

Workstation Consoles for SLC*

J. BOGART, A. HUNTER, M. SULLENBERGER, S. KLEBAN, N. PHINNEY, N. SPENCER

*Stanford Linear Accelerator Center
Stanford University, Stanford, California, 94305*

ABSTRACT

A workstation console based on the DEC VAXstation GPx has been developed for the SLC control system. The three video monitors (touchpanel, color graphics display and terminal screen) of the original SLC console are replaced by four windows on the single workstation monitor, and standalone diagnostic programs may run concurrently. In order to minimize software effort and operator training, it was a design requirement that higher-level control and applications code, particularly that in the SLC Control Program, be identical whether the program executed on the MicroVAX or on the host VAX. The code for the graphics is necessarily different for the MicroVAX, as is the lower-level code used to access resources resident only in the host. A server program on the host acts as intermediary between programs running on the workstation and these resources.

Introduction

The original design for the SLC control system^[1] included two console types with very similar user interfaces. The console normally used by operators, the Console On Wheels (COW)^[2] consists of a touchpanel as the primary means of operator input, a terminal for dialogue and messages, up to 8 knobs, and a color monitor for plots and text displays. The other console type, used primarily for program development and hardware checkout, provides most of the functionality of the COW on a simple terminal.

Although the COW includes an 8088 single-board computer to manage graphics and input for the touchpanel, display monitor, and knobs, the operator program itself (known as the SCP, for SLC Control Program) runs as a process in the host computer (a VAX 11/780 or VAX 11/785), along with other identical programs and a variety of special purpose processes, such as a database manager and a process to handle slow feedback loops. The actual SCP program is the same whether the console used is a COW or a terminal, but in the latter case the SCP program handles its own touchpanel, display and knob I/O.

By fall of 1985 two areas of possible improvement in this architecture were identified. There was some concern that the VAX CPU might become overloaded as the mode of operation changed from the traditional "hands on" approach to one dominated by substantial computation to model the SLC's behavior. At the same time, it seemed a worthwhile investment to improve operator efficiency by modernizing the user interface. A new workstation console type was proposed as a possible solution to both problems. The SCP program would run on the workstation CPU, thus reducing the computation load on the host VAX (though increasing its I/O load). A high-resolution color monitor with vendor-supplied windowing software would provide the tools for the new user interface.

Workstation Hardware and System Software

The hardware chosen was the DEC VAXstation GPx. It includes a high-resolution (1024x864) color monitor, three-button mouse, keyboard, MicroVAX II CPU with 7 megabytes of memory, a 71 megabyte hard disk, and Ethernet. It should be stressed that the selection of hardware was largely conditioned by software requirements. The operating system running on the workstation had to be compatible with the one running on the host VAX (VAX/VMS) so that the applications code contained in the SCP, written in VAX Fortran, and making free use of the non-standard features in the compiler, would carry over to the workstation with little or no change. Also both operating system and graphics package had to support a real-time application with asynchronous interrupts from several sources. The compatibility requirement effectively limited the choice of hardware to DEC products. Only the VAXstation GPx, just going into beta test, met all the additional requirements; we were fortunate in being chosen as a test site. Two graphics system were available for the GPx: DEC's implementation of the Graphics Kernel System (GKS)^[3] and DEC's own User Interface System (UIS)^[4]. Primarily because of the lack of support for asynchronous input, the version of GKS (Iib.) available was ill-suited to our needs; therefore all of our graphics was built on top of UIS.

Access to Resources

In the configuration of the SLC control computers (Fig. 1), the MicroVAX workstations (SLCu1, ..., SLCun) are connected to the host VAX computers (SLC, MCC) by an Ethernet network, and the host computers are connected to the 8086 Microcomputers via a broadband communication line. A SCP that is running on the workstation (remote SCP) must get access to the same resources available to a SCP running on the host computer (local SCP).

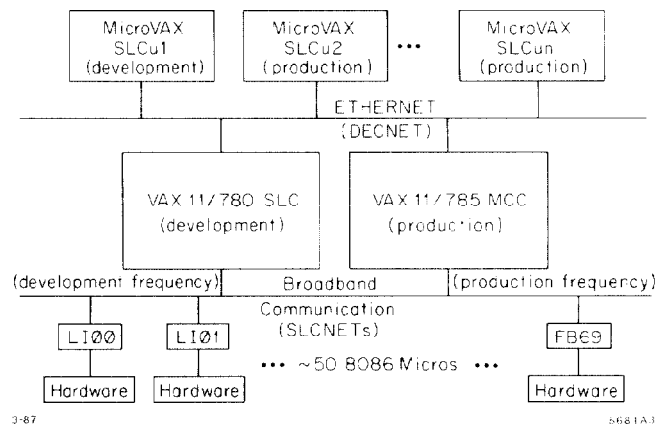


Figure 1. Computer configuration for SLC Control System

* Work supported by the Department of Energy, contract AC03-76SF00515

The resources needed by a SCP are:

- SLC database and other items in shared memory.
- SLC Micro Communication Network (SLCnet).
- Support processes (for example ERRTALK, which forwards error messages to those concerned).
- The actual code for the SCP process.
- Support files on the disk (for example a file containing the source for the touchpanel displays).
- Printer.

