THE FERMILAB ACCELERATOR CONTROL SYSTEM*

James Patrick#, Fermilab, Batavia, IL 60510, U.S.A.

Abstract

The Fermilab accelerator complex supports simultaneous operation of 8 and 120 GeV fixed target lines, a high intensity neutrino source (NUMI), antiproton production, and a 1.8 TeV proton-antiproton collider. Controlling all of this is a single system known as ACNET. ACNET is a based on a highly scalable three tier architecture and features a central database, extensive parallel data logging, security, accountability, a states facility, and a flexible sequencer for automation. In recent years the system has been enhanced to support the increasing operational demands of the current run, and also to reduce dependence in the upper layers on the obsolete VAX/VMS platform. A Java based infrastructure has been developed, and is now used for most middle layer functionality as well as some applications. A port of most of the remaining VMS code to Linux is nearing completion. Key features of ACNET are described, as well as experience with recent migration efforts, which had to be done without interrupting accelerator operation.

INTRODUCTION

The Fermilab accelerator complex consists of a 400 MeV linac, 8 GeV booster, 120 GeV Main Injector, 980 GeV Tevatron, an anti-proton collection facility, and an 8 GeV anti-proton "recycler" ring in the Main Injector tunnel. In March, 2001 a Tevatron collider run ("Run II") began with substantial upgrades from the previous 1992-96 run. In the past, Fermilab had never mixed collider and fixed target running. However, in late 2001 an 8 GeV fixed target experiment ("miniBooNE") began operation, followed by 120 GeV fixed target experiments in early 2003, and the NUMI neutrino beam in late 2004. The control system was required to support all these operation modes simultaneously. Also, at the start of the run all high level software ran on the obsolete VAX/VMS platform. An additional challenge during the run has thus been to migrate the control system to more modern platforms.

CONTROL SYSTEM OVERVIEW

The Fermilab control system is known as "ACNET". ACNET is a unified control system for the entire complex, controlling all accelerators and technical equipment. It was originally developed for the start of Tevatron operations in 1983. Over the years it has undergone substantial evolution in field hardware, computing platforms, and communication infrastructure. ACNET follows a 3-tier model, with application, central, and front-end layers, as shown in Fig. 2.

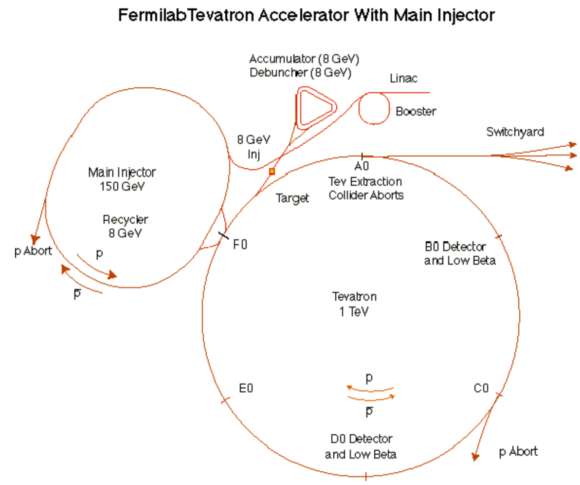

Figure 1. Fermilab Accelerator Complex

DEVICE MODEL

The ACNET system employs a flat device model with names restricted to 8 characters. Each device may have one or more of a fixed set of properties including reading, setting, digital status and control, and alarms. Reading and setting properties may be single values or arrays. While mixed type arrays are not transparently supported, this is often done with the required transformation done in library code. Device descriptions for the entire system are stored in a central database. There are approximately 200,000 devices with 350,000 properties in the Fermilab system.

