# THE SUCCESS AND THE FUTURE OF EPICS

M.E. Thuot, Los Alamos National Laboratory
M. Clausen, Deutches Elektronen-Synchrontron
L.R. Dalesio, Los Alamos National Laboratory
T. Katoh, KEK National Laboratory for High Energy Physics
M.E. Kraimer, Argonne National Laboratory
R. Mueller, Berliner Elektronenspeicherring Gesellschaft fur Synchrotronstrahlung
H. Shoaee, Stanford Linear Accelerator Center
W.A. Watson, Thomas Jefferson National Accelerator Facility

## Abstract

During the past five years, the control system software toolkit called EPICS (Experimental Physics and Industrial Control System), has developed from a comparatively small code co-development effort between Los Alamos National Laboratory and Argonne National Laboratory into an international collaboration on real-time distributed control systems. The wide application of this set of tools is the result of a combination of high performance, scaleable distributed control and well defined open interfaces between system layers that encourage users to add extensions. These extensions can subsequently be reused by others, adding to the utility of the tools. This paper will describe the architectural features that have supported these extensions, some of the new extensions produced by the 58 projects currently using EPICS and some of the new functions and interfaces we are planning to add to this control system toolkit.

## Introduction: EPICS' primary functions and architecture - a basis for collaboration

The development of EPICS started with a few fundamental user- driven requirements. Heading the list was the need for a control system that could be expanded easily and could concurrently support subsystem development and commissioning as well as integrated operation of an entire accelerator facility. A maximum level of functional flexibility in the control system software without additional programming was also a prime requirement. Further, the software needed to provide excellent real-time performance to close control loops in and between subsystems. These and other functional requirements were met in the original design of the EPICS control system toolkit that employs a client/server architecture for data transport to all system components and a modular, "process model" database for data acquisition and control.

## Two structures fundamental to EPICS success

The two structures are: a distributed control system architecture based on client/server data communications and a modular "process model" database. The requirement to provide independent sub-system operation as well as integrated system control in a scaleable control system was addressed by the first of these fundamental structures, a

distributed control system architecture-based on a client/server data communications design[1]. The server module resides in each one of the system controllers along with that controller's portion of the distributed database. From there, the server is able to send any value, calibration, alarm status, etc. to any client application, e.g., a display, archiver, etc., that requests that data by name. The multiple instances of the server in a control system respond to a request for data by searching for the named data in their local portion of the distributed database. If a server has the requested data in the local portion of the database, it responds to the request and the client then initiates a connection to that data. This allows any of an arbitrary number of data-using clients to receive data from any of an arbitrary number of data-supplying servers without the client needing to know the location or other attributes of the data. The result is an easily scaleable control system which can operate as separate independent controllers in an integrated system.
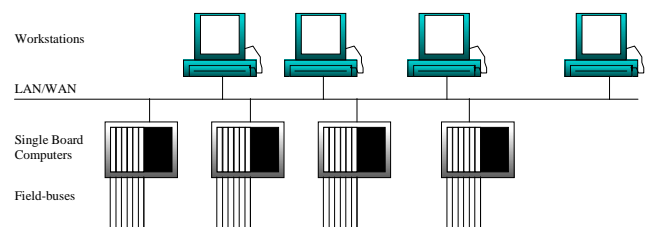


Fig. 1. Distributed Control System Architecture

The functional flexibility requirements were met by EPICS' second fundamental structure, a modular "process model" in the EPICS database[2]. A process model was embodied in a unique active database distributed to the many controllers in a control system. The process model consists of a set of input, process and output records in the database that are scanned by time or event. An input record for any type of data source can be selected from an extensible set that includes discrete (binary), analog and waveform (array) input records. These input records, when scanned, will interact with software drivers and acquire data from the facility through a

wide variety of supported VME, VXI, CAMAC, or industrial data acquisition modules. The data measurement thus acquired can be linked to one or more of a set of process records that can transform the data, or can contain a feedback controller (e.g. PID), or can generate a sequence of outputs. These outputs can be linked to discrete, analog, positioner, etc. output records which provide feedback to the facility. Input and output records may be used independently for supervisory control or linked with process records in an almost unlimited variety of combinations to provide automatic control for most of a facility's data acquisition or control needs.

The EPICS database is generated by data-driven configuration tools and for most data acquisition and control no additional programming is required. Using these tools, the control system implementers can respond rapidly to changes in control requirements. Using EPICS client/server data communication called "channel access", an X-Windows based operator interface can display any of the input, process, output, status, alarm, etc., values in the database from anywhere on the control system network. These fundamental structures and a set of tools for creating process databases, for creating operator displays and for archiving data were in the initial software toolkit that was the basis for the EPICS collaboration.

