

Accelerator Simulation and Operation Via. Identical Operational Interfaces*

J. Kewisch, A. Barry, R. Bork, B. Bowling, V. Corker, G. Lahti, K. Nolker, J. Sage, J. Tang

Continuous Electron Beam Accelerator Facility

12000 Jefferson Avenue
Newport News, VA. 23606

ABSTRACT

The CEBAF accelerator contains approximately 2500 power supplies, 340 klystrons, and 800 beam monitors. The operation of such a complex machine requires a control system which can provide a high degree of automation with strong support by simulation and modeling programs.

We present the architecture and first results of a control system which allows one the use of identical operation procedures and interfaces for operation of the real accelerator and high-level accelerator simulation programs. The interfaces were developed using TACL (Thaumaturgic Automated Control Logic) control software, developed at CEBAF for accelerator control. This setup provides the capability to: (1) test and debug the various operation procedures before the completion of the accelerator, (2) execute machine simulations under realistic environmental conditions, and (3) preview and evaluate the effectiveness of operational procedures during run time. The optimized simulation program adds only two seconds to the normal TACL operational cycle.

INTRODUCTION

The CEBAF control system consists of a programming environment referred to as TACL^[1] and a large number of user applications. TACL offers the following services to the user:

- Easy access to the hardware. TACL contains a database for the computer and CAMAC configuration. The user only needs to know the signal names for an access.
- Logic transformations of signals. TACL computes signal transformations in the "logic array" which is user generated with a graphic editor. This signal processing is used for example to convert units (*i.e.*, bits into amperes), to combine signals (*i.e.*, magnet current and beam energy into deflection angle) and for feedback loops (cryogenic controls).
- User processes for more complicated operations. TACL provides methods to execute user written programs which may or may not be synchronized to the logic cycle of TACL. These programs are used for the calculation of optical (Twiss) parameters, orbit corrections, etc.
- User interface. TACL handles the operator interaction: input via mouse, knobs or keyboard and output via numbers, bar graphs, dials and icons. User written programs can be executed in a window for non-standard presentations. A graphic editor exists for the layout of screens.
- Networks. TACL provides a two layer network for the transport of signals between computers.

These capabilities make TACL the ideal environment for an accelerator simulation system which runs in parallel to the real accelerator. The concept of this system is described in this paper.

THE TACL SYSTEM

The CEBAF control system uses two levels of computers. The supervisor level employs graphic workstations used for operator interaction and running the high level operation procedures. The local level performs hardware operations and CAMAC access using rather inexpensive computer hardware.

The TACL software is identical for both types of computers. A set of system files determine the configuration of TACL for each computer. Therefore it is also possible to perform all software tasks on a single computer. Figure 1 shows a complete TACL system for one computer.

The heart of TACL is a shared memory segment which is used for all communication between the different processes. The data in this segment are inputs and outputs to and from the operator or operations procedures. They represent not only readings and settings of the accelerator, but also control signals for sequencing of procedures. RESOLVER is the set of function calls used in all TACL and user programs to access this shared memory.

Four TACL processes are active at run time:

- RUN (RUNtime) program is the main operator interface. It has various built-in basic features to display data from the shared memory, and allow for input of data from keyboard, mouse and knobs. For more complicated problems, user-written programs can be executed in a "JWindow" on the RUN screen which allows shared use of the input and output devices. The layout of the operation screens is defined by DISPLAY FILES produced with the DISPLAY EDITOR.
- LOGIC is the main operation program. The actions of LOGIC are controlled by the logic array. Each node of the logic array has a value and a function to generate this value. The large number of built-in functions include reading from and writing to CAMAC and the shared memory, and can range from simple mathematical functions to complex operations like PID-control loops. The "user process" function allows activating user written programs to handle nonstandard devices and complicated operation sequences. The logic array is generated manually with the LOGIC EDITOR, or from INGRES database information, and is stored in the SYSTEM FILES.
- LLAN (Local LAN) and SLAN (Supervisor LAN) handles the communication on the local and supervisor net, respectively.

*Supported by D.O.E. contract #DE-AC05-84ER40150

The DISPLAY FILES and SYSTEM FILES can be created not only with the TACL editors, but also by special programs using information from the INGRES database.

TACL AS A ENVIRONMENT FOR CONTROLS AND SIMULATION

The unique capability of the TACL system to perform controls and simulations within one system lies in the architecture of the data flow. The LOGIC program acts as a data switchyard. Data can easily be redirected by changing functions of the logic array. The structure of the logic array remains unchanged for this purpose.

