ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOCPR02

THE EPICS COLLABORATION TURNS 30

L. R. Dalesio, Osprey DCS LLC, Ocean City, USA
A. N. Johnson, Argonne National Laboratory, Lemont, USA
K.-U. Kasemir, Oak Ridge National Laboratory, Oak Ridge, USA

Abstract

At a time when virtually all accelerator control systems were custom developments for each individual laboratory, an idea emerged from a meeting between the Los Alamos National Laboratory developers of the Ground Test Accelerator Control System and those tasked to design the control system for the Advanced Photon Source at Argonne National Laboratory. In a joint effort, the GTACS toolkit concept morphed into the beginnings of a powerful toolkit for building control systems for scientific facilities. From this humble beginning the Experimental Physics and Industrial Control System (EPICS) Collaboration quickly grew. EPICS is now used as a framework for control systems for scientific facilities on seven continents. The EPICS Collaboration started from a dedicated group of developers with very different ideas. This software continues to meet the increasingly challenging requirements for new facilities. This paper is a retrospective look at the creation and evolution of a collaboration that has grown for thirty years, with a look ahead to the future.

HISTORY

This year is noted as the 30th anniversary of the Experimental Physics and Industrial Control System (EPICS). We consider 1989 to have been the start of the collaboration between Los Alamos National Laboratory (LANL) and Argonne National Laboratory (ANL), formed by Mike Thuot, LANL, and Marty Knott, ANL, after discussions at the 1989 ICALEPCS and a subsequent technical APS evaluation of toolkits that had been developed at Los Alamos.

The precursors to EPICS were the Ground Test Accelerator Control System (GTACS) started in 1985 [1-4] and the Los Alamos Accelerator Control System (LAACS) presented in 1989 [5, 6]. The name Experimental Physics and Industrial Control System (EPICS) was adopted to reflect the collaborative nature of the toolkit.

Most of the core ideas were presented and evolved at two meetings that were the predecessors of the ICALEPCS series. The first was held in Los Alamos and included a number of other international laboratories. A second meeting was held at CERN in 1987. In 1989 these meetings started the ICALEPCS conference series. In 2013 the ICALEPCS International Scientific Advisory Committee presented the early organizers Peter Clout, Axel Daneels, Shin-Ichi Kurokawa, Dave Gurd, Daniele Bulfone and Ryotaro Tanake with a lifetime achievement award.

Channel Access [7] was the key network protocol that connected the distributed EPICS components, and initially the version number of the communication protocol also determined the major EPICS version number. EPICS version 3 started with the third iteration of the Channel Access protocol, but in 1991 a modification to the Channel Access code introduced a minor protocol version number field and started a long history of network compatibility between released versions of the toolkit.

From the start of this collaboration, all source code was meant to be shared. Code was co-developed, inherited and developed serially or competitively. This sharing of software was done openly from 1989 on, prior to the now familiar notion of Open Source Software (OSS) development. Lacking the commonly adopted OSS legal structures, joining the EPICS collaboration initially required signing an agreement to gain access to the source code. Until 2004, over 100 organizations had signed this agreement. Finally, the legal departments of APS and LANL allowed EPICS to become OSS, with all source code and documentation made freely available from web sites hosted by collaboration members. Since that time, there has been no record kept of the number of facilities using EPICS.

TECHNOLOGY ENABLES SCIENCE

The EPICS collaboration has been supported by international science facilities. The science that is produced in these facilities has been at the forefront of discovery including: high brightness X-Rays, Neutrons, Gravitational Wave Sensors, Radioactive Ion Beams and Nuclear Fusion, as well as Optical, Radio, and Infrared Telescopes. Innovation in the capacity, precision and speed of electronics, networks, sensors, and storage have all advanced the ability to gather data for analysis. The expectations of a control system have grown with these capabilities and the collaboration model allows facilities to build on the work of others. EPICS is now being used to control large and small scientific projects on every continent (including Antarctica).