## Collaboration with large facilities produce new requirements and developments

The next phase of EPICS development began with code development collaborations with several large facilities then under construction. During control system development for these large facilities, new requirements were identified and the set of interfaces in EPICS were extended.

## Client Extensions

The original client/server channel access data communication network defined an interface between the hardware, driver, data acquisition, control, and database functions on the server side and other control system functions such as data display, operator control, archive, analysis, etc., on the client side. This clean client/server interface was the basis for the first major expansion of EPICS when we began our collaborative co-development effort in 1989. The Advanced Photon Source (APS) at Argonne National Laboratory, uses EPICS to control a linac, a positron accumulator ring, a booster, a storage ring and a large number of x-ray beamlines. To respond to facility requirements to monitor the alarm status of 160,000 data channels distributed over more than 100 controllers, APS designed and implemented an alarm manager client to group, sort, log and acknowledge alarms as well as a knob manager to connect a hardware knob to a controllers and a general purpose monitor and control tool, Probe. The APS team also interfaced the EPICS channel access client software to several commercial data analysis and presentation packages including PvWave and WingZ enabling the analysis of any data item in the

system. Later, they implemented a MOTIF based operator interface to add drag and drop capabilities to the operator interface. Outside Paris, SACLAY added SL-GMS as an operator interface client based on their need for a richer set of display objects to provide the control required for the Tesla Test Facility injector development program. A client interface to Dataviews was implemented at Lawrence Berkeley Laboratory (LBL), for use on the Gamma Sphere project while the University of Chicago developed and contributed an interface to Interactive Data Language. The channel access client interface continues to support a large and growing set of extensions to the EPICS toolkit.

## Database Record and I/O driver extensions

The EPICS database records comprise another clean extensible interface, especially after the APS team defined and implemented an interface to separate the input/output records and the hardware driver software layer. Many new record types have been developed including signal select, data compression, timer, sequence, interlock, fanout, magnet setpoint, beam position monitor, etc. Almost every institution that has subsequently used EPICS responded to the clean hardware driver interface by contributing dozens of additional drivers for a multitude of hardware types and modules. APS produced the GPIB and Bitbus drivers supported under EPICS or a small independent operating system (HideOS). The CAMAC driver was built and then improved by a collaboration between TJNAF (formerly CEBAF), the Duke University FEL team and LANL. CAMAC is used as the primary I/O interface for the 100K channel central control system for the recirculating beam superconducting linac at TJNAF and for the proton storage ring at LANL. ProfiBus and Siemans HI bus drivers were produced by DESY for use on the cryogenic controls at HERA. The CANbus has been implemented by DESY, Royal Greenwich Observatory (RGO) and BESSY for use on the Synchrotron Light Source and industry pack support has been produced by DESY and LANL.
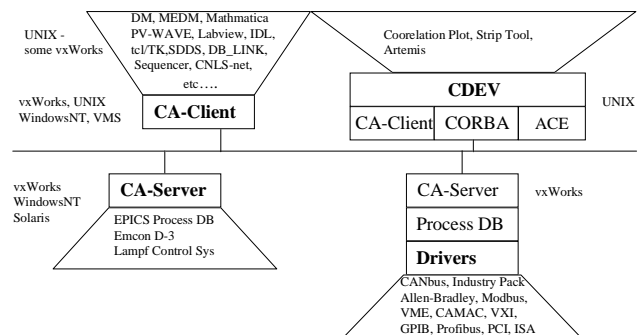


Fig. 2. Clean Interfaces Support Independent Development and Provide an Incremental Upgrade Path to Avoid Obsolescence

## Multiple platform support developed

Many programs that were using EPICS to build and integrate control systems needed to develop and run the code on their local Sun, H/P, DEC, UNIX workstation platforms. Many ports of EPICS to different platforms resulted: The port of the channel access client codes and development platform to Solaris was done at the RGO; to HPUX at TJNAF; to DEC UNIX at the University of Chicago and SLAC; the database is being ported to *LynxOS* at NASA Langley, and the channel access client was ported to Windows 95 and Windows NT at LBL. Changes to the original Sun UNIX installation procedures were extensive and resulted in clean, documented and easy to use procedures to build and install EPICS systems at new sites. Today, APS provides EPICS software distribution to many R&D institutions and a significant part of the documentation support on the Internet for the EPICS community. The balance of the EPICS documentation is served on the WWW from LANL and TJNAF.

