The Control System Database for the DØ Detector

Laura Paterno, Stuart Fuess, Stan Krzywdzinski
Fermi National Accelerator Laboratory
Batavia, IL 60510, USA

Harrison B. Prosper
Florida State University
Tallahassee, FL, 32306, USA

ABSTRACT

A central control system database has been used in the monitoring and control of the DØ experiment. The database stores all of the essential attributes of the hardware and provides on-line data acquisition processes with the access information required to read from or write to the hardware. It contains information on detector systems, low and high voltage power supplies, cryogenics monitoring, argon purity monitoring and environmental conditioning. The database serves as the master copy of local databases which reside in front-end computers. It is a relational database, implemented using DEC VAX Rdb/VMS. Utilities were provided for updating and accessing the database, either interactively or in a batch mode. To speed up access in the on-line environment, a read-only copy of the database is maintained and updated every day; detached server processes with sufficient resources provide processes with fast uninterrupted read-only access. Merits and drawbacks of the present system are discussed along with possible enhancements for the future.

1. INTRODUCTION

The DØ detector [1] at the Fermi National Accelerator Laboratory was constructed to study proton-antiproton collisions at a center-of-mass energy of 2 TeV. The detector consists of three nested shells. The innermost shell contains the central tracking and transition radiation detectors; surrounding these is the calorimeter, and surrounding the calorimeter is the muon detector. Combined these detector subsystems consist of ~100,000 channels. The monitoring and control of the channels of the detector as well as collection of physics data was accomplished through the DØ data acquisition system (DAQ), which was designed with two independent data paths for communication [2]. We will not discuss the physics data path in this paper.

Because of the large number of channels to control and monitor, a mechanism was needed to keep track of all the information for each channel: the DØ control system database. The database, known as the Hardware database (Hdb) [3,4], is a relational database implemented using DEC VAX Rdb/VMS. It was designed to provide access path descriptions to Control Data Acquisition (CDAQ) [5] processes as well as nominal settings with their associated tolerances, and descriptions of each device. Currently, the database contains ~5,000 devices which describe the channels of the detector.

The information from the database is downloaded to front-end processors which are either Motorola 68020 processors in VME crates or IBM PC processors. Each processor has its own local database which is updated from the master Hdb database. The local databases contain the access information and alarm conditions for all the devices that the processors monitor and control. The processors use the alarm information to monitor the hardware for failures and notify the on-line data acquisition system when changes of state occur in the system, that is, when a device enters into or leaves the alarm conditions.

In the following sections we discuss the organization of the Hdb database, the utilities used to maintain and access it, the merits and drawbacks of the current database and future enhancements under consideration for the 1999 run of the DØ detector.

2. DATABASE ORGANIZATION

2.1 Structure of the database

The database is relational in form and has a variety of capabilities that the DEC Rdb/VMS environment provides (concurrency control, data integrity, security, data independence, a data manipulation language, query optimization, remote access, etc.) as well as management utilities (RMU, RdbExpert, DECTrace). The database maintains data in a two-dimensional tabular format with 'flat' computer files. A table (or relation) contains a number of rows (or records) which are instances of table elements; the rows consist of columns (or fields) which are data elements and represent attributes of the table.

The Hdb database, as seen in the Entity-Relationship diagram in Figure 1, currently contains 39 tables which describe the hardware devices and their relationships in the detector. The boxed area contains the tables which hold all of the access and alarm condition information for the hardware devices. The other tables are used to describe relationships between the devices.

