COACK APPLICATION FOR THE BEAMLINE INTERLOCK SYSTEM AT THE PHOTON FACTORY

Takashi Kosuge, Yuuki Saito, Kazuyuki Nigorikawa
Kenji Ito, Isamu Abe and Shin-ichi Kurokawa
High Energy Accelerator Research Organization, Japan

Abstract

Component Oriented Accelerator Control Kernel (COACK)[1][2], originally designed for controlling accelerator complex, has been adopted for a new central control system for a complexes of more than 20 beamline interlock systems, installed at the 2.5-GeV electron storage ring of the Photon Factory. We have described the details of the entire system, emphasizing the hardware with which the new system was actually developed with COACK.

1 INTRODUCTION

About 20 beamlines are installed at the 2.5 GeV electron storage ring at the Photon Factory for using synchrotron radiation efficiently. Each beamline is equipped with an interlock system, beamline interlock system (BLIS), to avoid vacuum troubles and to protect the users from radiation hazards. A programmable logic controller (PLC) is used for the control of the BLIS, which is connected with a central control system (CCS) by optical fiber, as shown figure 1. The CCS monitors the status of beamlines through the BLISs and sends control signals to the BLISs. In figure 2 is shown a block diagram of the present CCS, which consists of a DOS based PC, a CAMAC system, and a custom-made control panel. This old-fashioned CCS was constructed more than one decade ago, and should be replaced by a new one sooner or later. Furthermore, one of shortcomings of the old system is that one cannot access to the PC for the system maintenance during the storage ring operation, 24hrs over 24 hrs and 7 days over 7days, due to a huge program running only in one PC with a small capacity.

COACK (Component Oriented Accelerator Control Kernel) has been developed as new software for controlling huge accelerator systems. It has been found, after severe performance tests, that COACK is a very powerful tool not only for controlling accelerators but also for other complicated systems, such as a control of all the beamline interlock systems. In addition, COACK has a big advantage in a budgetary point of view; we need low price personal computers, and a commonly used network to construct a whole system. This is why we adopted COACK as a new central control system replacing the old-fashioned central control system.

The application of COACK to a new central control system has not been straightforward. The renewal of the old system should meet the requirements; 1) the shortcoming of the old system should be conquered, 2) the renewal should be done in a short period, and not be so complicated. In this contribution we will describe how we have developed a new system with these requirements satisfied.

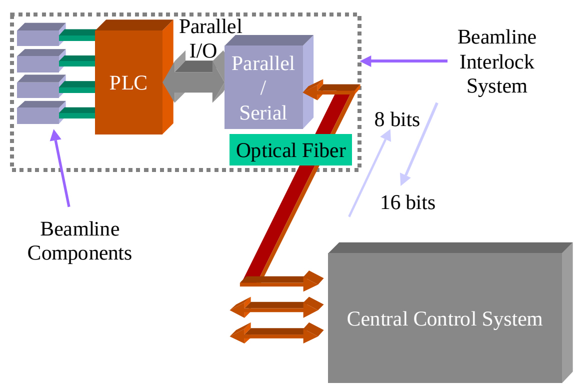

Figure 1: Layout and connection of beamline interlock system

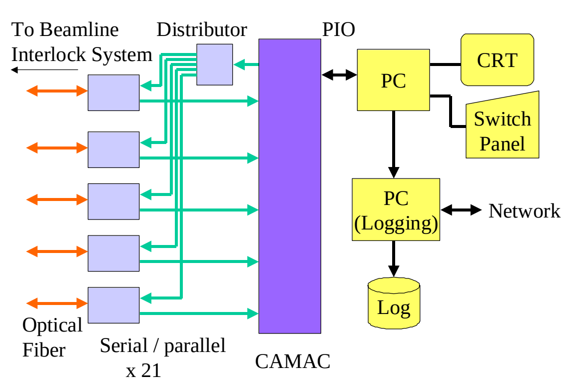

Figure 2: Layout of an old central control system

2 NEW SYSTEM WITH COACK

A schematic of the newly constructed system with COACK is shown in figure 3, in which a conception of control systems using COACK can be learned. The most important PC is a COACK server, in which a virtual system, virtual beamlines in this case, is constructed. We have a PC in the CCS, a PC for operators, and a PC for program development, which are connected with the COACK server by a LAN network.