## Operational reliability demonstrated

This fully distributed system with no single point of system failure took long strides toward demonstrating reliability in continuous facility operation. EPICS, used for HERA cryogenic controls for monitoring 6000 channels by 5 IOC's, has run without rebooting for their eleven month operational cycle. At Argonne and CEBAF the majority of IOCs run reliably[3][4]. One reliability issue occurs when the front end controller is left with too little available memory. This is easily resolved by dividing the application into two IOCs or adding memory. The other issue occurs when sub-bus controllers, particularly the bitbus controller, gets hung and requires a reset on the VME bus. This problem has been traced to a defect in the controller's bitbus interface chip. Most installations do not experience these problems. The EPICS collaboration and toolkit is currently in use at over 50 installations and has provided improved control system reliability and significant cost savings.

| Project | #signals | #IOC | #workstations | status |
|---------|----------|------|---------------|--------|
| APS | 175,000 | 130 IOC | 20 ws | Mostly reliable* |
| HERA Cryo | 6,000 | 4 IOC | 1 ws | reliable operation |
| Tesla Test Inj | 600 | 4 IOC | 2 ws | reliable operation |
| KeckII | 1,500 | 2 IOC | 2 ws | reliable operation |
| PSR | 2,500 | 4 IOCs | 6 ws | reliable operation |
| Duke FEL | 2,500 | 6 IOCs | 3 ws | reliable operation |
| PEP II RF | 8,400 | 8 IOCs | 2 ws | 80% complete |
| BESSY | 15,000 | 15 IOCs | 4 ws | construction |
| 1APS Beamline | 15,000 | 22 IOCs | 10 ws | reliable operation |
| CEBAF | 125,000 | 40 IOCs | 8 ws | Mostly reliable** |

\* a bitbus controller mishandles comm. errors and requires a VME reset
\*\* memory fragmentation causes ca-client connections to be refused when IOCs are run with over 7,000 process records on 16 MegRAM

## Other functional additions

During the period where large user facilities were joining the collaboration, many incremental changes occurred to make EPICS easier to configure, more secure, and more portable. Channel access improvements supported data communication connections across a wide area network allowing users to access data even though the data source (server) was not on their local area network. This provided support for more widely distributed control networks as well as providing the ability to improve system communication bandwidth by isolating network traffic on sub-nets while still providing integrated access to the data. Shortly afterwards access control was added to channel access to provide a level of data and control security. The original access control evolved at APS and LANL to provide data security for facility parameters that need to be protected from read or write access by some users or locations during specific modes of operation. Graphical tools for database definition with hierarchical capability and multiple instantiation were developed to simplify and speed the conversion of the nearly 100,000 channels in the TJNAF control system to EPICS. The ability to have a database make references to data in other databases without concern for their location on the network, the ability to change channel addresses and data links during operation and to directly load database and record types in an ASCII format all provide more powerful and flexible logic definition to the application engineers. This work was done in a collaboration with the controls groups of KEK, LANL, and APS. These functions became even more important as large user facilities became operational.

## Support for commissioning and high-level control

The early EPICS tools were adequate for construction and testing physics research facilities, however, they were not adequate for commissioning complex accelerator facilities and telescopes. New facility commissioning tools were needed. In response, the APS community developed a suite of command line tools based on Self Describing Data Sets (SDDS)[5] with modular functions that could be linked together to provide quick turnaround data acquisition, data analysis, visualization, and high-level control, such as bumps, global orbit correction and ad-hoc experiments. Tcl/TK user interfaces have been built on top of these SDDS tools to make them more friendly for operators. TJNAF is developing interactive graphical tools to provide similar data analysis functions modeled on the high level tools at SLAC[6]. In addition, TJNAF has developed a model server called ARTEMIS and integrated this beamline model with their EPICS real-time data system. The late conversion of the TJNAF control system to EPICS delayed the development of these commissioning tools until recently. KEK on the other hand, has already interfaced their modeling code, SAD, to the EPICS system for use during operation at TRISTAN in preparation for the KEK B-factory[7]. These commissioning tools have been successfully used to identify and determine the root cause of beam control problems within an operations shift and to optimize beam control parameters.

