EXPERIENCE WITH EPICS IN A WIDE VARIETY OF APPLICATIONS

Martin R. Kraimer
Argonne National Laboratory, 9700 South Cass Ave., Argonne, Illinois 60439
Matthias Clausen
Deutsches Elektronen Synchrotron (DESY), 85 Notkestrasse, Hamburg, 22603, Germany
William Lupton
W. M. Keck Observatory, 65-1120 Mamalahoa Highway, Kamuela, Hawaii 96743
Chip Watson
Thos. Jefferson National Accelerator Facility, 12000 Jefferson Ave., Newport News, Virginia 23606

Abstract

Currently more than 70 organizations have obtained permission to use EPICS, a set of software packages for building real-time control systems. In this paper representatives from four of these sites discuss the reasons their sites chose EPICS, provide a brief discussion of their control system development, and discuss additional control system tools obtained elsewhere or developed locally.

1 INTRODUCTION

The Experimental Physics and Industrial Control System (EPICS) [1] is currently in use at over 70 sites in Asia, Europe, and North America. It is used in accelerator facilities, astronomy, commercial industry, and industrial research. EPICS is the successor to the Ground Test Accelerator Control System (GTACS) that was developed at Los Alamos National Laboratory (LANL). EPICS resulted from a collaborative effort between the GTA controls group and the APS controls group. Over the last few years other sites have started using EPICS and contributing to its development.

Since EPICS is a collection of software components that can be used to develop a control system rather than a "turnkey" system, each site normally extends EPICS at the low level—i.e., custom record, device, and driver support—and adds support for high-level applications. This paper describes how four sites have used and extended EPICS.

2 ADVANCED PHOTON SOURCE

The Advanced Photon Source (APS) is a third-generation synchrotron light source located at Argonne National Laboratory. With regards to APS, this paper discusses the following: 1) APS/LANL cooperative development on input/output controller (IOC) software, 2) APS/LANL cooperative development on general-purpose high-level tools, 3) implementation of the APS accelerator control system, and 4) high-level applications developed by the APS accelerator physicists.

2.1 APS/LANL Cooperative IOC Software Development

During the last three months of 1989, APS evaluated GTACS by having a member of the controls group study the code and give tutorials explaining what he was learning. Most of the time was spent evaluating both the code which runs in the IOCs and Channel Access [2], the code which provides network-independent access to the IOCs. The result was a decision to use GTACS to build the APS control system. APS and LANL agreed to start cooperative development on improvements and on new general-purpose high-level tools [3].

While studying the GTACS code, APS identified many places where improvements could be made. The most important was the ability to add new record, device, and driver support without modifying core software. Although LANL also wanted the same or nearly the same improvements, GTA had already started commissioning and the GTA developers did not have time to implement these improvements. At that point the LANL and the APS staffs met and agreed on a design. APS then implemented the changes with input from LANL.

During the early years a lot of new hardware support was developed. If support for a device was developed at one site it was normally used at the other site. Improvements were folded into the common system. A VXI resource manager was implemented at LANL and used extensively at both sites. GPIB support was developed at APS and used at both sites. There was almost no duplication of effort in any area from Channel Access down to the device/driver software.

2.2 APS/LANL Cooperative Development on General-Purpose High-Level Tools

APS and LANL jointly decided on high-level tools which were needed but not yet available. The first tools to be developed were a general-purpose display manager/editor, an alarm handler, and an archive tool.

At that time a display manager/editor already existed but it ran only on VAX/VMS using a proprietary windowing system. LANL had already started design of an X-based replacement. When LANL/APS agreed to cooperate, the one point on which agreement could not be reached was whether this tool should be built directly on Xlib or on Motif. LANL, for performance reasons, wanted just Xlib. APS, for ease of development, wanted Motif. The end result is that two display managers/ editors were developed and are being used successfully. Each EPICS site normally chooses only one.

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.

The alarm handler tool was proposed and designed by APS, LANL reviewed it before development was started, and APS developed the tool. It is used at most EPICS sites.

The archiver was designed and implemented at LANL, reviewed at APS before development began, and implemented at LANL. It is used at many EPICS sites. Archiving is a difficult and many-faceted subject. Some EPICS sites use the LANL-developed archiver but many sites use other archive tools. New archive tools are currently being considered by several EPICS sites.

