Overview of the Experimental Physics and Industrial Control System: EPICS

Steve Lewis
Lawrence Berkeley Laboratory

A note for current and prospective users.

Introduction

The collaboration

The Experimental Physics and Industrial Control System (EPICS) is three things at once: an architecture for building scalable control systems; a collection of code and documentation comprising a software toolkit; and a collaboration of major DOE labs and industry. EPICS is primarily the work of the Accelerator Technology (AT-8) group at Los Alamos (LANL) and the Advanced Photon Accelerator (APS) at Argonne (ANL). The Computer Systems Group at Berkeley (LBL) and the Continuous Electron Beam Facility (CEBAF) have recently joined. More than a dozen other laboratory, university, and industrial facilities use EPICS. Three industrial partners have signed technology transfer agreements. Thus EPICS comprises a full collaboration, with shared, coordinated, tested and documented releases by LANL, ANL, LBL, and CEBAF; a supported, but no-cost "product" by some; and a collection of source code to be used as a starting point by others. For those requiring support, it is supplied as work-for-others among DOE labs, or through regular purchase agreements (but no license cost for DOE labs) with the industrial value-added partners. EPICS is distributed in source form. This ensures that every site has full control of its future locally, whether a full collaborator or not.

Current EPICS systems range in scale from single crate/single workstation sites with a few hundred "channels," up to 100 crate/30 workstation sites with about 30,000 channels. Before it's termination, the Superconducting SuperCollider Laboratory would have extended EPICS to the scale of 2000 crates, 100 workstations and approximately 1,000,000 channels.

Standards

EPICS relies on standards at every layer. This achieves the goals of: no dependence on specific hardware vendors; no assembly code or other dependence on specific instruction sets; the ability to upgrade incrementally both within a layer and across layers to take advantage of new equipment and vendors; seamless performance increase; and the ability to make cost/performance trade-offs with no software penalty. Use of public-domain software is strongly emphasized to minimize costs; to date, within the 'core,' only the operating systems are purchased commercially.

Architecture

Architecturally, EPICS embodies the "standard model" of distributed control system design. The most basic feature of EPICS is that it is fully distributed: It requires no central device or software entity at any layer. This achieves the goals of easy scalability, of robustness (no single point of failure) and incremental operation and upgrade. Thus it is natural for parts of a total EPICS system to be in production, while other parts are in development, and yet other parts are shut-down. Of particular note is the absence of any run-time entral name-server; EPICS uses broadcasting to implement a "discovery" process for name-to-location resolution.

Hardware Layers

EPICS comprises three physical layers and five software layers. The physical front-end layer is built from VME/VXI hardware crates, CPU boards, and I/O boards. The latter drive the hardware plant directly or through a variety of standard field buses such as IEEE-488 ("GPIB"), Bitbus, RS-232, and Allen-Bradley 1771. The CPU boards are generally from the Motorola 680X0 series and run the VxWorks real-time kernel. The physical back-end layer is implemented on popular workstations from Sun, HP, and others, running Unix. (VAX/VMS is also supported.) These layers are connected by the network layer, which is any combination of media (such as Ethernet, FDDI, ATM) and repeaters and bridges supporting the TCP/IP Internet protocol and some form of broadcast or multicast.

Client Layer

The software layers start with the paradigm now called "client-server." The client layer usually runs in the workstation physical layer and represents the top software layer. Typical generic clients are operator control screens, alarm panels, and data archive/retrieval tools. These are all configured with simple text files or point-and- click drawing editors. Other generic clients such as the WingZ spreadsheet, Mathematica, the PVWave visualizer, the Tcl/Tk graphical scripting language, and a NODAL interpreter are also EPICS clients. Of special note is a client called the Sequencer, which uses a high-level language called the State Notation Language to implement cooperating finite-state machines. A very useful client is the knob manager: It allows traditional tuning by operators through a standard Sun Dialbox.

EPICS clients are very high in performance. For example, operator screens with 1000 objects are brought up in less than one second and can update dynamically about 5000 objects/sec. The alarm panel can react to 1000 changing alarm conditions/sec. The archiver can sustain 5000 transactions/sec to disk. All of these levels are achievable on a the lowest cost workstations. The operator screens provide both mimic diagrams (sometimes called synoptic displays), tabular data, and simulated "meters," "buttons," and "sliders."

