THE EVOLUTION OF JEFFERSON LAB'S CONTROL SYSTEM*

M. Bickley, W. Watson, K. S. White Jefferson Lab, Newport News, VA, USA

Abstract

Thomas Jefferson National Accelerator Facility's (Jefferson Lab) accelerator controls were initially implemented as a proprietary in-house system. During machine commissioning, problems were encountered leading to a decision to migrate to the Experimental Physics and Industrial Control System (EPICS). Since then, the accelerator and all other laboratory controls have been successfully converted. In addition to implementing Jefferson Lab's controls using EPICS, new data visualization tools have been developed and existing programs have been enhanced with new capabilities. In order to provide a more generic interface for high level applications development, a device abstraction layer, called Common DEVice (CDEV), was implemented. These additions have been made available to other laboratories and are in use at many sites, including some that do not use EPICS. Control system development is not limited to computer scientists; operators, engineers and physicists frequently add capabilities using EPICS, CDEV, Tcl/tk, and other tools. These contributions have tailored the control system for many different types of customers. For the future, we envision more intelligent processing and more capable tools for data storage, retrieval and visualization.

1 INTRODUCTION

Over the past six years, the control system of the Continuous Electron Beam Accelerator Facility (CEBAF) at Jefferson Lab has evolved from an in-house system using proprietary graphics and network protocol, to a distributed client-server system using X-Windows and TCP/IP. The original system, know as Thaumaturgic Automated Control Logic (TACL), in addition to using proprietary features, did not scale well due to the tightly coupled system components and the lack of flexibility of the control logic and graphics displays. The TACL system, was in use at the Cryogenics Facilities, Injector and Accelerator Arcs and Linacs. During this time, controls for the three Experimental Halls were in the design phase and were not based on TACL.

As TACL based accelerator commissioning activities increased, it became obvious the system wide data update rate was unacceptably slow. This led to a more in depth look at the system, reprogramming the network data transfer software and ultimately, a decision to adopt a different control system. The new system, based on EPICS, was being developed by a collaboration of national laboratories. Since that decision, our controls have been converted to EPICS, which is working well and has scaled appropriately.

The conversion was accomplished primarily during machine commissioning by phasing in new software one system at a time and initially using EPICS controls with the original TACL display pages. In addition to the conversion of existing applications, many new control applications have been developed as the machine has been upgraded and new experiments have been installed in the halls. We have also developed new EPICS tools and enhanced others that are in use here and have been contributed back to the collaboration. Due to the flexibility and name-based nature of the EPICS system, it is easy for users to create new back-end software. Operators routinely contribute user interface programs developed using EPICS tools and the Tcl/tk toolkit. Accelerator physicists develop beam applications using EPICS, CDEV, Tcl/tk and other tools. Such flexibility has helped the system become more suited to the needs of different types of customers.

For the future, we envision adding tools to facilitate more intelligent processing on the back-end computers, resulting in better information and automation for the operators. We are also working on better tools for data storage, retrieval and visualization supporting a variety of different data formats. In order to provide better data integrity, and centralize the storage of critical operational data, we also plan a central database. This paper discusses the technical reasons that led to replacing the control system, how the conversion was accomplished, our successes, challenges and additions to EPICS, and our plans for future development efforts.

2 TACL

2.1 Philosophy

TACL was developed as a Control System Toolkit, providing a set of tools used to develop application specific control algorithms and display pages, and an environment for runtime execution and monitoring. [1] The distributed system, based on the standard model, [2] provided the standard set of control system functions including signal read and write, graphical user interfaces, alarms and data logging. The computers used for both local hardware control and high level operator display functions were Hewlett Packard (HP) workstations, and

* Work supported by the U. S. Department of Energy, contract DE-AC05-84ER40150

the software was developed using HP's proprietary Starbase graphics library and Link Level Access network protocol. The system provided control via CAMAC hardware and GPIB crate controllers.

