

CLS: A FULLY OPEN SOURCE CONTROL SYSTEM

E. Matias, R. Berg, T. Johnson, R. Tanner, T. Wilson, G. Wright and H. Zhang
Canadian Light Source, University of Saskatchewan, Saskatoon SK, S7N 0X4 Canada

Abstract

The Canadian Light Source (CLS) is one of the first major accelerator facilities to adopt a fully open source control system. The control system is based on Experimental Physics and Industrial Control System (EPICS) in use at many other facilities. From the outset CLS utilised Real Time Embedded Multiprocessor System (RTEMS) and Linux as the underlying operating systems for real-time control computers, operator interface computers and servers. When communicating with PLC and other intelligent devices CLS has also adopted a policy of using open communications protocols where possible. Combined, these strategies have led to a system that can easily evolve over the life of the facility without being tied to specific hardware or software suppliers. The operational experience over the past few years has indicated the selected architecture is sufficiently robust and reliable.

INTRODUCTION

The Saskatchewan Accelerator Laboratory (SAL) operated from the 1960s to the late 1990s when the pulse stretcher ring was decommissioned and the accelerator was reconfigured as part of the injector for a new 2.9 GeV third generation synchrotron light source [1]. At that time the existing control hardware was replaced and new control software developed. EPICS, RTEMS and Linux were selected as the primary platform for this system.

When selecting third party software, open source software was placed on an equal footing with commercial software and technical capabilities of the software were emphasised. The end result was a control system where the run-time environment was nearly completely open-source based and a significant portion of the development tools were also open source.

MAJOR OPEN SOURCE SOFTWARE COMPONENTS AT THE CLS

EPICS 3.14 beta 1

EPICS is in use or planned for accelerator facilities in North America, Europe, Asia, the Middle East and Australia. This widespread adoption and acceptance of EPICS was a key motivation in selecting EPICS at the CLS. By selecting EPICS, CLS has benefited from being able to reuse a significant number of libraries, drivers, and application programs.

EPICS core builds on contributions from various organizations, most notably Los Alamos and the Advanced Photon Source at Argonne. In addition CLS has reused the following EPICS extensions: (1) Extensible Display Manager (EDM) and the Channel Archiver from the Spallation Neutron Source (SNS); (2) a

modified version of the Knob Manager from Los Alamos and (3) LabCA from Stanford.

One of the challenges faced in the use of EPICS has been compatibility of applications originating from other laboratories and configuration control. In several cases applications are written for a specific version or series of versions of EPICS, care is required when integrating, testing and deploying these applications. These difficulties will hopefully be mitigated by the recent publication of the Channel Access (CA) Protocol and ongoing work by the developers of EPICS core to control changing and maintain backwards compatibility.

RTEMS

Single board computers (SBC's) originally developed for use at SAL were chosen for use at the CLS [2]. RTEMS support existed early for these Input/Output Controllers (IOC), and is now used for IOC applications running on these SBCs. The earliest RTEMS based IOC was installed in 2000. Today there are in excess of 150 IOCs within the facility based on RTEMS.

RTEMS has proven to be a very stable and mature platform. A version of RTEMS was selected for the Motorola 68360 based IOCs in 2000. However, each upgrade to EPICS (3.14 alpha1 to 3.14 beta1, and 3.14.7 is anticipated in the near future) has necessitated an upgrade to RTEMS.

The CLS was the first facility to use EPICS on RTEMS in a significant production environment. The price paid was the necessity of redeveloping software that other facilities used exclusively on VxWorks, impacting the benefit of drivers developed at other facilities.

CLS is in the process of deploying RTEMS on the Intel based IOCs for real-time control of VME based equipment.

Linux

Linux was selected as the primary development and production platform. Many other facilities were in the process of increasing their Linux Operator Interface (OPI) support, and Linux could be used as an OPI platform, as an IOC platform, and also as a general server platform for the control system. The dedicated servers provide network services, EPICS services, such as Process Variable gateways and channel archiving. Linux Operator Interface (OPI) computers are provided in the control room with quad-screen displays. Linux IOCs perform such tasks as communicating with Modicon PLC's via the Ethernet, or running orbit correction on a VME-capable PC, or running beamline stepper motors.

