

NEW CONTROL SYSTEM FOR THE KEK LINAC

N. Kamikubota, K. Furukawa, K. Nakahara and I. Abe
 National Laboratory for High Energy Physics (KEK)
 Oho, Tsukuba, Ibaraki 305, Japan

Abstract

A new control system for the KEK Linac started operating in September, 1993. The new system comprises Unix-based workstations and front-end computers with a VME-bus. All of these computers communicate with each other using the TCP/IP protocol. Control software has been developed to provide transparent functions for controlling the basic accelerator equipment.

1 Introduction

The KEK Linac provides 2.5-GeV electron and positron beams to the rings, Photon Factory and TRISTAN. The old control system, which comprised eight minicomputers and hundreds of microprocessors[1, 2], has been successfully operated since 1982. However, since the system capability has become inadequate for increasing demands, we decided to replace the old system with a new one.

The new control system started operating in September, 1993. It consists of three components:

- (a) TCP/IP network as a main communication network,
- (b) Unix-based workstations as resource servers, and
- (c) VME-bus-based computers as front-end stations with field communication networks (LOOP-II and LOOP-III)[1, 4].

In this article, we discuss the details concerning the control software for our Linac equipment. The hardware architecture, such as available workstations and VME-bus computers, is presented elsewhere[4].

2 Control Software

2.1 Software model

The control software for the new system was developed based on a simplified communication model (Fig. 1). A client who wants to control (or obtain the status of) one part of the Linac is requested to send a control message to the control system. The control message contains a pre-defined command (usually abbreviated as *com*), the name of the object to be controlled (*name*), as well as the property of the object (*value*), if necessary. The control

software accepts the control message and prepares a reply message, including a return code. In the case of success, the return code is set to zero. The object property may be associated with the reply, depending on the specified command. When an unrecoverable error occurs, the return code has a negative value. One can find the reason in the reply message why the request failed.

The example given in Fig. 1 shows that an operator (a client) asks the control system to change the property of the object "TEST2". After the property of "TEST2" is changed from 12 to 25, the operator is notified of the success by a reply message.

The use of this model over all types of accelerator equipment enables us to use simplified communication software. This model always requires the round-trip of a control message for one cycle function, while the old control system was developed based on the one-way transfer of a control message.

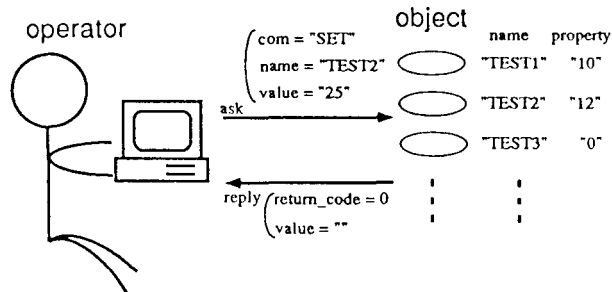


Figure 1: Simplified model for the new control system.

2.2 Flow of a control message

As shown in Fig. 2, a message from a client is received by a "message distributor" in a unix workstation. The message distributor re-directs the received message to a corresponding VME computer without changing the contents of the message. Finally, a server process at the destination VME receives it. The reply message goes back along the same path to reach the client.

A message transfer is realized by using the TCP/IP socket library. Details concerning the communication software are presented elsewhere[3].

2.3 Tables and log files (database)

The control software uses table files as a parameter database. A table contains the relations between the object names and the corresponding hardware parameters (for example, the board address and the network node-name of the VME system). Since the parameter tables are ASCII files, one can easily update them.

The control software produces several kinds of history records. They are recorded into log files for later analysis. The information levels (save all history records, save error information only, etc.) are tuned according to the control equipment concerned.

These tables and log files are commonly used for all of the workstations and VME computers with the NFS file-sharing technique. The relation between the table, the log and the control system is shown in Fig. 2. Since this system is a very simple database developed by ourselves, we have a plan to utilize a commercially available database in the future in order to realize higher functionality and easier maintenance.