2.2 Reality

The TACL system provided effective tools for all essential control system requirements. The graphical logic and display editors were easy to use, and programmers and engineers could quickly develop control algorithms and graphical user interfaces for many different accelerator systems. The local LAN and super LAN network programs shipped data from local computers controlling hardware in the field to all supervisory computers hosting operator displays. The system performed reliably and efficiently on small systems such as the Cryogenic Test Facility, Injector Test Stand, and RF Test Stand that were developed in preparation for the larger operational cryogenics plant and full accelerator. However, as various systems of the accelerator completed the construction phase, controls were installed for far more control points than originally estimated. As a result, the performance degraded, and system-wide updates became too slow.

The Logic program was responsible for executing predefined control algorithms on the local computers. The Logic Editor provided a rich set of functions for customizing control logic, but the runtime Logic program only had the capability to run the algorithm as one large monolithic block, preventing control or variation of the speed of execution. Additionally, the execution of the control logic was implemented in an interpreted fashion, limiting the overall speed of execution. A later version of the Logic program implemented a compiled execution option, speeding up execution by a factor of 4-10 times, depending on the CPU type. This improvement, however, did nothing to help with the network data update speed. The GPIB interface to CAMAC hardware proved reliable and performed quickly enough, but limited the control system to one type of hardware interface.

The graphical user interface tool provided an easy to use editor, and good run-time execution, but unlike newer systems, had no windowing environment and therefore could only display one control page per monitor.

3 ALTERNATIVES

3.1 Options

With the growing realization of the limitations of the existing control system, several options were proposed to ensure the control system would meet the future needs of accelerator commissioning and physics beam delivery. One proposal was an upgrade of the TACL system to overcome the performance and feature deficiencies. The other two options centered on importing a control system and making necessary additions to customize the system for Jefferson Lab. For this proposal, the long used control system from Fermilab, Accelerator Controls NETwork (ACNET), was considered along with EPICS, at the time a fairly new control system, in use at Los Alamos and Argonne National Laboratories (LANL and ANL).

Due to the short time available, and the feasibility of using an existing control system, we did not consider starting another in house programming effort from scratch. In years past, virtually all laboratories developed their control systems this way, before software sharing became a viable and efficient option. Once it was believed possible to effectively share and co-develop a control system with other labs, the benefits in terms of code reusability, wider expertise and shared labor were too attractive to ignore.

With three different paths to choose from, it was not at all clear which would provide the best solution and could be accomplished within the short time frame needed to support the lab schedule for commissioning and operations. Each alternative seemed to involve a considerable amount of effort, and each had different strong points working in their favor.

3.2 TACL Upgrade

An upgrade of the TACL system would allow usage of the already developed device control, utilize all existing hardware and minimize procurement and installation costs. Additionally, virtually all of the in house expertise was with Unix systems and TACL, so no additional training would be required. The toolkit approach was favored to reduce the required programming effort for adding new applications. Unfortunately, this option involved labor intensive changes deep in the TACL system that would in fact duplicate much of the effort already invested in EPICS.

3.3 ACNET

Fermilab's control system, ACNET, was a VAX/VMS based system, with many years of reliably running a large control system in its favor. This option would have involved replacing much of the existing control hardware in addition to replacing all Unix machines with VAX/VMS systems. Changing to a new operating system would require retraining developers. Additionally, this system involved custom coding each system application, with no tools available to make this task more efficient. In general, this system was viewed by the lab as a very robust, but aging system that did not take advantage of newer technologies. In the initial analysis, ACNET's existing base of high level applications was seen as a plus, but even after importing the base system, these applications would have required significant rework to function for the CEBAF accelerator.

3.4 EPICS

The EPICS system [3], a more recent development, began at LANL, and was adopted by ANL along with their commitment to collaborate and continue development. The Unix based system utilized a toolkit approach, but had no proven success running a large control system. This option involved changing a limited set of hardware and redeveloping device control applications using EPICS tools. The system had many existing device drivers, but no CAMAC drivers and no existing high level applications, so these would also need to be developed.

In the end, EPICS was selected for Jefferson Lab's control system. Sharing the TACL toolkit philosophy, taking advantage of the best existing technology and supporting multiple hardware interfaces all factored in EPICS' favor. Additionally, the growing strength of the fledgling collaboration was seen as an effective way to get more and better software with fewer programmers, reducing duplication of effort, and benefiting all participating labs.