## A higher-level interface is defined

As these high-level tools have been developed, the need to provide data abstractions at a level higher than the individual control system channels provided by channel access and to integrate non-real-time data, such as beamline configuration, became apparent. TJNAF developed an application program interface (API) called common devices, CDEV[8], that provides this level of abstraction. For example, using CDEV all the separate data and control channels used to power, cool, control and monitor a power supply and a beamline magnet could be represented by a higher level abstraction that can be named as a single object, e.g., "quad n". This level of abstraction simplifies the interface between a beamline model and the control system by hiding the hardware and data dependencies associated with a particular instance of the power supply connected to a coil with its associated cooling and power, etc. CDEV currently supports the channel access client as well as CORBA bindings. It also provides the ability to re-map the control system name space thus keeping high level applications from being modified when the low level organization changes. This interface is viewed as a mechanism to make high level applications shareable even at installations that do not use EPICS. A distributed client/server based archiver and the X-based display manager are both slated to use this interface. CDEV itself continues to evolve with a higher level of organization in the form of collections of devices in a future release.

## Taking advantage of a multi-lab collaboration

Learning to work together effectively was a significant ingredient in the success of this work[9]. It became evident that research programs needed to be able to do independent system development to meet the requirements of their projects. The clean interfaces that make many of the extensions simple to add are a portion of this success[10]. The resource to reintegrate this work is provided by the APS controls group. The support for learning, installing, extending and trouble shooting the system is provided by all experienced members of the collaboration. KEK supports other projects in Japan, APS supports their many beamline users, CEBAF supports their experimental users, central project groups in large distributed collaborations like the Gemini telescope and the BaBar detector support the multitude of universities that are associated with them. It is impossible to quantify the value of this working relationship, but nearly all members will point to this cooperation and support as one of the major benefits of using EPICS.

## The results of collaborations with large facilities

During this phase of the collaboration, we successfully learned how to effectively co-develop software to port the tools to various platforms and environments, to provide clean interfaces for extending the tools, and to effectively employ the expertise of the collaboration members. The fundamental architecture was demonstrated to scale up to a 100,000 signals distributed over more than 100 single board computers with up to 30 operator workstations. 60 Hz closed-loop control, distributed interlock implementation, the use of on-line modeling and adaptive control were all demonstrated to be supportable in the EPICS point-to-point distributed client/server model. As the toolkit became more completely developed, projects began to experience the cost advantages in using the shared development of EPICS. Most of the initial development costs of implementing a control system could be avoided by starting with EPICS.

## New system requirements and future EPICS developments

Now that APS and TJNAF are completing commissioning and beginning operation as user facilities, there is a need for hundreds of experimenters to have simultaneous access to current accelerator parameters and historical data. Since these data requests connect directly to the front-end computers controlling the process in the present architecture, the additional data communication load could impact the operation of the facility. Providing a gateway to concentrate requests for current data from multiple clients would allow many users to access data from the primary control network while limiting the impact of these many connections on the control network. This gateway has been implemented by APS and is under test. Hundreds of users will be able to observe data from the gateway while only one connection is made to each of the data sources in the control network. The additional latency introduced by a gateway is not critical as these users do not require high performance, real-time operation. The collection and distribution of archived data raises other issues. DESY has successfully used CORBA to distribute archived cryogenic system data. More stringent performance requirements may lead TJNAF to develop new archive data visualization tools that can correlate large numbers of channels of current and historical data.

## Extended support for alternate data sources

Another major change in EPICS will soon create an interface in place of the previous tight bond between the process database and the database server. Responding to user requests for the ability to serve data from legacy data stores, to serve data from the various workstations on a control network, and to serve data from alternate operating systems; a portable channel access server[11] has been developed under a distributed computing environment, DICCE, grant. This module defines a new system interface that allows legacy or other non-EPICS control systems to serve data to any EPICS client. It also enables processes, such as analysis or modeling programs, running in workstations on the control network, to supply data to displays, archivers or other clients. In the current alpha release, this server runs under Solaris and provide the ability to connect other data stores into EPICS. The portable server is being ported to VxWorks to replace the existing channel access server. With this new capability, it

will be possible to integrate existing systems and other data-stores (such as relational databases) and to port the EPICS database into other operating systems such as Windows NT, LynxOS, or UNIX.

## Lower cost small data/control systems

Small experiments or independently developed components need a control system solution that is packaged as a single box with no expensive components required for software or hardware, yet which is able to integrate into a larger facility or distribute it's data. Presently the minimum sized EPICS system is one client (workstation) and one server (a single board CPU running VxWorks) with some I/O modules. In the future, we plan to develop a single computer solution for small data/control applications while maintaining the client/server architecture so that the single computer application can be easily integrated into a larger control environment if and when it becomes necessary.

