UCRL- 98992
PREPRINT

SEP 1 1988

Distributed Computer Controls
for Accelerator Systems

T. L. Moore

This paper was prepared for submittal to
the Tenth International Conference
on the Application of Accelerators
in Research and Industry
November 7-9, 1988
Denton, Texas

September 1988

Lawrence
Livermore
National
Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISTRIBUTED COMPUTER CONTROLS
FOR ACCELERATOR SYSTEMS*

UCRL--98992

DE89 001130

T. L. Moore
Physics Department
University of California
Lawrence Livermore National Laboratory
Livermore, CA 94550

Abstract

A distributed control system has been designed and installed at the Lawrence Livermore National Laboratory Multi-user Tandem Facility using an extremely modular approach in hardware and software. The two tiered, geographically organized design allowed total system implementation with four months with a computer and instrumentation cost of approximately \$100K. Since the system structure is modular, application to a variety of facility is possible. Such a system allows rethinking and operational style of the facilities, making possible highly reproducible and unattended operation. The impact of industry standards, i.e., UNIX, CAMAC, and IEEE-802.3, and the use of a graphics-oriented controls software suite allowed the efficient implementation of the system. The definition, design, implementation, operation and total system performance will be discussed.

* Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48. Reference herein to any specific commercial products, processes, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply endorsement, recommendation, or favoring by the United States Government or the University of California.

1. INTRODUCTION

The quest for higher quality and repeatable control of accelerator systems can be addressed with a distributed approach to control architecture. Recent industry communication and operating systems standards, coupled with high performance computer systems available at reasonable prices, have brought the required controls technology within the resources of many laboratories.

A distributed architecture approach puts computers and instrumentation close to the devices being controlled, minimizing the investment in cabling plant and improving the overall quality of the signals being monitored and controlled. Computers located at the lower level of the structure allow truly parallel processing in the execution of control algorithms. This organization permits the construction and testing of autonomous subsystems of the control system prior to the actual installation of the major system components.

This paper describes concepts and implementation of the distributed control system in place at the LLNL Multi-user Tandem Laboratory (MTL) [1] including hardware and software selections. Detailed descriptions of the MTL and its missions are presented by J. C. Davis [2] and I. D. Proctor [3] in other papers at this conference.

2. IMPLEMENTATION OF MTL CONTROL SYSTEM

The present MTL control system is implemented in a two-tiered structure, local and supervisory. We chose UNIX-based Hewlett-Packard 9000 series computers for our system. They have good I/O performance through well documented drivers, a robust implementation of UNIX, support a variety of high quality graphics devices and have proven to be very reliable.

2.1 MTL Control Instrumentation

All control to the devices in MTL is through IEEE-583 (CAMAC) instrumentation. Each local computer is connected to CAMAC through an IEEE-488 (GPIB) connection. All analog to digital (a/d) digital to analog (d/a), digital input and digital output is accomplished through commercially purchased CAMAC modules. The choice of CAMAC as our instrumentation base was founded on the belief that the upgradable aspect of the system will be in the computer technology, not in the a/d's or d/a's that are connected to our systems. CAMAC also provides a degree of electrical isolation from the computer backplane and in effect is an I/O extension of the computer. Through the use of a GPIB CAMAC crate controller from AEON Systems, we achieve >300 Kbyte/sec transfer rates to the crates. The crate controller is capable of >700 Kbyte/sec rates but our computers pace the throughput. Since the amount of data from a fully stuffed crate of a/d's is only about 1 Kbyte, the total transfer times are dominated by the I/O driver overhead of the software (~200 μ s). Data transfers with the crates make use of address scan techniques. Address scan allows the reading or writing of an entire crate full of modules with one I/O command. Address scan is a selectable mode of the CAMAC crate controller.

Associated with the choice of CAMAC was the decision to use computer systems separated from the device I/O. We felt that we were not in the computer building business. Other options that were open to us would force us to custom configure the hardware and operating system. The results would be a unique system, unsupportable by any computer vendor. We decided that we did not want to be in the position of supporting unique structures. We would purchase the support or updates when necessary, leaving the quality assurance to the vendor. With CAMAC, we have a computer independent solu-

tion using industry standard interfacing allowing us to grow and update in the area where the technology is most volatile, computers.

