DISTRIBUTION OF COMPUTER FUNCTIONALITY FOR ACCELERATOR CONTROL AT THE BROOKHAVEN AGS*

A. Stevens, T. Clifford, R. Frankel
Accelerator Controls Section, AGS Department,
Brookhaven National Laboratory
Upton, New York 11973

Summary

A set of physical and functional system components and their interconnection protocols have been established for all controls work at the AGS. Portions of these designs were tested as part of enhanced operation of the AGS as a source of polarized protons and additional segments will be implemented during the continuing construction efforts which are adding heavy ion capability to our facility. Included in our efforts are the following computer and control system elements: a broad band local area network, which embodies MODEMS; transmission systems and branch interface units; a hierarchical layer, which performs certain data base and watchdog/alarm functions; a group of work station processors (Apollo's) which perform the function of traditional minicomputer host(s) and a layer, which provides both real time control and standardization functions for accelerator devices and instrumentation. Data base and other accelerator functionality is assigned to the most correct level within our network for both real time performance, long-term utility, and orderly growth.

Introduction

The Accelerator Controls Section in the AGS Department at BNL is currently engaged in implementing a distributed controls system for the Heavy Ion Transfer Line (HITL) Project which will connect BNL's Tandem Van de Graaff to the AGS allowing ions up to sulphur to be accelerated to ~15 GeV/A. This control system has evolved from the system originally planned for ISABELLE/CBA. Some parts of this system have already been installed and tested in support of the polarized proton physics program at the AGS. We discuss below the major components of this system and the functionality of these components. The assigned functionality leads naturally to a model of the distributed data base which will be briefly discussed.

System Architecture

A schematic representation of the basic architecture is shown in Fig. 1. The system features a broadband local area network communications link called RELWAY. This LAN utilizes a pair of high bandwidth low loss coaxial cables as the transmission medium with passive directional couplers at each node. These components, commercially available from the CATV industry, offer low cost and high reliability. In the polarized proton running, data is communicated on one channel (T-11 at ~33 MHz) at 1 M baud. This data communications channel is being extended for HITL and an additional channel (T-7 at 9 MHz) will be added to carry reset signals. More details on RELWAY, including communications and contention resolution protocols, are given elsewhere.[1-2]

Also represented in Fig. 1 are host/console computers and three types of microprocessor based systems called Comboxes, Stations, and Device Controllers (DC). The next section discusses the functionality and physical realization of these entities.

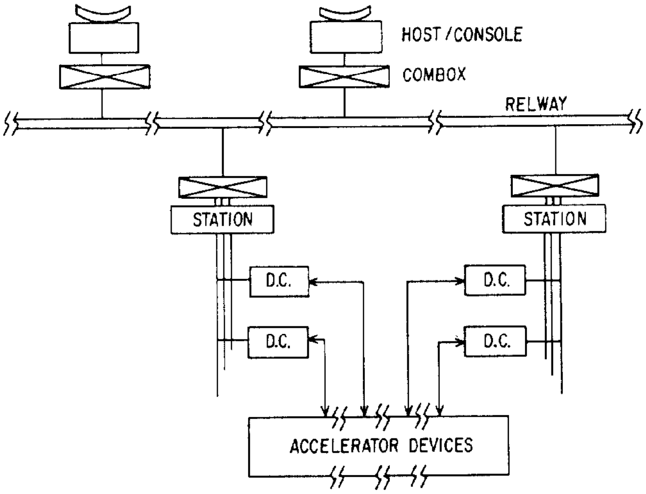

Figure 1 - Schematic representation of distributed control system discussed in text.

Assigned Functionality

In their control function, hosts act as sources of commands and setpoints and sinks of readbacks and status reports. Although a host may be of any size, most will have a disk based operating system. Prominent features of hosts include their console function and their ability to provide a computational resource. For polarized proton running our host is the AGS PDP-10 interfaced to RELWAY via a PDP-11/23 computer. For HITL we will employ Apollo 32-bit workstations as hosts. Figure 2 shows the configuration of the general purpose Apollo based control console. The Apollo workstations run a modern (UNIX like) multitasking operating system which supports multiusers. It is a demand paging system with up to 16 Mbyte address space for each process. The large high resolution bit mapped display, mouse, and display manager windowing software offers a versatile operator interface. The system supports dynamic assignment of the mouse, keyboard, or other input devices to different tasks which allows an operator to switch quickly from one operating task to another. Software productivity is greatly enhanced by the friendliness of the Apollo development environment which includes multiwindows, interactive multiwindow debugger, screen editor, code management system, and good high level language support by the OS primitives.

