Fermi National Accelerator Laboratory

FERMILAB-Pub-93/324

Control and Monitoring of the DØ Detector

S. Ahn, J.F. Bartlett, N. Denisenko, S. Fuess, A. Jonckheere, S. Krzywdzinski, L. Paterno, H. Prosper and R. Raja

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

J. Featherly and B. Gibbard

Brookhaven National Laboratory
Upton, New York 11973

L. Rasmussen

State University of New York
Stony Brook, New York 11794

November 1993

Submitted to *Nuclear Instruments and Methods in Physics Research Section A*

Operated by Universities Research Association Inc. under Contract No. DE-AC02-76CHO3000 with the United States Department of Energy

Control and Monitoring of the DØ Detector

J. F. Bartlett, S. Ahn, N. Denisenko, S. Fuess, A. Jonckheere,
S. Krzywdzinski, L. Paterno, H. Prosper and R. Raja
Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

J. Featherly and B. Gibbard
Brookhaven National Laboratory, Physics Department, Upton, NY 11973, USA

L. Rasmussen
State University of New York, Physics Department, Stony Brook, NY 11794, USA

Abstract

The DØ experiment at the Fermilab Tevatron Collider is currently studying proton-antiproton collisions at a center-of-mass energy of 1.8 TeV. The data-acquisition system for the experiment was designed with two independent pathways for communication between the front-end computers, which interface with the detector electronics, and the host-level computers, where the event data is logged and where the experiment is monitored and controlled. The event data path employs multiple, high-bandwidth, unidirectional busses while the control path, which uses standard local area network components, handles all control, configuration setting, alarm, and monitoring communications. This paper primarily discusses the software components associated with the control communication path, with emphasis upon the design philosophy, user interface, database, and networking issues.

1 Introduction

1.1 Physics Goals

The DØ detector[1] at Fermilab was constructed to study proton-antiproton collisions at 2 TeV center-of-mass energies. The principal physics goals of the DØ experiment are the study of high mass states and large P_t phenomena. These include the search for the top quark, precision studies of the W and Z bosons to give sensitive tests of the Standard Electroweak Model, various studies of perturbative QCD and the production of b-quark hadrons, and searches for new phenomena beyond the Standard Model.

1.2 Control System Structure

There are two independent pathways for communication and data transfer between the detector and the host data-acquisition computer cluster as shown in fig. 1. Physics event information passes from the detector digitizer buffers over eight parallel, high-rate, unidirectional data highways to the level-2 filter, which is composed of 50 VAX processors, and thence to the host cluster for recording on 8mm video tapes. The bi-directional polling of detector hardware elements and front-end software processes, the collection and distribution of control events (including alarms), and the downloading of run conditions and tables to the front-end processors and settings to the detector electronics uses a completely independent pathway.

2 System Development Environment

2.1 Design Methodology

The control system was designed using Yourdon's Structured Analysis/Structured Design methodology[2] (SASD). The context, data-flow, and entity-relationship diagrams from Structured Analysis and the state-transition diagram from Structured Design were the principal graphical tools used. Early experience using the SASD methodology was very encouraging and it produced software which, relative to earlier experience, had better internal structure, fewer errors, and shorter development times. However, lack of a robust CASE tool for SASD during the first few years of the project was a major problem in maintaining up-to-date documentation.

2.2 Software Tools

A majority of the control system software is written in the Pascal language using DEC extensions. Pascal was selected for its readability, support of data structures and pointers, and relatively strong typing. Other languages used were FORTRAN and C. The host-level VAX computers use the VMS and VAXELN operating systems while the front-end computers use PSOS (M-68020 computers) and DOS (IBM PC computers). Various standards such as X-Windows, MOTIF, and POSIX threads were incorporated as they became available. Some fraction of the current development activity is devoted to retrofitting these standards into processes which were written during the early stages of the project. All source files are kept in CMS-managed libraries which are organized on a project basis. The DØ experiment has an extensive code distribution and management system for maintaining up-to-date copies of source files, object libraries, and executable images on the VAX systems of our highly dispersed, collaborating institutions.