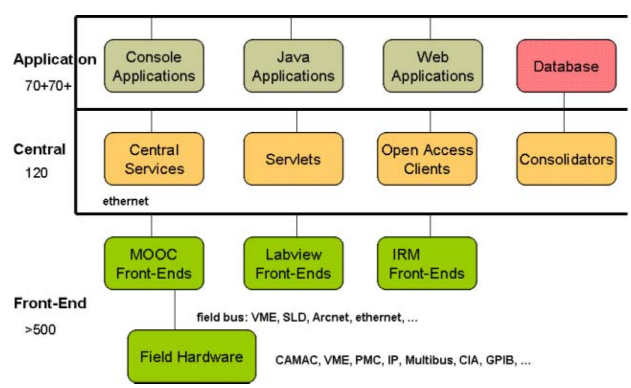

Figure 2. Control System Overview

COMMUNICATION

Communication among the various parts of the control system uses both general and fast real-time networks. General communication flows over Ethernet and uses a custom protocol also known as ACNET. This is a layered, task-directed protocol developed in the early 80's. It supports standard get and set device requests, with either

*Work supported by the Universities Research Association, Inc. under contract No. DE-AC02-76CH03000 with the U.S. Department of Energy.
#patrick@fnal.gov on behalf of the Fermilab Controls Department

a single or periodic return. In addition, a “Fast Time Plot” protocol support high data rates while minimizing network traffic. Readings acquired at rates up to 1440 Hz are grouped into blocks and returned by the front-end every 0.5 seconds. Also, a snapshot protocol forms blocks of up to 2048 readings at a time taken at any rate supported by the front-end hardware. Although ACNET was developed long ago, it does have many features found in later standard protocols. It has proven to be very robust, efficient, and extensible and has continued to work well as the control system has grown over the years. The primary deficiency is a lack of support for automatic translation of mixed type data across different platforms.

Real-time information is passed over several dedicated networks. Clock events (TCLK) are used to trigger events around the complex. A Machine Data (MDAT) link transmits 36 frames of data at 720 Hz to control machine ramps as well as other functions. Beam Sync signals derived from RF systems provide precise timing references where needed.

APPLICATION ENVIRONMENT

There are two major frameworks for applications, the original VAX based “console” framework and a newer Java based framework. Both environments provide all applications with a standard look and feel, and a single place from where all applications may be launched, known as the “Index Page”. In addition, in recent years some utility applications have been developed that run entirely in a web browser. There are generic applications for monitoring sets of devices (“Parameter Page”), plotting live and archived data, manipulating the device database, alarms, and synoptic displays. Many of these have console, Java, and web applications. Figure 3 shows the Parameter Page in the Java application framework. Most machine specific applications still run under the console framework. This is in the process of migration from VAXes to PCs running Linux and is discussed later in this paper. Java applications are run under Windows, Linux, Sun/Solaris, and MacIntosh platforms.

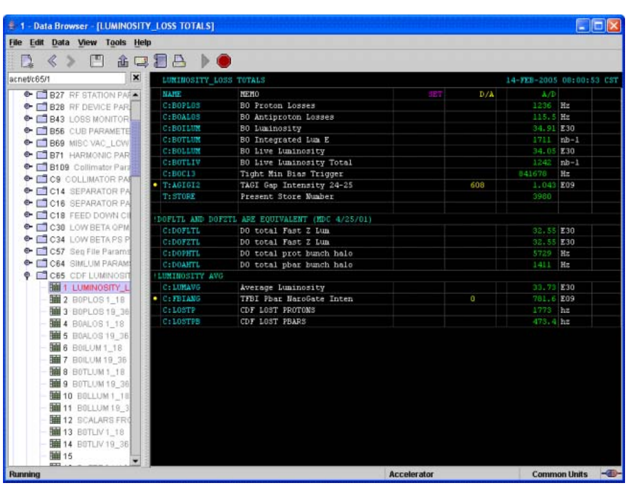

Figure 3. Parameter Page in Java framework

CENTRAL LAYER

The central layer contains a database, a number of persistent processes and some Java servlets to support web based applications.

Sybase is used for the primary database. It contains all device and node information, application data, save/restore data, and Tevatron shot data. MySQL is used for data logging and some other specialized purposes.

Most central services are in the form of “Open Access Clients” (OACs). These are virtual front-ends, that is they obey the standard ACNET communication protocol. However they run on a high level operating system and so have easier access to the database and other system features at the cost of hard real-time response. There are several classes of OACs:

- Utility OACs such as data loggers, scheduled data acquisition, virtual devices
- Calculational OACs that perform computations on devices across different front-ends
- Consolidation of related devices from different front-ends into a single array device
- Pulse to pulse feedback for fixed target lines
- Bridges to Ethernet connected instrumentation

There are over 100 OACs in the system; nearly all now run in a newer Java framework. Hardware platforms employed are both Sun Netras running Solaris, and PCs running Linux.