*Work performed under the auspices of the U.S. Department of Energy.

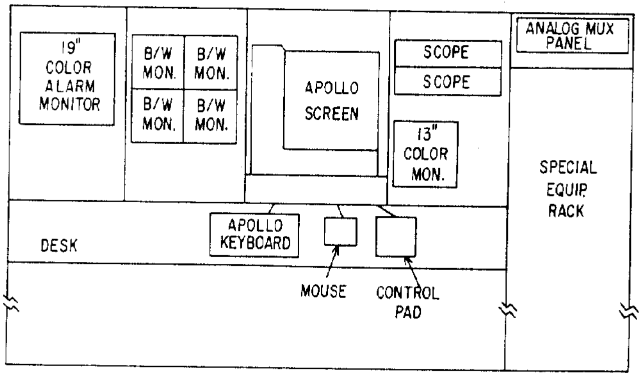

Figure 2 - The Apollo based HITL control console layout. Not shown in this sketch, but present in the actual layout, are telephone and intercom facilities and a variety of signal-selecting switches. Two of the black and white monitors are used for video display and two for alpha-numeric status display. The control page will appear on the 13" color monitor.

A Combox provides the access channel to RELWAY. It has the following attributes and/or performs the following functions: cable interface from main line (RELWAY) couplers; MODEM receiver; MODEM transmitter; status display of main line, branch line and Combox operation; control of main line contention resolution; control of branch line; buffering of both incoming and outgoing main line data; flow control and buffering of branch line traffic.

Physically the Combox consists of three or four EUROCARD modules supporting MODEM functions, an Intel single board computer (typically an SBC 80/30), and a BNL designed communication board supporting main line and branch line service.

Because the Combox is transparent to users, we do not regard this element as a hierarchal layer. From this point of view our system then consists of three hierarchal layers; Host, Station, Device Controller.

A Station performs the following functions: connects via a branch line (now implemented with IEEE-488) to a controlling Combox; connects via a secondary branch line (either a physical IEEE-488 branch or a logical multibus shared memory branch) to one or more Device Controllers; accepts for local use a set of watchdog tables; polls the Device Controllers for readbacks and formats these readbacks into reports when requested by the Host; generates unsolicited alarm reports by use of the watchdog compare tables; issues commands and/or setpoints generated by a Host for use by Device Controllers. The Station is cycled synchronously (usually to the AGS) and has parametric limits on the period within a machine cycle that commands may be sent or readbacks acquired. The Station also appends or strips Network level destination protocol headers. The Station software is PROM based with all stations in the system running the same code. The Stations are programmed in PL/M, a high level language supported by Intel. Currently the Stations employ the multitasking RMX 88 real time executive operating system.

Device Controllers have the functionality of acquiring accelerator information and/or controlling accelerator devices or instruments. Real time synchronization is accomplished at this (lowest) hierarchal level. As mentioned above, Device Controllers may have a shared memory/multibus or an IEEE-499 connection to a controlling Station, which polls the Device Controllers. Finally, Device Controllers are required to translate accelerator information into well defined data structures. Explanation of this functionality requires a discussion of the accelerator data base which is given below.

The functionality described above could, in principle, be rearranged. Specifically, we have considered transferring the Station functions to other components (Host, Combox, Device Controller) in an attempt to reduce our 3 level hierarchy to a conceptually simpler 2 level hierarchy. Although such a redistribution might be advantageous for the HITL project considered by itself, we believe that eliminating the Station might hamper our growth potential in the long term. A Station now provides a convenient point for local control of a geographical area. In addition, although not now implemented, a Station could serve as a process controller for a geographical area should that become necessary. Finally a large network might be flooded without the data blocking and packing potential of the Station.

