EPICS: RECENT DEVELOPMENTS AND FUTURE PERSPECTIVES*

R. Lange†, BESSY, 12489 Berlin, Germany
J. B. Anderson, A. N. Johnson M. R. Kraimer, W. E. Norum, ANL, Argonne, IL 60439, USA
L. R. Dalesio, J. O. Hill, LANL, Los Alamos, NM 87545, USA

Abstract

EPICS (Experimental Physics and Industrial Control System) has become a widely used control system toolbox in the accelerator and astronomy communities. New projects are continuously pushing the collaboration to incorporate new technologies, extend limitations and improve the system design. This paper describes recent key developments and presents development steps planned for the near future.

1 INTRODUCTION

1.1 The EPICS Collaboration

The group of institutes and companies that form the EPICS collaboration [1] has been growing constantly during the last years: today it consists of more than 140 collaborators from 19 countries. About ten of these institutes, the main users, are also contributing most of the development work.

Table 1: EPICS Licensees by Region

North America	86	China	6
Europe	29	Asia (other)	3
Korea	8	Australia	2
Japan	7	South America	1

The ever growing still always outdated list of projects that use EPICS for their control systems shows that about half of the projects are accelerators, while the other half is spread almost evenly between detectors, telescopes and commercial projects.

1.2 Growing Demands

A user community of that size places a high responsibility onto the developers: code changes have to be introduced very carefully and tested thoroughly on all available platforms, design improvements have to be discussed and coordinated with extraordinary diligence. All necessary measures have to be taken to avoid developments that introduce incompatibilities or side and after effects which would get multiplied by the number of users. On the other hand the growing community also increases the pressure to utilise new technologies and programming languages. In order to match the demands of new collaborators and projects more and newer hardware platforms have to be supported, existing limitations have to be extended or removed, and new reliable releases have to be shipped frequently.

2 PORTABILITY

Originally EPICS was designed to run on VME front-end systems using VxWorks as the underlying real time target operating system for the input output controller (IOC). Thus VxWorks and VME specific calls were scattered over the code, and the target system was extensively using VxWorks specific features.

An increasing number of projects, especially those hosted by universities as well as small installations, found acquiring the necessary VxWorks development and run-time licenses being well outside the tight limits of their budget. Support for additional host and target systems, namely from the public domain, became highly desirable.

2.1 The OSI Layer

An Operating System Independent layer was introduced into iocCore (the EPICS target system), completely encapsulating operating system dependent functions and resources, such as semaphores, threads, sockets, timers, and a symbol table containing function and variable names. [2,3] All hardware-specific support was unbundled (see also Section 3.1 below) and is now being built as separate modules. The EPICS build system was restructured to decouple operating system, architecture, and compiler specifics: the IOC components are now built targeting any supported OS, architecture and compiler permutation. So today the target iocCore runs on recent versions of VxWorks, RTEMS, Solaris, Linux, HPUX, Darwin and Windows.

A number of projects have chosen to use RTEMS [4] as the target real time operating system for their EPICS IOCs. [5] It has been shown that RTEMS provides a real time behaviour that is comparable to VxWorks [6], which makes it an interesting and promising low-cost alternative to VxWorks.

2.2 Host-based IOCs

As a side effect of the OSI layer introduction the EPICS iocCore now runs on typical workstation or desktop systems, its multiple threads usually being executed within a single system process. For these host-based IOC systems the functionality of the VxWorks target shell has been integrated into the iocShell, which was developed into a full-fledged replacement that even exceeds the abilities of its predecessor in some aspects. [7]

*Work supported by US Department of Energy under Contract Nos. W-7405-ENG-36 (LANL) and W-31-109-ENG-38 (ANL), the German Bundesministerium für Bildung und Forschung and the Land Berlin (BESSY).
†Ralph.Lange@bessy.de

Depending on certain TCP/IP implementation details, many of the supported operating systems even allow running multiple iocCore processes on a single machine using the same IP address and UDP port number. This has proven to be very useful in situations where databases that are not directly hardware-related can be started and stopped on demand.