2.2 MTL Computer Architecture

There are three local level computers; one located at the ion source, and one each located at the low and high energy ends of the tandem. All computers run discless UNIX, each directly connected to one CAMAC crate (a total of 15 crates are possible). The computer run-time control algorithms perform I/O and formatting of CAMAC data, bi-directional pass-through of parameters to the supervisor, and closed-loop control of various local parameters such as magnet currents and ion source power supplies.

The local computers are HP 9000, Model 319 with 4 Mbyte and RAM, LAN interface, GPIB interface, RS-232 interface, high resolution (1024x768) 16" color monitor, keyboard and mouse. When operating within the system, the monitor, keyboard and mouse are removed with all communication accomplished across the network from the supervisor. Under maintenance conditions, the monitor and keyboard can be plugged in and the computer operated locally.

One supervisory level computer is in the system. The supervisor performs the local computers coordination task and is the primary operator interface. The supervisor computer provides operator input through keyboard, mouse and knobs (analog tuning) and display through the 19" color monitor. The local computers operate in a discless environment with the supervisory computer as the UNIX disc server.

The supervisory computer is an HP 9000, Model 350CH with 8 Mbyte of RAM, LAN interface, two GPIB interfaces, RS-232 interface, high resolution (1280x1024) 19" color monitor, 571 Mbyte disc, 9-knob box, cartridge tape drive, small printer, keyboard and mouse. UNIX, networking software, graph-

ics software, C compiler, etc., came complete as a bundled package from Hewlett-Packard.

All computers are connected through an IEEE-802.3 ThinLan RG-58 cable. We experience good data throughput (>200 Kbyte/sec) and the system has tolerated many tank spark and power outage transients without incident. The networking software and interfaces appear very robust and were supplied with the computer bundle.

2.3 MTL Control Software

MTL makes use of a controls software package being developed jointly by Livermore and the Continuous Electron Beam Accelerator Facility (CEBAF), a large superconducting recirculating linac under construction in Newport News, Virginia.

The concept was developed by the author and has been implemented and expanded by R. G. Bork and staff of CEBAF. It is written entirely in C under UNIX. We continue to work with CEBAF in the further development of the controls package.

The software consists of five primary components; LAN communications, algorithm execution, operator interfacing, algorithm graphics editor and a display graphics editor.

All programming of the package is accomplished through extensive graphics editors. The software assumes that all instrumentation is CAMAC based, but other I/O is possible. All definitions of signal connections and control algorithms are entered through the editors in graphics form.

Algorithms are built by assembling function primitives, through the graphics editor, into a schematic representing the calculation and control flow. The editor enforces a unified way of describing the control logic and

calculations. In the effort it takes to describe the control process, the user has done the programming without writing one line of code. If a special function is required, a simple module can be written and added to the function list.

The operator interface editor is molded in the same scheme as the logic editor. The people operating the finished system can develop their own custom control screens oriented to the specific function they may want to perform. In MTL, there are many screens available to the operator. Many of the screens have duplicated information, plus special constructs of control added for a particular screen function. For example, MTL has a screen displaying the condition of the tandem accelerator and the specific subset of parameters associated with one of the ion sources. Another screen displays the same tandem accelerator parameters but has the subset of parameters associated with beam transport through the high energy end optics. These are operator defined control screens. Custom screens can be built by an operator for any specific purpose using the graphics editor tools, not unlike a CAD (computer aided design) system for controls.

Within the software package described above, there are three run-time programs potentially operating within each computer. These programs communicate to each other through shared memory pages defined through UNIX. All I/O, operator inputs, and LAN variables are available through this shared memory. The first program is called "lanmaster." It sets up the LAN communications among the computers and establishes the appropriate shared memory pages. When running in the supervisor, it interrogates each of the local computers for data by sending updated control parameters. In response to this communication, "lanmaster" running in the local computer updates the shared memory parameters of the supervisor. The second program to run is

called "logic." "Logic" is the program that actually runs the local CAMAC I/O and calculates the control algorithms. As mentioned above, the algorithms are generated through the use of a graphics editor. The third program is called "run." Its task is to provide the operator interface through the graphics display, keyboard, mouse and knobs. With this program structure, it is only necessary to run the programs required for a specific operation. As an example, MTL local computers run "lanmaster" and "logic" only. Since there are no operators at the local computers, "run" would have nothing to do. For maintenance purposes, control screens are available to be run at the locals, but normal operation does not require it. The supervisor runs all three programs. In MTL, its I/O is totally across the LAN, although a CAMAC crate could be attached. With the computers we have installed, MTL enjoys a systemwide data update rate at the supervisor of eight times per second. The local computers update their instrumentation from 20 to 40 times per second. The locals perform at this speed even when being interrogated by the supervisor. All computers have sufficient RAM to allow the run-time programs to execute without having to go to virtual disc memory, thus maintaining good operating performance.