4 CONVERSION TO EPICS

The first step in the conversion process was to select a portion of the existing control system, and attempt to provide similar controls functionality for that portion [4]. This would serve as a prototype, providing the accelerator controls staff with training and experience in EPICS, and demonstrating the feasibility of converting the remainder of the controls. The plan was to perform the control system conversion while accelerator commissioning was under way. At this point in time, roughly one third of the beam line was already in use, and there was a strong disincentive to halt the accelerator development schedule for a prolonged period of control system rework.

The part of the accelerator selected for the prototype was the superconducting RF system. One reason for beginning with the RF system was its relative independence from the rest of the control system, running on dedicated front-end computers. The RF is a significant component of the entire control system, representing roughly one-half of all I/O points. Additionally, it is highly repetitive, so once the complex controls were developed for one cryomodule, the logic could be replicated, substituting proper names, for the other 21 cryomodules. Demonstrating the replication capability was key to showing TACL systems could be quickly converted for EPICS. Success in this prototyping effort would provide convincing evidence that the entire project was feasible and help us develop a schedule for proceeding.

A small group of software developers were selected to participate in the RF prototype. They served as a vanguard in learning about EPICS, immersing themselves in the new control system with the idea that they would mentor other developers as work continued to convert other systems. In order to provide on the job training and speed the RF prototype development work, Jefferson Lab established a close relationship with the controls group at LANL, where EPICS was first developed. LANL provided on site support during the prototyping project. Their EPICS expertise, immediate availability and insight into controls development kept the project moving forward. This proved to be very valuable assistance, preventing potential roadblocks from impeding progress. After 3 months of work, the prototype was complete, and we were able to control the RF system using EPICS.

Having shown that EPICS was a reasonable option for Jefferson Lab, we chose to make the conversion as smooth and non-invasive as possible by providing a link between the existing TACL controls and new EPICS controls. By providing a live interface between the two control systems, accelerator operators were able to control hardware connected to EPICS through TACL interfaces, and vice versa. This provided users the opportunity to examine TACL and EPICS screens, side by side, and verify proper operation. The integration process was made straightforward by the fact that both control systems use name-based management of process variables. A single Unix process served as the locus of communication between the two control systems, so that all name translation and conversions could be made in one place.

Once the RF system and the EPICS/TACL interface were working well, we used the information we had learned to plan for the conversion of the remaining accelerator systems. The CHL was left for last because of its more stringent requirements for high availability. A failure of CHL control usually results in the loss of liquid helium, so a few hours of down time can be quite costly. The cryogenic system was small enough that TACL performed well and reliability was very good. Given these factors, the cryogenic personnel wished to see at least a year of operational success with EPICS in the accelerator before they would consider their own conversion. EPICS continued to perform well during the remainder of the accelerator conversion, and afterwards the conversion of the cryogenics systems also went smoothly. This completed the implementation of EPICS for Accelerator Division systems. In the Physics Division, the existing control applications were also ported to EPICS, and new developments began using EPICS. Using the same control system labwide has proven very beneficial especially as we have been able to share control system programmers across divisions as needed.

5 IMPROVEMENTS AND UPGRADES

As we began to work with EPICS, we added some IOC features to support conversion of our existing systems. After introducing EPICS for machine operations, we began improvements on the console side of the system to meet the needs of our customers.

In order to make EPICS work with our existing CAMAC hardware; a driver was written according to the ESONE standard, handling I/O for all types of CAMAC modules. Once most of the existing TACL CAMAC based applications were converted to EPICS, many new systems began to take advantage of EPICS' broad hardware support. We were able to import, from other EPICS sites, VME device support for several modules and we developed device support for 20 additional VME cards. Serial and GPIB applications were also developed.