In other applications commercial libraries and systems that are limited to a certain OS (typically Windows) can now be interfaced to in a very efficient, straightforward way. This has been shown for the connection to OPC Servers [8] and LabVIEWS systems [9].

For systems that do not need hard real time capability, support for a number of PCI based interface cards has been developed allowing inexpensive PC hardware running Linux to be used as IOCs.

3 CODE MANAGEMENT

An ever increasing number of source code lines and modules was making it very hard to apply changes and do the necessary tests. Preparing and shipping new releases became such a time-consuming procedure that steps had to be taken to ensure that the release process did not block developments.

Tracking known problems and the steps to their solution was another field that required some managing effort, as developers' mental notes did not always offer the necessary transparency to the release process.

3.1 Code Unbundling

During the porting process described above, all non-essential and hardware specific support modules were removed from EPICS Base and modified for installation as independent modules. Responsibility for these modules was shifted from the core development group to other developers within the collaboration.

This effort had two major effects:

1. Hardware dependent code was moved out of Base, which was thereby reduced to the core database and communication engine.
2. EPICS Base can be developed, tested, and released independently without the need for immediate testing of all support modules.

3.2 Release Procedure

Formal schemes, procedures, and checklists for releasing a new version are currently being worked on. This development is aimed at making test and shipment of new EPICS Base releases a more manageable task.

3.3 Error Tracking

A web-based bug reporting and tracking database has been set up at the central EPICS documentation web site. [10] The core developers have started using it successfully on a regular basis, while users from the collaboration still prefer the classical way of writing mail to the Tech-Talk mail exploder, followed by core developers creating entries in the database as necessary.

4 LICENSING

While EPICS was available only to non-profit organisations under a collaboration agreement or under a paid license for the longest part of its history, the licensing policy has been changed recently. Current versions of EPICS Base and related components are available under licenses that are similar to open source definitions.

While these changes had little effect on the existing network of collaborating institutes, it opened completely new ways to create complete subsystem solutions and new tools for the EPICS toolbox.

4.1 Vendor-Supplied EPICS Subsystems

The Swiss Light Source has successfully acquired complete system parts from external vendors – including the EPICS controls for the supplied subsystems. To ensure seamless integration into the SLS control system, the vendor company received an extended EPICS software package tailored by the SLS that included parts of the framework in which their product was intended to run. In addition to that, vendor specialists were given EPICS training by the SLS. [11]

4.2 Development by Commercial Partners

A number of large projects of the EPICS collaboration combined their efforts and funding to create a unique partnership with a commercial software company, Cosylab [12] from Slovenia. Cosylab developed and improved the standard visual database configuration tool for EPICS, VisualDCT. [13] After some initial administrative resistance was overcome concerning providing public money to a private company for creation of open source software, the endeavour has proceeded efficiently and the results were very convincing. New groups of sponsor institutes formed and specified additional packages containing improvements and extensions of VisualDCT that were handed out to Cosylab.

During this process, it became obvious that large projects with commensurate funding strength are in the position to get their demands into prominent parts of the specification, while the larger part of the collaboration played a more passive role with less control but nevertheless benefiting of the improvements and newly added features.

5 MAJOR TOOLS FROM THE TOOLBOX

The EPICS toolkit contains a large number of different tools, applications, and interfaces. From this huge collection a set of generic, easily adaptable core tools emerged, covering the fundamental functionalities of a control system. Some of these tools are traditional, highly stable applications with a long development history, others are new developments that promise to be future standards. Any new installation of EPICS will use the majority, if not all of these applications, which will cover most demands of the development and commissioning phases:

The **edm** (Extensible Display Manager) displays and manages operator screens (panels).

VisualDCT [13] is the standard graphical tool to create EPICS databases.

The **Channel Archiver** [14] is the most-widely used tool for long-term archiving of data that is available through Channel Access, the EPICS data communication protocol.

The **alh** (Alarm Handler) displays, logs, and manages the alarm status of a hierarchy of channels.

