

NEW CONTROL SYSTEM USING COACK FOR KEK SLOW-POSITRON ACCELERATOR

A. Shirakawa*, T. Kosuge, K. Nigorikawa, J. Kishiro, S. Kurokawa, T. Shidara, T. Kurihara and I. Abe
 KEK, 1-1 Oho, Tsukuba, Ibaraki, 305-0801, JAPAN

Abstract

KEK slow-positron project is one of the applications of electron linac (linear accelerator), and is the facility to be opened for user experiments. We have made the control system for its primary electron linac implementing COACK (Component Oriented Advanced Control Kernel) [1].

The new system consists of COACK server & clients, LAN, and PLC-contained controllers as IOCs. COACK provides the cost/time saving for coding control programs for various types of accelerator components. Even though some parts of the system are not implemented, the virtual machine concept in the COACK makes it very efficient to develop some components for the remote control.

1 INTRODUCTION

KEK slow-positron experiment had originally started in 1991, using the 2.5-GeV electron Linac, which was an injector both for TRISTAN and PF-Ring. After a few transitions, last year we started to construct a new facility possessing its own linac in the same building. Though user beam lines are still under construction, the primary linac has prepared and started its operation. We had to make up everything in a short time, with low cost, and with little manpower.

As for the control system, we have much experience in controlling the KEK 8-GeV (upgraded from 2.5-GeV) electron linac by PC, especially with Visual Basic (VB) programming. On the other hand, a new powerful technology COACK has been establishing. Combination of them shows very good performance in constructing a new system.

2 SLOW-POSITRON FACILITY

Positron is the anti-particle of electron and has unique properties. Low energy positrons are useful as a probe for

material science. By using an electron linac we can obtain high intensity positron beam, which is hundreds or thousands times stronger than that from available isotope. One of the expected experiments is 'positron microscope.'

The new facility comprises an 8-meter primary electron linac, a metal target to create positrons, and a beam line for user experiments. Fig.1 shows the current outline of this facility. Although the scale of the machine is rather small, it contains many types of accelerator components. In the spring of 2002 we succeeded in accelerating electrons and observed positrons at the end of the positron beam line.

3 OUTLINE OF CONTROL SYSTEM

Table 1 shows devices and numbers of I/O signal lines to be controlled. These are temporary numbers; more I/O signal lines will be added according to upgrades. Furthermore, when the experimental devices are completed, the scale of whole system will be twice or third times bigger.

Table 1: Devices to be controlled in the primary Linac

Series	Devices	I/O lines
Safety interlock	Interlock switches & Beam operation	60
Electron gun	4 power supplies etc.	40
Beam transport magnets	15 power supplies	180
Vacuum	10 vacuum pumps, 2 gauges, 2 gate valves	110

Figure 2 shows components in the control system. Most of devices are connected to controllers containing Yokogawa PLC. Connection between them is analog I/O

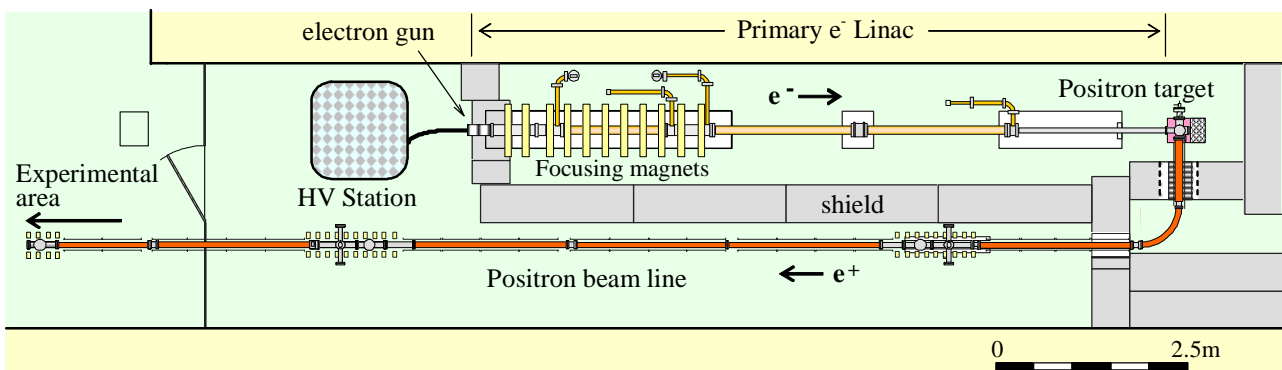


Figure 1. Outline of KEK slow-positron facility.