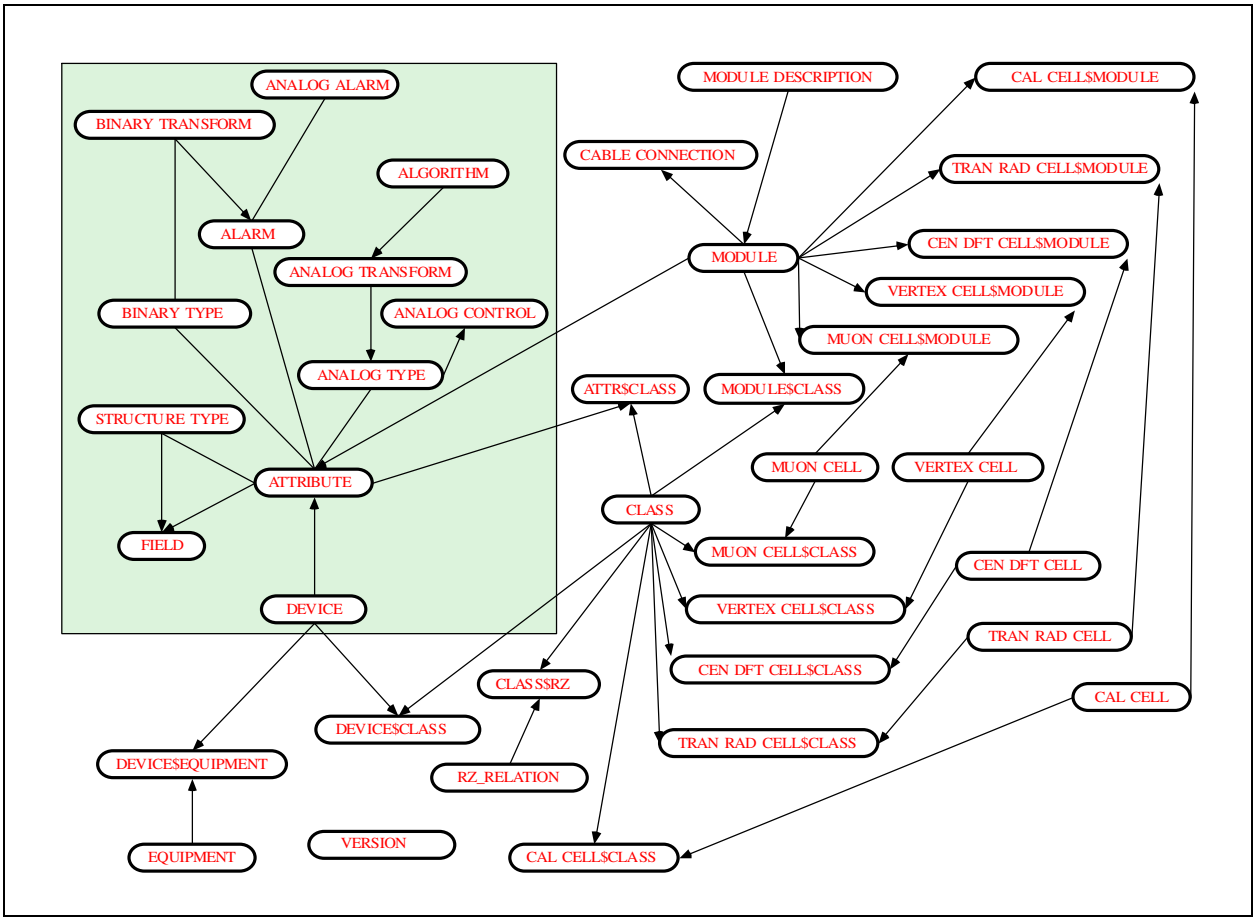

Figure 1 - The Entity-Relationship Diagram of the DØ HDB Database

The individual boxes represent the tables of the database and the connecting lines represent the relationships between the tables. A line with no arrows indicates a one-to-one relationship; a line with arrows means a one-to-many relationship. For example, a device can have one or more attributes associated with it but an attribute can have only one alarm associated with it.

2.2 Devices

Central to the Hdb database is the notion of a *device*. A device may represent any physical hardware object in the detector or in the on-line software. The device may contain any number of monitoring and/or control channels. For example, a power supply is a device because it has voltage or current output which can be monitored and controlled. Only the access information necessary to read and/or write to the physical device is represented in the database. It can (and does) contain some default values used to control the device.

Distinct properties of a device are represented as *attributes*. An attribute may describe analog or binary monitoring and control points of the device. It contains access information which enables CDAQ processes to request readings or settings to devices through the front-end processors. They also have associated alarm information which is loaded into the front-end processors. Devices and their associated attributes were carefully named to provide an easy association between the database objects and the physical devices they represented.

