EPICS Architecture*

L.R. Dalesio (LANL), M.R. Kraimer (ANL), A.J. Kozubal (LANL)
Mail Stop H820, Los Alamos National Laboratory Los Alamos, New Mexico, 87545
Argonne National Laboratory Argonne, Illinois , 60439

Abstract

The Experimental Physics and Industrial Control System (EPICS) provides control and data acquisition for the experimental physics community. Because the capabilities required by the experimental physics community for control were not available through industry, we began the design and implementation of EPICS. It is a distributed process control system built on a software communication bus. The functional subsystems, which provide data acquisition, supervisory control, closed loop control, archiving, and alarm management, greatly reduce the need for programming. Sequential control is provided through a sequential control language, allowing the implementer to express state diagrams easily. Data analysis of the archived data is provided through an interactive tool. The timing system provides distributed synchronization for control and time stamped data for data correlation across nodes in the network. The system is scalable from a single test station with a low channel count to a large distributed network with thousands of channels. The functions provided to the physics applications have proven helpful to the experiments while greatly reducing the time to deliver controls.

I. INTRODUCTION

EPICS is currently being co−developed by the Accelerator Technology controls group at Los Alamos National Laboratory and the Advanced Photon Source controls group at Argonne National Laboratory. Its architecture provides a wide range of functionality, rapid application development and modification, and extensibility at all levels to meet the demands of experimental physics. The hardware and software for each functional subsystem was seleceted to meet these requirements. The subsystems are: the Distributed Database, the Display Manager, the Alarm Manager, the Archiver, the Sequencer, and Channel Access [figure 1]. Technology changes, to be incorporated, will further extend the performance of the EPICS subsystems. Programs at various installations have applied EPICS successfully with no modifications to the software, demonstrating adequate performance, functionality and extensibility.

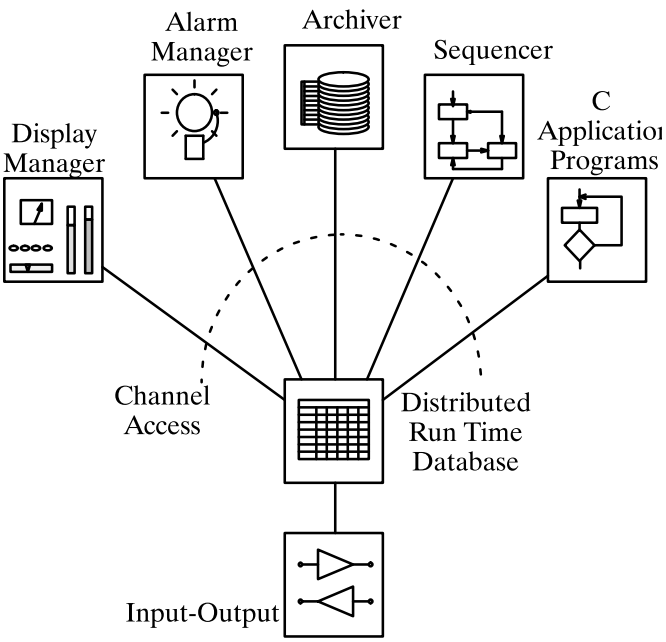

Figure 1. Functional subsystem layout of EPICS.

II. THE DISTRIBUTED DATABASE

A distributed database is used to provide local control. A portion of the distributed database is loaded in each I/O Controller (IOC). The database provides data acquisition, data conversion, alarm detection, interlocks, and closed loop control[1][2].

The IOCs, in which the database segments are loaded and executed, consist of a VME or VXI backplane, a 68020 CPU running the vxWorks real−time kernel, an ethernet connection, and an optional complement of I/O [Figure 2]. The I/O buses supported are: VME, VXI, Allen−Bradley Industrial I/O, GPIB, BITBUS, and CAMAC. The benchmarks for the database scan tasks show the ability to close 4,000 analog loops per second leaving about 40% of a 68020 available for the sequencer and channel access

* Work at LANL supported and funded under the Department of Defense, US Army Strategic Defense Command, under the auspices of the Department of Energy.

Work at ANL supported by U.S. Dept. of Energy, Office of Basic Energy Sciences, under Contract No W−31−109−ENG−38.

services. The maximum periodic repetition rate currently available is 60Hz. Using interrupt on end−of−conversion hardware allows for rates higher than 60Hz. The fully distributed database allows I/O controllers to be easily added to take on additional loading.

