CEBAF CONTROL SYSTEM

R. Bork, C. Grubb, G. Lahti, E. Navarro, J. Sage
Continuous Electron Beam Accelerator Facility*
12000 Jefferson Avenue, Newport News, Virginia 23606

T. Moore
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA 94550

Abstract

A logic-based computer control system is in development at CEBAF. This Unix/C language software package, running on a distributed, hierarchical system of workstation and supervisory minicomputers, interfaces to hardware via CAMAC. Software aspects to be covered are ladder logic, interactive database generation, networking, and graphic user interfaces.

Introduction

A logic-based application software package, Thaumaturgic Automated Control Logic (TACL), is presently in development at CEBAF for the control of machine operations. TACL is a tool for building custom control software for a hierarchical, distributed process control system. It is premised on a ladder-logic configuration, with a series of editors for relational database generation. Once developed, this database is used by run-time programs which perform the actual control of the system as specified in the database.

TACL is designed to operate with Hewlett-Packard (HP) series 300, 500, and 800 computers and with Computer Automated Measurement and Control (CAMAC) equipment. At this time, TACL uses a control system architecture of two hierarchical levels: an upper level known as the Supervisory Control Level (SCL), and a lower level or Local Control Level (LCL). The system may be configured with a maximum of ten SCL computers, each with a subsystem of at most 20 LCL computers. Each LCL computer may have up to 15 CAMAC crates attached.

Overview

TACL allows a user to develop a control software system via a menu-driven, graphical interface. The package offers menu selection of pre-defined data I/O functions, internal control algorithms, and run-time display options. In combining these selected functions, the user builds up a complete, flexible, modifiable control database. Scientists and engineers can quickly develop control system applications and modify their control databases or run-time displays.

A block diagram of the software components of TACL is shown in figure 1. Two components, the logic and display editors, are used off-line to build the required databases. During run-time, other programs use these databases to implement the specified control algorithms, handle data I/O, and manager the user interface.

Off-Line Editors

Logic Editor

The logic editor, used to develop the primary database for operation of the control system, consists of two sections. One section defines all I/O interfaces to the computer. The second section is used to develop the internal data manipulations and control algorithms.

The interface section development screen is a menu-driven development page used to define CAMAC and LAN interfaces for the database. Among the capabilities of this section are:

1) Adding/deleting computers. In a distributed process scheme, computers may be added or deleted from the database with necessary addresses and locations defined.
2) Adding/deleting CAMAC crates.
3) Adding/deleting/moving CAMAC modules.
4) Defining I/O points. User-specified names are assigned to I/O channels to/from CAMAC modules and LAN.
5) Maintaining the CAMAC module library. The library function is used to add data about new types of CAMAC modules as they are introduced into the system.
6) Saving/loading files. Stores or retrieves system files.
7) Producing documentation. Produces hard-copy documentation for specific user application.

Once this portion of the editor has been used to define the interfaces to the system, the actual control algorithms can be built.

The logic development page permits the user to define data flow and data manipulation in a schematic fashion through the use of function symbols. Parts are chosen from a menu listing and then placed on a screen grid, in much the same manner as drawing an electronic circuit on a CAD computer. Each part is a predefined, functional software object, which is placed into the logic ladder represented by the grid.

Specific functions/objects may be chosen from the following categories:

1) Input from and output to CAMAC, data storage, operator interface, and LAN. The first portion of the editor, described previously, was used to define interfaces to move data into and out of the computer. The objects defined here move data into and out of the logic analysis portion of the run-time program and perform the necessary conversion of raw data to the desired engineering units.
2) Data ties. These provide for data flow from one section of logic to another, as direct connections or feedback loops.
3) PID. This function is a proportional-integral-differential control algorithm, common in process control. It allows for the setting of the PID constants, control conversions, sample time, high/low rate and value, loop setpoint, and various other items particular to PIDs.

* This work was supported by the U.S. Department of Energy under contract DE-AC05-84ER40150.

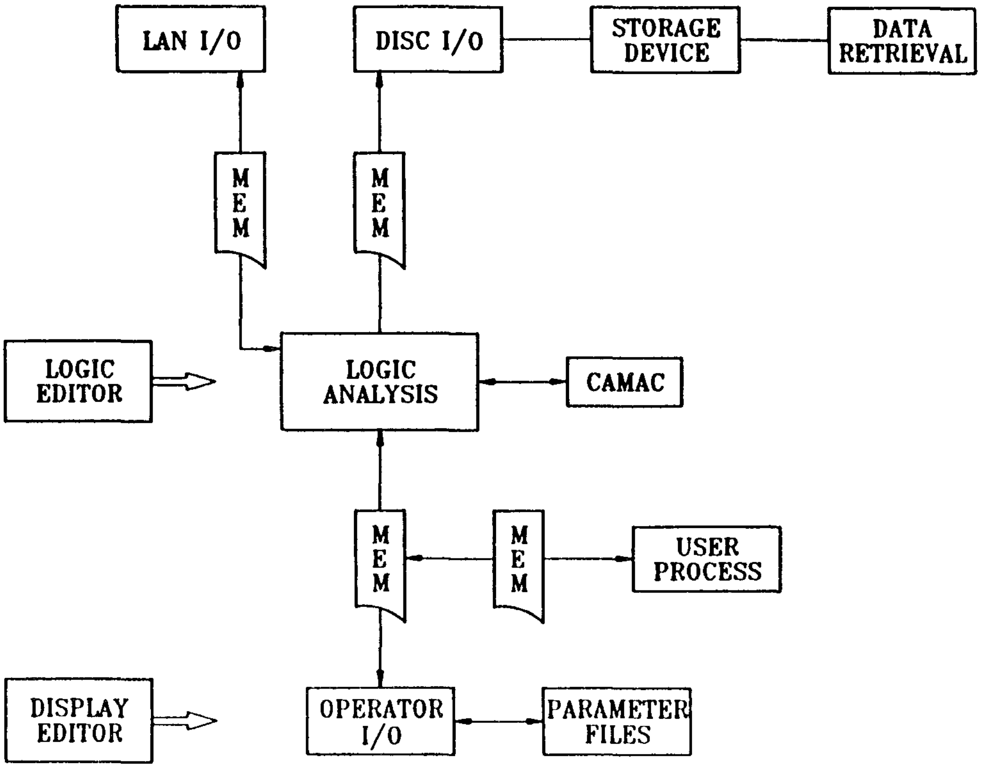

Figure 1 Instrumentation and Control Software System Architecture

4) Normally open or closed contacts and switches. The data used to operate these software switches can come from external or internal devices.

