Console Software for the PLS Storage Ring Control*

Jae W. Lee, Jae H. Lee, Jong H. Lee, Eun H. Lee, and Sang C. Won
Pohang Accelerator Laboratory
POSTECH, San 31, Hyoja-Dong,
Pohang 790-784, Korea

Abstract

Console software for the PLS(Pohang Light Source) storage ring control has been developed at Pohang Accelerator Laboratory. Sun Microsystems' SPARCstation2GX™s have been used to develop the software which will be actually run on SPARCstation10/51GX™s and SPARCstation10/514MP™. This paper describes an overall structure of console software including database software, communication software, man-machine-interface software, and application software.

1 INTRODUCTION

The upper level network of the PLS storage ring control system is a baseband Ethernet which links 4 Console Computers, one Host Computer, 3 Xterminals, 2 personal computers, and 4 SCCs(Subsystem Control Computer). Console Computers are for operators to monitor and control the machine, while Xterminals display the machine status continuously. Computing power of the Xterminals rely on the Host Computer. Two personal computers are provided on the control network for machine status display. They also generate computer graphics for TV broadcasting. Four SCCs are frontend computers of data acquisition system. They supervise Closed orbit correction system, RF system, Timing system, and MPS, Vacuum and Beam diagnosis system. Console software consists of database management programs, communication programs, man-machine-interface programs, application programs, TV broadcasting programs, and miscellaneous utility programs. Man-machine-interface programs have been developed using OSF/Motif™, Xt-Intrinsics™, and Xlib™.

2 STRUCTURE OVERVIEW

All machine control processes in Console Computers fetch necessary data from the central database residing in the same computers. The central database consists of Static database partition, Dynamic database partition, and several shared memory partitions. They all are actually shared memory segments. The internal structure of the Static database is very similar to that of SPEAR database in SSRL. The Dynamic database has unique internal structure of our own, whereas others are typical CERN MOPS(Multiple Object Partitioned Structure) shared memory partitions. The Static database partition stores all sorts of static information to manage machine control signals. The Dynamic database partition consists of the area for storage of the setpoint values and the one for the readback values of the control signals. Current setpoint values of the machine components and their readback data are updated in realtime basis. The CERN MOPS memory partitions are used to share the beam parameter data among processes in Console Computer. Man-machine-interface processes and Application processes run on the Console Computers and the Host Computer to keep the machine in proper condition and allow operators to supervise the machine. The man-machine-interface processes display a current or historical status of machine components and electron beam graphically. They also provide useful tools for operator to control the machine. The application processes log the important monitoring information, diagnose the machine components, make slow feedback control, and search an alarm situation. Communication processes interface the lower layer of VME-bus based data acquisition systems which are connected via Ethernet. They forward the control commands of the Console Computer processes to the data acquisition system through TCP/IP socket connection, or continuously refresh the Dynamic database partition with the present machine component data. Other utility programs have been developed for database management, operating configuration management, and local tests.

3 DATABASE SOFTWARE

The Static database management utility programs are borrowed from Stanford University to save the developing effort. They were originally written in Fortran to run on VAX/VMS™ for SPEAR control. They have been rewritten in C language here with some minor modifications to conform to our control system environment(SunOS™ and SPARC™ workstations). Some utility programs have been developed for our own Dynamic database management while CERN MOPS is borrowed for sharing information among processes running on the Console Computers and the Host Computer. MOPS, a data structure management system was originally developed for CERN LEP/SPS control system and was designed to support the programmers in organizing their data in a structured fashion. Including static database download program, most of the database management utility programs and access

*Work supported by the Ministry of Science and Technology of the Republic of Korea and Pohang Iron & Steel Co., Ltd.

libraries have been developed. RDBMS(Relational Data Base Management System) interface will be developed to improve the data compatibility between our machine control database and commercial RDBMS in the future.

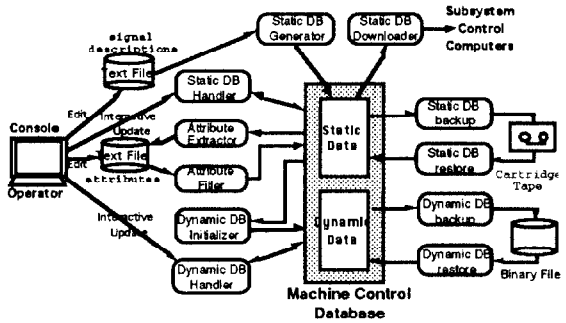

Fig.1 Database software on Console Computer

4 COMMUNICATION SOFTWARE

Communication software enables programmers and operators to control and monitor the machine transparently. No one should be aware of where the device locates or which SCC supervises the signal he wants to control. It provides the means with which control commands are to be forwarded to the right MIU(Machine Interface Unit) and with which realtime machine information is updated onto the Dynamic database partition. Currently, all programs of this sort use TCP/IP socket stream to establish the communication channel.