2.3 Implementation of the APS Accelerator Control System

The accelerator control system currently has 165 IOCs containing approximately 160,000 records. The main control room (MCR) has 10 Ultra Sparc workstations each equipped with two video monitors. The control system network has ten satellite hubs which are dual connected to a central hub using FDDI. Each IOC is connected to two hubs using a redundant fiber Ethernet transceiver. Each MCR workstation is connected via FDDI to a network hub.

The APS accelerator control system was developed by control engineers using the EPICS-supplied toolkit. The controls group assigned engineers to subsystems based on the APS technical groups, i.e. linac, rf, diagnostics, power supplies, and vacuum. For each subsystem one engineer was assigned overall responsibility for all controls-related activities, i.e., hardware and software. This engineer interacted closely with the group responsible for the subsystem itself. For example, the engineer discussed requirements with the people in the group, attended group meetings, etc. Controls group engineers are responsible for all controls-related activities from engineering-related display screens to the hardware that interfaces to the equipment.

In addition to EPICS-supplied support, the control engineers needed custom record, device, and driver support for their IOCs. Although they sometimes asked the APS EPICS developers for help, they developed most of the support themselves, implementing 13 special record types, 48 device support modules, and 2 drivers. In addition, a number of sequence programs and subroutines for the subroutine record were written.

The APS controls system makes extensive use of GPIB devices. BITBUS is used as a low-level subnet. Many of the special record and device support modules were created for GPIB and BITBUS devices. One of the EPICS tools developers created general-purpose GPIB support and helped with the BITBUS driver, but the rest of the support was developed by the control engineers.

An APS-developed event system, consisting of VME event generators and event receivers interconnected with 100-Mbaud fiber optic links, provides the means of distributing medium resolution/accuracy timing events throughout the facility. Special record support was developed to integrate the event system with EPICS, including the ability to synchronize time stamps across IOCs.

Other major APS control applications developed using the EPICS toolkit include fast timing, BPM control and monitoring, charge and current monitors, and the orbit feedback system.

2.4 High-Level Applications Developed by the APS Accelerator Physicists

The APS accelerator physicists have developed a set of high-level tools grouped under the collective name of Self Describing Data Sets (SDDS). The SDDS toolkit can read and write data in EPICS database records. Though most of the applications do rather simple operations, the combination of these and others from the SDDS postprocessing toolkit allow arbitrarily complicated analysis of data and control of the accelerators at the Advanced Photon Source. These tools are general and can be applied to devices other than accelerators under control of EPICS.

Initially adopted for complex physics simulation programs, it was clear that the SDDS file protocol would excel in data-collecting software as well. Typically, an EPICS tool would write EPICS data to an SDDS file with each readback written to a column of names corresponding to the EPICS database record name. Single value data that describe the experimental conditions might be written to the file as parameters. Once collected, the EPICS data can be further analyzed and plotted with any of the SDDS tools. One can regard the SDDS tools as the layer between the EPICS control system and more functional analyzing tools and scripts, with SDDS protocol files as an intermediary.

The various tools are joined together via a scripting language, originally C shell, now Tcl/Tk. Most of the accelerator physicists can write Tcl/Tk scripts to diagnose problems or to solve new controls problems. Before new control strategies become operational, a Tcl/Tk-based tool is prepared for use by operations.

3 DEUTSCHES ELEKTRONEN SYNCHROTRON (DESY)

Shortly after commissioning the HERA (Hadron-Electron Ring Accelerator) storage ring it was obvious that some of the components developed for the cryogenic control system needed immediate performance and reliability improvements (the MKS-2 group is responsible for the cryogenic systems). One major problem was related to random crashes of one of the subsystems responsible for transfer and conversion of data between the in-house fieldbus called SEDAC and the cryogenic process control system. It was obvious that the replacement component should not only match the requirements for this special application but also open the door to a new standard of control applications tools. In 1993 EPICS was chosen because of its process control and record-oriented structure.