*akihiro.shirakawa@kek.jp

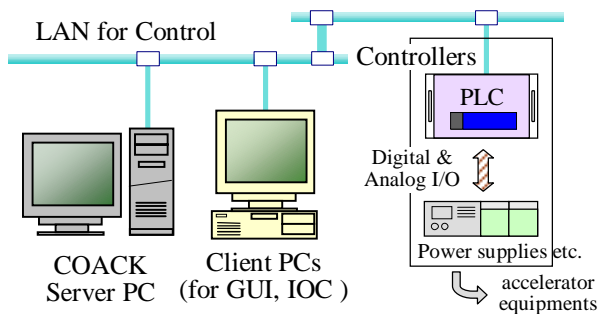


Figure 2. Hardware components in the control system

or digital I/O (DC24V or TTL level.) Each device series has one or two controller(s). PLCs are on Ethernet line, and communicate with PC by UDP/IP. Actually some part of LAN and IOCs has not implemented; they will be completed soon.

We have COACK server with 1.8-GHz Pentium-IV single CPU and memory of 1 GB. The OS is Windows 2000 Server. We have two client PCs, one for handling IOC-PLCs and the other for operator GUI. We are using 433-MHz machines for them.

4 SOFTWARE ARCHITECTURE

COACK, as a core of the system, serves many useful component-wares in the accelerator control: DB, cache, alarm-system, logging, etc. Communication components are also available, if you choose VB for coding and DCOM for server-client connection. We did so, and what we had to do were:

- To build up object (device) class structure based on our facility at COACK server.
- To draw GUIs for operator.
- To write IOC-handling programs.
- To make PLC sequence programs.

Building up class needed about one week. Most of the period was spent for analysing object relations. Class Builder Tool by COACK was quite useful for the work. The class structure was determined as shown in Figure 3.

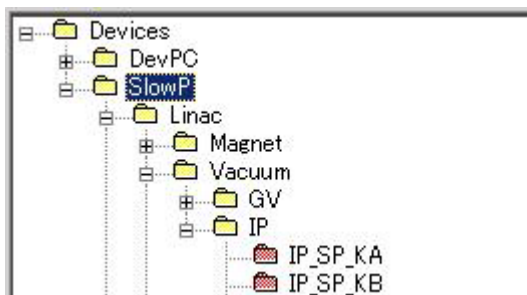


Figure 3. Image of class structure of Slow-positron device. Vacuum nodes are shown for example.

- What we did for preparing operator GUIs were
- opened new VB file and positioned some buttons, text boxes, graphs and figures;
 - implemented DCOM communication component from COACK;

iii) linked picture and DCOM component.

A set of GUIs for one device series needed only a few days to complete by one programmer. At linking, just four kinds of commands are set: 'Get Cache', 'Set Cache', 'Flag On' and 'Flag Off.' These commands are already prepared in server. We can access to server cache and get event notification from server easily. We have two or three kinds of GUI windows for each device series. They are shown in linac main control room, sharing with conventional linac.

At IOC-handling part, it polls to PLC periodically (once in a second, typically) to get status data. If/when data changes, it will set cache data to server. It also flags on to server to get request for data-setting. When request event arrives, it sends set command to PLC. Message format to PLC is proper to Yokogawa's, on the UDP/IP protocol. For the protocol we have interface .dll package, which we had developed before. For non-existing controllers, we treat them as virtual objects. Handling code for them is arranged a little as if they exist.

In making PLC sequences, we referred to PLCs of conventional linac. Since equipments under both PLCs are almost the same, we made copy of the existing program and had a slight adjustment to the new facility. It saved time.

Total time for completing GUIs and IOC programs for one device series was about one week.

Figure 4 shows the structure of software. We can see how many components and functions COACK provides.

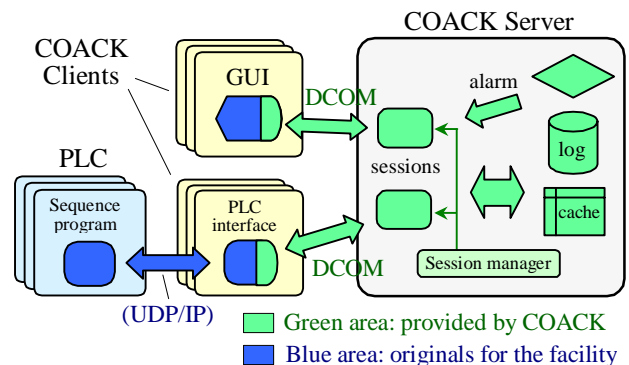


Figure 4. Software structure of new control system.

