

COMPARISON BETWEEN AN IN-HOUSE VS. A COMMERCIAL CONTROL SYSTEM FOR BEAM LINE CONTROL*

Martin Pieck[†], Kay-Uwe Kasemir,
 Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Abstract

Two independent magnet and vacuum control systems were implemented in the Low-Energy Demonstration Accelerator (LEDA) facility. The first one, for the High-Energy Beam Transport (HEBT) in context of the Accelerator Production of Tritium (APT) project, used the in-house Experimental Physics and Industrial Control System (EPICS). The second one, for the high-power proton beam-halo experiment [3] (HALO-Channel) as an extension to the existing accelerator, is based on commercial products (National Instruments, LabVIEW). This paper will evaluate our experience during implementation, operation, and maintenance of these systems. It will include a discussion of various problems encountered and solutions implemented.

1 INTRODUCTION

As part of the linac design for the Accelerator Production of Tritium (APT) project, the first 6.7-MeV portion of this 100-mA proton accelerator was assembled at the Los Alamos Neutron Science Center (LANSCe) in 1999 and was in operation for over one year. In 2000, a 52-magnet lattice to measure high-power proton beam-halo formation extended this Low-Energy Demonstration Accelerator (LEDA) that is in operation since fall 2000.

The first one, for the High-Energy Beam Transport (HEBT), used the Experimental Physics and Industrial Control System (EPICS), developed in a collaboration of Los Alamos and Argonne National Laboratories with Thomas Jefferson National Accelerator Facility and others. The second one, as an extension to the existing accelerator, is based on commercial products (National Instruments, LabVIEW).

In principal both system comprise the same type of machinery (powers supplies (PS), ion pumps (IP), ion gauges (IG), themocouples (TC), gates (GV), and beam line valves (BLV)) even so the quantity of the devices are different based on the length of the acclerator section that they control. Table 1 gives an overview of the quantity of devices that are controlled by both control systems.

Table 1: Beam Line Device List

DEVICE	HEBT	Halo-Channel
Quadrupole PS	4	52
Steering PS	4	20
IP	1	8
IG	2	3
GV + BLV	3	3
TC	16	40

EPICS, used for the HEBT, is a toolkit for building distributed control systems, now used by a collaboration of over 100 institutions [1]. It is the basis for operator controls interfacing. Commercial controls applications and visualization tools have been integrated by using the EPICS Channel Access (CA) communication protocol.

National Instruments (NI), used for the HALO-Channel, is a commercial company specializing in easy-to-integrate products. It achieves this by providing high-level software functions that communicate with their own off-the-shelf hardware products as well as products that fulfil industry standards.

Provided below is a summary of the control system components for the HEBT and HALO-Channel. Brief descriptions of hardware system architecture as well as software components are presented.

2 CONTROL SYSTEM HARDWARE ARCHITECTURE

Figure 1 shows the configuration of the HEBT control system (CS) and how it is connected to the controlled devices. File server and workstations are SUN Microsystems Ultra 5s running the Solaris operating system. They are connected via Ethernet and the CA communication protocol to the personal computer (PC)-input/output controller (IOC) with a 200Mhz Intel Pentium II processor.

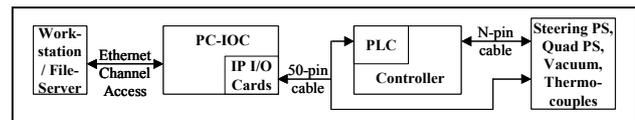


Figure 1: HEBT Control System Architecture.

The PC-IOC hosts 4 GreenSprings ATC40 carrier cards in ISA slots. Each of those carrier boards can host four so-called IndustryPac (IP) I/O cards used for data acquisition. They are available from different brands. For the HEBT the IP cards from GreenSprings (16-ADC/analog in, DAC-SU/binary out, and UniDig-I-E/binary in) and Systran Corporation (ADC128F1/analog out) have been used.

As indicated, ribbon cables connect the PC-IOC and the controller to the accelerator hardware as well as to each other. The controller hosts an Automation

*Work supported by US DOE, Office of Defense Programs, and the Office of Nuclear Energy, Science and Technology.

