to work, there are still a lot of challenges remaining before the control system can be considered complete.

Upcoming Challenges

Until now controlling the subsystems has needed a big number of manual actions, to be performed in correct sequence. While this is natural and prudent in the early stages when the subsystems and the machine are new and it is not known how they will work in different situations, it cannot be sustained in the subsequent phases. As soon as component performance has been verified and is well known, routine operations should be automatized to reduce the operator's workload and increase the efficiency and the reliability of the operation. This can be done in many ways, but most of the automation should be implemented in the lowest practicable level, to make the behaviour consistent. Relying on e.g., scripts can cause inconsistent behaviour if the system is manipulated via some other method at the same time. In this area there is still a lot of work to be done and the best methods of implementation have to be worked out.

A related but different issue is operational sequencing, namely bringing the complicated machinery from a cold start to neutron production for users, and do that in a repeatable manner. Developments towards this goal have been already started but there is still a long way to reach a sustainable state. The challenges of managing large numbers of similar component systems as larger abstract entities are still to be solved.

LESSONS LEARNED SO FAR

On the way to this stage there have been several issues which have taught us important lessons. There are too many to be listed so I take up only a few, concentrating on my own point of view. These are not technical but rather managerial ones; in the end the technical issues are mostly easier to solve.

Difficulty in communication that arises from lack of a common language or vocabulary has caused more delays and friction than any technical issue. One should not assume that your message was understood as you expected. Cultivating personal contacts and lowering the threshold to ask even difficult questions is important.

In development, management push will only go so far. When engineers are empowered and supported, one can expect magic to happen. Empowering does not mean handing out a blank check, though. As a manager you should stick to your principles but be prepared to adjust course when needed. But also take time to explain your principles and make sure they are understood.

Be prepared to compromise – but do it for common good, not to pretend to be nice.

Be careful with timing of introducing new tools or ideas. Too early will frustrate and scare people off. Too late will be – too late, as costs to redo work that has already been done may be too high, or will not be accepted.

CONCLUSION

A lot has happened since the development of the control system started. While there has been a number of changes along the way, many things and in particular the fundamental ideas from the original design [22] have survived until today. There have been many changes in details but the basics have not changed a lot, and many of the original goals have been achieved. EPICS 7 has matured and we can employ it fully in production, and it provides a good basis for development in the coming years. Hardware standards are still as valid as they were at the time of selection, and are still evolving.

The organisation has grown from a modest start to a functioning organisation, of course not without its challenges but has been able to demonstrate great capabilities.

We are still in the middle of development on our way towards a fully functioning facility, but the achievements by now create confidence that we can not only reach the goals that have been set but also be able to evolve in the following years and stay on the forefront of development.

ACKNOWLEDGEMENTS

All the results here are due to the hard work of my colleagues at ESS, in the Integrated Control System Division as well as in other divisions, that were too many to include in the author list; instead, I have referred to the contributions in this and earlier conferences in the article. However, there is a lot more work that could not be referred to, either because there simply was not enough space to mention, or the work is still in progress and has not yet been published.

We are also indebted to the in-kind partners of ESS for their contributions.

Last, but certainly not least, we would not be where we are without the EPICS community, whom we owe a big thank you.

REFERENCES

[1] https://www.picmg.org/openstandards/microtca/

[2] F. Chicken *et al.*, "Development of Standard MicroTCA Deployment at ESS", presented at ICALEPCS'23, Cape Town, South Africa, paper THMBCMO21, this conference.

[3] https://en.wikipedia.org/wiki/EtherCAT

[4] http://www.beckhoff.com/twincat/

[5] T. Gahl *et al.*, "ECMC, the Open Source Motion Control Package for EtherCAT Hardware at the ESS", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 71-75. doi:10.18429/JACoW-ICALEPCS2017-MOCPL05

[6] A. Rizzo *et al.*, "Full Stack PLC to EPICS Integration at ESS", presented at ICALEPCS'23, Cape Town, South Africa, paper THMBCMO11, this conference.

[7] R. Lange *et al.*, "Integrating OPC UA Devices in EPICS", in *Proc. ICALEPCS'21*, Shanghai, China, Oct. 2021, pp. 184-187. doi:10.18429/JACoW-ICALEPCS2021-MOPV026

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-FR1BCO01

[8] https://epics-controls.org

[9] R. Lange *et al.*, "Five years of EPICS 7 - Status Update and Roadmap", presented at ICALEPCS'23, Cape Town, South Africa, paper TH1BCO01, this conference.

[10] http://e3.pages.esss.lu.se/

[11] https://controlsystemstudio.org/

[12] https://slacmshankar.github.io/epicsarchiver_docs/index.html

[13] B. Bolling *et al.*, "Dynamic Control Room Interfaces for Complex Particle Accelerator Systems", presented at ICALEPCS'23, Cape Town, South Africa, paper TUMBCMO07, this conference.

[14] B. Bolling *et al.*, "Multi-Dimensional Spectrogram Application for Live Visualization and Manipulation of Large Waveforms", presented at ICALEPCS'23, Cape Town, South Africa, paper TUMBCMO12, this conference.

[15] R. Fernandez et al., "Controls Configuration Database at ESS", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017. doi:10.18429/JACoW-ICALEPCS2017-TUPHA156

[16] A. Nordt *et al.*, "Verification and Validation of the ESS Machine Protection System-of-Systems (MP-SoS)", presented at ICALEPCS'23, Cape Town, South Africa, paper TU2BCO06, this conference.

[17] S. Pavinato *et al.*, "The ESS Fast Beam Interlock System - Design, Deployment and Commissioning of the Normal Conducting Linac", presented at ICALEPCS'23, Cape Town, South Africa, paper TUPDP01, this conference.

[18] T. S. Richter *et al.*, "Time Synchronization and Timestamping for the ESS Neutron Instruments", presented at ICALEPCS'23, Cape Town, South Africa, paper THMBCMO24, this conference.

[19] J. Cereijo García *et al.*, "Timing System at ESS" in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017. doi:10.18429/JACoW-ICALEPCS2017-TUPHA088

[20] A. Gorzawski *et al.*, "EPICS NTTables for Machine Timing Configuration", presented at ICALEPCS'23, Cape Town, South Africa, paper TUPDP094, this conference.

[21] G. de Souza Fedel *et al.*, "LLRF and Timing System integration at ESS", presented at ICALEPCS'23, Cape Town, South Africa, paper THPDP051, this conference.

[22] T. Korhonen *et al.*, "Status of the European Spallation Source Control System", in *Proc. ICALEPCS'17*, Melbourne, Australia, Oct. 2015, pp 1177-1181. doi:10.18429/JACoW-ICALEPCS2015-FRB3O02