IEEE Transactions on Nuclear Science, Vol. NS-30, No. 4, August 1983

THE USE OF A COMMERCIAL DATA BASE MANAGEMENT SYSTEM IN THE LAMPF CONTROL SYSTEM*

Stanley K. Brown
Los Alamos National Laboratory
Box 1663, MS-H810
Los Alamos, NM 87545

Abstract

The control system for the accelerator at the Los Alamos Meson Physics Facility (LAMPF) is currently being upgraded to run on a DEC VAX 11/780. To unify the several disparate control hardware protocols into one consistent software protocol requires that the hardware devices be addressed logically rather than physically, the physical connections being supplied by the control system. To accomplish this, the control system must have some link between the devices' logical names and their physical connections. This link is supplied by a data base which is managed by a commercial data base management system. This paper discusses the reasoning behind the choice of a commercial system, the particular DBMS chosen and some of the pros and cons of using the DBMS as well as some of our experiences trying to join tools of the commercial world with the real-time world.

Background

The control of the LAMPF accelerator was accomplished through the use of an augmented Systems Engineering Laboratory (SEL) 840 computer. This computer was augmented in several ways: its memory size was increased and several instructions were added to its repertoire. This work was all done locally so that now the 840 is a unique computer. To control the accelerator, all data points are interrogated and all set points are manipulated through actions originated in this central computer. Over the years, the capabilities of the 840 as well as the capabilities of the original data and control system, known as RICE (Remote Instrumentation and Control Equipment), became severely taxed. The data handling and control functions were further augmented by the addition of DEC PDP 11/10's which use CAMAC devices for the instrumentation interface and locally designed CAMAC devices for communication. A certain amount of intelligence was built into these 11/10's so that today the control system supports four distinct and dramatically different instrumentation protocols. To make matters worse, the details of these protocols are usually found only in the computer programs.

To allow for future expansion and to replace the SEL 840, which has become something of a maintenance liability, the control system is presently being converted to run on a VAX 11/780. To provide for a smooth changeover without a dramatic interruption in the accelerator operation, we decided to configure the system to allow control from both the 840 and the VAX. The details of the conversion can be found in [1]. Figure 1 indicates what the present system interconnection scheme looks like. The box labeled DIVA11 is a PDP 11/34 which supports mass storage for the SEL 840. By implementing a VAX port we have access to operating data from either computer. The box labeled RIU-11 is another PDP 11/34 which provides a port for both computers into the RICE Interface Unit (RIU) and ultimately the major portion of the data and control system, the RICE. Also dual ported through this computer is a box labeled MT. This is the master timer, the device which provides all timing signals for the whole accelerator. Finally, there is a box labeled NET11. This is a PDP 11/34 which provides both control computers with access to the previously mentioned PDP 11/10's, of which there are presently five.

*Work performed under the auspices of US Department of Energy

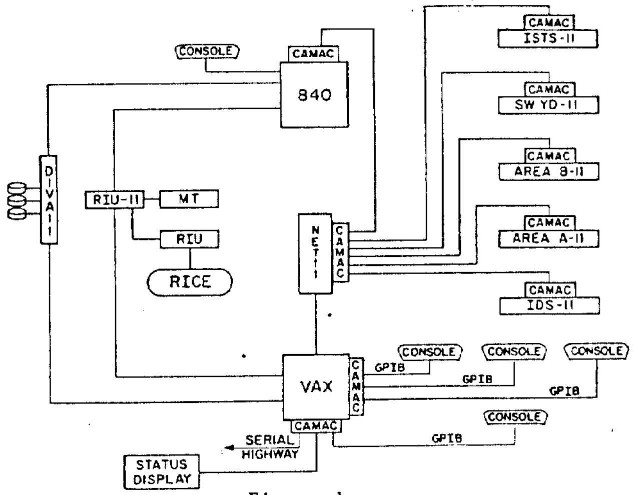

Figure 1

Addressing the problem of the many different instrumentation protocols as well as making the programs independent of the specifics of the hardware in the data and control system, we decided to use the philosophy of an operating system and view the control computer as a computer with approximately 12,000 peripherals. When a user wishes to write information on a disk drive he "opens" a file. The action of opening the file allows the operating system to provide the hardware specific information to read or write the file. This information is obtained from an operating system input-output device "data base." The particular I/O request is routed through the appropriate device driver and the reading or writing takes place. When the file is no longer needed it is "closed," thus informing the operating system that the hardware information can be flushed.