[†]pieck@lanl.gov

DirectLogic 205 PLC with analog and binary in/out modules. Analog signals may use Ultra SlimPaks for isolation purposes and binary signals use a separate binary relay board from and to the PC-IOC. Neither is shown in the drawing.

The HALO-Channel control system extends the LEDA controls hardware architecture by 2 PC-IOCs as shown in Figure 2. Two IOCs were necessary since the amount of I/O channels exceed what one could handle. Both are connected via the same Ethernet (like the HEBT) and CA communication protocol to the workstation/file server. The first PC-IOC (I) is a 266MHz Intel Pentium II processor that hosts 2 PCI-GPIB communication boards from NI, each controlling 10 PS.

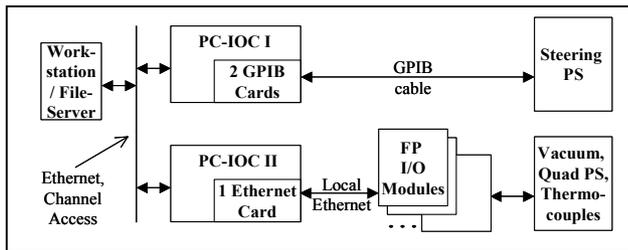


Figure 2: HALO-Channel Control System Architecture.

The second one is also a PC-IOC with a 500Mhz Intel Pentium II processor, hosting a second Ethernet Network card, which creates an independent network that is just local to that PC-IOC II. This separate high-speed Ethernet network has 10 NI FieldPoint (FP)-1600 Ethernet network modules that interface with up to six FP I/O modules (analog/binary in/out) each. With up to 100 Mb/s data communication rates and event-driven communication, the FP-1600 delivers a high-performance network connection to the PC-IOC. The FP-1600 network module communicates between the host PC and the I/O modules. It also provides several diagnostic and auto configuration features.

3 CONTROL SYSTEM SOFTWARE ARCHITECTURE

3.1 Network Configuration and User Interface

The communication between the PC-IOCs of the HEBT and HALO-Channels control systems in the accelerator equipment area and the Workstations/File Servers in the control room is built on a TCP/IP-based network and uses EPICS CA as the primary protocol.

In order to provide an operator interface (OPI) on top of the EPICS CA communication protocol, two main applications were employed: the display editor (EDD) and the display manager (DM). DM is the actual real-time interface, the interface that an operator uses to monitor and control through the EPICS CA system, while EDD is used to design and configure the displays that DM uses. Therefore, to the operator supervising both control

systems, it provides the same look and feel beside the fact that the implementations use two different software applications.

3.2 Software Application and Operating System

The HEBT uses EPICS and the HALO-Channel uses LabVIEW. Even though both fall in the category “graphical programming”, one of the biggest differences is that the heart of EPICS (the runtime-database) requires the “configuration” of predefined function blocks (called records) while LabVIEW is much closer to a real programming language with symbols that provide a more straight forward implementation of the intended data operation.

The EPICS IOC runs the real-time operating system (RTOS) VxWorks while the LabVIEW IOCs run on Windows NT. The uniform OPIs for both control systems run on SUN-Solaris operating system (OS).

3.3 Application Development System

The application development environment (ADE) for the HEBT is a set of tools that is not truly a part of EPICS. It is a piece of software that sets up the software application environment for an EPICS application. To set up EPICS applications, it is helpful to use those tools. Though powerful, detailed knowledge is required to use their features and avoid errors. The Open-Source Concurrent Versions System (CVS) tool was used for source code control. CVS maintains the changes between versions. It supports group collaboration by merging the files from each programmer therefore providing organized collaborative development opportunities.

Though this was not used for the HALO-Channel, LabVIEW (only the Professional Development System edition) has a source code control (CCS) tool that can integrate with third-party source code control systems.

3.4 Control Strategy

The implementation of an appropriate control strategy required two different approaches based on the difference in hardware that was used for the HEBT and HALO-Channel [2].

The EPICS configuration is twofold. A runtime database models data flow: Interlinked “Records” specify hardware or software inputs and outputs, scan rates, conversions, display parameters etc. On the IOC, the EPICS base software reads this configuration, creates threads for the necessary scan rates and handles read and write requests via CA. This approach is less useful for the implementation of sequential control strategies.