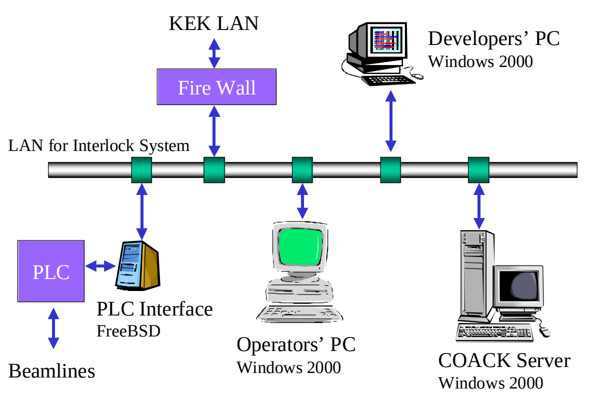

Figure 3: schematic of newly constructed system with COACK

2.1 Replacement by PLC

As already shown in figure 2, we monitor 16 different status from each beamline, and these 16 bits are received by CAMAC I/O registers, which send the data from "20 beamlines sequentially in time to the PC. On application of COACK to the new system of beamline interlock system, CAMAC has been replaced by PLC as shown in figure 4, because CAMAC is already out-of-date as an interface tool.

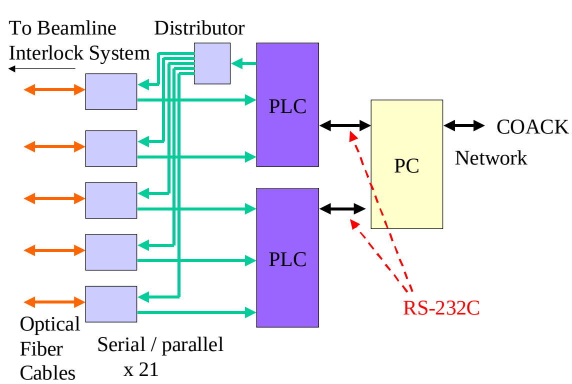

Figure 4: new CCS with two PLC instead of CAMAC

As we have over 20 beamlines, and are obliged to install two PLCs, which are connected with a PC by RS-232C (9600 bps). We have chosen RS-232C, for we are able to adopt most of OS for the PC. In monitoring the beamline status, it is important to know in what sequence (in mil-sec order) the status changes. In the new CCS, the two PLCs make such an important role. Another advantage for using PLCs is a status buffering function with PLC, which is shown in figure 5. When the 16-bit information from each beamline changes, it is stored in data memory area by using the Xor function of the PLC.

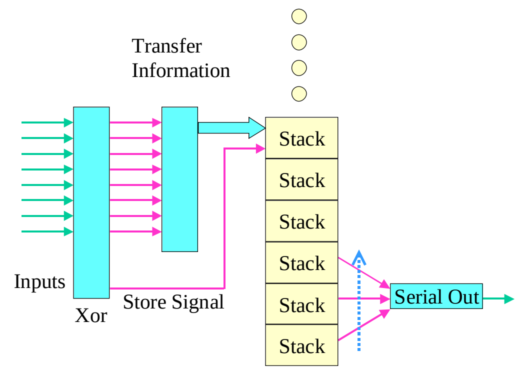

Figure 5: Status buffering with PLC

2.2 Hardware and OS

As shown in figure 3, we have the PC in the CCS, the PC for operators, and the PC for program development, beside the PC for the COACK server. We adopted FreeBSD 4.0 Release for the PC in the CSS, which is used as an interface with beamline interlock systems. Windows 2000 Professional is used as an OS for the PC for operators. We can avoid the shut- down of the system on developing or fixing programs with the PC for the development.

2.3 Performance

Specification of the PC for the COACK server is as follows;

- CPU: Pentium III 500MHz,
- Memory: 256Mbytes,
- OS: Windows 2000 Professional.

As a feasibility study of the COACK server and the newly constructed system (figure 3), we sent 598 event messages in 24 seconds from the PLC to the COACK server to see how heavy the CPU load in the COACK server is. As shown in figure 6, the CPU load attains to a value of 100 % right after sending the messages, keep it for 35 seconds, and is reduced to a normal value. This feasibility study shows that the newly constructed system works properly. For future development, we intend to include more PC and devices in the system. Thus, the present COACK server should be upgraded. At present, we are planning to introduce a dual CPU PC as a new COACK server by the end of this year.

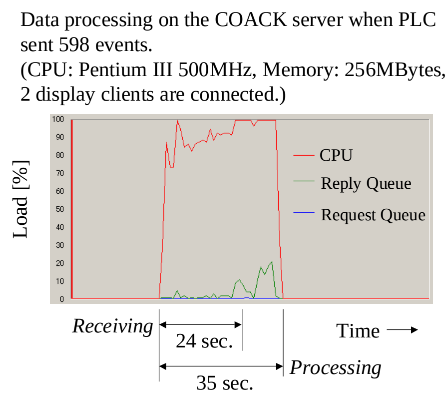

Figure 6: CPU load test of the COACK server

2.4 *Software*

A single big program was used in the PC of the old CSS. Functions of the program are as follows;

1. Check input data from beamline interlock system,
2. Save log data to archive files,
3. Display beamline status on a CRT,
4. Send permission data to beamline interlock systems.