The initial installation of the MTL controls followed the structure outlined above but was implemented with HP 9000, Model 310 computers running under HP BASIC. HP BASIC is an advanced language that has the features of FORTRAN 77, ALGOL, and PASCAL with powerful I/O and graphics extensions. The downside of it is that it is quasi-interpretive and did not support the LAN. Intercomputer communications were accomplished across a dedicated GPIB network with the local computers set up as devices on the bus. The I/O drivers considered the locals as any other kind of GPIB device and sent messages without a hitch. An LLNL developed interrupt handler written in

HP BASIC was utilized to operate the network. Special keyboard handling routines were written to ensure that the computer did not pause waiting for operator input. Mouse and knob routines were also written along with CAMAC I/O utilities. Two types of main programs were written, local and supervisory. Although the local programs (three each) were unique, they followed the same structure, allowing customization from one to another a straightforward and uncomplicated task. The supervisor computer had the primary operator display/interaction interface and the intercomputer I/O tasks. MTL experienced a systemwide update rate of a few per second.

The power of this design structure lies in the effort to implement and update the system. The initial system was designed and in operational test within four months of project start. Total software investment to get to operational testing was about eight man-weeks (mainly because of the extremely interactive nature of HP BASIC). After installation within MTL, further refinements were made in the operator display and in the interrupt handling routines. An additional man-month (across three months of calendar time) of software time was added before MTL was ready to be turned on through the full control system. Although the system worked admirably well, we knew the growth of MTL would eventually overwhelm the computers.

In June 1988, MTL upgraded to the UNIX based system described above. This upgrade entailed the replacement of all of the computers previously installed in the system. Since the instrumentation was CAMAC based and already installed, we did not have to address ourselves to new instrumentation in the upgrade. The entire new system was installed adjacent to the old system to allow simultaneous operation while we developed new control algorithms using the graphics-oriented software. After a few self-inflicted false starts, the entire system was cut over within five weeks and with only

two weeks of MTL down-time experienced for this upgrade. The new system was running the MTL in late July 1988.

The entire electronics systems installation, BASIC system development, testing and upgrade were accomplished with one electronics technician directed by one part-time engineer, one physicist and assisted by two additional contract fabrication technicians. Most of the initial BASIC software development was accomplished by Thomas Zimmerman, the lead electronics technician.

3. CONCLUSIONS

The distributed approach to controls is a viable, cost effective way to implement control systems on a large range of applications. We have found it to be a logical structure to implement, quick to install, and, because of the modularity, easy to manage. The architecture has a growth path oriented to computer developments through the use of communication, instrumentation, operating system and interfacing standards.

Because of the flexibility and the potential expansion of the distributed processing role, major rethinking of traditional facility operations can be addressed, such as unattended operation. The quality of operation possible with this system is improved by an order of magnitude over the more traditional, hard-wired approach.

The maner in which new facilities are designed is impacted by this architecture. We believe major savings occurred by not having to invest in much facility wiring. We did not have to build extensive cableways and shielded cable runs. We minimized the amount of long-run wiring typically installed in accelerator laboratories. The modular approach allowed the design, fabrication and testing of major controls subsystems well in advance of the installation of the mechanical hardware they would be controlling.

Because of its low demand on the physical aspects of a facility, expansion to additional capabilities can be considered more easily. The flexibility and fast implementation of the system's architecture open expansion opportunities that were prohibitively costly in the past.

The control system presently implemented provides a high quality, highly stable platform from which to operate the MTL. Additional work will be accomplished this next year to address unattended operation of MTL for overnight processing of AMS samples. This will entail improvements in our overall instrumentation in addition to the development of more sophisticated control algorithms.

References

- [1] I. D. Proctor, J. C. Davis, M. R. Eaton, J. I. Garibaldi, T. L. Moore, B. J. Schumacher, J. R. Southon and T. A. Zimmerman, Nucl. Instr. and Meth. A268 (1988) 344.
- [2] J. C. Davis, paper at this conference.
- [3] I. D. Proctor, paper at this conference.

1543A-9/28/88