FRONT-ENDS

Front-ends interface the control system to field hardware. There are three primary types in the system:

- MOOC – Minimally Object-Oriented Communication. This framework runs in VME or VXI Power PC based processors running the VxWorks operating system. Support is provided for a wide diversity of field hardware in many form factors including CAMAC, VME, VXI, Multibus, CIA, and Ethernet connected. There are about 350 MOOC front-ends in the system.
- IRM – Internet Rack Monitor. This is a turn-key front-end system with a processor, 64 channel ADC, DAC, and digital I/O channels. There are about 125 IRMs in the system.
- Labview – Labview is used for one or few of a kind instrumentation. It interfaces with the main control system via the standard ACNET protocol and is controlled by standard applications. Labview GUIs are generally only used by instrumentation experts. There are about 25 Labview front-ends in the system.

SEQUENCER

The Sequencer is an accelerator application that automates the very complex set of operations required to operate the Tevatron collider, as well as other parts of the complex. Operations are divided into “Aggregates” for each major step, such as "Inject protons", Inject pbars", "Accelerate", etc. Operational steps in each aggregate are described in a custom command language. This is easily readable and modifiable by machine specialists. Commands may set or check devices, wait for conditions, execute a script, or start a program to perform more complex functions.

The Sequencer application allows the operator to run complete aggregates, single step them, and restart from failed commands. In addition, an editor is provided that simplifies creation of aggregates by giving the operator a list of commands and options to select from.

Distributed Intelligence

A philosophy of distributing intelligence in the control system greatly simplifies the work that must be done by the sequencer. There are separate sequencer instances for the Tevatron, Main Injector, and pbar sources. Many fine details are performed by lower level programs; the sequencer sends only general instructions to them. Custom function generators are used which have preloaded all possible required waveforms, the appropriate function is triggered by specific clock events.

States

The control system has a "States" facility that is used for synchronization among the various pieces during operation. A State value is contained as a virtual ACNET device owned by a STATES OAC. Any element in the system may request setting a state device. This setting is reflected to the control system via Ethernet multicast, and by setting a list of listener devices stored in the database. Synchronization among the various sequencer instances uses states, also states are used by the sequencer to transmit information to lower level programs.

SDA

SDA is a system that collects and analyzes data for Tevatron collider shots. It simultaneously refers to Sequenced Data Acquisition for the collection part, and Shot Data Analysis for the analysis part. A dedicated OAC stores accelerator data at each stage of the injection process, and every 10 minutes during a collider store. This allows computation and tracking of transfer efficiencies, emittance growth, peak luminosity etc. A number of web accessible reports and plots are automatically generated during this process. A variety of tools exist including custom applications, Excel, and Java Analysis Studio [1] that may be used for more detailed analysis of the data.

DATA LOGGING

To accommodate the large and diverse needs of the complex for archiving data, a parallel strategy is employed. Seventy data logger instances run in parallel on separate service layer computers. Data are stored in MySQL databases used as circular buffers on each node. One or more are assigned to different departments to use as they wish, there is no central management. Device data may be collected at fixed rates up to 15 Hz, or on clock or state events. Currently about 48,000 different devices are logged. To avoid losing data when a circular node database wraps around, each day data at a maximum rate of 1 Hz is extracted to a "backup" logger on a dedicated machine and stored indefinitely.

To view the archived data, a variety of tools exist. There are standard plotting applications in both the console and Java frameworks, web applications, Java Analysis Studio [1], and APIs with which custom applications may be written.

SECURITY

The control system is on a dedicated network that is inside a firewall that restricts access both in to and out of it. In such a large and diverse system, it is also desirable to control setting capability even among accelerator personnel. To accomplish this, "roles" are defined and applied to both people and remote consoles. Specific devices can only be set by those with certain roles. Furthermore, applications run from outside the main control room start with settings disabled, they must be manually enabled by the user. These features reduce the probability of accidental settings disrupting operations.

In addition to trying to prevent inadvertent settings, the control system saves a considerable amount of usage information to aid in diagnosis of problems that might occur. This includes all settings, device errors, database queries, application usage, clock events and state transitions. Most are viewable via web-based reports.