MOTIVATION BEHIND THE DEVELOPMENT OF EPICS

The EPICS toolkit approach had these goals [8]:

- Allow independent subsystem development
- Reduce programming costs on projects by 85%
- Provide the best possible performance
- Allow extensions at every level of the toolkit.

The seamless integration of new Input-Output Controllers (IOCs) with self-contained application directories satisfied the independent development goal. For every newly developed component, performance measurements were performed. Designs were iterated to improve throughput and reduce latency. Narrow, well defined interfaces were provided at many levels of the system to support extensibility. EPICS dramatically reduced the time to produce control system applications. But the overall cost of control

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOCPR02

engineering dropped only by about 40% as engineers spent the time saved in the basic software implementation phase by now being able to provide system engineering and integration support. One side effect was a significant improvement in reliability, as the primary parts of the code ran in many facilities and different environments [9].

In experimental facilities it is common for requirements and specifications to develop late in the project cycle. In EPICS, control system tasks that once required implementing new software components have been replaced by the configuring of existing, shared software components. This configuration-based approach includes both the front-end computational nodes as well as operator interfaces [10]. Configuration of a control system in place of programming supports rapid modification with much less risk of software errors. All common code was carefully optimized to get the maximum performance possible as the demands of the experimental facilities have typically exceeded the available electronics. Clean interfaces were developed to support new hardware interfaces and expand the set of tools available to collect, save, view and manage data. The proliferation of drivers and clients developed under EPICS V3 suggests that this was done successfully.

THREE ASPECTS OF EPICS

The EPICS Architecture

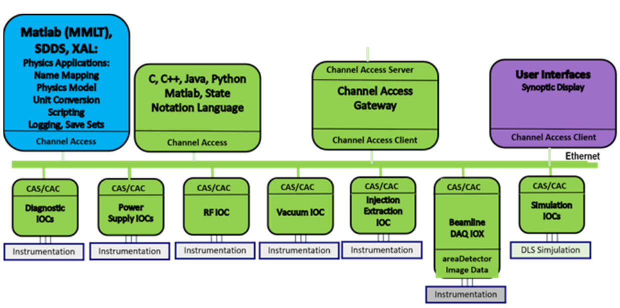

Figure 1: Fundamental EPICS Client/Server Architecture.

The EPICS Architecture (see Fig. 1) followed what was referred to as "the standard model." This is a distributed control system with Front End controllers that integrate hardware into the control system, and network clients to provide Supervisory Control and Data Acquisition (SCADA) tools such as synoptic displays, time series archiving, and alarm collection and aggregation. The EPICS interfaces for clients were greatly exploited to develop environments for physics applications. Some of these include the MATLAB Middle Layer Toolkit, Self-Describing Data Sets (SDDS), and the accelerator physics software platform XAL.

In the EPICS database, real-time "records" represent input signals, outputs, and algorithms to implement most steady state control and data acquisition for scalar values. The State Notation Language sequencer provides high performance state machine support. The more recent Area Detector collection of low-level drivers and waveform records builds on the EPICS database to provide a comprehensive and highly flexible N-dimensional array and image processing pipeline environment, used for cameras and detector data collection. The EPICS Architecture has been used and expanded successfully since 1989.

The EPICS Toolkit

The EPICS Toolkit consists of many client applications and one primary data service, the IOC which executes the runtime database. Initially the IOC and the Channel Access server ran only on the commercial VxWorks real-time operating system. At the start of this millennium, the EPICS 3.14 series releases added support for additional operating systems. EPICS supported VMS and Sun OS/Solaris while these were viable, and still runs IOCs on VxWorks as well as RTEMS, Linux, Windows and MacOS.

IOCs can connect to devices via Ethernet, RS-232, RS-485, I^{2}C, Modbus, Control Logix, VME, VXI, CAN, USB, µTCA, ATCA, PCI, cPCI, PCIe, PC/104, MIL-STD 1553 and IndustryPack. It has outlived CAMAC and GPIB.