We developed custom record support for our more complex systems, Beam Position Monitors and Magnets. EPICS records were also developed and contributed back to the collaboration for use by other EPICS sites. These include the Big Subroutine record, MultiBit Binary Input Direct, MultiBit Binary Output Direct, Array Analog Input and Array Analog Output. The Big Subroutine record was added to provide support with more inputs and outputs than the existing EPICS subroutine record allowed. The MultiBit Binary Direct and Array Analog records were added to provide records for straightforward bit manipulation, needed for our RF system which relies heavily on information packed as groups of bits into words.

The cryogenic operators depended heavily on a TACL graphs program that continuously charted up to ten selected signals at regular time intervals. EPICS had a similar program called Striptool that provided this function but did not meet all the needs of the cryogenic engineers. As a result of customer feedback, we rewrote the Striptool program improving its timing, reliability and efficiency and adding new features, such as support for detailed configuration files and data exportation. This new Striptool is widely used in the EPICS community. TACL also included a custom PID record specific to the needs of our cryogenics plant. An EPICS record was written to duplicate the behavior of the TACL PID record, allowing the cryogenic applications to be converted and run with the same result as the original system.

Due to the need to quickly replicate applications with the same logic but different signal names, we worked with LANL to improve the method of using Capfast to develop hierarchical EPICS databases and scripts to automate the replication process. Along the same lines, a library of subroutines was developed to facilitate programmatic generation of control screens. This development allowed programs to be written to create the definition files needed for complex but repetitive screens, often containing thousands of signals represented as graphical objects. Not only did these two methods simplify the tedious work of creating and replicating databases and screens, they also dramatically reduced the effort required to maintain the databases and screens when devices were added, moved, changed or removed.

Jefferson Lab was the first EPICS site to exceed 100,000 operational EPICS records, an important milestone for demonstrating the scalability of the system. Because of the large size (~120 IOCs) of our EPICS installation and the large number of device control programs (~150), we also developed a configuration management system [5] for our IOC applications. This system provides features for multiversioning and quick application rollbacks, and has proven critical during system upgrades. As an operational facility, our maintenance and test time is usually very limited, and the machine must be quickly restored to an operational state after testing new code. Ensuring that we can always quickly install and boot the last good operational version of any application has allowed the controls group a good deal of latitude to test new software or upgrades during short machine downtimes. Without this freedom and insurance, it would take significantly longer to change the control system due to the long duration between major machine shutdowns.

With a large number of process variables in our system, we experienced a performance problem with EPICS due to the demanding nature of screens developed to control the RF and Magnet systems. These screens requested name resolution of up to 2000 signals upon startup. When multiple users brought up this type of screen, the result was a short, but noticeable spike of CPU utilization on all IOCs. With all the CPU cycles occupied responding to name resolution requests, system updates were temporarily delayed. This is due to the broadcast method EPICS uses to dynamically locate signal names, asking all IOCs if they host a requested process variable. In order to minimize this effect, we wrote a program called the CEBAF Channel Access Nameserver. This program learns the locations of all control system signals at IOC boot time. The most name resolution intensive programs, MEDM, BURT and ALH were then recompiled to get signal locations from this nameserver instead of by broadcast. These changes effectively eliminated the IOC CPU spikes and significantly reduced the time needed to resolve signal intensive screens, making operators much happier with control system performance.

Another area in which Jefferson Lab contributed enhancements to the original version of EPICS is the archiver. While the lab is currently working with LANL to develop a new archiving engine, we had an immediate need for a more intuitive, simple to use graphical viewer for archived data. With that as a goal, the lab designed and built a new tool for examining its data archives. It is extremely easy for users, because all they need to know is the names of the channels they wish to view. A few mouse clicks enable them to have a plot of the data against time, and the viewing window enables them to pan and zoom through time. The tool was built in a modular fashion, so that the data access portion of the code can be easily replaced with another set of code that provides the same calling interface. This has in fact been done, with our support, so that the archive viewer now provides visualization for several types of data stores.

We have also made efforts to enhance the EPICS synoptic display tool, MEDM. The original MEDM requires the licensing of a commercial product, XrtGraph, in order to use the X-Y plot widget. We needed more flexibility than the XrtGraph tool provided, and found it onerous to require the licensing of commercial software for this functionality. In order to resolve this, we reverse engineered the MEDM functionality of the XrtGraph library. New functions, with identical calling interfaces, were written so that reliance on XrtGraph was eliminated.