Data Base Model

We have defined data structures based on a data model which, in turn, is based on the assigned functionality of the hierarchal components (Host, Station, Device Controller) of the control system architecture. The basic building block of this model is the "logical device" which may or may not correspond to a real device. Every logical device has an associated host resident data structure and an associated device controller resident data structure. The fundamental data structure associated with a logical device consists of command fields, status fields, setpoint fields, and readback fields. This fundamental structure is resident in the Device Controller. Logical devices have been divided into two classes; simple and complex. A simple logical device (SLD) by definition is a logical device with at most one setpoint and/or readback which is monitored by the Station. A complex logical device (CLD) by contrast can have any number of setpoints and/or readbacks and is not monitored by the Station watchdog function. Note that an SLD is readily mapped to a line on a page: it is amenable to control from a page display program.

Because SLDs are monitored, they have associated Station resident data structures. These structures contain, in addition to the fundamental structures described above, data fields which contain information necessary for the watchdog function. This information is either sent directly from the Host or is computed from data sent from the Host. Data structures in the Station associated with CLDs contain only that information necessary for Station-Device Controller communication.

At the Host level, logical devices have associated data structures which contain essentially "passive" information. Examples include addressing information, allowed commands and status, conversion factors, etc.

The distributed accelerator data base is composed of the fundamental logical device data structures defined above. For SLDs, this data base resides in Station memory; for CLDs in Device Controller memory.

The discussion of logical devices has been, thus far, rather formal. We give now a few examples, not meant to be accurate in detail, of logical devices in an attempt to elucidate the motivation of our data model.

An operator might like to consider a "logical device" to be a line on a page. This line should have a name descriptive of its functionality (e.g., HITL QUAD 12), a state (e.g., ON) and, perhaps, a setpoint and readback. This conceptual entity clearly, and not accidently, corresponds to an SLD as defined above. On the other hand, there is a great deal of accelerator information that an operator is normally not interested in. We take, as an example, a function generator with 1,000 setpoints and, we will assume, a corresponding 1,000 readbacks. Not only is an operator normally uninterested in this information, but it is clearly unnecessary to monitor these 1,000 points every AGS pulse. Such data clearly forms a CLD as defined above. On the other hand, the operator certainly wants to know if the device related to the function generator suddenly ceases to behave properly. One way to monitor this device is to calculate (in the Device Controller processor) the sum of the differences between the setpoints and the corresponding readbacks and to allow the Station to monitor this single number. The data structure associated with this construct is an SLD with no commands, status, or setpoints, but with one readback whose expected value is zero.

Our data model might, then, be summarized as follows. Every physical Device Controller has available to it a certain amount of information. This information must be arranged into data structures called "logical devices". A given physical device, in general, will have several associated logical devices. A subset of logical devices called Simple Logical Devices are amenable to page display control and represent quantities which are monitored (or are capable of being monitored) by the Station. Complex Logical Devices, by contrast, are not monitored. In general they represent information which is accessed infrequently.

Project Status

Modern control systems design is sufficiently complex that a "top down" definition of hardware and software interface functions is required prior to the design of any modules. Detailed documentation describing this functionality now exists for Device Controllers in the form of a technical note,[3] and for Hosts, Stations, and Comboxes in a less formal collection of internal notes.

We have completed formal specifications for instrumentation controllers, power supply controllers, analog multiplexing controllers, and translating controllers. The latter controllers convert formats of commercial RS 232 and IEEE-488 equipment to a format compatible with our network. Working code exists for operator interface to Hosts, network maintenance and management functions, Combox data buffering and Station intertask scheduling. A standardized physical package, multibus based, but capable of supporting 3-U high EUROCARD modules has been designed and employed for numerous Device Controllers, Comboxes, and Status. Most of our Device Controller applications can be supported with Intel 8088 class microprocessors and several systems based on SBC 88/25 cpu's have been designed and are nearing unit test. It appears thus far that the data model described above adapts well to a large range of accelerator control applications.

References

1. J. Skelly, T. Clifford, R. Frankel, "A Broadband Accelerator Control Network", IEEE Trans. Nucl. Sci., NS-30, 2155 (1983).
2. R. Frankel, "Tailoring a CSMA Local Network to Research Needs", Data Communications No. 9, 145-154 (1984).
3. R. Frankel, "Device Controller System Specification and User's Guide", ACS/TN/85/001, (1985, unpublished).