CONF-951036--1

SLAC-PUB-95-7047
November 1995

Development of Operator Requested Control System Applications: Experience with the SLC Control System at SLAC*

M. Stanek,

Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309 USA

The SLC Control system at SLAC has evolved into a powerful tool for operation of the accelerator and for troubleshooting the unique problems encountered in extracting maximum performance from the SLC. The evolution has included the development of many custom applications and user interface features generated from accelerator operator and accelerator physicist requests. These applications are written and maintained primarily by the Controls Software Engineering group, and not by the users themselves. The process of developing and supporting user requested control systems applications at SLAC is described, including the effects of organizational structure, formal and informal procedures, and control system architecture.

The SLC Control System in general terms [1] consists of:

- a SLC Control Program (SCP) that operators use from an X-windows compatible terminal. The terminal may be equipped with a touch-sensitive screen to address the SCP's menu.

- the database describing accelerator devices.

- a host VAX which runs the SCP, database and other applications.

- standard SLC 80386 microprocessors that run the Intel RMX operating system, and are geographically distributed throughout the accelerator to control and monitor devices.

- the SLCNET proprietary network and network software.

Details of the system have been described in many previous papers, and will not be covered here [2].

The standard SCP, run from an operator terminal, is a menu oriented interface that provides status and control for all accelerator devices. It also runs all the special applications required to make measurements and optimize the performance of the accelerator. The SCP is built with sharable images [3], which has meant that tight control over code must be maintained to avoid errors from casual users impacting the main control program. One result of this somewhat "closed" architecture is that fewer hardware engineers, physicists, and operators have become directly involved in writing applications. The bulk of the responsibility for developing new applications as the accelerator has evolved has been shouldered by the Controls Department Software Engineering group.

Because of this focus of responsibility on a small group of specialists, the Controls department has been able to put processes in place to insure proper control of the SCP code. These include formal project and code review, and development and testing procedures [4]. In most cases, software management has assigned responsibility for each class of applications to more than one engineer. This allows operational support to be more consistent, with fewer lapses in coverage when key individuals are absent. Forward-thinking software engineers have also been able to design code with flexibility in mind, allowing future changes and upgrades to be made more easily. Maintenance responsibility for software problems is clearly defined, and managers are able to focus resources efficiently to address critical problems that affect the accelerator program.

With these positive characteristics come some difficulties that must be overcome to keep the customers satisfied. Anyone who has written even the simplest code realizes quickly what an iterative

*Work supported by Department of Energy contract DE-AC03-76SF00515.

Presented at the International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS '95), Chicago, Illinois, October 30-November 11, 1995

process it is. Iteration is much easier when the person specifying the code and the person developing it are the same. When the responsibility is split, communication between the requester, user and developer becomes extremely critical. Breakdowns in communication may result in delays, wasted time, or beautifully written applications that don't quite do what the user wanted, and rarely get used.

The organizational structure in SLAC's Technical Division is not optimized for ease of communication between Software Engineers and Operators. Software Engineers are part of the Controls Department, and the Operations and Accelerator Physics groups are part of the Accelerator Department. Both departments report to the head of the Technical Division, but the users (and requestors) of the SCP are quite removed from the software engineers in the formal organizational structure. For example, a control room operator and an applications programmer would go up four levels, to the Associate Director of the Technical Division, to find a common manager.

Despite the problems with closed architecture, limited software engineering staff, and organizational separation, software projects and upgrades are successfully commissioned at SLAC. The continued forward progress is due to several factors. Below are some of the measures taken at SLAC (some planned, others serendipitous) that have helped keep control system improvements on track.

1) Close physical proximity. The software engineering group is housed in the Main Control Center building and its adjacent trailers. This allows quick and easy consultation between operators or physicists and software engineers.

2) Flexible priorities. With close physical proximity, the software engineers need to develop a tolerance for momentary interrupts from curious or confused control room staff. SLAC has been fortunate that our software engineers are extremely helpful, and are willing to shuffle priorities to answer questions or help troubleshoot problems. Software management has also been supportive in shifting long-term project priorities to adapt to the changing needs of the accelerator physicists and operators. By the same token, the users must clearly communicate priorities for new requests, and realize the inefficiency inherent in constantly changing priorities.

3) Liaisons. Knowledgeable people from both the users and software groups have been dedicated to act as liaisons for changes to software and development of new projects. Even small changes to the database or user interface displays have benefitted from review by representatives from both groups. For example, software contact persons are able to identify potential errors or conflicts in minor changes, and operations contacts are able to suggest small improvements to "front end" interfaces that make displays less confusing. At SLAC, these liaison interactions may include informal, "on demand" consultations several times per week between reviewers, focussed meetings to address a particular problem, or ongoing formal project review meetings. Knowledgeable operations staff have occasionally been assigned to work part-time on the commissioning of particularly large applications. In the past few years, complicated control system upgrades for the accelerator timing system and machine protection system have been successfully brought online. A major reason for the smooth commissioning was the contribution of two operators, assigned to these projects during their "backup" or "off-shift" time. These operators did not write code, but prior to the code release, they were able to help configure and test the applications with realistic accelerator scenarios in mind, and were able to suggest minor changes to the interface packages that eased confusion when their colleagues first encountered the applications online.

