INTEGRATING COMMERCIAL AND LEGACY SYSTEMS WITH EPICS

J.O. Hill, LANL P.O. Box 1663, Los Alamos, NM 87545 USA
K.U. Kasemir, Universität Osnabrück, Fachbereich Physik, D-49069 Osnabrück, Germany
J.B. Kowalkowski, ANL 9700 South Cass Avenue, Argonne, IL 60439 USA

Abstract

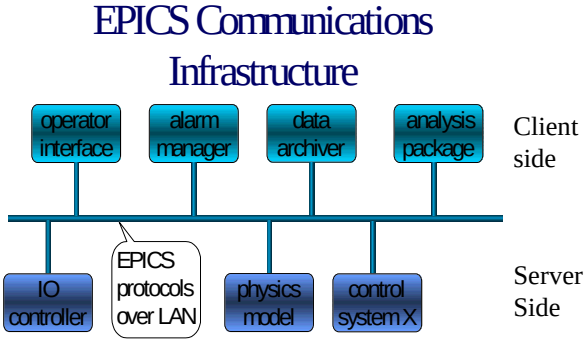
The Experimental Physics and Industrial Control System (EPICS) is a software toolkit, developed by a worldwide collaboration, which significantly reduces the level of effort required to implement a new control system. Recent developments now also significantly reduce the level of effort required to integrate commercial, legacy and/or site-authored control systems with EPICS. This paper will illustrate with an example both the level and type of effort required to use EPICS with other control system components as well as the benefits that may arise.

1 INTRODUCTION

1.1 What is EPICS

EPICS is a process control and data acquisition software toolkit in use at over 70 sites world wide. The software is designed for general utility and has been successfully installed into a wide range of applications including particle accelerators, experimental physics detectors, astronomical observatories, municipal infrastructures, petroleum refineries, and manufacturing. A scalable, fault-tolerant system that follows the "standard model"[1] can be created with the toolkit. Compilers and filters are used to instantiate control algorithms in front-end computers from function block and state-machine formalism-based input. EPICS communication occurs within a software layer called channel access (CA) that follows the client server model and employs the internet protocols (Figure 1).

Figure 1



A mature set of client-side tools provide operator interface, alarm handling, archival tasks, backup, restore, state sequencing, and other capabilities. There is also an expanding library of hardware device drivers that have been written for use with EPICS. Recently we have seen a number of sites working on generic physics and control theory applications that will interface directly with EPICS. All of these components taken together form a toolkit that allows control system installation with a minimum of low level coding. Details can be obtained on the world-wide web[2] and from previous papers[3][4][5][6]. EPICS is very unusual among control system software packages in that it has been developed by a collaborative effort of several laboratories and industrial partners[7].

2 INTEGRATING WITH EPICS

2.1 Manipulating EPICS Process Variables

Many 3rd party products have been integrated with EPICS. These include LABVIEW, PVWAVE/IDL, SLGMS, WING Z, MATLAB, TCL/TK, DATAVIEWS, MATHMATICA, and the SDDS toolkit. All of these facilities have the ability to read and write external named data. To integrate with EPICS, a simple code that translated these requests into requests to read, write, and monitor an EPICS process variable must be written. With this translation code installed, the 3rd party product becomes an EPICS client side tool that can directly manipulate EPICS process variables.

Recently, many components of EPICS have been ported to the PC environment[8]. This has opened new opportunities to integrate EPICS with the many inexpensive "shrink-wrapped" software components that are available for the PC today. PC versions of the EPICS libraries have been packaged as Microsoft Dynamic Link Libraries (DLLs). This allows these EPICS components to be called directly from almost any PC program with macro language capabilities such as Excel, Visual Basic, and many others. This does however require writing small amounts of code.

A Microsoft Direct Data Exchange (DDE) server for EPICS has also recently been written. Many PC based programs can read, write, and monitor named data through this interface. Many 3rd party PC products can now integrated without writing *any* additional code. For instance, to access an EPICS process variable with Microsoft's Excel we simply add a line as follows to our spreadsheet.

==CaDDE|Get!'HighlyImportantMagnetCurrent'

0-7803-4376-X/98/$10.00 © 1998 IEEE

2.2 Your Data Can be an EPICS Process Variable

Recently a server level "Applications Programmers Interface" (API) has been added to EPICS. After writing a small amount of translation code conforming to this API, any named data appears in the EPICS process variable name space and can be directly manipulated as an EPICS process variable. This allows a legacy system to leverage the many EPICS client side tools that are capable of directly manipulating EPICS process variables. For example, the LANSCE control system at LANL and the D3 control system at DESY both can export their data into EPICS using this mechanism.

To export data into EPICS as an EPICS process variable one must provide at least the functions in Table 1.

Table 1

	Function
1	named process variable exists in subsystem test
2	create (or locate) instance of process variable
3	read from process variable
4	write to process variable

It is also necessary to inform EPICS when a process variable is modified so that any clients monitoring that variable will receive an update. To be compatible with the widest range of EPICS clients a subsystem must also be ready to respond to client requests for a standard set of process variable attributes such as limits, units, time stamp, and alarm status.

2.3 Adding New Function Blocks

A typical EPICS system has a real-time database resident in a processor with direct access to the process IO. This real-time database is made up of interconnected function blocks which implement the logic specific to a particular application of the system. A wide range of function blocks are supplied with the system. They provide execution semantics associated with various types of process IO, calculation, multiplexing, fan-out, delay, data reduction and many others. This list can be extended by writing a record support module that plugs into well defined software interface within the EPICS system.