Many of the DØ detector electronics systems contain the same kinds of physical devices (power supplies, temperature monitors, etc.). The monitoring and control channels for these devices are described in the database using structured attributes [6]. The common data for the similar devices was placed into a set of 'elementary devices', which were only entered once. These elementary devices are then referenced by structured attributes of a device through the field table. This made the database more modular in structure and required less data to be entered into the database.

3. UPDATING AND ACCESSING TOOLS

3.1 Updating Tools

Three programs were used to update the Hdb database: HDBatch [7], HDBrowser [8], and HDB Entry [9]. They were written in C or FORTRAN with either embedded RDML or SQL statements used to access the database. The programs provided a variety of services to their target clients including browsing, listing, entry, modification, and deletion of the devices in the database.

The initial loading of the database was done using HDBatch. This utility uses text files which may be created with the HDBedit command which invokes the VAX Language Sensitive Editor (LSE) with a special environment which contains all the HDBatch commands. The files described the devices to be loaded into the database and HDBatch updates the database either interactively or through a batch job running under the VAX/VMS operating system. The utility was written in C and calls low level access routines [4,10,11] to create, delete, modify and list requested devices.

The HDBrowser program uses the VAX Forms Management System (FMS) to display forms whose fields represent data fields in the database. Because the program is interactive, users can see what they are currently entering or have previously entered into the database in a layout which is easy to understand. The program also uses the same low level access routines as HDBatch.

The HDB Entry program was designed to allow easy creation of multiple devices of identical type. It was written in FORTRAN with embedded SQL statements. A portion of the code was generated from the structure of the database. In other words, if the database was changed, the code which dealt with the actual access to the database could be regenerated automatically. The program has both an interactive and batch mode interface. The interactive mode was written using the VAX Screen Management facility (SMG).

3.2 Database Access Utilities

There are two layers of routines provided to access the Hdb database. The first consists of low level routines: create, delete, modify and view procedures for each table in the Hdb database. The routines exist in a shared image library. Thus, if changes need to be made to the implementation of any routine, all clients of that routine automatically get the changes without recompiling or relinking their code. All of the routines are used by the HDBatch and HDBrowser programs; only the view routines are used by the second layer of access routines.

The second layer of routines is called the Database Services (DBS) package [12]. The DBS routines were written in PASCAL with calls to the low level C routines. They were designed to make it easy for the CDAQ processes to

get the information they need. There are also a few routines used by non-CDAQ processes to obtain descriptions of the devices in the database for the purpose of displaying that information. This has helped experimenters to diagnose hardware failures that have occurred during data taking. The Hdb server processes described below also use these routines.

4. SERVER PROCESSES

During the early running of the DØ detector it was discovered that access times to the database were slow. Client processes which downloaded detector constants to the front-end processors often doubled their normal execution times. The problem was traced to writers making frequent modifications to the database. They effectively locked out all processes interested in just obtaining information from the database while they were writing. Fast access to the database also necessitated processes having more system resources and quotas available to them.

Two improvements to the system alleviated these problems and reduced program execution times to acceptable levels. Since restricting modifications to the database was not an option, a read-only copy of the database was created and then updated daily. It solved the problem of writers affecting access times and also served as a natural backup to the master read-write database. The second improvement was the creation of the Hdb server processes. The servers are detached processes which have the necessary resources and quotas. They keep the database permanently opened to clients. Normal DBS access routine calls in a client application are replaced by ones which send messages (via an InterTask Communication (ITC) package [13] layered over DECNET) to the server. The server performs the database access and returns the results to the client process.

5. APPLICATION PROGRAMS

5.1 Downloading

Before data taking (running) commences, the detector control system must be downloaded with the appropriate hardware device settings. 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 the commands and accesses the database through CDAQ routines to perform the settings. Typical cold-start downloading times average 7 to 9 minutes. Run-to-run changes can be performed in less than 3 minutes.