3.1 Conversion of Cryogenic System to EPICS

Along with EPICS, a new UNIX operating system environment (replacing VMS) and VxWorks real-time kernels (instead of iRMX) had to be learned within six months in order to replace the existing subsystem. From installation of the various software packages to the start of the first running period of an EPICS system at DESY took about eight months. The HERA cryogenic controls system must be extremely reliable because the superconducting magnets run 24 hours a day, every day for at least six months at a time. The first EPICS system met these stringent requirements and was considered an unqualified success.

Once the EPICS IOCs were in operation, it was possible to make use of the new freedom these systems provided the system designers and the operators. Online access to all the individual data was now possible as well as the chance to establish new ways of archiving process data for long-term storage. This led to the first new development for the EPICS system. An archive record, the archiver (still under VMS), and a viewer (based on IDL) were programmed to improve the quality of the archived data, storing on significant change instead of at a fixed time (i.e., every 60 sec). The analysis of problems was now possible with a much higher resolution than before. In addition, other EPICS tools such as display managers (first edd/dm, later medm) became so popular that the operators became involved in the process.

3.2 DESY Utilities

The next major use of EPICS at DESY was for monitoring utilities. The installed system of Siemans program logic controllers (PLCs) had improved the stability of HERA operations; however, it lacked adequate operator interfaces and tools like an archiver and alarm handler. In addition, the distributed nature of the installation necessitated that displays be distributed throughout the DESY site and into the halls of the 6.3-km-long HERA tunnel. EPICS was chosen to provide these services. The initial goal was to provide tools to the utilities group that would enable them to create their own environment for supervisory control. Students were trained to create hundreds of graphic screens. Special tools (scripts) were prepared to create EPICS databases from the PLC configuration databases. But the most difficult job was preparing the interface to the H1 network of the PLCs. Nearly half a man-year was spent writing the software to interface H1 and the common driver interface (CDI), which is DESY-developed software that provides an easy-to-use EPICS interface to various fieldbuses.

This system has grown continuously since it began operation. Even more important than this growth is the acceptance of the new control environment by the operators who, until EPICS arrived, had not had any contact with computer-based controls. The HERA water system is now monitored by engineers who use the archive viewer to supervise their water pumps and to tune their control loops.

Recently new systems went into operation to monitor the quality of DESY's main power supply. Three 10-kV transformers are monitored to catch any power glitch and freeze the current reading into an EPICS waveform record, which is then read by an archive process and stored in a raid array for later analysis.

In addition to the supervisory controls for the HERA water systems (based on H1 communication) another fieldbus, Profibus, is supported. Profibus is a German standard widely used for small- and medium-sized PLC applications. At DESY the Profibus-based PLCs control the air condition systems and small general-purpose systems. An EPICS interface was quickly implemented due to experience gained on the previous work.

3.3 Collaboration with Other Institutes

In addition to the utilities group, the machine controls and process controls groups were demanding more support from the MKS-2 group. The modular nature of the EPICS toolkit made it possible to distribute the various jobs to collaborators outside DESY like the Institutes of High Energy Physics in Protvino, Russia and Beijing, China. DESY staff trained the new collaborators in order to coordinate the work and reintegrate the new software into EPICS. Thanks to their excellent collaborative efforts, many new features were developed and are now in operation. One of these areas was the controls for the magnet power supplies for the Tesla Test Facility (TTF) which is the test bench for the next generation of linear colliders at DESY based on superconducting cavities.

To implement the TTF cryogenics controls, several functions necessary for process controls, such as the PID algorithm, had to be added or modified in the EPICS toolkit. All these modifications have been in operation for several months and are providing stable operations for TTF cryogenics.

3.4 ORACLE

The need to handle EPICS databases for multiple IOCs and different versions of databases for different EPICS releases was the driving force behind creation of a relational database for configuring and storing record information. An ORACLE database was designed to support both the old report structure and the new ASCII type of EPICS databases. The interface to configure individual records as well as record prototypes and prototype record groups was written in ORACLE-Forms to facilitate configuration of complete databases without requiring in-depth knowledge of the record structures. New records are created by merely supplying default values.

In operation since February 1997, this tool is the basis for a new phase of database generation at DESY since it is now possible to train people quickly. The database functionality was recently expanded to integrate configuration data for the alarm tools and the global database structure

following the idea that a record belongs to a *device* within a *location* of a *facility*. This type of information can also help generate information for the CDEV API from Jefferson Lab (see section 4), a feature that is attracting growing interest at DESY.