3 Data Acquisition

3.1 Control Data Acquisition Services

The procedures in the Control Data AcQuisition services library[3] (CDAQ) hide the details of the actual data-acquisition and network transport protocols by translating a simplified user's view, which consists of a device with one or more readable and/or settable attributes, to the required access path description via queries to a central hardware database. Requests in the form of multiple device/attribute pairs, which may reference several front-end nodes, are assembled into message packets, called frames, which are destined for the individual front-end nodes. These requests are issued either in single-execution or periodic mode. The CDAQ services interface manages the synchronization and assembly of replies from multiple front-end nodes, arbitrates contention for shared resources, and automates recovery of all allocated resources, both local and network-wide, when a request is deleted or the user's process terminates abnormally.

3.2 Gateways

Requests sent over the Ethernet LAN which are destined for nodes on the token-ring LAN and returning replies pass through one of three network gateways. Each gateway, which is a microVAX computer, is capable of handling 24 independent circuits simultaneously. Since the encoding or byte order of of some data items differ on the two networks, the gateways also perform the appropriate transforms under the control of a format descriptor block contained in each message frame. Some processes, such as the central event distributor, connect to the gateway in one of the multi-cast modes and receive all messages of that class. The peak capacity of a single gateway is between 50 and 80 kilobytes per second depending upon the record size. The peak load on the token-ring LAN occurs when trigger conditions are changed between runs and is approximately 30% of its capacity.

3.3 Front-End Computers

The detector electronic systems are interfaced to front-end computers which are either Motorola 68020 processors in VME crates[4] or IBM PC processors. Some detector sub-systems are connected directly to the VME bus while those located on the detector platform communicate via a MIL-1553B serial link. The 68020 processors can connect to multiple VME slave crates by means of a Vertical Interconnect bus of several meters length.

4 Databases

A central hardware database is used in the monitoring and control of the experiment. The database is in relational form and is implemented under DEC Rdb which offers a variety of capabilities (concurrency control, data integrity, security, data independence, data manipulation language, query optimization) as well as management utilities (RMU, RdbExpert). The database defines all of the essential attributes of the hardware (LAN, node, operation code, data format, rack, crate, slot, limits, status and control bits, alarms specifications, etc.) and provides CDAQ with the information it needs to establish communications with a hardware device and to convey its description to the front-end node in appropriate form for direct access writes and reads. It also contains information on all detector systems, support systems (such as low and high voltage power), cryogenics and argon purity monitoring, and environmental conditioning systems.

The central hardware database also serves as the master copy of the databases which reside in the front-end computers. A separate loader program searches the database for the devices which reside on a particular node and loads the database tables in that node using CDAQ services to perform the settings.

In addition to the interactive SQL provided with Rdb, other facilities are provided for accessing the hardware database, including interactive and batch interfaces. These facilities were written in Pascal, C, and FORTRAN using SQL statements embedded in the host languages. In practice the database is accessed through a detached, server process which has been allocated sufficient resources to permit rapid access to the database files. A read-only copy of the database is maintained and is updated every day from the read-write version. This provides readers with a fast access to the database, uninterrupted by writers.

5 Alarm System

An important section of the monitoring and control system for the DØ experiment is dedicated to the production, distribution, display, and logging of alarms, or, more generally, any events which are significant to the experiment[5].

At a 15 Hz repetition rate the front-end processors compare the current readings to the local database of analog channel nominal and tolerance values or digital nominal bit values. Upon a transition from inside or outside of the allowable range of readings, the front-end generates an asynchronous alarm message containing the node, channel or bit number, the associated hardware device (database name and identifier field), and the reading producing the state change. The message is placed on the token-ring LAN using a group-functional (multicast) code.

The host level application tasks comprising the alarm system are built upon a package of generalized client-server routines. The principal component of the host system is the central event distributor task, configured as a server for clients which may send and/or receive events. The central event distributor maintains a list of all devices which are out of tolerance at the current time. It also monitors special events designated as heartbeats from critical processes and internally generates an alarm when a heartbeat is missing. Other significant event messages, for example the beginning and ending of data acquisition runs, are also distributed via the central event distributor task. The central event distributor task continuously maintains a connection to one of the Ethernet/Token-Ring gateways to receive event messages which are generated by the front-end computers.

The gateway itself connects to the event distributor as a client capable of generating significant event and heartbeat messages. Other host level client tasks open DECNET connections to the central event distributor to either transmit or request significant event messages. At the time that a client connects to the event distributor it specifies a set of criteria which is used to select the event messages which are sent to that client.