## System Data Available to Every Desktop

If we are to make all the control system data available to the desktops of physicists, commissioners and engineers, the current support for data to PCs through an X-client will need to be expanded to include native visualization tools under windows NT and windows 95. Extremely large systems, such as slow control for detectors with over a million signals, present new challenges in configuration management and distributed name resolution. The challenges in performance and labor cost for systems another order of magnitude larger than those operational today is being studied. Graphical configuration tools and relational database configuration tools have replaced some of EPICS original text-based tools to improve user friendliness and to respond to the demands for configuration control at large facilities. Specialized tools are being developed for accelerators, detectors and astronomy applications. Tools to provide data to users on any platform on the network are also being developed. There is continual improvement aimed at reducing the manpower required to define and maintain applications while providing users with more convenient desktop access to their data.

## Support for new hardware technology

As industry provides higher speed and higher accuracy hardware as well as lower cost platforms, the collaboration continues to take advantage of these developments. Originally, we used VME, GPIB, and Allen-Bradley industrial I/O. As other institutions joined, industrial offerings changed, and existing systems were integrated, support for CAMAC, VXI, Bit-Bus, and serial devices were added. Future support will be provided for the PCI and compact PCI bus with the IOC software running in industrial PCs. The extensible driver interface in EPICS has enabled any user to add I/O support. The APS has taken advantage of FDDI and Ethernet hubs to increase network bandwidth. ATM is also becoming a cost effective answer to increasing network bandwidth. As a result, any new and promising technology (as well as old existing technologies) can be easily integrated.

## Conclusions

The collaboration has successfully extended the base set of tools by taking advantage of well defined interfaces and productive cooperation. Today's core set of EPICS tools has been successfully used at green field sites and existing facilities with up to 160,000 signals controlled by 100 single board computers with 30 operator interface terminals spread over 3 kilometers in accelerator control, particle detectors, astronomy and other research areas. Minimally funded projects have been able to use high performance, distributed control technology. A large and diverse group of projects have found a common set of capabilities, but more importantly, they found the means to extend the core set and then reintegrate the extensions into a more capable toolset. The modular architecture allows continuous developments that eliminate obsolescence. Through the combined efforts of many institutions we continue to extend control system capabilities, reduce costs, and employ new technologies.

## References

[1] J.O.Hill, Channel Access: A Software Bus for the LAACS, International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS), October 30 - November 3, 1989, Vancouver, British Columbia, pp. 352-355.

[2] L.R.Dalesio, The LAACS Database: A generic Instrumentation Interface, ICALEPCS, Oct. 30 - Nov. 3, 1989, Vancouver, B.C..

[3] D.J. Ciarlette, Operational Experience from a Large EPICS based Accelerator Facility, presented at ICALEPCS, Oct. 1995, Chicago, Illinois. Session Th2-c*.

[4] K.S.White, Operational Experience with the CEBAF Control System, poster presented at ICALPECS, Oct. 1995, Chicago.,IL. Session F-PO-64*

[5] M.Borland, Doing Accelerator Physics Using SDDS, UNIX and EPICS, presented at ICALEPCS, 1995, Chicago, IL. Session Th3Be*

[6] S.Schaffner, et. Al., A Generalized Correlation Plot Package for CEBAF, poster presented at ICALEPCS, Oct. 29 - Nov. 3, 1995, Chicago, IL Session W-PO-31*.

[7] T.Mimashi, et.al., Feedback Controls for the TRISTAN Light Facility, poster presented at ICALEPCS, Oct. 29 - Nov. 3, 1995, Chicago.,IL. Session W-PO-28*

[8] J.Chen, An Object-Oriented Class Library for Developing Device Control Applications, presented at ICALEPCS, Oct. 29 - Nov. 3, 1995, Chicago.,IL. Session M4Ba*

[9] M.Knott, Communication in Support of Software Sharing and Collaborative Development, presented at ICALEPLCS, Oct. 29 - Nov. 3, 1995, Chicago.,IL. Session Th3-Bc*

[10] L.R.Dalesio, et.al., Distributed Software Development in the EPICS Collaboration, presented at ICALEPCS, Oct. 29 - Nov. 3, 1995, Chicago, IL Session Th3Ba*

[11] J.O.Hill, A Server Level API for EPICS, presented at ICALEPCS, Oct. 29 - Nov. 3, 1995, Chicago, IL. Session W1Ab*

* at: http://adwww.fnal.gov/www/icalepcs/abstracts/icalepcs_abstracts.html