There is client support for C, C++, Python/Jython, Java, MATLAB, and C#. There are gateways to LabVIEW, OPC, OPC-UA, and others to integrate existing control systems. Perhaps the most evolving area is in the display technology, where EPICS started by supporting X11. EPICS display tools survived NEWS, Motif, TK and Eclipse/RCP/SWT. At this time, actively supported User Interface technologies include: X11, Qt, and JavaFX.

The EPICS Collaboration

Figure 2: The first collaboration meeting at LANL.

The strongest element of EPICS is the collaboration. In the beginning, collaboration meetings (for example Fig. 2) focused on developers, and occurred at least four times a year. There is a cost in time and energy to listen and understand the concerns of others, with a real opportunity to learn. While the work of each developer is out of necessity primarily driven by the needs of his or her home institute, participants in the EPICS collaboration try to come up with solutions to local problems which are transferrable or adaptable to the needs of other EPICS sites. Contributions can be as small as suggestions for new features and bug reports, or as large as a new client tool. The plethora of supported control system hardware and interfaces is a di-

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOCPR02

rect result of the EPICS collaboration. Facilities were willing to support developments to the extent that they handled more than their own requirements. For example, one controls group at KEK received funding to support EPICS training throughout Asia, which enabled many laboratories in that region to successfully use EPICS. Anyone adopting EPICS is the beneficiary of 100's of Full-Time Equivalent (FTE) years of development and testing. More importantly, they have the support and goodwill of those that have also chosen to work collaboratively. All of the developer meetings and many of the talks in the collaboration meetings provide frank insight into the benefits of this approach.

Multiple collaboration email lists with full message archives provide access to a world-wide knowledge base. As a community, people have been able to move between laboratories with all of the skills and experience needed to start contributing to new projects with zero start-up time. The opportunity to work with such talented people is unprecedented, and the comradery in this community continues to provide real satisfaction for most participants.

MIDLIFE CRISIS

When EPICS reached the age of 20, it became obvious that it could benefit from better support for complex data sources. A series of meetings was held to collect ideas about the direction that should be taken for a new version of EPICS or an alternative toolkit. System architects from the ICALEPCS community, including astronomy and particle physics control system developers, joined these meetings to freely share their opinions and ideas.

It became clear that many SCADA applications collect data that would benefit from a network API; the separation of data collection and viewing would provide a cleaner interface. Should EPICS evolve from a signal-based API into a device-based API, from a channel protocol to an object broker protocol? Intelligent devices that use Ethernet as an instrumentation bus were recognized as a huge shift from bus-based signals. At the same time, a clear statement from our user community was that EPICS V3 is not "broken." Existing interfaces for integrating I/O devices must not be abandoned in the course of this upgrade.

A very small group started developing prototypes for a new EPICS network protocol and data representation. One challenge was to send more data while keeping high performance and robust operations. The initial work was funded by a Small Business Innovation Research Grant from the US Department of Energy.

The new protocol PVAccess that resulted can fundamentally handle arbitrary data structures, unlike the earlier Channel Access protocol. In addition to the read, write, and subscribe operations already supported by Channel Access, it adds command/response operations to support the needs of middle layer services. Subscriptions are optimized to only send changed structure elements over the network.

NSLS II at Brookhaven National Laboratory and the European Spallation Source provided funding and people to apply this architecture to a set of middle layer services and the EPICS IOC. As part of this effort, a set of Normative Data Types was developed to mimic the Channel Access "DBR Types," and to also allow the transfer of Area Detector images as well as Tables and statistical samples.

The CS-Studio operator interface toolkit fully supports PVAccess, for example it can directly display images received from Area Detector via this new protocol [11].

PVAccess is still new. Details of the API are evolving as the initial interfaces were deemed difficult for most programmers to use. Nevertheless, it is already being used successfully in several production environments alongside Channel Access [12].