The SLC database, SLCnet, and support processes could not be removed from the host computer and it was not possible to distribute these objects to all of the workstations, so a new server process, the DNET Server, was written to run on the host computer and act as proxy for access to these resources (Fig. 2). The SCP's executable code and some of the other support files could and in fact had to be duplicated and distributed to each of the workstations, and a distribution system was designed to provide this capability, described in a later section. The printer on the host computer could be accessed directly by the workstation across the DECnet (Ethernet) network, so no additional software support was needed.

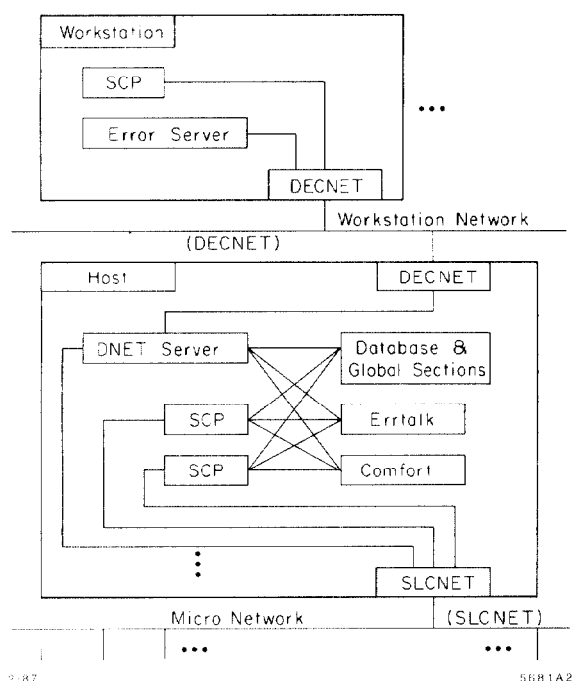


Figure 2. SCP communication paths.

The server system consists logically of two parts: one part, a set of subroutines collectively known as the DMSG Routines, is on the workstation as part of the remote SCP; the other, the DNET Server, is a process on the host computer. Where a local SCP has direct access to a particular resource, for example the SLC database, the remote SCP has to use its DMSG Routines to package its database request, send the request across the DECnet to the DNET Server which then accesses the database, packages and ships the results back to the remote SCP, where the DMSG Routines unpack these results.

The DNET Server process has five parts.

- Background server to process synchronous requests to the database and other global sections.
- Foreground server that serially (per workstation) handles communication requests to the SLCnet or other host process.
- SLCnet listener that asynchronously handles return communication from the SLCnet network.
- Mailbox listener that asynchronously handles return communication from the host processes.
- Error server that forwards error messages generated on the host to the Error Server on the workstation for display.

The DNET Server is implemented as a multi-threaded process which handles all requests at software interrupt level. This means that the Server is capable of simultaneously executing many requests from different workstation and it also guarantees that a long request from one workstation will not appreciably delay a request from another. The number of separate streams of activity (threads) within the Server grows or shrinks as needed. A single thread forks into two when a new action must be started and old threads are killed off when their work is done. The Server maintains a separate set of variables for each thread in order to keep each thread independent.

User Interface

The highest priority for the workstation project was to port the SCP to the workstation. Although ultimately it might be preferable to let the workstation user-interface diverge substantially from the COW, initially coding time and operator training time could be minimized by making a SCP on the workstation look rather similar to one on a COW. Our aim was to support all the standard utilities for writing to or reading from the touchpanel, dialogue window, display or knobs in such a way that applications code calling those routines could be moved unchanged to the workstations. This was accomplished, although the outward form of the display and touchpanel were somewhat different and the outward form of the knobs was radically different from their realization on the other console types. Figure 3 shows a workstation running a SCP with a typical configuration of display, dialogue, touchpanel, and error windows.

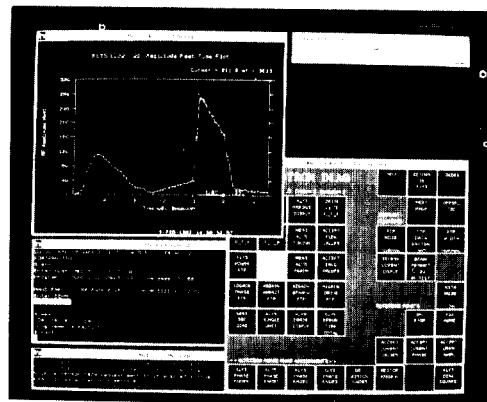


Figure 3. SCP running on Workstation.