Cryogenic operators without any knowledge of the EPICS control architecture are now preparing the EPICS database, creating graphics, and providing the information to create sequence programs. This would not have been possible without the ORACLE-based record configuration tools.

3.5 Current and Future EPICS Projects

Another project that is nearly finished is providing EPICS services to PC-based fast ADC boards. This will allow access to PC-based information from any X-terminal or PC. In addition, the data stored in the ADC boards will be written to the central raid-array so that everyone can retrieve the data.

While the above-mentioned projects are relatively small, the next big project is in the planning phase: The original process control system for the HERA refrigerator will be phased out within the next two to three years. The replacement time schedule will be very tight. Thousands of lines of sequence programs will be converted into the native EPICS language of the sequencer; the internal structures of the DDC controls will be converted into EPICS records; and the functionality of the partly redundant processors will be analyzed and, if necessary, implemented for the IOCs.

The network is based on distributed Ethernet switches on an FDDI backbone. The Ethernet workstation connections are evolving towards switched 100baseT connections, again on the FDDI backbone.

EPICS installations at DESY are still growing because of the reliability and rich functionality of its software. The software distributed to the collaborators is very stable and can be used for production after making local modifications, and the collaboration provides help and support for all installed software. Common development efforts, such as the joint Argonne-DESY- CERN improvements to the medm graphical user interface, are made known to a wide audience.

4 THOMAS JEFFERSON NATIONAL ACCELERATOR FACILITY

Thomas Jefferson National Accelerator Facility (Jefferson Lab, formerly CEBAF) made a decision to adopt EPICS, following an extensive review of its control system, in 1993. At that time it was recognized that the in-house-developed TACL control system would have difficulty scaling up to full machine operation without extensive work, and it was decided that there would be immediate and long-range advantages in joining the EPICS collaboration [4]. Conversion began in the midst of accelerator commissioning.

As much of the existing control system (over 90%) was CAMAC-based, Jefferson Lab quickly implemented support for CAMAC based I/O devices, using the IEEE standard 758-1979 to define the interface between EPICS and an arbitrary CAMAC interface. This allowed the use of existing IEEE-compliant software device drivers available for parallel crate controllers. Further development of IEEE routines for a CAMAC serial highway interface allowed integration of the remaining I/O.

In migrating from TACL to EPICS, much of the design of the original system was preserved, and support for CAMAC interrupts was not needed (i.e., not used in the original system and not implemented in the conversion). The following year, LANL modified the CAMAC driver for the serial highway to add support for interrupts and for multiple serial highways (controllers). Jefferson Lab further extended CAMAC capability by adding support for multiple processors sharing a single serial highway for compute-bound configurations.

Over the past three years, a wide variety of hardware has been integrated into the control system, including GPIB, VME, and RS-232 devices. In most cases this has required writing additional device drivers; in approximately 15-20% of the cases device drivers were obtained from the EPICS community. EPICS has proved very flexible in this regard.

Because CEBAF was further into its commissioning schedule than APS, it was the first site to scale EPICS up to a single integrated system with tens of thousands of database records, hitting the 100,000 mark in 1994. With only a few minor problems, this more than tenfold increase in the size of the largest system was accomplished smoothly, giving every indication that an additional order of magnitude is achievable.

The current configuration of the control system contains 70 IOCs and 160,000 records providing an interface to 50,000 hardware I/O points. Compared with APS, Jefferson Lab has fewer single-board computers (by more than a factor of 2), and has consequently had to deal more with loading issues (this is partly budget driven, as this is the second control system for a relatively young site). Many problems were encountered in the first two years related to poor behavior of the EPICS software under low memory conditions. This has been solved with additional memory and some software improvements. Some remaining problems still under investigation may be related to CPU loading and/or network loading.

4.1 Accelerator Controls

The EPICS toolkit is quite flexible and is not specific to accelerator operations. It provides extremely flexible name-based control of a wide variety of hardware I/O points. Much of the work done by the controls group at Jefferson Lab has been in adding accelerator-specific software on top of, and alongside of, EPICS.