EPICS 7 – COMBINING EPICS V3 WITH PVACCESS V4

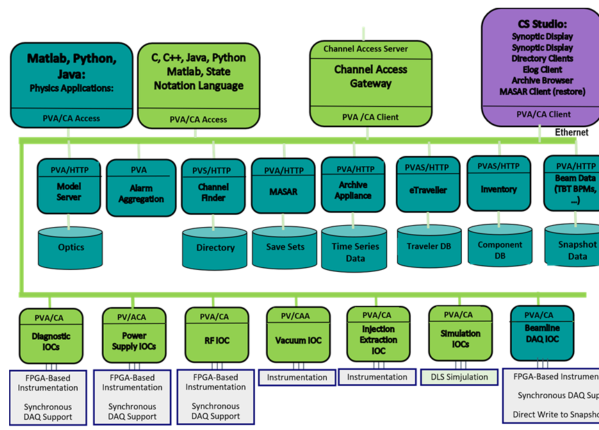

Figure 3: EPICS 7 adds PVAccess and middle-layer services to EPICS.

PVAccess was originally considered to be part of a rewrite of EPICS that was being called EPICS V4, but we wanted to stay 100% compatible with existing installations that use and will rely on Channel Access for the foreseeable future. EPICS 7 thus combined all of EPICS V3 along with the new capabilities of the V4 modules [13], see Fig. 3. The IOC process database and drivers are still supported without changes. Channel Access is still the primary protocol used by the IOC, but users can also access IOC data via PVAccess. The Normative Data Types extend the original EPICS timestamp with a user-defined identifier, and the EPICS Alarm metadata type now offers an alarm message string.

Custom PVAccess servers, implemented in C++, Python or Java, can provide middle-layer services for arbitrary data structures. Examples that benefit from the new command/response mechanism include the Directory Service which provides a structured view of the flat name space. It offers PV name lookup based on physics application needs, geographical location, device association or any arbitrary grouping. These services have been integrated into the CS-Studio environment to, for example, perform automated directory lookups while prompting the user for a PV name, as shown in Fig. 4.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOCPR02