5.2 Monitoring and control

There are a variety of programs which perform general utility or subsystem specific monitoring and control functions. Many of these programs access the database through CDAQ services. Examples include: (1) a program to periodically monitor a set of devices in the detector and log the readings for later trend studies, (2) programs to monitor and control various power supplies, (3) low level programs which monitor and access any device and display the readings as a bar graph or a time trend on a strip-chart graph, and (4) a program that synchronizes the triggering of the detector. Another program, which displays significant event changes in the system, accesses the database directly through the DBS package.

6. ADVANTAGES AND DISADVANTAGES OF THE CURRENT DATABASE

The current database system provided very stable, reliable, and efficient service through the 4 years of DØ running. No data corruption occurred during this entire period which can be attributed to the use of a well documented and supported commercial database. The database server process enabled clients to get quick, read-only access to device information without the need for additional resources in order to obtain fast access times. This helped to improve the downloading times for the DØ detector and reduced the amount of physics data which would otherwise have been lost if downloading times were slower. The use of both a read-write and read-only database enabled writers to update the database without slowing down the readers of the system. This also improved the downloading time as writers may block out access to readers while updates are going on.

There are of course disadvantages to the system as well. Only 12 out of the 39 existing database tables are actually used. This was because not enough manpower was available to update them and the need for the other information was never critical. The database updating programs also have flaws. They are either too slow (HDBatch and HDBrowser), the interface can only be understood by an expert (all), or a full set of services was not provided (HDB Entry).

7. FUTURE CONSIDERATIONS

The present system has been reliable due to the capabilities and data integrity provided by a commercial database product. Experience with non-commercial databases was not always positive. The products experienced data corruption problems and required much more manpower than the commercial solution. A commercial database package with all the present features of DEC Rdb will be a critical requirement for the future database.

The DØ collaboration is considering what upgrades can be made to the experiment. The upgraded detector will have approximately 10 times more channels to monitor and control. This puts strong requirements on database access times to ensure fast downloading of all detector constants. Server access to the database for time-critical clients will be essential in view of the success with the present database server and our decision to switch to a multi-platform environment.

The database itself will be restructured in line with to the new detector subsystems. These new systems will have thousands of identical channels with only slightly different address access. This will require rewriting all the database updating and accessing software. A single integrated tool that is intuitive to the users (hardware experts) and not modeled on the database definitions would be preferable. The tool should be written in a single language with one database query language embedded for ease of maintenance. It will also have to run on the different platforms that DØ supports.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the considerable contributions of the members of the Accelerator and Computing Divisions of the Fermi National Accelerator Laboratory as well as those of the DØ Collaboration.

REFERENCES

[1] S. Abachi, et al., (Fermilab PUB93-179-E) Nucl. Instr. and Meth. **A338** (1994) 185.
[2] S. Ahn, et al., (Fermilab PUB93-324-E) Nucl. Instr. and Meth. **A352** (1994) 250-253.
[3] M. Frey and A. Waller, DØ Hardware Database, DØ Note 821 (1988) unpublished.
[4] M. Frey and A. Waller, DØ Design Notes-The Hardware Monitor Database, DØ Note 824 (1988) unpublished.
[5] J.F. Bartlett, S. Fuess, & H.B. Prosper, User's Reference Manual for the DØ Control Data Acquisition Services, DØ Note 866 (1988) unpublished.
[6] S. Krzywdzinski, Structure Type Devices in HDB Database, DØ Note 1279 (1991) unpublished.
[7] A. Waller, HDBatch: Batch Entry Program for the Hdb Database, DØ Note 915 (1990) unpublished.
[8] M. Frey, HDBrowser: Interactive Entry Program for the Hdb Database, DØ Note 879 (1989) unpublished.
[9] L. Paterno, H.B. Prosper, & R. Raja, The DØ Hdb Entry Program, DØ Note 1512 (1993) unpublished.
[10] M. Frey, Hdb Database User's Guide - C version, DØ Note 822 (1988) unpublished.
[11] M. Frey, Hdb Database User's Guide - FORTRAN version, DØ Note 823 (1989) unpublished.
[12] S. Krzywdzinski, User Reference Manual for HDB Database Services, DØ Note 1280 (1993) unpublished.
[13] J. Featherly, InterTask Communications package (1989) unpublished.