Mixed results were experienced with Linux. Through the build project there was an ongoing need to upgrade

Linux distributions to support newer PC hardware. This was a costly activity.

There continues to be an assortment of compatibility issues between different releases of Linux.

Matlab Accelerator Toolkit

Matlab is used as a platform to interact with the machine using several tool sets, these toolsets were the result of a collaborative effort between ALS and Spear of the past several years. Accelerator Toolbox (AT) [3] is used to model the storage ring lattice and perform simulations. LOCO [4] is an algorithm for debugging linear optics in the storage ring. The Middle Layer [5] is a set of scripts and functions that allow the accelerator tools above it to become machine and control system independent. Matlab Channel Access (MCA) [6] is used to connect the other tools to the EPICS control system. Matlab served as an indispensable platform during commissioning of the CLS. Many key measurements such as transverse orbit response and quadrupole centering were performed using this package.

The accelerator tool box, LOCO, Middle Layer and MCA are open source. Matlab is of course a commercial product. Matlab was selected primarily to take advantage of the rich toolset for accelerator control that has been developed at ALS and SPEAR III. Matlab has since been applied as a scripting tool to aid in the commissioning of beamlines.

Glade

Glade is a general purpose graphical user interface environment on the Linux platform. With a few modifications to support EPICS calls, Glade was able to be used to build operator interface applications on both the accelerator and beamline environments.

The motivation for the introduction of Glade was the need to do extensive file handling and editing of tables of machine configuration data. An example of one of these applications is the accelerator timing system, where operations staff need the ability to load, save and edit ring fill patterns.

freeTDS

CLS uses a series of Microsoft SQL servers as the basis for the facilities central relational database systems. These servers support the financial, stores, engineering, maintenance, work management and user office systems within the facility. Over time it has become advantageous to develop tighter integration between the control system and the other work management systems. To achieve this integration a means is needed to interface with Microsoft SQL servers.

Borland Kylix is in use to develop custom desktop applications for both the controls and beamlines where database integration is required. This was achieved through a Linux ODBC library and the freeTDS package [7] that communicated with the Borland Kylix dbExpress interface. The existing Borland Kylix widgets were automatically extended to use the additional database

interface, allowing custom code development that runs on either a Windows system or a Linux system.

Due to recent moves to discontinue the Borland Kylix development environment CLS is examining alternative compilers for freeTDS based applications.

SYSTEM ARCHITECTURE

The CLS control system architecture is based on a distributed control system. The heterogeneous collection of computers is interconnected predominately using Ethernet. Modbus communication is transmitted over Ethernet. Some data-links over RS232 and Profibus can also be found in the system.

As shown in Figure 1, there is an implicit hierarchy to the system. Operator Interfaces (OPI) at the top tier provides a means for the operator to interact with the system. Input Output Controllers (IOC) hardware provides a mapping between device level hardware and the operator interface. The device layer implements tight loop control and controls field equipment. Additional more general support equipment such as servers, timing systems and the networking infrastructure support the various layers in the control system.

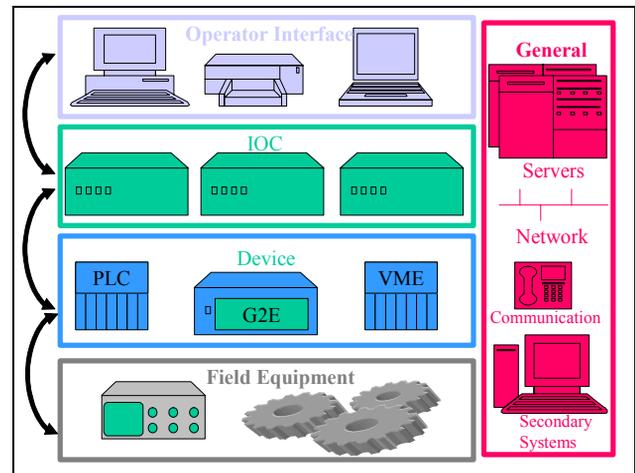


Figure 1: CLS Control System Architecture.

GENERAL SERVICES

Servers

Dell PowerEdge servers running RedHat Linux 7.2 are used to provide control network services. The main server for the control network has been in continuous operation for nearly five hundred days. Some file server functions are provided by a Storage Area Network. Operational experience suggests this to be a robust and reliable platform.