The **sequencer** runs finite state machines on the IOC that connect to the database through Channel Access.

StripTool is a generic application to create multi-channel on-line x/t charts with browsing and zooming capabilities.

Also the development of these mainstream EPICS tools is influenced most by the large projects that typically fund them, while the rest of the collaboration shares equally in the benefits.

In addition to these main tools, the toolkit contains a wide variety of modules: special solutions for site-specific problems, new developments with a more experimental character, and complete highly featured subsystems that cover a complete functionality range. EPICS users can freely select, build, test, and evaluate any of these tools and then decide to keep, drop, watch, or improve it by adding the missing features. The EPICS tools follow strictly the Bazaar development model. [15]

6 CURRENT DEVELOPMENTS

6.1 WAN Capabilities of EPICS

The larger EPICS installations are running their control systems in a wide area network (WAN) environment with several IP subnets interconnected by routers. Channel Access (CA) was originally developed for systems running in a single IP subnet. Its name resolution is accomplished using IP broadcasts, and therefore large sites must be aware of scalability and configurability issues.

Today, the CA Gateway component [16] can be used to break large systems into manageable subsystems and the CA Nameserver component [17] can be used to reduce broadcast traffic. However, there are remaining WAN issues such as server state-of-health beacons, resource location monitoring, and wild card queries which need to be considered in the context of global name resolution services. [18,19]

6.2 Channel Access Protocol

The first steps have been made to prepare the design of the future Channel Access protocol. [18,20] This design relates to the newly designed object oriented data abstraction interface Data Access [20,21] which will be used in a future Channel Access API. Ultimately Channel Access will transport any user data across the network that can be introspected with the Data Access interface.

6.3 Native Java Channel Access Client

A number of collaboration members have asked for a native Java implementation of the Channel Access client library. As a first step in that direction efforts have been started to create a full official documentation of the Channel Access protocol.

6.4 Re-engineering of Core Libraries

An effort has recently started to re-engineer some central libraries of iocCore to provide a native C++ interface. [22] This will provide a strong foundation for future changes.

6.5 OS Independent Device Support

Introducing the OSI layer into Base created the necessary environment to provide operating system independent device and driver support for existing and future I/O equipment.

While the first support modules are being re-engineered and changed to be OS independent, it is getting more and more obvious that this process is not simple and may need some additional support from Base libraries.

6.6 Development Framework

The host-based tools that are used to create IOC databases and configurations for the different EPICS tools (see Section 5) are separate applications with few interconnections and interoperability. There is no integrating framework that allows for quick and easy creation of a database and the matching panels and tool configurations. At the recent Spring 2003 Collaboration Meeting at Diamond a first prototype of a possible framework application was presented [23], allowing the creation of a small EPICS application with the look-and-feel of an Integrated Development Environment.

7 TRENDS

A small email survey covering the ten largest EPICS installations shows current trends for hardware and operating system choices:

On the front-end level, most large institutes are using VxWorks on VME bus systems. New systems are entirely built with PowerPC based controllers, existing 68k based controller boards are gradually being replaced. Only few institutes are using RTEMS, while some more are evaluating it and wait to see how strong it will be supported by the community. Linux on Industrial PC and PC-104 hardware is being used and evaluated for slow and low channel density applications.

On the operator console level, there is a strong movement away from the traditional Sparc/Solaris and PA-RISC/HP-UX workstations towards using Linux on PC hardware, as well for graphical workstations as for file, computing and database servers. Especially HP’s future as a control system development platform seems limited.

8 LONG-TERM PERSPECTIVES

A task force group has been formed that organises a set of meetings called EPICS-2010 to explore actual and possible demands of control system users and collect ideas for future control systems in general and for EPICS specifically. The aim is to create a visionary sketch of the control system design and features that are desirable in a time range of about seven years. At this conference a status report on these meetings will be presented. [24]

9 CONCLUSION

After more than ten years the development of the EPICS toolbox had come to a point where the demands of a large user community and the mere size of the source code made it necessary to introduce some profound changes into the system design and code management.