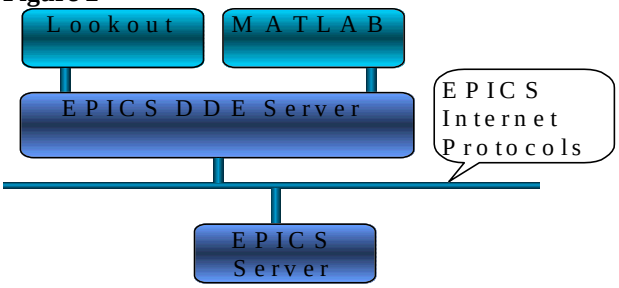
2.4 Adding New Device Drivers

There are also well defined interfaces within EPICS for adding new device drivers. To install a new device driver the programmer must write a set of functions conforming with the device support API of a particular function block.

3 AN EXAMPLE

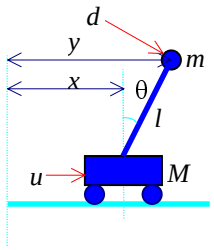
A demonstration of the famous nonlinear cart and pendulum problem from control theory was implemented on a portable laptop PC using only EPICS, the EPICS DDE server, and demonstration copies of two commercial products for the PC (Figure 2).

Figure 2



An evaluation copy of National Instrument's "Lookout" was used to implement the operator console for the demonstration. An evaluation copy of "MATLAB" from Math Works was used to implement a model of the cart and pendulum dynamic system (Figure 3), and also an LQR feedback compensator for this system (Figure 4). The dynamics of the system can be seen in Equations 1 and 2. All communication between MATLAB and Lookout occurred with the assistance of an EPICS server and an EPICS DDE server. The model and the compensator were updated in "pseudo real-time" in response to force input *u* from the operator console. The demonstration system can be controlled in a manual mode (without the aid of the LQR compensator) and also a closed loop mode (with the aid of the LQR

Figure 3

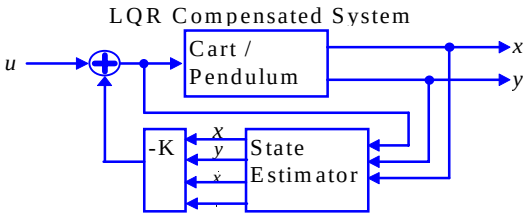


$$(M+m)\ddot{x}+ml(\ddot{\theta}\cos\theta-\ddot{\theta}^2\sin\theta)=u \quad 1$$

$$m(\ddot{x}\cos\theta+l\ddot{\theta}-g\sin\theta)=d \quad 2$$

compensator). The model is updated within MATLAB 100 times a second in response to a periodic DDE update for the force input *u* from the EPICS server.

Figure 4



The demonstration was implemented in 2 man weeks by an individual integrating these products together with EPICS for the first time. Most of the labor was devoted to drawing the operator screen within Lookout, modeling the linearized cart and pendulum system in MATLAB, and designing an LQR compensator using the MATLAB control systems toolbox. No new EPICS code was written for this demonstration. Figure 5 shows the response of the

compensated system to a step on the *u* force vector on the cart.

4 CONCLUSIONS

EPICS is an open modular system that was designed for extension. Nearly all components within the system are replaceable. It is possible to manipulate EPICS process variables from a system that knows very little about EPICS. Likewise a system mostly unaware of EPICS can export its data into EPICS as EPICS process variables. Within two weeks one of the authors was able to integrate two commercial PC based components and create a non-trivial control theory demonstration that was updated in pseudo real time and hosted solely on a laptop PC platform. The public interface to EPICS only supports sending read, write, and monitor messages to a process variable and its associated attributes. This interface is both primitive and concise. It has been suggested that this interface is therefore old technology and not suitable for new projects, and we will counter that it is compatible with a large number of commercial and legacy systems that exist today. When optimizing and debugging a control process it is important for operators to be able to directly examine and record the internal state of the system. We have positive expectations for emerging high level object oriented standards, but we also see the continuing value of a high performance process variable based interface.

[1] B. Kuiper: 'Issues in Accelerator Controls', Proc. ICALEPCS, Tsukuba, Japan, 1991, pp 602-611.

[2] W. McDowell et al.: 'EPICS Home Page', "http://www.aps.anl.gov/asd/controls/epics/EpicsDocumentation/WWWPages/EpicsFrames.html"

[3] L. Dalesio et al.: 'The Experimental Physics and Industrial Control System Architecture: Past, Present, and Future', Proc. ICALEPCS, Berlin, Germany, 1993, pp 179-184.

[4] L. Dalesio et al: 'The Los Alamos Accelerator Control System Database: A Generic Instrumentation Interface', Proc. ICALEPCS, Vancouver, Canada, 1989, pp 405-407.

[5] J. Hill: 'Channel Access: A Software Bus for the LAACS', Proc. ICALEPCS, Vancouver, Canada, 1989, pp 352-355.

[6] J. Hill: 'EPICS Communication Loss Management', Proc. ICALEPCS, Berlin, Germany, 1993, pp 218-220.

[7] M. Knott et al: 'EPICS: A Control System Software Co-development Success Story', Proc. ICALEPCS, Berlin, Germany, 1993, pp 486-491.

[8] Chris Timossi performed the initial port of several of the EPICS components to the PC environment.

Figure 5