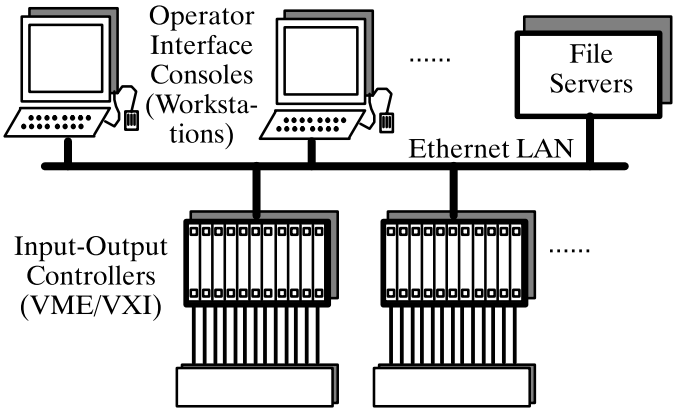

Figure 2. Physical subsystem layout of EPICS.

The database for each I/O Controller is configured offline, using the Database Configuration Tool (DCT). DCT is a menu based configuration tool written in 'C' using the Curses library. Channels are added and modified either interactively through DCT or through a text file. The commercial database package, PARADOX, has been used to produce text input to DCT. A variety of reporting facilities to help document the application is also provided in DCT. The database successfully provides data acquisition, conversion, notification, and closed−loop control without any applications programming.

The scan tasks and drivers supporting the distributed database can be easily extended. The device driver library is continually expanding to support new I/O buses and additional modules on existing I/O buses.[3] A database record is provided which invokes 'C' subroutines in the context of the scan tasks. New record types can be added to support complex device types or new algorithms. Outside the realm of the database, the IOC can be extended by writing sequence programs or tasks running directly under vxWorks that interface to the database using channel access.

III. THE DISPLAY MANAGER

The **display manager** provides an interface between the operator and the control system[4]. The display manager is capable of monitoring or modifying any field in any database distributed throughout the network via the channel access communication bus. The display manager allows the application engineer to build display hierarchies that use the windowing capabilities of modern workstations and personal computers.

The display editor and display manager run under UNIX and the X−window system. Running the display manager on the SUN IPC, display call−up of 1,000 static graphic elements and 100 dynamic graphic elements has been benchmarked at less than two seconds with an update rate of 10,000 monitor elements per second. The SPARC station I and II and graphics accelerators can be used to provide even higher performance. Using a client that is not in the same machine reduces the performance of the client by around 3%. The display manager and editor have been run on X servers on the VAX/VMS under Decwindows and the 80386AT/MSDOS under MicroSoft Windows 3.0 using the HCL−eXceed/W X−server. EPICS imposes no limitation to the number of operator interfaces that are on the control network.

The screens for display are configured using a graphic display editor (EDD). EDD can be used through text input or interactively. The X−based graphics editor provides a wide range of helpful functions for producing professional displays quickly and easily. Non−technical people have been trained to produce displays in less than 1 hour. Process displays are produced in hours. The displays are automatically connected to the operational parameters by name. No knowledge of the I/O location is needed at display configuration. Display elements are predefined for monitoring and modifying any parameter on the network, producing display hierarchies, strip charting, and plotting synchronous events. A single display can be built, then invoked many times for different instances of a similar device or subsystem. For instance, a screen can be built for one vacuum station and invoked for each vacuum station, greatly reducing the creation and maintainence of the displays. The ease of use has greatly reduced the time to create and maintain displays.

Many devices have been added to the display manager. In addition, X applications can be written to run in conjunction with the display manager. Using the X windows environment also allows users to take advantage of other packages that are available commercially and through EPICS.

IV. THE ALARM MANAGER

The **alarm manager** is used to create fault trees for alarm presentation to the operator[5]. The warning and alarms are configured in the database. The alarm manager monitors and groups these alarm conditions into a fault tree for presentation to the operator. On selecting the highest group, the alarm manager automatically traverses the fault tree until arriving at the unacknowledged alarm or the first ambiguous branch. The groups can be set up to disable on given machine parameters such as operational mode. The alarm manager provides a useful tool for steady state operation, allowing the operator to react to excursions from normal operations.

The alarm manager currently runs under UNIX. The interface to the control system is channel access. The alarm manager can be run on many workstations where each instance can run any alarm configuration.

