DESIGN OF SPring-8 LINAC CONTROL SYSTEM USING OBJECT ORIENTED CONCEPT

H.Sakaki, H.Yoshikawa, Y.Itoh, A.Kuba, T.Hori, A.Mizuno, H.Yokomizo
JAERI-RIKEN SPring-8 Project Team, Kamigori, Ako-gun, Hyogo 678-12 JAPAN

Abstract

At the present software technic, the methodology of Object Oriented Programming(OOP) has become the mainstream. Because it will be gave the efficient development of programming and the easy maintenance. OOP makes many parts of program that called Object, and these Objects are assembled into main program. When many Objects are reserved on the system development, so that the future development will be easy by using stored Objects.

On the accelerator, we consider the system as assembled parts. And it will always be improved or replaced. So OOP is suitable for the accelerator system [1], then we try for using OOP. First of all, all component of SPring-8 Linac were abstracted and modeled, and we designed a Super Class which is core of Object.

In this paper, the concept of our system and its present status are described.

I. STRUCTURE OF CONTROL SYSTEM

For the data acquisition and low level control of devices, we adopt the VME system. The hierarchy of control system have two layers. One is Control Process layer(VME), another is Operation Process layer(WS). These VMEs have MVME147s board(Motolora 68030 CPU) and run on OS-9 operating system. The WS is going to use HP9000 series. Table I shows I/O boards which are used in the VME system. All of them are commercial products. However device drivers that connect between OS and I/O boards were written by in-house staff. The VME system will locate in the second floor where is called the klystron gallery. And we will have many accidents at the control, because of the klystron EMI. Now we are examining about its [2].

Table I
SPring-8 Linac I/O boards

Name	Function	Number of boards
MVME147s	CPU(68030)	20
AVME9350	AI(16ch)	34
AVME9210	AO(6ch)	25
DVME-DIN3	DI(64ch)	53
DVME-DOUT3	DO(64ch)	31
VPAK601	MotorControl(2ch)	55
EVME-GPIB21	GPIB	18

II. STRUCTURE OF SOFTWARE

A. Layout of Object

Fig 1 shows a layout of Objects in Linac. Objects in the VME system, as one to one projection of real devices. So that we can control as if the Object is a real device. When a real device is replaced a new specifical device, the Object is only replaced the new Object, that's all. This is different from the conventional structured programming, therefore we can get high programming efficiency. On the Man Machine Interface(MMIF), now we are designing.

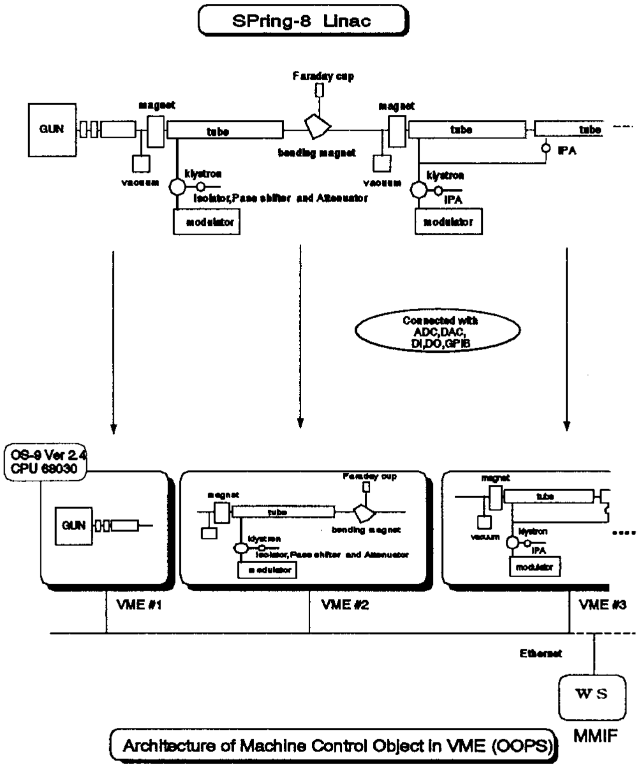

Figure 1. Layout of Object

B. Design of Super Class "MACHINE"

Assuming that Linac is constructed by similar simple components, we design the Super Class "MACHINE" that is core of Object. It has common characters of all devices.

Fig 2 shows the Super Class "MACHINE" and the Sub Classes of it. Main attributes of "MACHINE" are the "parameter and status" and the "Behavior". The "parameter and status" means common elements of Object, and the "Behavior" means the transitional status of Super Class (all Objects). Further, "Behavior" correspond to part of SPring-8 Linac machine Control Commands(SCC), the inter-Object message for SPring-8 Linac machine Control system, so that all Objects are became the "Behavior" status by SCC. On OS-9, Object Oriented Languages(like C++) have not been commercially available yet. So we have to use traditional C-Language for our Objects.