Channel Access Layer

The second software layer that connects all clients with all servers is called "channel access" (CA). Channel access hides all the details of the TCP/IP network from both clients and servers. CA also creates a very solid "firewall" of independence between all client and server code, so they can run on different processors, and even be from different versions of EPICS. CA mediates different data representations, so clients and servers can mix ASCII, integral, and floating (as well as big- endian and little-endian) types where each uses its natural form.

The design of CA provides very high performance, allowing throughput rates on the order of 10,000 "gets" or "puts" per second under heavy load, yet minimizing latency to about 2 msec under light load. If the medium allows it, many clients and servers can simultaneously sustain these rates. Since EPICS is a fully-connected and flat architecture, every client and every server make connections with no "relay" entities, so there are no bottlenecks beyond the physical limits of the medium. CA also uses a technique called "reporting by exception" or callback. Once a client has expressed an interest in certain data to a server, the server notifies the client only when the data changes. This not only minimizes traffic, but signals both the health of the server and the freshness of the data.

All data carry time-stamps. Any client can obtain data with guaranteed matching time-stamps from multiple servers. Data also carry validation information based on both the quality of connection, and validity down to the hardware layer as explained later. Thus, a critical client implementing a global feedback loop can assure it is operating only with fully validated data.

Server Layer

The third software layer is the server layer. The fundamental server is the channel access server that runs on the CPU embedded in every VME/VXI crate. It insulates all clients from the database layer below. This server cooperates with all channel access clients to implement the callback and synchronization mechanisms already described. Note that although channel access clients are typically independent Unix programs that call channel access routines through a shared library, the channel access server is a unique EPICS task. One copy of it runs on every CPU board.

Database Layer

The fourth, or database layer, is really the heart of EPICS. Using a Unix tool, the database is described in terms of “records.” About 50 record types exist (and can be extended) for performing such chores as analog input and output; binary input and output; building histograms; storing waveforms; moving motors; performing calculations; implementing PID loops, emulating PALs, driving timing hardware; and other tasks. Records that deal with physical sensors provide a wide variey of scaling laws, allow smoothing, and accept independent hysteresis parameters for display, alarm, and archive needs.

The fundamental entity in the EPICS database is the “channel.” A channel is a path to a record in the database. Besides its name, a record has a value, and perhaps other attributes such as units, maximum/minimum, and so forth. The record name and attribute is the only way that CA connects a client to a server.

Record Activity

Record activity is initiated in several ways: from VME hardware interrupts, from software “events” generated by clients such as the Sequencer, when fields are changed from a “put,” or using a variety of periodic scan rates. Records support a great variety of data linkage and flow control, such as sequential, parallel, and conditional. Data can flow from the hardware level up, or from the software level down. Provision is made for local and global simulation, in which additional records stand in for hardware input/output records. Records validate data passed through from hardware and other records as well as on internal criteria, and can initiate alarms for uninitialized, invalid, or out-of-tolerance conditions. Although all record parameters are generated with a configuration tool on a workstation, most can be dynamically updated by channel access clients, but with full data independence.

Of course, full customization is possible by defining new record types, and a bread-board capability exists in the form of a generic "subroutine" record, using C code. This is one of the few places where new C code often appears. Largely, new EPICS implementations require toolbox programming, rather than C programming. All of EPICS is coded in C.

Linkages between records are not restricted to a physical CPU board, but can span multiple boards in a crate, and multiple crates. Various DOE labs have produced a few custom VME boards to enhance multi- crate synchronization to the sub-microsecond level. Record processing activity is highly efficient, achieving typical rates of nearly 4000/sec on a 33MHz 68040 class CPU.

Device Layer

The fifth, bottom layer of software is the device driver layer. A large and rapidly growing list of drivers has been written for many popular VME and VXI interface boards. Drivers are written in C, and the collaborating labs regularly collect new drivers from all the EPICS partners and redistribute them.

Summary

In summary, EPICS provides a software toolkit for implementing control systems following the "standard model" paradigm. DOE Labs, industrial partners, and other users augment the toolkit. EPICS control systems can achieve modularity, scalability, robustness, and high speed in hardware and software, yet remain vendor and hardware-independent. EPICS provides seamless integration of several data acquisition bus standards. The software development environments for intelligent local controllers and workstations can be identical. Good documentation, training, and support are available. Standard systems can be configured with text editors and other simple tools, yet full customization is available to sophisticated sites.