The alarm hierarchy files are currently configured using a text editor. The format of the text files is predefined. The configuration files are converted to a binary file at initialization. The system engineer who sets up the alarm configuration can put in guidance for the operator. A screen in the display manager can also be invoked from the alarm manager to aid the operator in handling the exception.

V. THE ARCHIVER

The **archiver** is used to collect data from the control system to disk. Data collection for each data request is initiated and discontinued based on time or the condition of any parameter in the distributed database. More than one request can be active on each workstation. The archive data retriever allows users to select channels for retrieval and examination. Data that is older than one minute is available through the archiver during operation. The archived data can be plotted against time or as functions of other channels. The plots can be converted to postscript and printed, or they can be converted to encapsulated PostScript and included in word processors like Interleaf. Data can also be formatted as input to either LOTUS or EXCEL. The archiver provides a program interface into archived data as well. This feature allows the user to write data analysis codes that directly interface to the archived data. The methods for accessing archived data facilitate data analysis for experimenters and application engineers.

The archiver runs under UNIX and can collect 5,000 channels per second. The archiver uses channel access to access parameters of interest on the network. Many instances of the archiver can be run on the control network.

The archive requests are currently configured using a text editor. The format of the text file is predefined. The configuration file is converted to a binary file at initialization.

VI. THE SEQUENCER

The **sequencer** executes state programs[6][7], which run on the I/O controllers. The user can build state programs using the state notation language that are compiled and loaded into the I/O controller at initialization. The state notation language takes the I/O controller through modal changes. It can be used to switch between different operational modes (i.e. warm up, ready, operate, shut down) or to handle exceptions (i.e. vacuum leak detected – go to safe state). The state notation language has a syntax designed to make implementing state transitions easy. The state notation language features transitions for time and events. Connections to the database on the network are handled by assigning a database name to a variable. 'C' statements can be placed in the state program. Using the state notation language alleviates the need to know channel access or vxWorks internals, reducing the source code by approximately 75%.

VII. CHANNEL ACCESS

Channel access is a 'software bus' provided to communicate between various EPICS subsystems[8]. It allows the system components and user extended components to perform channel connections, gets, puts, and monitors on any field of the distributed database using the TCP and UDP protocols. Channel connections are performed by connecting a unique channel name to the IOC containing the channel. Channel access provides notification to its clients when a connection is broken and another notification when it reconnects. This connection management is used to keep all subsystems informed of the status of the other IOCs, on which it may depend for parameter information.

Channel access services are available under UNIX on the 68xxx and SPARC series processors, under VMS, and under vxWorks on the 68xxx based processors. The interface routines are in 'C'. These routines are used by the I/O controller, display manager, archiver, alarm manager, and the sequencer for communication.

VIII. EVENT SYNCHRONIZATION

EPICS is equipped with a timing system that provides event synchronization across the network. Through a hardware/software solution, events can be accurately correlated in distributed I/O controllers. These events provide data with time stamps that are used by the archiver and operator interface to accurately analyze synchronous data sets in real time and for historical data. In addition, the timing system provides the ability to process data on sub−harmonics of the base rate of the physics machine. This allows data taking to occur on the same Nth event across IOCs on the network. The timing system is designed to recover from communication failures and IOC initialization during operation. The timing system has undergone rigorous testing before being used on the Ground Test Accelerator.

IX. COMMERCIAL IMPROVEMENTS

Upgrades to the EPICS hardware and software are becoming commercially available, and require little effort to integrate. The IOC will soon upgrade from the 68020 to the 68040 processor. An FDDI backbone for the distributed network is currently available, but has not been required at any of the EPICS installations. Version 5.0 of vxWorks provides performance enhancements to the network communication, task switching, and semaphore handling. The standards upon which EPICS is built, are continually improving in performance.

X. EPICS SOFTWARE EXTENSIONS

Upgrades to the EPICS software are underway for making the system easier to configure and diagnose, and provide more functionality. Upgrades are being implemented to provide interactive X−based configuration tools for the database, archiver, and alarm manager. Extensions are planned for the display manager to include polygons. The display editor is being extended to include object grouping. Devices and database record types will be added as required.

XI. CONCLUSION

EPICS has been successfully applied to many applications[9−13]. The tools that are available have been used to provide data acquisition, supervisory control, closed−loop control, sequential control, and data archiving. All functional requirements have been met with the existing architecture. Extensions are underway to make the configuration of EPICS easier and further improve the throughput. The EPICS collaboration continues to provide useful tools for producing better results with less expense to each individual program.