Using this model, the accelerator device hardware information for the 12,000 devices is kept in a data base. Each device is given a "logical" name which acts as the primary key for its retrieval. When a device is "opened," the information about that device is fetched from the data base. Driven by the contents of that device's "record," subsequent operations are directed to the appropriate hardware driver. The accelerator control system opens the data base as part of the boot process and keeps it open, making item retrieval as efficient as possible.

Choice of DBMS

To manage this data, we chose to use a commercial Data Base Management System (DBMS) rather than writing one of our own. This decision was based on a few rather simple points. First, a commercial system has probably had more effort put into it than we could afford, to maximize its throughput as well to make its user interaction simple and intuitive. Second, we would obtain maintenance and reporting features as part of the package. If we had implemented our own system, these would have been features we would have had to provide. On the other hand, we find our application to be much more complex than a typical commercial application. Because it is used to determine how to read and command hardware devices, changes made to the information must be handled at the same time that the system uses the information for control purposes. Further, since a device may be being monitored by several applications at once, these changes must be global in effect. This means the DBMS must support a true multi-user environment while maintaining very user-friendly editing and maintenance capablility. User friendliness is mandatory because it is our intent to have the information maintained by a secretary, unknowledgeable in the ways of computers or control systems. Therefore, our system tends to use the DBMS to its fullest extent, pushing its capabilities beyond what the vendor has probably even tested.

Several months prior to our decision to use a commercial system, a laboratory task-force had been formed and charged with the responsibility to find and choose a commercial DBMS that could be used on both PDP-11's and VAXes and that would fill the laboratory's varied needs, thus to become the laboratory-wide standard. Various requirements were imposed including the availability of fill-in-the-forms video screen input editing capability, a report generator, an interface to the DBMS from some other language like FORTRAN, and multi-user capability. The particular data base system chosen was DRS, a product of Advanced Data Management. DRS supports a data base using the hierarchic model. This is fortunate since our accelerator device data base tends to look hierarchic. DRS was the only DBMS that could meet all the specifications in the Request for Quotation.

Design of Data Base

Each record (a record being defined as the collection of all the information about a single device) in the data base is logically divided into five different parts. These parts are not physically separated but are contained in the same level in the hierarchy. These parts are:

- o Header, containing:
  - Device Name, the main search key
  - Equipment Type, whether RICE equipment, CAMAC equipment, in a remote computer, or such
  - Function, whether it can be read, commanded or both
  - Whether or not it is an analog device
  - What kind of generic device it is, e.g., a magnet, a device on an actuator, etc.
  - What devices, if any, there are that are related in some way to this one
- o Conversion, containing any conversion tables. Converting, for example, from data returned from the Analog Data System to engineering data
- o Data Table, containing:
  - Hardware addresses where the data obtained from a device is found
  - The time, during an accelerator cycle, when the data item is present
- o Command Table, containing:
  - Hardware addresses where commands are sent to command the device
  - How long a command must be held for it to take effect
- o Device Dependent Table, containing:
  - Table to convert data to magnetic field strength if device is a magnet
  - Mounting geometry if knowledge of the device's position is required

Parts of the record exist at a lower level in the hierarchy. Typically, these are multivalued, e.g., the Conversion Table which for a single device may have up to 26 pairs of values to allow for linear interpolation between Analog Data System data and engineering data. A simplified diagram of this structure can be seen in Figure 2.

Applicability of DBMS

The data base just described must provide data to a piece of software which interfaces to our higher level applications. This software, called the Data System, provides subroutine calls which applications programs can use. These calls include an ASSIGN which connects the program to the DATA SYSTEM, an OPEN which fetches the data base record for the appropriate device, a READ which gets data from a device, and a COMMAND which commands a device. Since a given program must be able to open a device and be assured, at that time, that no other program can also control the same device, a locking mechanism must somehow be provided. Once locked, the device must also be unlocked even if the application program crashes. This requires that the DATA SYSTEM operate in a higher level (kernel mode in VMS, the VAX operating system) than that in which the calling program operates. The method provided by VMS to catch just this sort of circumstance was found not to work in kernel mode. It was for this reason that we abandoned the idea of using the data base itself for the actual data that programs could