5 OPERATION STATUS

5.1 Safety interlock

Full control components have been prepared for safety series. They are working very well without troubles. Required functions for safety interlock series are as follows:

- Beam operation sequence
- Interlock sequence for radiation safety
- Display information on operator's console

All the program sequences are in the PLC. GUI on PC is only for showing information for operator; no action from computers is permitted. Beam operation is done with hardware switches. Besides the logging system in server, we have another small log at console, which is also a

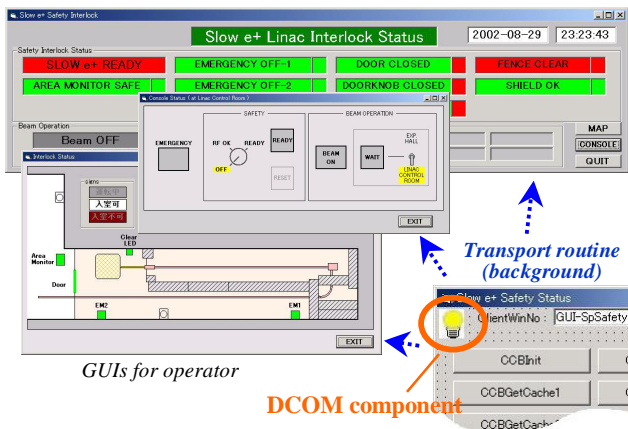


Figure 5. Images of GUIs for safety interlock.

client of COACK. In this series, message traffic is less than ten packets per second at the most. Image of safety interlock GUIs are shown in Figure 5. DCOM communication routine at background receives messages from server and then takes action to GUI windows.

5.2 Beam transport – as virtual machine

Since IOC for beam transport series has not prepared, we treated it as a virtual machine and coded IOC-handling programs. But GUIs are ready to use as if everything is ready.

Program complex that requires CPU load is the one for this series. When transferring a set of magnet parameters, CPU load and network traffic will increase.

We arranged IOC program for performance test. When we transfer data for all of 15 magnets at once, CPU load instantly arises to 100%, because ‘Set Cache’ commands and event notification for each magnet occur simultaneously. Use of a multi-CPU server PC would be very effective for such case. Or even with single CPU, if we pack 15 pieces of information in one packet, CPU load and traffic will decrease radically. It will not be a big problem, however, because we do not have much chance to change all the parameters at once in operation.

5.3 Others – Vacuum and Electron gun

For vacuum and electron gun series, every control component is OK except for Ethernet line. Each program has been prepared and tested. In these series the

frequency of commands or events is low. There should be no problem.

6 DISCUSSION

In our case, we referred to control system of 8-GeV linac. It surely helped us, but it is relatively small contribution compared with the use of COACK. One of noticeable advantages is that one can create client programs easily even if she/he does not understand insides of COACK well.

Programming IOC-handling parts takes most time in constructing system. If any convenient software package is provided for them, it will be much easier [2].

Thinking about unification of control systems in the future, non-DCOM interface will be required to contact with KEKB EPICS system [3][4].

Machine CPU needs to be upgraded, especially when the system covers all the slow-positron facility.

7 SUMMARY

We have made a new control system for KEK slow-positron linac. It is working stably. This is one of the application projects of COACK use. Various functions and tools of COACK save the cost/time for coding. The virtual machine concept helped us to complete the control system, though the whole set of interface has not installed yet.

8 REFERENCES

- [1] I. Abe et al., “Recent Status of COACK and it’s Functions”, PCaPAC2000, Hamburg, Germany.
- [2] M. Mutoh et al. “Development of Generalized Device Layer for the COACK System”, this conference.
- [3] T. Kosuge et al., “COACK Application for the Beamline Interlock System at the Photon Factory”, PCaPAC2000, Hamburg, Germany.
- [4] T. Kosuge et al., “Development of Simple Transmission and Retrieval System for Control Systems”, Proc. of Symposium on Technology in Laboratories by Department of Engineering and Technical Services, May 2002, Toki, Japan (in Japanese).