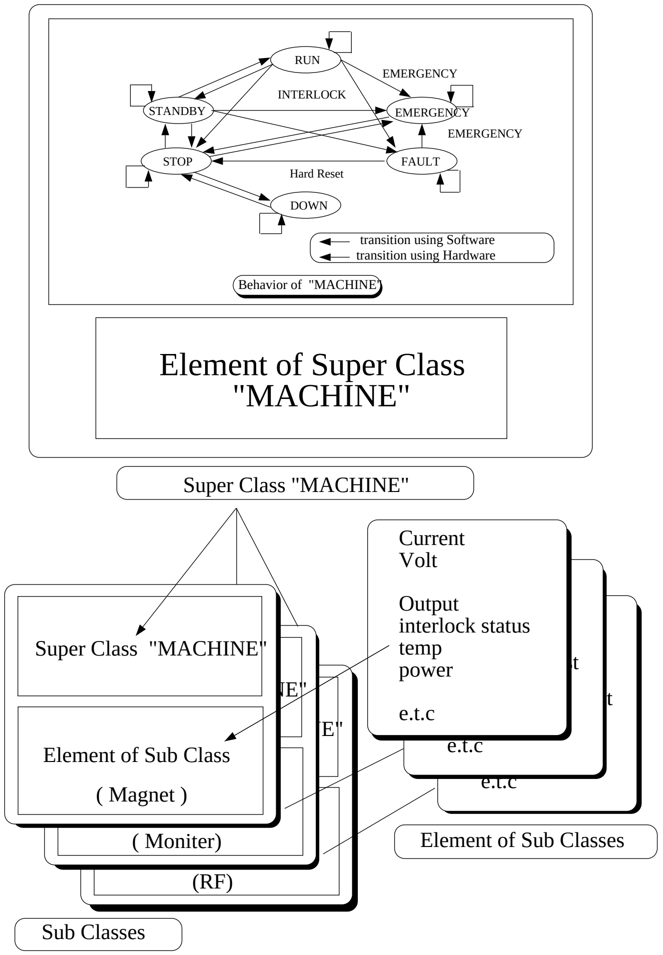

Figure 2. Design of Sub Class

C. SPring-8 Linac machine Control Datagram(SCD)

SCD means SPring-8 Linac machine Control Datagram [3]. The traffic of our network has single and rare fragmentation of data. Then we tried to find the suitable protocol for our requirement that it has to be standardized, be real-time feature and be matured as a reliable protocol.

But we could not find such a protocol, then we decided that TCP/IP is tentatively adapted to our protocol. Though, the connection-type service(typically TCP) should get down the efficiency of communication and take much CPU power for open/close connections. We want to reduce the over head of the communication service.

Other hands, the connectionless type service(UDP/IP) has smaller over head, and it is faster than connection-type service in our traffic condition. But it will happen that transport error, loss of packets and mistake, because of the ease of error check in UDP/IP. It is necessary to reinforce the error check. So it motivated us to develope the SCD protocol. The characteristics of SCD protocol are description follows.

- **Simple:** SCD is connectionless, open/close procedures are not needed, then the number of packets and the network traffic are reduced.
- **reliable:** The time-out sequence get higher reliability. The lower of protocol is UDP/IP which has enough reliability and is used all over the world.
- **flexible:** So MAR procedure is implemented, it is flexible to be adapted to the modification of the machine configuration.

SCD protocol is fifth layer as the session layer of OSI layer model(fig3). Data flows from upper layer to lower layer, and each layer build/divide the packet. In time domain, data is sent from head to tail.