Therefore, the State Notation Language (SNL) tool was used. It is a C pre-processor for modelling sequential control. The resulting program is loaded onto the IOC. While this approach requires knowledge of C as well as

SNL, it allows implementing arbitrarily complex algorithms and provides file access.

LabVIEW, on the other hand, allows implementing any type of data processing, sequential control logic, and file access with the provided functions. In this respect it is an advantage that one has to learn only one tool rather than bringing different tools together that fulfil the same purpose. Using different tools also bears the possibility of creating mistakes that are overlooked since there is not an obvious connection between one and the other tool. On the flipside, all data sampling, data flow, sequential operation and read/write requests from the network have to be coded and debugged from scratch in LabVIEW.

3.5 Device Support

At the end both control strategies control the hardware listed in Table 1. For this purpose, appropriate, off-the-shelf hardware was employed. For the HEBT EPICS system, a set of in-house device drivers had already been developed. The LabVIEW drivers for the HALO-Channel came with the corresponding hardware.

Noticeable during implementation was the flexibility of the EPICS system, given through the available source code. But the possible employment of other hardware was restricted since new drivers needed to be developed in-house, which would have been more time consuming.

LabVIEW, on the other side, provides a wide range of drivers for all sorts of hardware, even more appreciated is the support of many companies providing their products with NI drivers, which makes the integration much easier and less time consuming. On the down side, if those drivers show any kind of problems (as happened for the HALO-Channel) or do not provide the expected functionality one is at the mercy of the company to fix the problem.

3.6 Performance

The EPICS application runs on the RTOS VxWorks while LabVIEW runs on Windows NT. Neither of one is absolutely deterministic but the LabVIEW system shows more performance fluctuations depending on the CPU load of the PC-IOC. In particular this shows as a varying discrepancy between expected and observed scan rates.

3.6 Maintainability

Software, or program maintenance is an important and yet sometimes challenging job. Updating application programs in order to meet changing information requirements, such as adding new functions, changing data formats as well as fixing bugs and adapting the software to new hardware devices requires the understanding of the software design and architecture. Depending on who is maintaining the software, rather simple modifications can become more time consuming than necessary.

While someone unfamiliar with the PLCs might find the HEBT control system more complex, since it requires the understanding of PLC ladder logic and the EPICS database, there is on the other hand a precise distinction between data configuration/processing and functional operation. The control system for the HALO-Channel incorporates the configuration/processing as well as the functional operation within the LabVIEW system. Although it is more complex, all information is stored in one program. This might cause some confusion since related information is distributed over the program and not centralised as in the EPICS records. With respect to convenience, EPICS IOCs allow login and debugging from a remote control room, LabView IOCs require the maintenance personal to be physically at the PC.

4 CONCLUSIONS

In our opinion, the perfect toolset that serves all individual control system needs does not exist, but based on our experience we can make several claims.

EPICS seems to be better suited for larger research projects since usually requirements do not show up in a timely fashion and through its scalability it handles those issues better. On the flipside, if future hardware upgrades are of particular concern NI/LabVIEW has an advantage since it provides drivers as soon as a new product hits the market. Smaller projects are most likely better implemented in LabVIEW since the overhead of setting up the ADE and learning different tools is not necessary. Furthermore, the people producing the subsystem controls are going to be most efficient using tools that are familiar to them, but the integration requirements and system maintenance must then be considered from the design phase through completion.

Therefore, at the beginning of the project, one has to evaluate not only the control system requirements but also the people involved in building and maintaining it.

It is likely that only a compromise between in-house and commercial tools will minimize cost and time spent.

5 ACKNOWLEDGEMENT

We are very grateful to Matthew Stettler (Los Alamos National Laboratory/LANSCE-8) for helpful discussions during the preparation of this paper.

6 REFERENCES

- [1] M. Thuot, et al., "The Success and the Future of EPICS" LINAC96, Geneva, August 1996.
- [2] D. Moore and L.R. Dalesio, "A Development and Integration Analysis of Commercial and In-House Control Subsystems", LINAC98, Chicago, August 1998.
- [3] P. Colestock et al., "Measurement of HALO Generation for Proton Beam in a FODO channel", these proceedings.