The alarm display task[6], which uses the MOTIF windowing system for its displays, is a particular client which can be configured to receive, summarize, and display significant event messages. The default display summarizes the events in each of several categories. By using the mouse and cursor an operator may obtain detailed information on each individual event including descriptive fields from the central hardware database.

The event logger is another client of the central event distributor which logs all significant event messages to a disk file. An event scan task, which incorporates a filtering mechanism similar to the event distributor, provides selective access to the logged data. The final task in the alarm system is an intelligent alarm processor, currently in prototype stage. This task processes the significant event messages to determine if corrective actions can be automatically taken, and performs such actions. The present version of the process acts to suspend physics data recording when critical devices are found to be out of tolerance and to automatically allow resumption of recording when those devices return to safe values.

6 Application Programs

6.1 High Voltage System

The High Voltage sub-system uses a new computer controlled, VME-based power supply module developed at DØ. The modules have a relatively simple interface which permit setting the demand voltage and several limits for hardware-detected trip conditions. Two levels of Motorola 68020 processors provide a more complex channel entity defined by a sequential state machine which is capable of variable-rate ramps, detection of trip conditions based upon deviations from expected behavior, and the grouping of related channels which must ramp or trip in synchronism. As with other devices in the detector, these modules are controlled and monitored via CDAQ services. An application program, which may have multiple executing copies, controls the various detector HV supplies; however, no two programs are able to control the same channel simultaneously. It has built-in monitoring and logging features and a user-defined macro language for faster and accurate control. A second application program monitors the current and voltage readings of all the HV supplies in the DØ detector, displays the present state of the individual channels as an array of color-coded buttons, and logs out-of-tolerance conditions to a database.

6.2 Clock System

The DØ clock system, which is phase-locked with the accelerator clock, synchronizes triggering of the detector with beam crossings. The clock hardware consist of three module types: a phase coherent clock (PCC), a sequencer, and a selector fanout. The Clock Interface, which controls and monitors the current state of both the PCC and the sequencer modules, is based on the client-server model. Multiple clients can control and monitor the clock settings with potential conflicts resolved by the server. The selector fanout modules, whose settings rarely change, are initialized by the downloading program described below.

6.3 Downloading

Before a run commences, the detector must be downloaded with appropriate hardware device settings for the selected run conditions. A master synchronizing program accesses configuration files that describe the run parameters and converts them into a sequence of commands interpreted by multiple instances of the downloading program which execute in parallel. The downloading program parses these commands and uses CDAQ services to perform the settings. A complete set of run conditions can be downloaded to the front-end computers in 10 to 12 minutes and most run-to-run changes can be loaded in less than 3 minutes.

6.4 Monitoring Programs

There is a variety of other programs which perform either general utility or subsystem-specific monitoring and control functions. All of these programs use the common CDAQ services interface to access devices. Several utility examples are: (1) a program which can periodically monitor any device in the detector and log the readings in a database for later trend analysis, (2) a program which controls the various types of low voltage supplies, (3) several low-level programs which can monitor or control any device which is accessible over the control path and display the device readings as a bar graph or as a time trend on a strip-chart graph, and (4) a program which monitors all front-end nodes to ensure their accessibility from the control path.

Acknowledgements

The authors wish to acknowledge the considerable assistance of members of both the Accelerator and Computing divisions of Fermilab.

References

[1] S. Abachi *et al.*, Nucl. Instr. and Meth. (Fermilab PUB93-179-E) to be published

[2] S. Mellor and P.Ward, Structured Development for Real Time Systems (Yourdon Press, 1980)

[3] J. F. Bartlett, S. Fuess, H. Prosper, User's Reference Manual for the DØ Control Data Acquisition Services, DØ Note 866 (1989) unpublished