Due to the time pressure of ongoing accelerator commissioning in 1994-1995, it was necessary to rapidly pro-

duce a handful of accelerator applications to provide high-level accelerator setup and control. In particular, beam steering, linac optics setup, cavity phase cresting, and machine settings management were urgently needed. In order to provide operator-friendly tools, the Tcl/Tk scripting language and its graphics library were adopted. An interface between Tcl and the EPICS channel access network library was implemented, and this provided the basis for a rapid prototyping environment with name-based I/O and full graphics support [5].

Experience with this environment is extremely positive. While some of the more demanding applications, such as beam feedback and an online accelerator model, have been rewritten in C++ to improve performance and capability, additional Tcl- based tools continue to be written. The most recent work includes an automated turn on of the entire injector, written by the accelerator operations group (not software engineers), and an accelerator optics tuning package which looks at the Courant-Snyder invariant as a function of beam pathlength.

In creating these accelerator control applications, it was unavoidable that work done at other laboratories was to some extent duplicated. Several attempts were made to port software from other systems to Jefferson Lab, but most online applications had too many dependencies upon the other laboratories' control systems (non-EPICS). This was one of several motivations in creating a new framework for control system applications called CDEV [6]. Additional motivations include isolating high-level applications from the details of the low-level control system implementation, and organizing named I/O points into devices.

CDEV is a C++ and Java framework for developing device control applications and provides a definition of a virtual control system which is independent of local implementation. It allows applications to be shared between EPICS sites like Jefferson Lab and APS, and non-EPICS systems such as SLAC and CERN. One unique feature of CDEV is that it supports multiple protocols (concurrently, if desired), and in particular it includes support for the EPICS channel access protocol, allowing CDEV applications to access EPICS-named variables through a device-oriented API. CDEV has also been shown to map well onto the SLAC and CERN/PS control systems.

A small number of new CDEV applications have been written at Jefferson Lab (e.g., beam steering, a correlation measurement package), and a few of the EPICS applications have already been modified to call CDEV instead of Channel Access (stripTool converted at Jefferson Lab, alarm handler at APS, display manager an ongoing joint project between LANL and Jefferson Lab). As these tools are finished, they may be used immediately at CERN and other non-EPICS sites which have implemented the CDEV adapter.

It is hoped that as this suite of control system-independent accelerator applications grows, additional non-EPICS sites will contribute to creating a reusable suite of high-level accelerator control software [7].

For some of the non-hardware portions of the control system (e.g., on-line accelerator model, error log, and archive data log query processing), the "software bus" used in EPICS does not provide a sufficiently rich protocol, and Jefferson Lab has implemented an additional message oriented protocol under CDEV for these applications. The protocol is also implemented in Java, allowing Web-based applets full live access (typically restricted to read-only) to all control system data through a gateway program on the Web server.

A final significant area of accelerator-specific developments is a fast feedback system for beam energy and orbit corrections [8]. This system may eventually provide feedback at more than 2 kHz to fully suppress observed 60-Hz and 180-Hz noise. Feedforward is also under investigation. A prototype state space formalism-based tool has been implemented which executes at 60 Hz, with plans to scale this to 540 Hz in the next six months. These speeds are outside of the range at which EPICS is an appropriate solution, and the current implementation uses separate tasks running on the same processor as EPICS (a separate processor may be used for highest rates). These tasks are controlled and monitored through EPICS, allowing a uniform view of this mixed system. Early performance results are quite encouraging.

4.2 Physics Controls

EPICS is also being used in the control of the three major physics detectors at Jefferson Lab. The data acquisition system CODA is also accessible through CDEV, and joint development projects between the accelerator and physics divisions on the use of EPICS and CDEV are further improving software reuse within the extended EPICS community. The most recent development in this area is a new error message logging facility co-developed by these two teams.

5 KECK OBSERVATORY

The W. M. Keck Observatory operates the twin Keck 10-meter optical telescopes on the summit of Mauna Kea in Hawaii. Both telescopes have identical control systems, entirely implemented using EPICS [10].