Network

Several Virtual Local Area Networks (VLANs) are in use at the CLS running on a common hardware layer. The network based on CISCO technology with redundant core switches. The network infrastructure is shared

between the accelerator control system, beamline control, voice over Internet Protocol telephones, video and office networking needs.

Secondary Systems

There are some auxiliary secondary systems within the facility that are not integrated in the main control system. These systems provide control over the security system, video surveillance, some building ventilation systems, and standalone diagnostic systems. These systems are run by individual computers connected to the main control room.

OPERATOR INTERFACE (OPI)

Workstations

Workstations on the accelerator systems are Dell Optiplex hardware running RedHat Linux 7.2 while beamline workstations are running Fedora Core 1 Linux.

The use of personal computer (PC) hardware and the Linux operating system has generally resulted in a robust and reliable environment. To reduce testing and maintenance cost a single version of Linux (Redhat 7.2) was used. The use of Fedora Core 1 Linux was driven by the need to support slightly more modern hardware. The need to upgrade operating system to support newer hardware continues to be one of the fundamental challenges in using commodity hardware.

CLS is currently in the process of upgrading all of the user workstations back to a single version of Linux, likely based on Scientific Linux.

Touch panels

Two types of touch panels are in use. Siemens touch panel connected directly to PLC equipment is used in the cryogenics plant and some RF applications. CLS is also in the process of deploying Linux based touch panel screens that are true EPICS OPI devices running EDM screens.

INPUT OUTPUT CONTROLLERS (IOC)

Embedded Real-Time Output Controllers (EROC)

EROCs are Motorola 68360 based single board computers running RTEMS. The EROC was originally design for SAL. When the CLS project was started the board design was licensed to a local firm that then sold the hardware back to CLS.

When refurbishing the Linac many of the existing power supplies were also refurbished. These power supplies were upgraded to include embedded IOC hardware directly integrated into the power supply regulation and interlock circuits.

Many of the stepper motor controls in use at the CLS also make use of embedded EROC boards. The many devices controlled or monitored by RS232 are generally connected to EROC boards, and makes up the majority of EROC systems installed.

IOC Interface to VME Equipment

Connection to VME hardware at the CLS is not accomplished with a CPU board located inside each crate, rather a pair of fiber optic transceivers (1 PCI and 1 VME) [5] are used to connect a single PC computer to 1 or more VME crates. The fiber optic link is achieved by using a pair of cards SIS3100(VME) and SIS1100(PCI) installed into their respective back planes and connected with a fiber optic cable. Once connected, a Linux application is able to map the memory locations of any VME board installed in the VME crate into the application memory space of the Linux application. This allows the Linux application to access the registers or FIFO's of a particular board installed in a VME crate simply by reading a pointer to a memory location. The actual transmission and reception of data across the fiber optic link is transparently taken care of by the SIS1100 PCI driver. Figure 2 shows how a single EPICS application running on a Linux IOC is able to read and write data to 3 different boards located in 3 different VME crates.

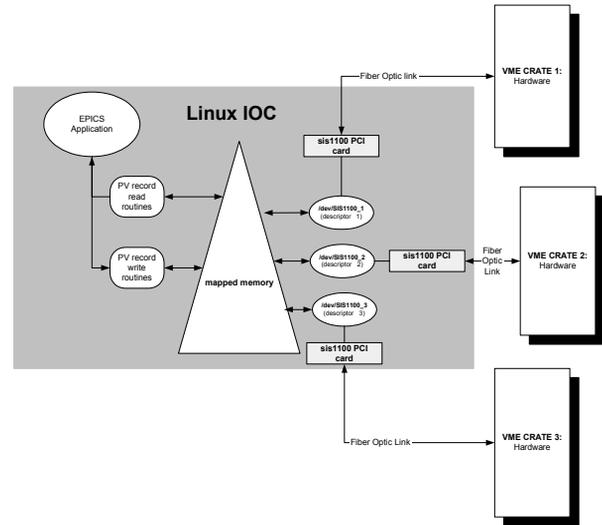


Figure 2: CLS Control System Architecture.

Work is ongoing on converting some of these IOCs to RTEMS where real-time requirements cannot be met by the standard Linux kernel.