4.1 Realtime Data Refresh Process

Specific parts of the Dynamic database partition can be continuously updated with the present machine data by this process. It is initiated by any Console Computer process which needs the fresh data realtime. One can request the refresh for the only specific part of the whole control signals to reduce the network traffic. For good system performance, duplicated or similar refresh request is designed to be cordinated automatically. Any process which wants realtime information for a group of control signals can requests the refresh by specifying the desired signal group and interval. Normally, the refresh is to be continued until the requester takes it back. During the refresh, any kind of trouble is detected and notified to the requester via interprocess signal(SIGUSR1). Any block data such as video data or bunch of measurement data can be selected to be refreshed continuously. Classification of control signal groups are predetermined in the Static database.

4.2 Control Command Forwarding Subroutine

Setting a specific signal to some value can be achieved by making a subroutine call with arguments such as control signal name and setpoint value. The programmer does not need to know the detailed information about the signal he wants to control. All he has to do is to define the name and setpoint value. The subroutine checks the validity of signal name and class(setpoint or readback). It also checks the setpointing range to prevent the trouble which may be caused by exceeding allowed setting limit. One can trace the issued commands later since every instruction is logged on a file.

4.3 Multiple Control Command Forwarding Subroutines

Similar subroutines were developed to set multiple signals to corresponding values at once. One can use data array in his program or a text file to define the signal names and corresponding setpoint values. Validity check feature is equivalent to the Control Command Forwarding Subroutine's except that no instruction is to be executed if any signal name and setpoint value pair has trouble. Location transparency is kept again and it is allowed to mix up signals which belong to the different SCC or MIU.

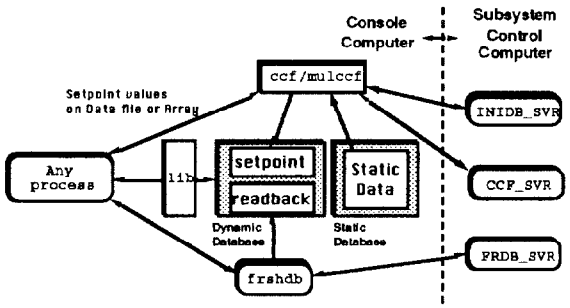

Fig.2 Process interaction on communication

5 APPLICATION AND MAN-MACHINE-INTERFACE SOFTWARE

Application software is to check the machine component status automatically, do the slow feedback control, log on the important machine status, or notify the operator of the critical failures. Man-machine-interface programs are to provide the operator with an easy-to-use operating environment for supervising and diagnosing the machine. All Console Computer processes may ask Realtime Data Refresh Process to update the corresponding Dynamic database partition for the signal group they are interested in. They use database access subroutines to get the interesting realtime data from the Dynamic database and display the machine status on Console screens. They also instruct Control Command Forwarding Subroutines to set some control signals to the desired setpoint values. OSF/Motif™ widgets, Xt-Intrinsics ™ and Xlib™have

been used to implement man-machine-interface on Console Computers and Xterminals.

6 UTILITY PROGRAMS

Beside the database management utility programs, several utility programs are needed for operating configuration management and component calibration. Several SCC emulation programs have been developed to help locally testing the Communication Software.

7 STATUS BROADCASTING SOFTWARE

Some information should be provided to the technical staffs and beam line users. Beam energy, current, lifetime, number of bunches, experiment beam line user status, and operation schedule belong to this kind of information to be broadcasted. This information will be provided from the Host Computer to the personal computers via communication channels. Broadcasting software running on the personal computers turn the information into computer graphics, which will be broadcasted through Cable TV network. Every staff and beam line user can see this with his home TV. This information can be accessed from the Data Bank on the Host Computer through the research network at the same time.

8 ACKNOWLEDGEMENT

The author would like to acknowledge the many people who have contributed to the design and implementation of Console software for the PLS storage ring.

9 REFERENCES

[1] S. C. Won, S. S. Chang, J. Huang, J. W. Lee, J. Lee, J. H. Kim, "Status Report on Control System Development for PLS", Proc. of the International Conference on Accelerator and Large Experimental Physics Control Systems, Tsukuba, Japan, Nov. 1991, pp. 147-150.

[2] S. Howry, T. Gromme, A. King, M. Sullenberger, "A portable database driven control system for SPEAR", SLAC-PUB-3618, SLAC, Apr. 1985.

[3] W. Herr, "MOPS - A data structure management system for the LEP/SPS control system", CERN SPS/89-8(AMS), European Organisation for Nuclear Research.

[4] Werner Herr, "MOPS User Guide for C programs", CERN/SPS/88-43(AMS), European Organisation for Nuclear Research.