5) Math Functions. Presently available are adders, subtractors, multipliers, dividers, as well as exponential, log, power, root, trigonometric, and polynomial functions.

6) Timers. Variable time settings in 1-second intervals at present.

This is a quick list of the functions presently available. Other object functions, such as look-up tables for magnet control, are in development. Since the source code for this editor is modular, new function objects can be developed and tested as a small unit outside of the editor program and then placed into the editor.

Display Editor

The display editor is used to create custom, graphic, run-time displays for 19" high-resolution, bit-mapped monitors. This editor, like the logic editor, is menu-driven. The user chooses predefined symbols (or creates symbols which then appear in a menu), draws lines, and makes labels to build a run-time display and user interface.

Menu-selectable items include:

1) Data read/write boxes. Read boxes show the operator a single data point. Write boxes provide for data entry via keyboard or knob box.

2) Status symbols. User-created symbols with or without an associated data box. These symbols do not change in run-time although the data will change to reflect conditions.

3) Dynamic symbols. User-created with or without a data box. These are actually three symbols in one (tri-state). The symbol which appears on the display at a given time depends on the data at that time. For example, an open valve symbol may change to a closed valve symbol on receipt of a relay closing signal from CAMAC.

4) Labels. A means of putting text on the screen.

5) Lines. Color and width selectable. Normally used in flow diagrams, connections, or to generally enhance and highlight the display.

6) Predefined symbol macros. At present there are two, both used for the display and operation of PID loops.

In creating a symbol, the user chooses and assembles graphic primitives (circles, rectangles, lines, text), selecting color, size and placement by buttoning a mouse. The completed symbols are stored and may be retrieved later for this logic database or for any other database.

The association of a display symbol or display data box with actual run-time data is made when a symbol, read/write box or PID is placed on the screen in an editor session.

Provision has been made for deleting, moving, and examining symbol parameters if display revisions are required.

Run-time Programs

After the system and display databases have been generated using the editors, they are ready to be used by various run-time programs. Identical logic analysis, LAN control, and data logger programs run on all machines in the system. The operator interface program version is chosen to support the graphic capabilities of the particular SCL or LCL computer.

Logic Analysis Program

This program is the core of the run-time system. Its functions are: (1) handling all I/O to CAMAC, to data storage units, to the LAN control, and to operator interface programs, and (2) performing the data manipulation and control algorithms.

This program develops a threaded code from the database information which then functions to:

1) Read data in from CAMAC/LAN/operator.

2) Perform the control logic.

3) Write data out to CAMAC/LAN/operator.

4) Perform fault checking and correction. CAMAC or remote computer faults are reported to the operator, and when possible, automatic corrective action is taken.

Lanmaster

The LAN control program handles all data transmissions to/from other computers on the subsystem network and checks the integrity of the network for automatic fault handling and notification to the operator. It uses level-two access into the IEEE 802.3 hardware.

Run-time Display

This is the operator interface program which manages the display monitors for operator interaction at each computer. The previously created display files are continuously updated with new data coming from the logic analysis program via shared memory. Operator input is acknowledged locally on the display and transmitted to the logic program, again via shared memory. Input from the operator may be via knobs, keyboard, trackball, or mouse.

On the HP500 computers, a maximum of four different displays are kept current on two monitors. This is done by dividing the eight graphics planes of each monitor into two sets of four as foreground and background buffers. These four displays are held in frame buffers for virtually instantaneous page changing. In addition, other displays can be menu-selected during run-time to replace any of the original displays.

The right-hand portion of each display is reserved for system use. It shows the status of computers and CAMAC equipment in the system, and provides menus for display changes.

Graphs

The Graphs program provides for real-time and off-line graphing of data generated by the run-time processes. It permits a maximum of eight parameters to be graphed at a time on a single display in eight colors. The eight parameter names and variable data scales appear to the left with their data plots in the graphical area.

Manager and Master

The Manager program is the first program run on any computer. It activates a menu on the console which is used to load and activate the other application processes.

The Master program is a background process which allows two graphics displays to operate with a single mouse, knob box, and keyboard. It operates in much the same fashion as a window manager, which each display appearing as a separate window. Master tracks the current mouse/knob position and traps input requests and passes them to the appropriate process. It also presents keyboard requests to the process running the display that the mouse cursor presently appears on.

Present Status

TACL is presently operational at CEBAF on the Cryogenics Test Facility, a 2-K liquid helium plant, and the injector test stand. In addition, this package has been adopted and is being used to control the Multi-user Tandem Facility at Lawrence Livermore National Laboratory (LLNL) and was due for installation to operate the TANDEM accelerator at Yale University.

These systems all employ what is known as Version II of TACL. Version III, currently under development and due for release in the spring of 1989, will provide various extensions, including:

1) Logic and display software IC's for automated beam transport system control including beam modeling interfacing capabilities.

2) Additional interface capabilities for custom-written user software.

3) Additional graphics display capabilities and enhancements.