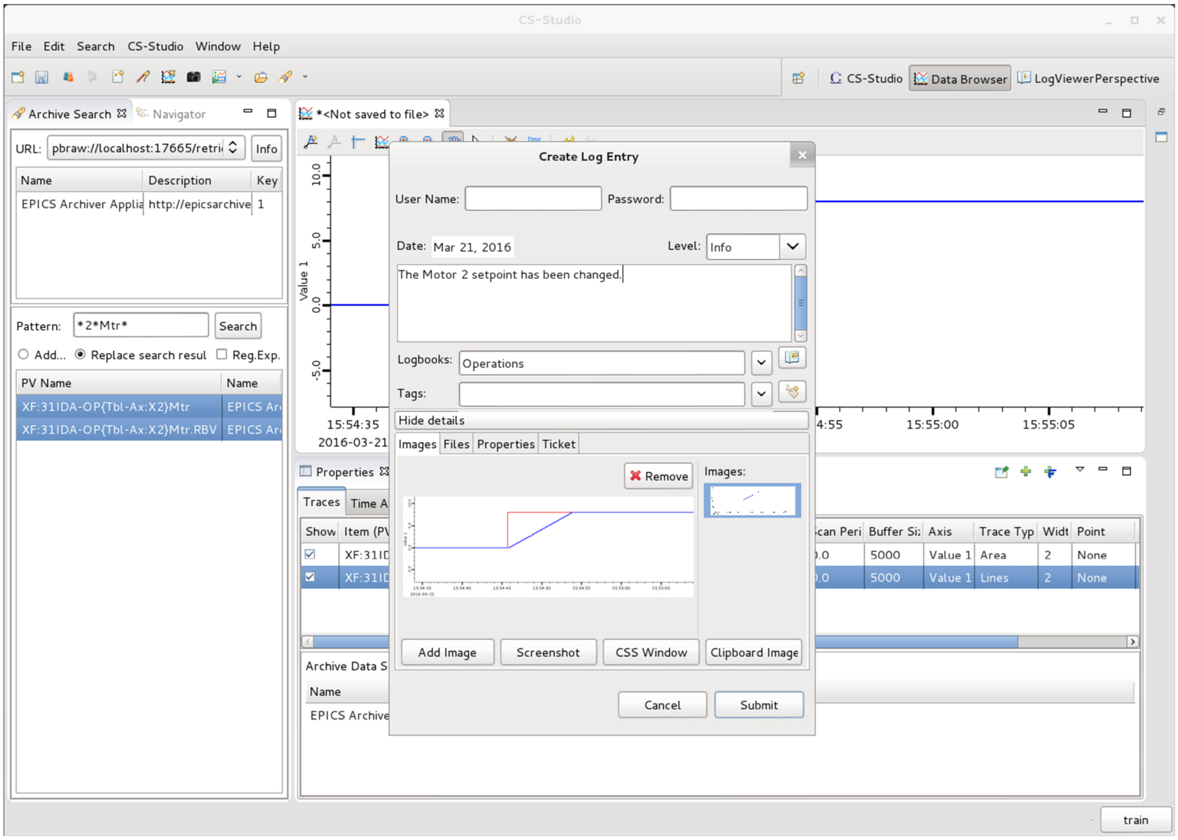

Figure 4: An example CS-Studio workflow: Locating channels by name, plotting the archived data, then submitting a log entry showing the results.

While the EPICS IOC now supports functionality via PVAccess similar to Channel Access, work remains to allow using the advanced features of PVAccess (multi-dimensional arrays, tables, statistics) inside the IOC process database. New records will be developed to support the new structured data types, to implement table merges and n-dimensional array manipulations for example. The completion of the PVAccess gateway has been funded by the Stanford Linear Accelerator Laboratory. The PVAccess Java and C++ implementations are being revised by experts in the respective languages to clarify locking and to create APIs that will fully benefit from their respective programming languages, funded by the Oak Ridge Spallation Neutron Source.

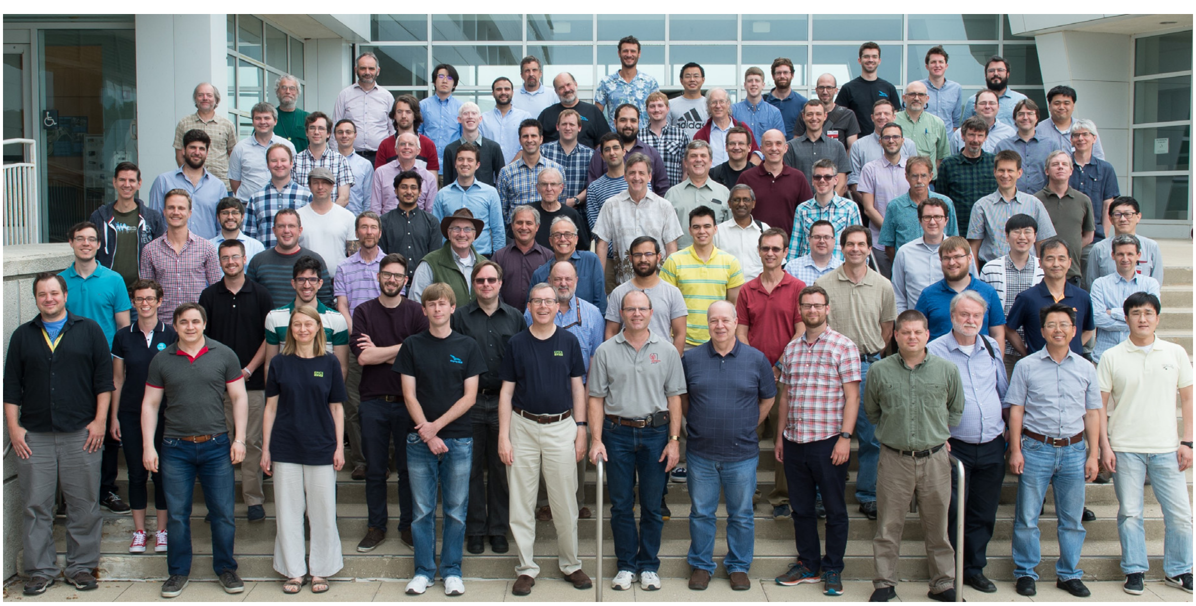

Figure 5: Collaboration meeting at Argonne, June 2018.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOCPR02