4) Formal request process. The operators have a formal process for requesting new software, or changes to existing applications. A text file of software requests is kept in a common account on the VAX. All operators have write access to this file, and are free to add requests to the file at any time. A typical request contains only one paragraph, describing the problem and a possible solution. A few operations staff members are assigned, with a supervisor, to periodically review the file, approve or deny each request, and assign a priority to each approved request. These annotations are included in a new file, renamed "active requests". Priorities are based on the expected utility of the new software, and on how difficult it might be to develop. The "active requests" list is then reviewed by a software engineering manager assigned as the operations liaison, who estimates the cost in man hours, and puts the requests in a queue, to be assigned to the engineers. The operations reviewers and the software managers meet periodically to track the jobs, clarify the requests, and possibly reassign priorities based on the estimated cost.

DISCLAIMER

Portions of this document may be illegible electronic image products. Images are produced from the best available original document.

This process has been effective in getting some operator requests satisfied, but it can be a bit of a black hole as well. Many requests are good ideas, but are only minor improvements, or are not easy to implement, and so get approved with very low priority. There are still active requests from 1992 in the queue, which leads to the perception that this process is too slow. The operations staff assigned to review and track the requests do this in addition to working rotating shifts in the control room, so the periodic reviews are not done on a regular basis. It takes a bit of extra momentum to get projects out of the queue and into the SCP. As a result, this process has evolved into a method of addressing minor, low priority improvements.

5) Debriefing sessions. The formal request process allows the operators to jot down ideas as they come up, and keeps the good ideas from getting lost. To focus the group more on critical improvements that would help them better address the routine problems faced in operating the accelerator, a debriefing or brainstorming session is held at the conclusion of every accelerator run cycle. All operators attend the sessions, and together they are encouraged to generate a list of problems. Software engineering managers have also attended these meetings, and have been helpful in keeping the discussions "grounded" in terms of what software improvements are reasonable to expect. The list is sorted, and reviewed with key managers from the accelerator physics group. Some of these problems are best addressed with software improvements or new applications, and after further discussion with operators, formal requests are written and submitted to the software engineering group. These requests are given a relatively high priority, since they are endorsed by representatives from both the accelerator physics and operations groups.

One example that illustrates the positive outcome of the debriefing process is an improvement to the SCP optimization scan packages. To keep the electron and positron beams in collision at the SLC interaction point, and to minimize the transverse beam sizes, specialized applications were written to scan optical elements in the Final Focus, and measure the resultant beam size. The data is automatically fitted, and an optimum value computed. The operator may implement with a single button push, the change in the scanned optical parameter to accept the computed optimum value. The underlying software package for performing the scan and making the measurements is a general purpose "Correlation Plot" application [5], and the additional features (canned setups for the scans, automatic fitting, computing optima, and accepting best value) were all developed by a team of accelerator physicists and software engineers. The operators realized that these additional features could be helpful in optimizing the performance of other areas of the SLC, and requested that the special features be made available for other types of optimization scans. These expanded optimization packages are now used routinely by the operators. A new set of tools was provided with relatively small cost, due to the clever design of the original application package, and the ability to keep communication paths open between the groups.

6) Documentation. Significant SCP software changes and new releases are accompanied by an article in the Software Engineering Newsletter, called the *Index Panel*, which is published bi-weekly and distributed to the user community. The articles vary in length from a paragraph to several pages, and may include graphics to illustrate the use of new applications.

The online Help facility on the SCP allows documentation to be easily retrieved by the users. However, a lack of standards for Help files has lead to an inconsistent implementation. Operators have learned not to rely on the existence of useful help unless they remember encountering a particular informative file in the past.

7) Tools for routine changes. Although the complexity of the SCP is an impediment to the average operator writing specialized code, minor changes to the database, touchpanels, and graphics displays can be accommodated with canned editing routines. The software engineering group, with help from one knowledgeable operator, have provided several interactive command files for editing many useful parameters, including touchpanel menus, help files, simple graphic displays, and some database constants. These command files lead the user through the required steps to perform and document their changes, and allow non-experts to contribute their ideas for improvement without becoming bogged down in the details of the control system.

Certainly there are many other factors, not discussed here, that contribute to the success or failure of a control system to meet operator needs. The above items are some that have worked well at SLAC, from the perspective of an operations manager, and former operator. With cooperation of the users and software engineering groups, and diligence to keep the communications channels open, continuous improvements to the SLC control system have been a major reason behind the successful physics runs at SLAC in recent years.

REFERENCES

[1] T. Lahey, et. al., 'Adding PCs to SLC Control System', Proceedings of the 1993 Particle Accelerator Conference, Washington, D.C. SLAC-PUB-6211.

[2] N. Phinney, et.al., 'The SLC Control System: Status and Development', Proceedings of teh 1987 Particle Accelerator Conference, Washington, D.C. SLAC-PUB-4215

[3] J. Bogart, et. al., 'SLC Control System Basic Users Guide', April 1995

[4] S. Clark, 'Conduct of Operations: Software Engineering' SLAC Document 18-05-91-00, 15 April, 1993

[5] L. Hendrickson, et. al., 'Correlation Plot Facility in the SLC Control System', Proceedings of the 1991 IEEE Particle Accelerator Conference. SLAC-PUB-5464

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.