The effort of migrating our control systems to EPICS sparked interest in a general solution to the control system integration challenges. We set out to create a system interface that would make similar kinds of projects more straightforward and allow non-EPICS applications to easily run with our EPICS system. The result, CDEV, is a very general abstraction of device interaction. It enables any control system device to be addressed through messages, with a common Applications Programming Interface (API). Two things are required in order to integrate a new control system under CDEV: a service to map CDEV messages to the control system, and a description of what devices are available through that service. This approach has been demonstrated by the use of CDEV at other laboratories, which have developed the appropriate services for their control systems and used CDEV compliant client software developed at Jefferson Lab to interact with their control systems.

Another capability that Jefferson Lab desired was the ability to view CDEV data with a synoptic display. A portion of our operational data was available as real-time CDEV objects, but was unavailable through the standard EPICS network protocol, Channel Access. Because MEDM was working well for us in operations, it was logical to enhance this tool. In order to accomplish this, the code was modified to support either Channel Access or CDEV communication, depending on a compile-time option. With an appropriate environment, in which unknown CDEV objects are assumed to be Channel Access devices, all existing screens can be used with the CDEV version of MEDM, and still display correctly.

The primary communication tool for Jefferson Lab operations is the Electronic LOGbook (ELOG). This program provides a shared tool to communicate information about machine operations and maintenance, and replaces paper logbooks. ELOG entries can be entered manually or automatically. It is easy to include snapshots of terminal windows and or scan in paper documents. The ELOG is viewable by anyone on site using a web browser, and entries can be cross-referenced in other electronic documents. This tool has been extremely popular due the fact that many users can simultaneously add entries and view the logs. ELOG is easy to replicate for other groups and we now have about 12 integrated ELOGs in operation.

6 FUTURE PLANS

Jefferson Lab is working closely with LANL in order to enhance the archiving capabilities of EPICS. While we have already created a new archive viewer, the existing data-taking engine remains limited in functionality, difficult to maintain and unreliable. LANL has developed an engine that is a distinct improvement, but is missing some of the flexibility that we require. The archive system has three major components: a data taking engine, an independent management and control interface to the engine, and an archived-data viewer. We have defined the interfaces between the components, so that work can now proceed on the pieces independently.

A key factor in the design of the archiving system has been flexibility. We wish to encourage collaboration with other labs in this effort. There are EPICS users with their own data-taking engines, and we want to ensure that their data is viewable with the common data viewer. Similarly, some sites may have a strong affinity to a particular data viewer, but may need a different data acquisition tool. Our goal is to make it as simple as possible for others to integrate their components, with a robust, well defined interface.

In an effort to facilitate more automation and intelligent data manipulation at the Unix level, development of the Automator, a configurable tool for monitoring and responding to changes in the control system using a CDEV server, is in progress. The framework of communicating with existing signals and creating derivative signals has been tested, and logical blocks to process the incoming data are under development. A beta version of this will be available in a few months.

The newest project, databases to manage data in both the operations and software development environments, is just beginning. Our efforts now are to evaluate our needs and see what can be reused from databases at other EPICS sites. We expect this to be a great opportunity for Jefferson Lab to collaborate with other sites in the continuing evolution of our control system.

REFERENCES

[1] R. Bork, et al., "CEBAF Control System" CEBAF-PR-89-013.

[2] M. E. Thout, L. R. Dalesio, "Control System Architecture: The Standard and Non-Standard Models", Proceedings of PAC93, Washington D. C.

[3] L. R. Dalesio, et al., "The Experimental Physics and Industrial Control System", Proceedings of ICALEPCS93, Japan

[4] K. White, et al., "The Migration of the CEBAF Accelerator Control System from TACL to EPICS", Internal Jefferson Lab Review, June 1994.

[5] S. Schaffner, et al., "Tools for Application Management at Jefferson Lab, Proceedings of ICALEPCS99, Trieste, Italy.