The graphics software includes a generalized set of windowing utilities, suitable for other programs which might someday run on the workstation, graphics and handling of asynchronous input for the workstation touchpanel and knobs, graphics for the error window, and a GPx device driver for the Unified Graphics^[5] package. Unified Graphics is a device-independent graphics package developed at SLAC which supports, among other devices, the COW monitor and touchpanel (output only) and the hardcopy devices in use for text and graphics output from the SCP. The Unified Graphics driver was not used for the touchpanel because of the difficulties involved in integrating the Unified Graphics output with handling of the input, but we were able to use the Unified Graphics driver for the workstation display window. In this way all applications displays written for the COW carried over without change to the workstation. The knob and touchpanel input handling depend heavily on the support in UIS of asynchronous mouse input.

We could not simply use a terminal window for display of error messages without losing the ability to provide certain desirable features. Instead, errors are displayed in a separate graphics window, with associated utilities to handle scrolling and expansion of the window.

Software Distribution and Maintenance

Prior to the introduction of workstations, the SLC control program had been implemented as a tree of shareable images, both to conserve memory and to promote modularity. The shareable images near the root of the tree contain graphics, utility and communication procedures, while those higher up the tree contain applications code. In implementing the workstation software it was only necessary to diverge from the existing software at the lower level. Hence some shareables have a different version for the workstation and the host CPU; the remainder have only a single version which executes on all machines. The one difficulty with this arrangement is that it further complicates the problem of software distribution, which would have existed in any case as we moved from an architecture with two similar machines (development and production VAX) to one with half a dozen or more machines of two different types. The solution took the form of a set of VAX/VMS command procedures. One procedure is used to create a new shareable image; if this image is one that has a different version on the workstations then both versions are created. Typically the programmer then tests the new image, and if he is not satisfied he modifies his code and creates another new image. Other procedures distribute (i.e., copy and install) the new shareable to the appropriate CPU's. In all cases the programmer invoking the procedure merely supplies the name of the shareable to be created or distributed; he need not be aware of how many files are in fact created or distributed, or where they go.

Each workstation when restarted checks its list of shareable images against the master list on the SLC computer, getting any new released images it lacks. There are also several disk support files that are used by the SCP and kept locally on each workstation. The distribution for these files is done with command procedures similar to the ones used for the shareable images.

Conclusions

Workstation consoles were integrated into the control system to reduce the CPU load on the main host and to provide a more attractive user interface. Another implicit goal was to keep a high degree of compatibility between the SCP program running on the workstation and the version running on the host VAX. In the time available to us we were able to move the SCP (and, with little or no extra work, several stand-alone diagnostic programs) to the workstation. Except for certain classes of utility routines, the workstation SCP and host VAX SCP are identical. We were able to make some improvements in the graphical interface, most notably in the touchpanel and in giving error messages their own window, separate from the dialogue area. The latter feature has proved so popular that the COW interface has been similarly modified.

The design of the DNET Server and the DMSG protocol were appealing because of their generality. It was clear that any resources resident on the host could be provided to a workstation process in this manner, and that any we left out initially could be added later with a small incremental programming effort. However, that same generality implies a certain lack of optimization. All resources are treated equally, but in fact different resources are used differently and in different amounts by the SCP. The SCP, at least as presently used, is not very CPU intensive; it is dependent on SLCnet and especially on database access. The latter, which takes negligible resources for a process running on the host VAX, is quite expensive for both the workstation and the host CPU when the process is running on a workstation. Plans are now underway to distribute the database to all workstations.^[6]

Acknowledgments

The integration of the workstation console into the SLC control system could not have been accomplished without help and advice from many people. We would particularly like to thank R. Beach for writing the Unified Graphics GPx driver and R. K. Jobe for help in the design of the workstation knobs.

REFERENCES

1. N. Phinney, "Report on the SLC Control System," *IEEE Trans. Nucl. Sci.* **NS-32**, 2117 (Oct. 1985).
2. E. Linstadt, "Dissecting the COW," *IEEE Trans. Nucl. Sci.* **NS-32**, 2115 (Oct. 1985).
3. "VAX GKS IIb, Software Reference Guide," Digital Equipment Corporation, Maynard, Massachusetts (Mar. 1986)
4. "MicroVMS Workstation Graphics Programming Guide," Digital Equipment Corporation, Maynard, Massachusetts (Mar. 1986)
5. R. C. Beach, "The Unified Graphics System for Fortran-77: Programming Manual," SLAC, **CGTM-203** Rev. 2 (Nov. 1985)
6. M. Huffer, "Distributed Database for the SLC Control System *Design Proposal*," Internal Document (Feb. 1987)