XII. ACKNOWLEDGEMENTS

The initial design and implementation of EPICS was done by the Controls Group of the Los Alamos Accelerator Technology Division. In it's previous incarnations EPICS was known as the Ground Test Accelerator Control System (GTACS) and the Los Alamos Accelerator Control Systems (LAACS). EPICS is currently being co−developed by the Accelerator Technology controls group at Los Alamos National Laboratory and the Advanced Photon Source controls group at Argonne National Laboratory. Specification, design, implementation, verification, and configuration control has been performed by the following group: J.B. Anderson (ANL), M.D. Anderson (ANL), B.C. Cha (ANL), R.A. Cole (LANL), L.R. Dalesio (LANL), C.E. Eaton (LANL), B.A. Gunther (LANL), J.O. Hill (LANL), D.M. Kerstiens (LANL), M.J. Knott (ANL), A.J. Kozubal (LANL), M.R. Kraimer (ANL), F.R. Lenksus(ANL), W.P. McDowell (ANL), G. Slentz (LANL), M.E. Thuot (LANL), R.C. Zieman (ANL). EPICS continues to evolve as new perspectives are provided and new challenges are met. The evolution is directed at providing better control tools to the experimental physics community.

XIII. REFERENCES

[1] Dalesio, L. R., "The Ground Test Accelerator Database: A Generic Instrumentation Interface," in *Accelerator and Large Experimental Physics Control Systems*, D. P. Gurd and M. Crowley-Milling, Eds. (ICALEPCS, Vancouver, British Columbia, Canada, 1989), pp. 288—291.

[2] Dalesio, L. R., "The GTA Control System Database: Configuration, Runtime Operation and Access," in *Proceedings of the 1989 IEEE Particle Acelerator Conference*. (PAC, Chicago, Il., USA, 1989), pp. 1693—1694.

[3] Arnold, N.R., Daly, R.T., Kraimer, M.R., McDowell, W.P., "I/O Subnets for the APS Control System," submitted to Particle Accelerator Conference, San Francisco, California, May 6—9, 1991. .

[4] Hill, J., Kerstiens, D., and Dalesio, L.R., "A Virtual Control Panel Configuration Tool for X−Window System, " submitted to ICALEPCS, Tsukuba, Japan, Nov. 11−15, 1991.

[5] Kraimer, M.R., Cha, B.C., and Anderson, M.D., "Alarm Handler for the Advanced Photon Source Control System," submitted to Particle Accelerator Conference, San Francisco, California, May 6—9, 1991.

[6] Kozubal, A. J., L. R. Dalesio, J. O. Hill, and D. M. Kerstiens, "A State Notation Language for Automatic Control," Los Alamos National Laboratory report LA−UR−89−3564, November, 1989.

[7] Kozubal, A. J., "Automation From Pictures: Producing Real Time Code from a State Transition Diagram," submitted to ICALEPCS, Tsukuba, Japan, Nov. 11−15, 1991.

[8] Hill, J. O., "Channel Access: A Software Bus for the LAACS," in *Accelerator and Large Experimental Physics Control Systems*, D. P. Gurd and M. Crowley-Milling, Eds. (ICALEPCS, Vancouver, British Columbia, Canada, 1989), pp. 288—291.

[9] Atkins, W., "Using the EPICS Sequencer Tool to Automate the GTA Vacuum System, " submitted to Particle Accelerator Conference, San Francisco, California, May 6—9, 1991.

[10] Wilson, William L., Marcus W. May, and Andrew J. Kozubal, "Rapid Development of a Measurement and Control System for the Advanced Free-Electron Laser," submitted to Thirteenth International Free-Electron Laser Conference, Santa Fe, New Mexico, August 25—30, 1991.

[11] Knott, M.J., McDowell, W.P., Lenkszus, F.R., Kraimer, M.R., Arnold, N.R., Daly, R.T., Gunderson, G.R., Cha, B.K., and Anderson, M.D., "The Advanced Photon Source Control System," submitted to Particle Accelerator Conference, San Francisco, California, May 6—9, 1991. .

[12] Eaton, C.E., Little, C.K., Martz, V.E., Glasscock, R.B., Carlson, G.W., "GTA RF Control," Los Alamos National Laboratory report LA−UR−89−948, November, 1989.

[13] Schindler, J., Bemis, M., Gonder, J., Oberg, D., Wright, R., "RF Control and Diagnostics for the Average Power Laser Experiment (APLE)," submitted to the 13th International Free−Electron Laser Conference, Santa Fe, New Mexico, August 25−30,1991