The Keck I telescope was commissioned in 1992 using an in-house control system. This system was functional but had various problems and was hard to extend. When the Keck II telescope was funded in early 1994, it was decided not to propagate the Keck I system to the new telescope but instead to produce a new control system that would scale to the planned future use of the two telescopes for optical interferometry. Since EPICS had already been selected by the Gemini 8-meter telescopes project, and there was already interest in EPICS throughout the astronomical community, it made sense to consider it for the Keck II control system.

Various low-level design studies were carried out in 1994 and the formal decision to use EPICS was made at a design review in March 1995. Development proceeded rapidly (with much help from LANL staff) and Keck II saw "first light" in January 1996, becoming fully operational in October 1996. Keck I was retrofitted with the new control system in March 1997 and both telescopes are now (independently) running identical software versions.

Unlike the accelerators described elsewhere in this paper, the Keck EPICS systems have relatively small numbers of IOCs, records, and I/O points. Each telescope currently has three IOCs, with about 2000 records total per telescope, of which about 850 are attached to hardware. In contrast to many accelerator systems, there is little duplicated hardware and a lot of algorithmic code runs on the IOCs, both in subroutine records (including two extended versions of the subroutine record supplied by the Gemini project) and in custom records (thirteen custom records have been implemented in-house).

In addition, EPICS is being used for the Keck adaptive optics program, which will add two or three IOCs per telescope. Looking further into the future, it will also be used for coordinating the two telescopes and several outrigger telescopes as an optical interferometer.

5.1 EPICS for Telescope Control

The main telescope control servo runs at 40 Hz, a low enough frequency to permit standard EPICS record processing to be used for reading hardware and performing the necessary encoder and servo processing. This processing has many inputs and outputs and it was decided early on that it would be necessary to write custom records with many fields. (Even if the new genSub record had been available at the time, it is probable that the same number of custom records would need to have been written.)

The time interval between the timing pulse that latches the telescope encoders (and triggers EPICS record processing) and the velocity demand being written to the drives is termed the critical region. This critical region is required to be less than 1 ms. In order to achieve this specification, the custom records use precalculated coefficients to perform critical processing as fast as possible and then forward link to the next record in the chain. Once critical processing is complete, the coefficients for the next iteration are calculated. As downstream records finish processing, the records that forward linked to them continue with their noncritical processing.

The main telescope control described above accounts for five of the thirteen custom records. Five more are used for the "pointing code," the rigorous astrometry that determines where the telescope should be at a given time. The three remaining records handle some standard conversions that, while logically appropriate for subroutine records, require more inputs and outputs than can be handled by such records.

In the early days, several LANL contractors wrote EPICS device and driver support for devices not already supported by EPICS. One generally-applicable product which has resulted is drvAscii/sioSup, which uses a C printf()-like format string in a record's PARM field to describe a prompt and an expected response. This allows control of many serial devices without having to write any C code (and is similar in concept to the ANL generic GPIB support).

5.2 Use of Standard EPICS Tools

Nearly all databases are generated using Capfast; there are still one or two snippets that use the LANL DCT tool or which are hand-edited in the ANL .db format. Experience with Capfast has in general been positive: the end result is wonderful but the drawing tool causes some gnashing of teeth.

At first, the LANL display manager was used extensively. However, use of Tcl/Tk and the commercial dataViews package—interfaced via the Keck-specific KTL library (described below) to Channel Access—for user interface screens is increasing. DM continues to be used primarily for engineering screens.

The EPICS alarm handler and archiver have been used extensively, and both have been invaluable. There have been some user acceptance problems with the alarm handler but this is probably due to insufficient attention being given to the configuration files. The m4 macro processor is used to combine subsystem alarm handler configuration files into a single telescope-wide file.

5.3 Locally-Developed Tools

Keck is only partially an EPICS shop. Owing to the freedom afforded the university groups who build its instruments, several different messaging systems and control APIs are in use. In order to alleviate this problem, the Keck Task Library (KTL) [11] API was defined in 1991. KTL shares some features with CDEV in that the API is independent of the underlying services and an internal API permits run-time activation of shareable libraries containing service-specific code. Like CDEV and Channel Access, all I/O is name-based, an object being identified by a service name (which identifies the shareable library) and a service-specific keyword name. Keck requires all telescope and instrument subsystems to provide a KTL library; as with CDEV, this permits KTL applications (mostly GUIs) to use a single API to control multiple heterogeneous underlying systems. Recent GUI development has used a Tcl/Tk/KTL extension called KTCL.