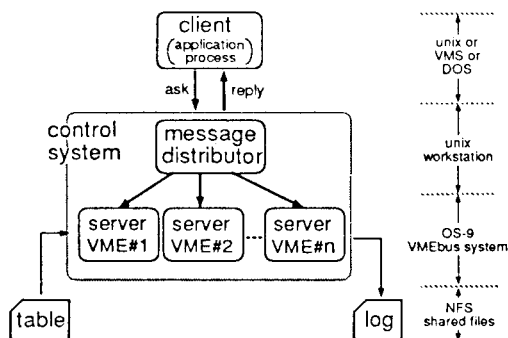


Figure 2: Flow of a control message. Though only one-way arrows are shown in the figure, reply arrows exist.

2.4 User's interface

Three kinds of services are available for actual control of the Linac equipment.

Utility commands Utility commands to control various accelerator equipment are available. Utility commands are useful for maintenance purposes. For instance,

one can obtain the currents of the magnet power-supply "Q3-4B" by a utility command "mg":

```
unix% mg cur Q3-4B /* get magnet currents
16.967 16.977 /* set-value and read-out
```

C functions The functions of C language have been prepared for more complicated needs or for cases when a faster response is required. An example of how to obtain a magnet current is shown here.

```
float curr[2]; /* magnet currents */
int rtn; /* return code */
rtn = mg( "CUR", "Q3-4B", curr );
```

Since our communication software has high availability among different computers[3], a user can use VMS or DOS machines as well as unix workstations.

Documentations for both utility commands and C functions are now being prepared. Part of the online manuals are already available at the unix workstations.

tcl language commands In addition to the previous services, we have tried to utilize the tcl/tk user interface available on X-Windows[5]. We already provide tcl built-in commands to access accelerator controls for operators. The formats of these tcl commands are the same as those of utility commands. Using the tcl/tk and control built-in commands, we can easily build operator interfaces.

2.5 Controllable equipment

Controllable equipment can be divided into two groups. The first group corresponds to VME modules, such as digital input/output and modules for LOOP-II/LOOP-III. The second group comprises various accelerator equipment, such as magnets and klystrons. The control software for the second group calls functions of the first group, but never makes direct access to a VME module. During the development stage for both groups, attention was paid not to include (or to include as little as possible) any hardware-dependent description in the user's interfaces. We put those into database tables.

It is worth noting that tables, log files, message distributors, and user's interfaces independently exist for each control equipment. The control messages forwarded to a specific equipment do not have any interference with those for other equipment. This feature is realized by assigning different socket servicenames for each equipment.

2.6 Security

A message distributor checks the client hostname each time a connection request arrives. Any requests from un-registered hosts are rejected. It is also possible to allow a limited number of commands (usually read-only commands) from registered network nodes. Also, the IP connection from outside the Linac control network is prohibited.

2.7 Surveillance processes and alarms

Surveillance processes, which always monitor specified equipment, exist in each VME computer. These processes do not send any control message, but monitor the equipment status by watching the online database at a shared memory area. When a surveillance process finds something changed, it makes an alarm message (a string) and writes it into a log file located in the shared disk. An operator receives alarms by watching the log file in the shared disk. This method is suitable for keeping the network traffic low.

In addition to the changes of the equipment status with control messages, the changes caused by an unexpected fault of a local controller are also saved in a log file. Such history records in the log are very useful when searching for the origin of a controller fault.

When each VME computer does not have sufficient CPU power, it may cause a considerable delay of alarm messages. Currently, 4-6 surveillance processes are running with lower priorities in each VME computer. Typical delay times are 1-4 seconds.

3 Acknowledgement

We wish to thank Professor I. Sato for his encouragement during the development of the new control system. We also express our best thanks to the Linac operators for support during the software development.

References

[1] K. Nakahara, I. Abe, R.P. Bissonette, A. Enomoto, Y. Otake, T. Urano and J. Tanaka, Nucl. Instr. Meth., A251(1986)327

[2] K. Nakahara, I. Abe, A. Enomoto, Y. Otake and T. Urano, Nucl. Instr. Meth., A247(1986)153

[3] N. Kamikubota, K. Furukawa, K. Nakahara and I. Abe, Proc. Int'l Conf. on Accelerator and Large Experimental Physics Control Systems

(ICALEPCS'91), Tsukuba Japan, November 1991, p.318

[4] N. Kamikubota, K. Furukawa, K. Nakahara and I. Abe, Nucl. Instr. Meth. A(1994) in press; KEK-Preprint 93-137

[5] "Tcl and the Tk Toolkit", J. Ousterhout, (1994, Addison-Wesley)