This feature allows an easy substitution of the hardware interface by a simulation program as shown in Figure 2. Figure 2a shows the setup of a supervisor computer in the operation of the real accelerator. The INGRES database contains a detailed description of the accelerator and is used for the assembly of the accelerator and the generation of the logic array. For the operation of the accelerator TACL writes the settings of the hardware to CAMAC and receives the readings on the same path.

Figure 2b shows the system in the simulation mode. The read and write CAMAC functions in the logic array are substituted by operator input and output functions. This automatically generates a new section in the shared memory. The simulation program is implemented as a user process and can access hardware settings via RESOLVER calls. Together, with the description of the machine (from INGRES), the simulation calculates the behavior of the beam and writes the simulated readings of the instrumentation back to the shared memory.

Except for a simple redefinition of the data flow in LOGIC, no change is necessary to convert the control system into a simulation system and vice versa. All operation procedures can be used in the same way. Besides providing theoretical support for the machine physicist, this system offers two major advantages:

- New operation procedures can be tested in a realistic environment.
- New operators can be trained without interrupting the operation of the machine.

SIMULATION OF THE CEBAF MACHINE

A simulation system as described above has been implemented on our simulation computer system. The simulation program is based on the optics code PETROS^[2]. PETROS calculates the central orbit of the beam including second order nonlinear terms. A special routine is added to handle the linac cavities and the energy dependance of the beam motion exactly.

The simulated system describes the CEBAF accelerator from the 5 MeV point to the end of the fifth pass and includes 376 main dipoles, 1612 correction dipoles, 806 quadrupoles, 72 sextupoles, 806 beam position monitors and 1690 cavities. Multiple passes through the same element are counted as multiple elements.

The logic array for our system allows the operator to use deflection angles, focal length and energy gains as inputs. Since the beam energy in a linear accelerator depends not only on the settings of the cavities but also on the path length of beam trajectory (which depends on the beam energy), LOGIC has been set up which follows the beam line element by element, calculating beam energy and magnetic fields for each element.

The system is implemented on a single Hewlett-Packard 720 computer. The following computing times were observed:

- Setting all magnets 0.3 sec
- Calculation of new orbit 1.5 sec
- Reading beam position and display 0.1 sec

Using a faster computer like an HP 730 for example would reduce the simulation time to a total of one second which is close to the expected response of the control system with actual machine hardware.

FUTURE EXPANSION

For use in the control room, expansion of the simulation system is planned as shown in Figure 3. Parallel to the real machine control, a simulated system will be available on a separate computer. A modified version of the SLAN program will allow initialization of the settings on the simulation system. The operator can use this system to test changes of the machine settings before they are applied to the real machine.

Although the simulation will predict changes to the machine correct in the first order, the absolute results will be different because of unknown machine errors (tolerances in fabrication and alignment). These errors cannot be measured directly; however, one can develop an approximate description from the effect of the errors on measurable beam parameters as beam position, beam size, etc.

This information will be collected during the commissioning and operation of the machine, and stored in the measurement database. An analyzer program could then be used to extract the information about machine errors from a large number of measurements.

This procedure can again be tested by substituting the real machine with a simulation program: A set of machine errors is generated for the real machine simulator using a random generator. The operation of the simulator produces a measurement database. The analyzer is then used to reconstruct the machine errors, which can be compared with the original random numbers.

CONCLUSION

The data flow concept of TACL makes it for the first time possible to operate a simulation with the same operator interface and operation procedures that are used to control the real accelerator.

REFERENCES

- [1] R. Bork: CEBAF's control system, CEBAF-PR89-013.
- [2] J. Kewisch: Diplomarbeit Universität Hamburg (1978).