The EPICS archiver can log binary data at high rates and is used for collecting 40-Hz servo data. This data is imported using arrIDL (described below) and analyzed using IDL. The archiver can also log ASCII data at specified intervals but this produces files which are verbose and hard to handle. To address these difficulties, a simple archiver (EZAR) was written. EZAR can log both events and periodically sampled data in a comma-separated ASCII format. It is useful where ease of access to archived data is more important than efficient collection of the data.

The ANL BURT tool was evaluated as a means of loading and restoring calibration data. However, it seemed more sophisticated than warranted by Keck requirements and instead a simple PVLOAD tool was written. PVLOAD parses a file of EPICS channel assignments (including support for array-valued fields) and then uses Channel Access to set the values. A complementary PVSAVE tool is planned but has not yet been implemented.

Sequence programs are used for managing complex interactions such as subsystem initialization and fault handling. They are also used, at a higher level, for managing groups of subsystems. When implementing the latter, it was necessary to add a feature to the sequencer program to permit queueing of Channel Access monitors. This change will be made available to the EPICS Consortium.

Finally, IDL is used extensively for the analysis of engineering data. Accordingly, an EPICS extension to import EPICS archiver data into IDL has been written. This extension, arrIDL, is based on the C code from the LANL arrtk extension.

6 CONCLUSIONS

As shown in Table 1, EPICS has been successfully used to implement controls for medium to large facilities. APS, TJNAF, and Keck are using EPICS for most of their controls. DESY is using it mainly for cryogenic controls and facility monitoring.

Each site uses the same EPICS core, but has implemented special low-level software to communicate with hardware and/or fieldbuses connected to other control systems. The EPICS design makes it easy to add such support without having to modify the EPICS core.

Similarly, each site has developed high-level software that exists above the EPICS core software; the existence of the Channel Access client library has made this possible.

Table 1: Control System Sizes

	APS	DESY	TJNAF	Keck
# of Workstations	~10	~8	~10	~6
# of IOCs	165	31	70	6

8 REFERENCES

[1] L. R. Dalesio et al., "The experimental physics and industrial control system architecture: past, present, and future," Proc. of ICALEPCS '93, Berlin, Germany, NIM A 352, pp. 179-184, (1994).

[2] J. O. Hill, "Channel Access: A Software Bus for the LAACS," Proc. of ICALEPCS '89, Vancouver, British Columbia, Canada, NIM A 293, pp. 352-355 (1990).

[3] L. R. Dalesio et al., "Distributed Software Development in the EPICS Collaboration," Proc. of ICALEPCS '95, Chicago, Illinois, pp. 360-366 (1997).

[4] W. A. Watson et al., "The CEBAF accelerator control system: migrating from a TACL to an EPICS based system," Proc. of ICALEPCS '93, Berlin, Germany, NIM A 352, pp. 118-121 (1994).

[5] J. van Zeijts, "Rapid Application Development Using the Tcl/Tk Language," Proc. of the 1995 PAC, pp. 2241-2243 (1996).

[6] J. Chen et al., "CDEV: An Object-Oriented Class Library for Developing Device Control Applications," Proc of ICALEPCS '95, Chicago, Illinois, pp. 97-104 (1997).

[7] W. Watson, "A Portable Accelerator Control Toolkit," these proceedings.

[8] M. Chowdhary et al., "A Fast Feedback System for CEBAF," Proc. of ICALEPCS '95, Chicago, Illinois, pp. 429-434 (1997).

[9] W. Lupton, "Software Infrastructure for the Keck II Telescope," in Telescope Control Systems, Proc. of SPIE, Vol. 2479, pp. 140-151 (1995).

[10] W. Lupton, H. Lewis, K. Tsubota, A. Honey, S. Quady, "The Keck II Telescope Control System," Optical Telescopes of Today and Tomorrow, Proc of SPIE 1996 (to be published).

[11] W. Lupton, A. Conrad, "The Keck Task Library (KTL)," ADASS '92, PASP Conference Series 52, pp. 315-319 (1993).