REMOTE ACCESS

While access to the control system is restricted by the firewall, several methods are provided for remote access. A Java Applet version of the console application index pages is available. A subset of applications are permitted to be run this way, however settings are always disabled. Special web browser applications have been written that allow access to parameter pages, data logger and SDA data, and the device database among other things. And finally a dedicated API, the "Secure Controls Framework" [2] has been developed for use by standard Java applications. This provides better performance and functionality than the web applications. Authentication by the standard Fermilab Kerberos system is required to connect to the control system by this method.

MIGRATION

Java Framework

By the late 1990's, nearly all VAXes at Fermilab had been shut down except for those in the accelerator control system. Whereas the complex was projected to run until at least 2010, some migration strategy away from the VAXes was essential. Not only has the hardware been unsupported for years, little third party software is current as well. As no long shutdowns were planned, this strategy would have to be incremental, allowing piece by piece replacement of the control system. As old and new parts would thus have to interoperate, the ACNET protocol would remain as the primary communication mechanism.

An architecture based on the Java language was developed, with the previously mentioned application and OAC frameworks. By the start of the collider run in 2001, the basic infrastructure was in place. The first major application in the new framework was SDA. Over the first few years of the run, most VAX OACs were converted to the Java framework. Also a number of Java applications written, some based on VAX applications as well as some with new functionality.

Console Framework Port

Progress on converting the VAX application base to Java proved to be slow for a variety of reasons. In late 2003, a division wide group was formed to assess the controls migration strategy. It was determined that the based of active VAX applications was more than 800 and it was not practical to convert all to Java with the available resources. Therefore a project was established in early 2004 to port the VAX console application base to Linux "as is", with no change in style or functionality.

For this port, an incremental strategy was again employed. First the console infrastructure and core libraries were ported. Following that applications were ported and put into operation following testing and certification. A dual-hosted console manager was developed to start an application on the proper console platform (VAX or Linux) enabling certification to be done one at a time. A web-based tracking system was essential to monitoring this very large project. Due to the large code base, application interdependencies, binary data in files and network communication, this project turned out to be more difficult than expected. Also, most of the C code was in the older K&R dialect and required modification to conform to the modern ANSI standard. Despite these difficulties, at this time the control system operates with around 500 applications on Linux. While there has been some impact on operations, this has been very small. Migration of the remaining applications is expected by the end of 2006, and the VAXes are expected to be decommissioned in the spring of 2007.

FUTURE PLANS

The Tevatron collider is scheduled to run for approximately three more years. At that time the focus of the complex will be on increasing the intensity of the NUMI neutrino beam. Parts of the pbar accumulation facility will become Main Injector pre-injectors. The ACNET control system will continue to be used for this upgrade.

Also under construction at Fermilab are various test facilities for the proposed International Linear Collider (ILC) [3]. These include horizontal and vertical superconducting RF cavity test stands, and a small accelerator consisting of an injector and three ILC cryomodules. These facilities will use a combination of the widely used EPICS system [4], and the DOOCS system [5] used at the FLASH (TTF) facility at DESY.

SUMMARY

Accelerator performance has steadily improved since 2001. Peak luminosity has exceeded 2.25 x $10^{32}cm^{-1}sec^{-2}$, a factor of 10 greater than the previous run. Nearly 2 pb^{-1} has been delivered to the collider experiments. Since beginning operation 10^{20} protons have been delivered to NUMI. The control system has successfully dealt with the challenge of multiple simultaneous operating modes. Its unified nature, with solid software frameworks has contributed to this success. And in addition to supporting operations, the control system has been substantially modernized over the course of the run. Incremental strategies were employed allowing switching small pieces of the system to new platforms. This allowed migration to occur both without long shutdowns and with minimal impact on operations. With the Java based framework, and the console software ported to Linux, the ACNET control system should be viable for the projected life of the complex.

REFERENCES

[1] http://jas.freehep.org/jas3/
[2] D. J. Nicklaus and A. Petrov, "Secure Client Tier for the Accelerator Control System", Proceedings of ICALEPCS 2005, Geneva, Switzerland.
[3] K. Saito, "SCRF Test Facilities Toward the ILC", Proceedings of PAC 2005, Knoxville, TN, USA.
[4] http://www.aps.anl.gov/epics/
[5] http://doocs.desy.de/