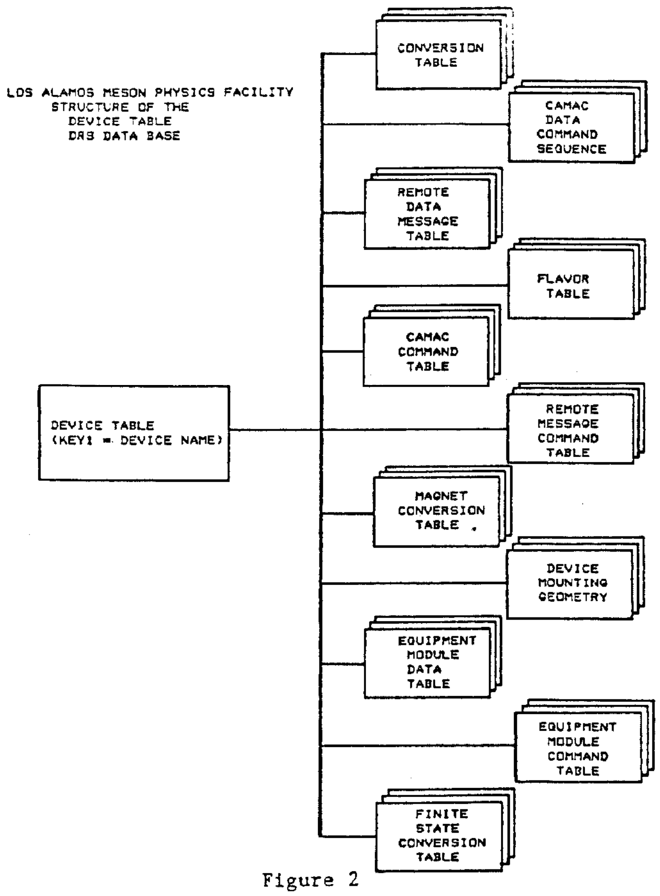
LOS ALAMOS MESON PHYSICS FACILITY
STRUCTURE OF THE
DEVICE TABLE
DRS DATA BASE
DEVICE TABLE (KEY1 = DEVICE NAME)
CONVERSION TABLE
CAMAC DATA COMMAND SEQUENCE
REMOTE DATA MESSAGE TABLE
FLAVOR TABLE
CAMAC COMMAND TABLE
REMOTE MESSAGE COMMAND TABLE
MAGNET CONVERSION TABLE
DEVICE MOUNTING GEOMETRY
EQUIPMENT MODULE DATA TABLE
EQUIPMENT MODULE COMMAND TABLE
FINITE STATE CONVERSION TABLE

Figure 2

retrieve. Instead, we translate the data base into a large pageable array, something VMS calls a "global section." We then wrote special routines which would retrieve the proper data from this global section, making it available to the DATA SYSTEM in its interaction with the program. This circumstance was perhaps fortunate in that later timing studies seem to reveal that the DBMS is far too slow for retrieving records for our application.

Other considerations have led us to believe that the particular choice of DBMS was not the optimal one. Most of the problems we've had difficulty resolving could be directly addressed by better documentation. Consequently, after struggling through the fundamentals, we augmented their documentation by writing a tutorial[2] which is directed at the programmer learning the system. Recently, ADM has attempted to improve their documentation by providing something called an Application Sampler that goes through various steps in data base design and use. Still, as our implementation progresses, we continue to run into things for which we have to discover work-arounds.

On the other hand, all is not negative. The reasons for originally using a commercial DBMS are still valid, and we find that DRS performs quite well in the areas where speed is not an issue. We have found that the ease of casual queries has proved very useful. Ad hoc modifications of data for which more formal procedures are either not available or are clumsy have also proved beneficial. Video screen forms are easy to program and are provided with a great deal of power. We have also found ways of coupling the forms capability with the Report Generator. This provides an intuitive method for obtaining the reports that are needed for informational purposes as well as maintenance of the data.

Having learned the methods of data base generation, creation, maintenance and backup that are unique to our particular DBMS, we find that our original assumptions were correct. Our system is relatively easy to use and flexible enough to support even the most casual of inquiries.

Acknowledgment

The author wishes to thank Phyllis Wallis for the use of her Data Base diagram.

References

[1] Schultz, David E. and Brown, Stanley K., "On-Line Replacement of a Particle Accelerator Control Computer," Proc. IEEE Real-Time Symposium, Dec. 1981, pp.78-82.

[2] Wallis, Phyllis, "Guide to Data Base Operation Using DRS," Unpublished Los Alamos National Laboratory Report. May 1981.