THE NEXT 30 YEARS

The EPICS architecture, which served the participants of the EPICS community well for the last 30 years, has recently undergone a substantial extension. The toolkit is well poised to support scientific exploration in the future, to continue to adapt to technological changes, and to drive future workflows. It is to be hoped that the collaboration will continue to commit resources to attract the critical mass of dedicated, top-quality people (e.g. Fig. 5) required to support the community.

ACKNOWLEDGEMENTS

We recognize an entire community for their contributions, small and overwhelming. The ability and the desire to help out others in this field speaks to the higher purpose of the science that is enabled by the technology that we have provided.

REFERENCES

[1] L. R. Dalesio, "The Ground Test Accelerator Control System Database: Configuration, Run-time Operation, and Access," Particle Accelerator Conference, March 20-23, 1989, IEEE 0.1109/PAC.1989.72896

[2] A. J. Kozubal, L. R. Dalesio *et. al.*, "Run-Time Environment and Application Tools for the Ground Test Accelerator Control System," in *Proc. ICALEPCS'89*, Vancouver, Canada, 1989.

[3] A. J. Kozubal, L. R. Dalesio *et. al.*, "A State Notation Language for Automatic Control", in *Proc. ICALEPCS'89*, Vancouver, Canada, 1989.

[4] A. J. Kozubal, D. M. Kerstiens, J. O. Hill, and L. R. Dalesio, "Run-time Environment and Application Tools for the Ground Test Accelerator Control System," in *Proc. ICALEPCS'89*, Vancouver, Canada, 1989.

[5] L. R. Dalesio, J. O. Hill *et.al.*, "Los Alamos Accelerator Control System: The Second Generation," in *Proc. ICALEPCS'89*, Vancouver, Canada, 1989.

[6] L. R. Dalesio, "The Los Alamos Accelerator Control System Database: A Generic Instrumentation Interface," in *Proc. ICALEPCS'89*, Vancouver, Canada, 1989.

[7] J. O. Hill, "Channel access: A software bus for the LAACS," in *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 293, no. 1–2, Aug. 1990, pp. 352-355.
https://doi.org/10.1016/0168-9002(90)91459-O

[8] L. R. Dalesio, M. R. Kraimer, and A. J. Kozubal, "EPICS Architecture," in *Proc. ICALEPCS'91*, Tsukuba, Japan, 1991.

[9] L. R. Dalesio and M. E. Thuot, "The Effects of the Toolkit Approach to Experimental Physics Control," in *Proc. ICALEPCS'91*, Tsukuba, Japan, 1991.

[10] J. O. Hill, L. R. Dalesio *et. al.*, "A Virtual Control Panel Configuration Tool for the X-Windows System," in *Proc. ICALEPCS'91*, Tsukuba, Japan, 1991.

[11] K.-U. Kasemir and M. L. Grodowitz, "CS-Studio Display Builder", in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17)*, Barcelona, Spain, Oct. 2017, pp. 1978-1981.
doi:10.18429/JACoW-ICALEPCS2017-THSH303

[12] K.-U. Kasemir, G. S. Guyotte, and M. R. Pearson, "EPICS V4 Evaluation for SNS Neutron Data", in *Proc. 15th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'15)*, Melbourne, Australia, Oct. 2015, pp. 947-949.
doi:10.18429/JACoW-ICALEPCS2015-WEPGF105

[13] L. R. Dalesio *et al.*, "EPICS 7 Provides Major Enhancements to the EPICS Toolkit", in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17)*, Barcelona, Spain, Oct. 2017, pp. 22-26.
doi:10.18429/JACoW-ICALEPCS2017-MOBPL01