DEVICE LEVEL

Programmable Logic Controllers (PLC)

Modicon Momentum, Siemens S7/300 and S7/400 PLC hardware for used for machine protection, mechanical services and RF systems. The IOC interface to the Modicon Momentum PLCs is accomplished using Modbus over Ethernet. The IOC interface to the S7/300 and S7/400 PLC hardware is accomplished using Profibus networking and a Profibus interface board. The CLS control system makes use of nearly thirty PLCs within the facility.

Other Device Level Equipment

GPIB devices are integrated into the control system using a GPIB to Ethernet adaptor. The IOC then communicates with the G2E device via Ethernet.

VME crates are connected to industrial PC hardware and have no active processing.

Serial devices are connected either directly to EROC hardware or to RS-232 to Ethernet adaptors that are then mapped into EPICS by an EROC or Industrial PC.

OPERATING EXPERIENCES AND LESSONS LEARNED

Reliability and Availability

The control system has demonstrated a high degree of reliability and availability. A very limited number of problems were found with RTEMS and EPICS during the early commissioning stages. The IOC computers have proven to be very reliable, and problems with OPI computers have proven to be minor. The main control system server has been in operating continuously for over 1 ½ years.

Configuration Management

One of the challenges in the use of open source is that there is an increased burden on the configuration management system. This is primarily a result of a significant increase in the number of source code files under management.

MKS Source Integrity Enterprise was adopted for configuration management. This package was primarily selected because of the integrated work-flow management capabilities and change-package tracking.

Layered and Interdependent Software

One of the difficulties encountered with the CLS control system has been managing the interdependency of different layers of software that make up the control system. It is not unusual for an application program to be dependent on a specific version of EPICS, for a specific version of EPICS to be dependent on a specific version of an operating system, and a specific version of an operating system to be dependent on a specific hardware configuration. This layering and interdependency problem would be encountered with either an open source based system or a commercial system. What must be stressed is the importance of release management and regression testing in dealing with issue.

Managing Diversity

In comparison to many other facilities CLS has a small development team. In order to limit the resources needed to deploy and maintain the control system discipline is required in the selection and adoption of new hardware and software components or packages. When new components were being considered for inclusion into either the accelerator or beamline system an assessment

would be made to determine that it truly added new necessary functionality. A strong emphasis was placed on commonality of design across the entire facility.

Management of Evolution

One of the difficulties faced in maintaining a control system based on commercial software is the degree of dependence that one develops on the vendors product life cycle. This can sometimes drive upgrades and even complete redesigns when the product is discontinued or phased out. CLS has already experience some difficulties in this respect with commercial control software that has been used at the periphery of the control system. One of the strong points in the use of open source is that the lab has far more control over the time line for evolution and upgrade of software components. This becomes increasingly important these system become more interdependent.

CONCLUSION

It was not the original intent at the beginning of the CLS build project that a fully open source control system be developed; nonetheless this was the outcome. The design has been the result of the careful selection of software components based on their ability to meet functional, reliability and maintainability requirements. The final resulting system has proven to be robust and positioned CLS well going into operations.

REFERENCES

- [1] L. Dallin, R. Berg, J.C. Bergstrom, X. Shen, R.M. Silzer, J.M. Vogt and M. de Jong, "Canadian Light Source Status and Commissioning Results", EPAC 2004, p. 2269.
- [2] E. Norm, "CLS Embedded Controller User's Manual", CLS Document 7.9.39.2. CLS Saskatoon Canada, May 2000.
- [3] A. Terebilo, "Accelerator Modeling With Matlab Accelerator Toolbox", SLAC, Menlo Park, CA 94025, USA. PAC 2001
- [4] J. Safranek, G. Portmann, A. Terebilo, Matlab-base LOCO", SSRL/SLAC, Stanford, CA 94309, USA C. Steier, LBNL, Berkeley, CA 94720, USA EPAC 2002
- [5] G. J. Portmann, J. Corbett, and A. Terebilo. "Middle Layer Software For Accelerator Control", SSRL March, 2004
- [6] A. Terebilo, "Channel Access Client Toolbox for Matlab", SLAC, Stanford, CA 94025, CA
- [7] B. Bruns and J. Lowden, "FreeTDS A Guide to Installing, Configuring and Running FreeTDS", 2004, <http://www.freetds.org>.