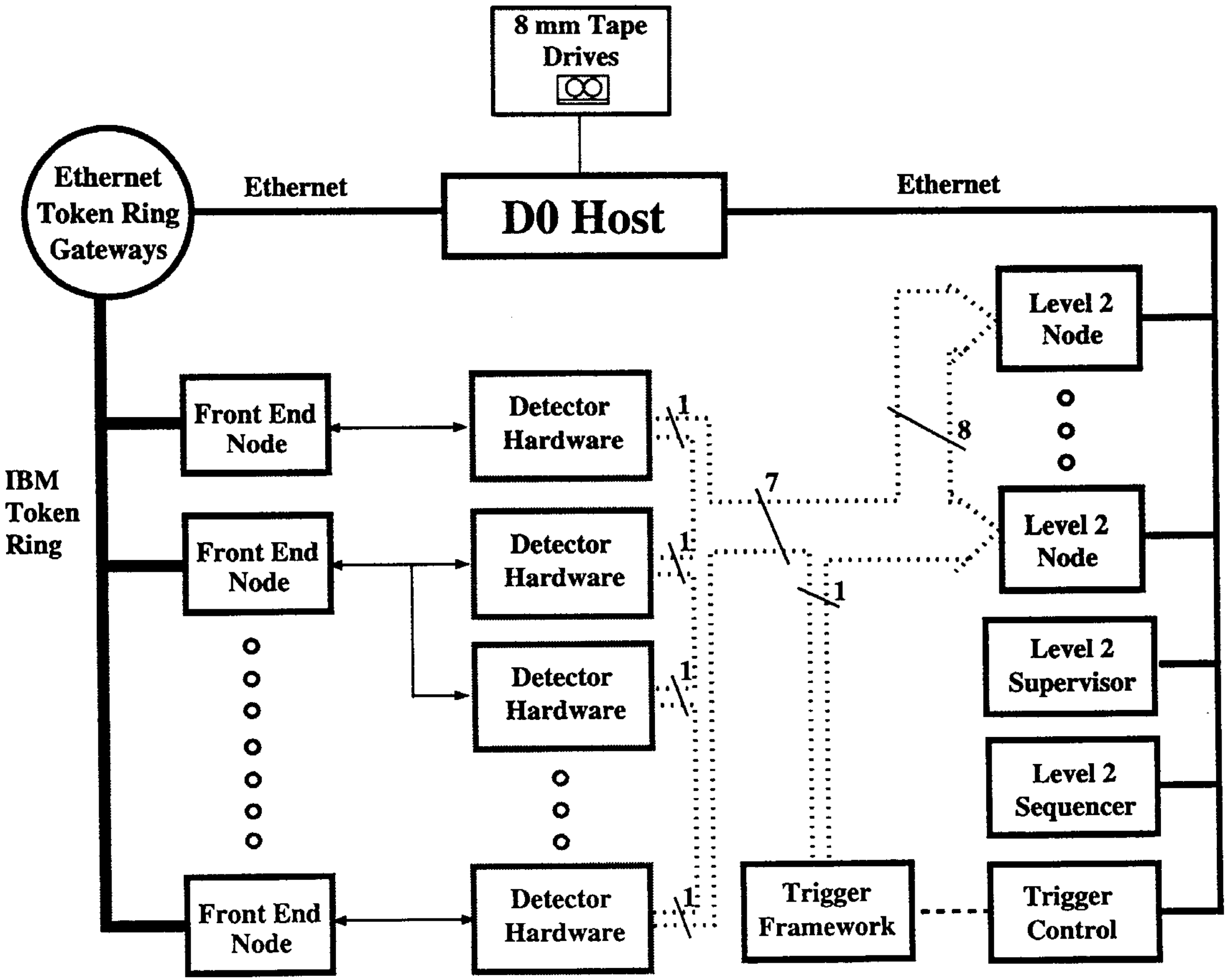
[4] R. Goodwin *et al.*, Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems, p. 125, Vancouver, British Columbia, Canada, October 1989

[5] S. Fuess, DØ Alarm System Programmer's Guide, DØ Note 1558 (1992) unpublished

[6] L. Paterno and S. Fuess, Alarm Display User's Manual, DØ Note 1922 (1993) unpublished

Figure Captions

[Figure 1] DØ data acquisition system configuration showing the separate physics and control communication network paths.


8 mm Tape Drives
Ethernet Token Ring Gateways
Ethernet
D0 Host
Ethernet
IBM Token Ring
Front End Node
Front End Node
Front End Node
Detector Hardware
Detector Hardware
Detector Hardware
Detector Hardware
1
1
1
1
7
8
1
Level 2 Node
Level 2 Node
Level 2 Supervisor
Level 2 Sequencer
Trigger Framework
Trigger Control