In the new system, each function is distributed to the three PCs, as shown in figure 7. Functions 1 and 2 are carried out in the PC in the CSS, and Functions 3 and 4 in the PC for operators. Function 2 should be done with the COACK server. However, a reasonable logging speed cannot be obtained with the present COACK server. We hope that the data logging will be carried out by the dual CPU machine replacing the present one by the end of this year. The PC for the COACK server covers main logic part of these functions. All programs running on the PC for operators are written in Visual Basic, which is useful for the graphic display for beamline components. An example of the graphic display is shown in figure 8.

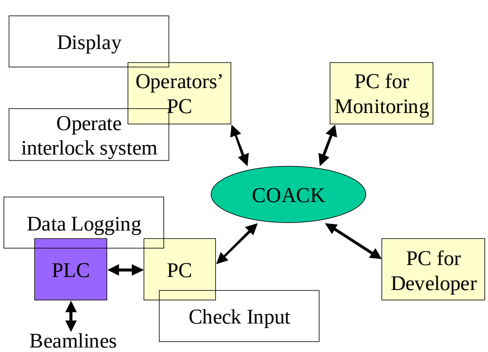

Figure 7: Software sharing for the new system

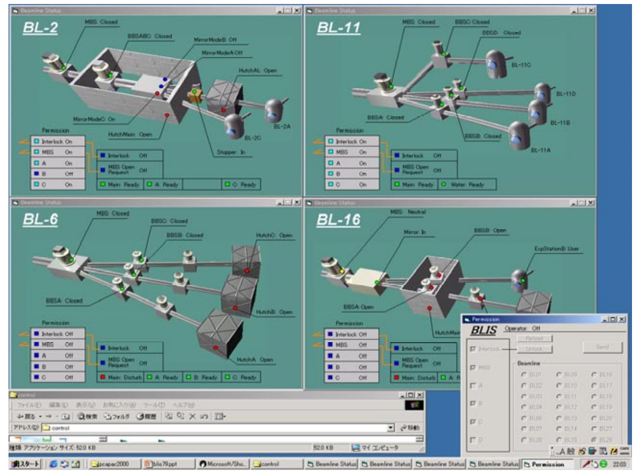

Figure 8: Example of graphic display at the PC for operators

2.5 *COACK interface for non-Windows*

An operating system of the PC in the CCS is FreeBSD, while other PCs, including the PC for the COACK server, use Windows 2000. COACK supports TCP/IP socket interface and XML massages for the communication between the COACK server and other clients. We have developed an interface program using TCP/IP Socket and XML for non-Windows systems. The program is written in Perl, and easily adoptable in various kind of operating systems, such as Red Hat Linux 6.1. The COACK interface program is very important for non-Windows systems, in which XML messages are frequently encountered.

3 ADVANTAGE OF COACK

We found advantage in introducing COACK into the new system for controlling the CCS and beamline interlock systems.

Firstly, we did not have to develop core program for the CCS. When we develop the system like the CCS, we usually have to spend a long time for developing the main part of the system (e.g. develop sending messages protocol). This main part already exists as COACK, which is developed for controlling huge accelerators. Thus, we are able to concentrate on programming the hardware drivers and user interfaces. In fact, it took only two months to construct a first version for the new system, while we spent about 6 month for programming for the old CCS.

Secondly, we can avoid a shut-down of the system on upgrading programs and editing databases, while we had to shutdown when updating user database or fixing bugs in the old system.

4 SUMMARRY

In summary, we have shown that COACK is useful as a control system for beamline interlock systems. We emphasize that we have developed the COACK interface program with non-Window systems. For the next development (figure 9), we are planning to install a PC in each beamline interlock system, which is directly connected with the PC for the COACK server by LAN. This PC runs FreeBSD as an operating system, and has the same function as the PC in the CCS. We believe that this plan will be realized at a low price owing to FreeBSD and these PCs consisting of CPU and I/O interface.

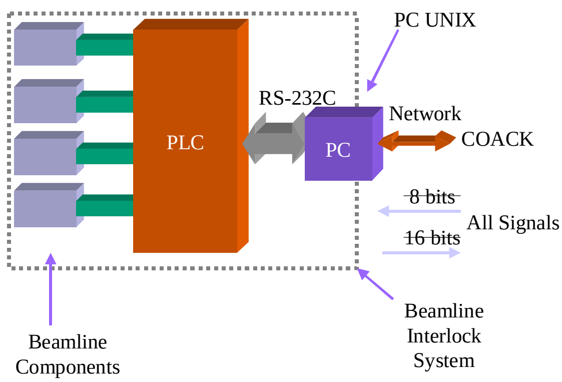

Figure 9: future development of beamline interlock systems with COACK.

REFERENCES

[1] I.Abe, et al., "COACK-II PROJECT ON ACCELERATOR CONTROL KERNEL DEVELOPMENT", ICALEPCS'99, Trieste, 1999

[2] I.Abe, et al., "Recent status on COACK project", This workshop, Hamburg, 2000