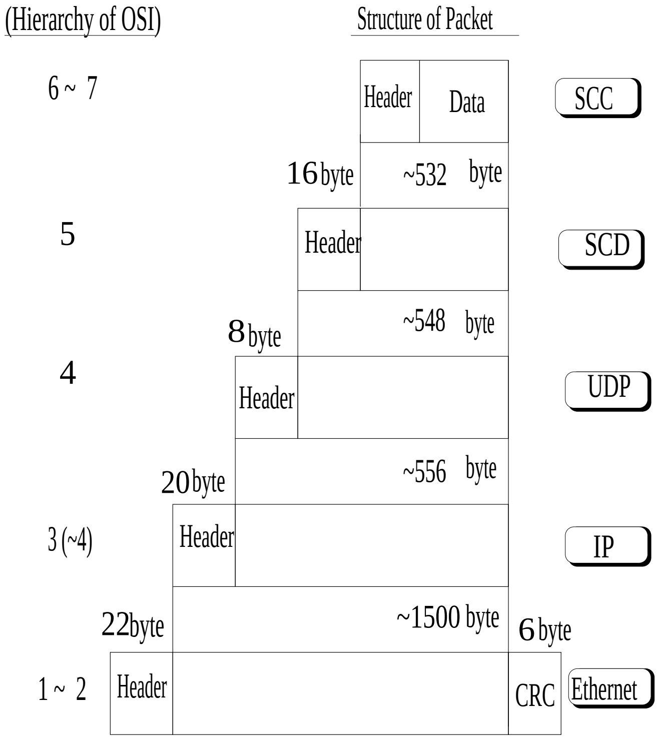

Figure 3. Layer of SCD

D. Communication Process

It is important for the machine control to pursue a data transfer speed. So, we designed the Communication Process following concepts on SCD.

- Using Object Oriented
(differentform Super Class ”MACHINE”)
- Self address learning type
- High flexibility

Fig 4 shows the Communication Process both WS side and VME side. This Process is implemented on HP-UX and OS-9. Between Communication Process and Objects, however, SCC is forwarded from Communication Process to Object by the Mailing system. When we want to send a parameter to Object, we can use API in the Communication Process. If we have never known the Object address, but Object name is known, Communication Process is able to search the address (MAR procedure) and to send SCC to the target Object. We never be worried about Object address. This looks like the Super Post Office.

This Process is also used Object Oriented concept and programming, so we will get a high maintainability.

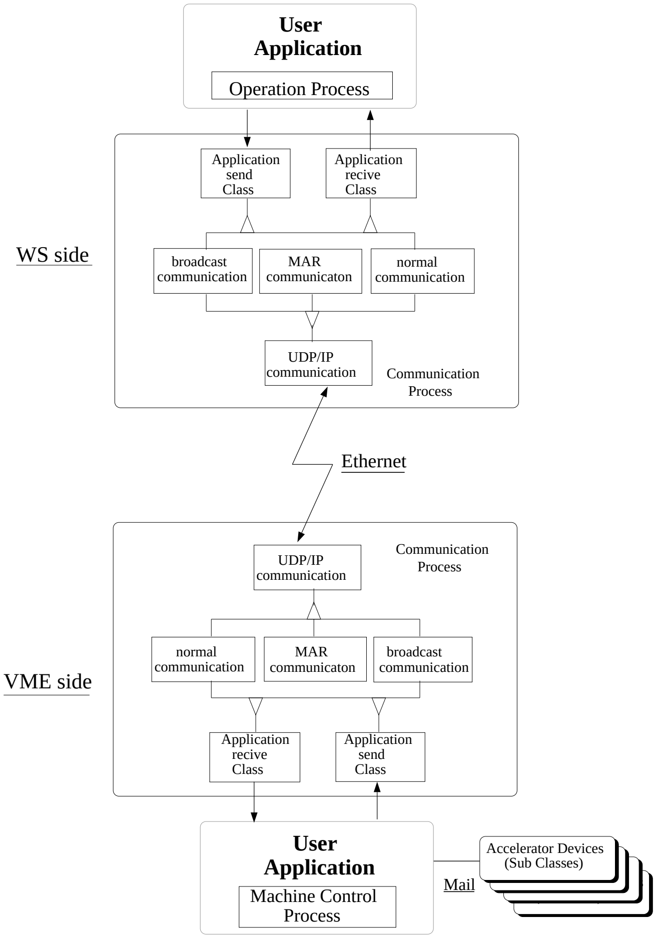

Figure 4. Layout of Communication Process

III. CONCLUSION

We designed the Super Class for SPring-8 Linac, so that we will be able to make the Sub Classes easily. However, as Object is written by traditional C-Language, it is felt slightly complex. Now, we are making all Objects on the VME system, and have to check the performance of our system. These Object are going to put together software and hardware on this year.

References

[1] H.Nishimura, ”Dynamic Accelerator Modeling Uses Objects in Eiffel” , COMPUTERS IN PHYSICS, Vol.6, No.5, 1992,p456.

[2] I.S.Ko et al., ”CONTROL OF PLS 2-GeV LINAC”, Presented at the 17th International Linac Conference, Tsukuba, Japan, August 21-26,1994

[3] H.Yoshikawa et al., ”Class structure of the Injector Linac control system of SPring-8”, Nucl. Instr. and Meth. in Phys. Res. A352 (1994) 216-217