A number of important steps have been made to modularise the system and improve the manageability, new ways have been created to utilise professional resources, and a process to analyse future demands for a modern control system has been established.

REFERENCES

[1] http://www.aps.anl.gov/epics/

[2] M.R. Kraimer, "EPICS: Porting iocCore to Multiple Operating Systems", ICALEPCS'99, Trieste, October 1999.

[3] M.R. Kraimer, J.B. Anderson, J.O. Hill, W.E. Norum, "EPICS: A Retrospective on Porting iocCore to Multiple Operating Systems", ICALEPCS'01, San Jose, November 2001.

[4] http://www.rtems.com/

[5] W.E. Norum, "EPICS on the RTEMS Real-Time Executive", presented at the Synchrotron Radiation Instrumentation 2001 Conference, Madison, Wisconsin, August 2001; published in the American Institute of Physics Review of Scientific Instrumentation, January, 2002.

[6] T. Straumann, "Open Source Real Time Operating Systems Overview", ICALEPCS'01, San Jose, November 2001.

[7] W.E. Norum, "New Capabilities of the EPICS IOC Shell", ICALEPCS'03, Gyeongju, October 2003.

[8] R. Fleischhauer, B. Kuner, G. Meyer, J. Rahn, C. Winkler, "Integrating PLCs with an OPC Interface into an EPICS-based Control System", ICALEPCS'03, Gyeongju, October 2003.

[9] D.H. Thompson, W. Blokland, "A Shared Memory Interface between LabVIEW and EPICS", ICALEPCS'03, Gyeongju, October 2003.

[10] http://www.aps.anl.gov/epics/mantis/

[11] S. Hunt, "Purchasing Accelerator Subsystems as Turnkey Components", ICALEPCS'01, San Jose, November 2001.

[12] http://www.cosylab.com/

[13] M. Sekoranja, S. Hunt, A. Luedeke, "VisualDCT – Visual EPICS Database Configuration Tool", ICALEPCS'01, San Jose, November 2001.

[14] K.U. Kasemir, L.R. Dalesio, "Overview of the Experimental Physics and Industrial Control System (EPICS) Channel Archiver", ICALEPCS'01, San Jose, November 2001.

[15] Eric S. Raymond, "The Cathedral and the Bazaar", first presented at the Linux Congress, 1997; published by O'Reilly, October 1999. http://www.catb.org/~esr/writings/cathedral-bazaar/

[16] R. Lange, "CA Gateway Update", presented at the EPICS Collaboration Meeting at JLab, Newport News, Virginia, November 2002.

[17] J. Sage, M.H. Bickley, K.S. White, "Using a Nameserver to Enhance Control System Efficiency", ICALEPCS'01, San Jose, November 2001.

[18] J.O. Hill, "Next Generation EPICS Communication Protocols", ICALEPCS'99, Trieste, October 1999.

[19] J.O. Hill, K. Furukawa, S. Hunt, A. Johnson, R. Lange, J. Sage, E. Williams, "EPICS and WANs: Tradeoffs between Isolation, Security, Robustness, and Transparency". ICALEPCS'03, Gyeongju, October 2003.

[20] J.O. Hill, R. Lange, "Next Generation EPICS Interface to Abstract Data", ICALEPCS'01, San Jose, November 2001.

[21] R. Lange, J.O. Hill, "Data Access – Experiences Implementing an Object Oriented Library on Various Platforms", ICALEPCS'01, San Jose, November 2001.

[22] A.N. Johnson, M.R. Kraimer, "Reimplementing the EPICS Static Database Access Library", ICALEPCS'01, San Jose, November 2001.

[23] S. Hunt, "Why aren't we all using Labview?", presented at the EPICS Collaboration meeting at Diamond, Abingdon, June 2003.

[24] M. Clausen, N. Yamamoto, M. Kraimer, H. Shoaee, L. Dalesio, "EPICS-2010 – A Visionary Look at the Evolution of a Collaborative Control Toolkit", ICALEPCS'03, Gyeongju, October 2003.