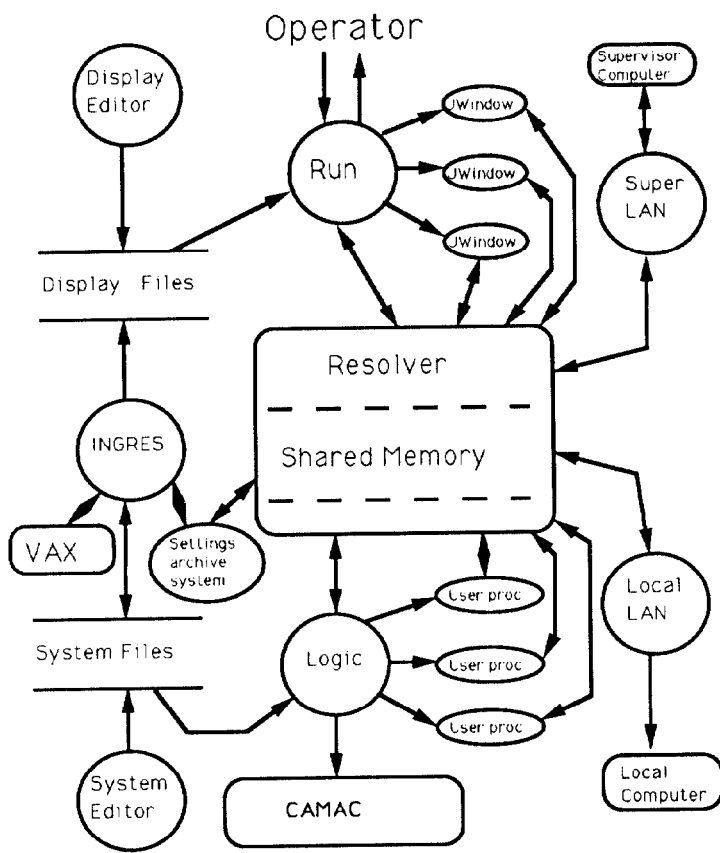


Fig. 1: TACL system overview

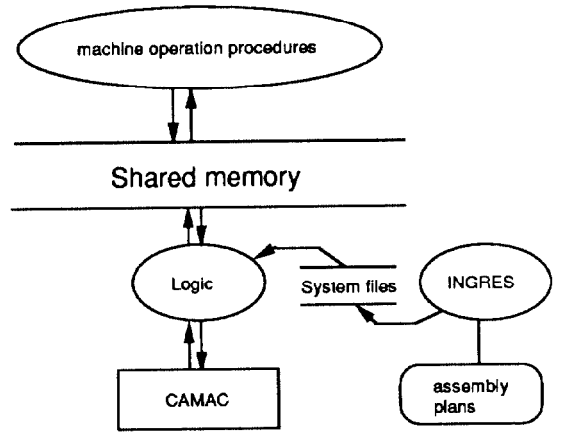


Fig 2a: TACL data flow for machine operation

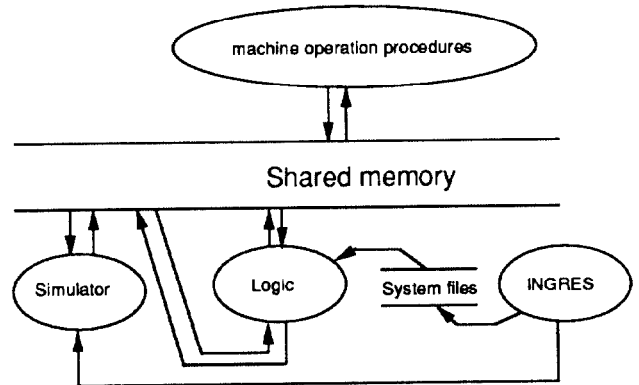


Fig 2b: TACL data flow for simulation

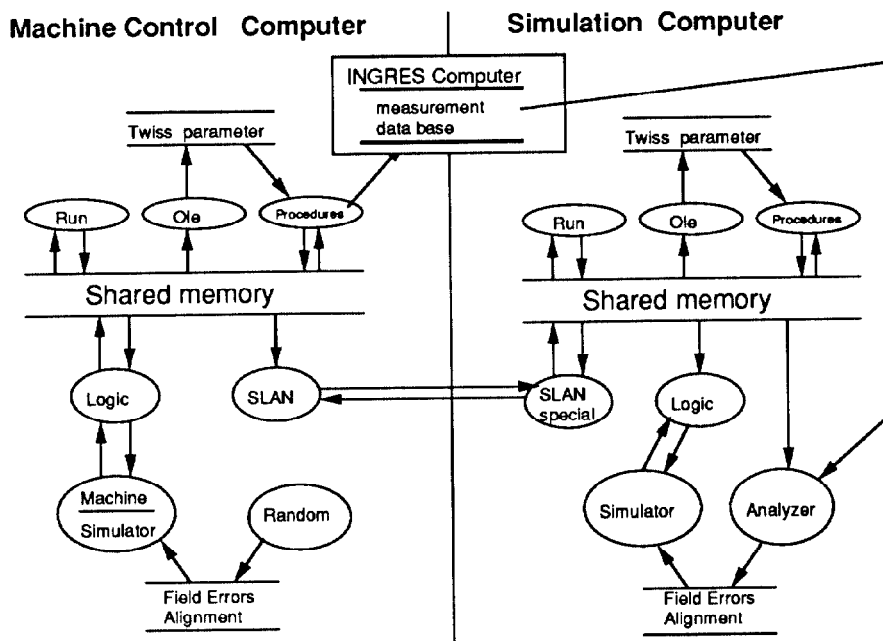


Fig. 3: Parallel operation and simulation with TACL