

CURRENT STATUS OF THE CONTROL SYSTEM DEVELOPMENT AT RIKEN RI-BEAM FACTORY

Toshiya Tanabe[#], Toshikatsu Masuoka, Koichi Yoshida, Keiko Kumagai, Misaki Komiyama, Takashi Emoto and Masayuki Kase, RIKEN, Wako, Saitama, Japan

Abstract

Deployment of the Common Object Request Broker Architecture (CORBA) based control system for RIKEN's RI-Beam Factory Project [1] has started. Vendor independence, software sharing and interoperability are the main aims of the design. Abeans R2 [2] developed at Jozef Stefan Institute (JSI) of Slovenia have been connected to our CORBA servers running on VMEs via plug-ins. Oracle9i running on a Linux is employed to store the data, operation loggings and device configurations. XMLs are utilized to set the configuration of the each device from the DB to the equipment. The power supply (PS) controls are firstly deployed as well as RF monitor via Programmable Logic Controller (PLC). A generic CORBA-GPIB object and CORBA-EPICS gateway server [3] have also been developed.

was also regarded as a product of 1980's design without object oriented features such as the use of Java in the GUI in integrated fashion. Unlike other big labs, the number of our staffs is extremely limited despite of the large budget for the project, and the outsourcings of programming for the EPICS had been costly as very few institutions besides KEK used it. Therefore, it was decided that the use of distributed object in the heterogeneous environment in the modern computer industries could be used in the control system for future accelerators.

CORBA is the industry standard for the deployment of heterogeneous distributed objects and recent versions of Java language includes its core libraries by default. It has already been used in other large scientific facilities in the world such as ANKA [5] and NIF [6].

OBJECTIVES

Since our existing system uses the Experimental Physics and Industrial Control System (EPICS) [4], one might ask why a new control system should be developed from scratch. At the time the R&D work for this project was started, EPICS was totally dependent on a particular vendor (Windriver, Inc.) and most of I/O boards with a device support are foreign products, which tend to cost twice in Japan compared in the country of origin. EPICS

SYSTEM ARCHITECTURE

Overall system architecture including the some future extensions is shown in the Fig. 1. The Java graphical user interface (GUI) programs work as a CORBA client only except for the alarm call-back function. The objects running in the application server (AS) can be both as a client and a server. Those running in the VME are mostly regarded as CORBA servers except when it gets the parameter data from the database. The relational database

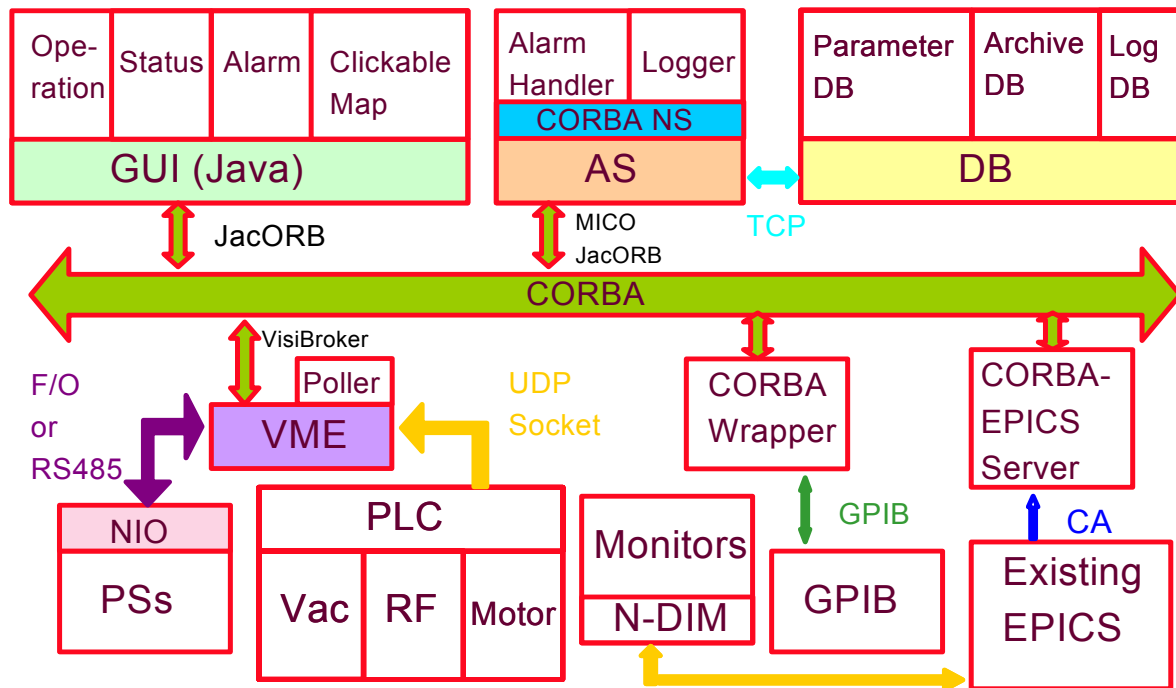


Fig. 1: Overall System Architecture

[#]tanabe@riken.jp

management system (RDBMS) functions as a back-end of the AS and cannot be accessed directly from the other objects. GPIB wrapper and CORBA-EPICS server are still in the prototype stage and they will be field-tested in the coming months. Table 1 shows the list of hardware and software used in the system.

Table 1: List of the used hardware and software (those in bold characters are commercial software).

Equipment	Items	Used Software
Client PC (Java enabled)	OS	Win2k/OS-X/Linux
	Language	Java1.4
	CORBA	JacORB 1.4.1
Database [Dell PowerEdge 2650]	OS	Miracle Linux 2.1
	RDB Software	Oracle 9i (9.0.1)
Controller (VME64x) [Motorola MVME2400]	OS	vxWorks5.4
	Language	C++ / C
	CORBA	VisiBroker-RT 4.1c
	XML Parser	libXML 2.5.2(modified)
Application Server [Dell PowerEdge 4600]	OS	RH Linux AS 2.1
	Language	GNU C++ / GNU C / Java1.4/ JDBC1.2
	Web server	Apache/Tomcat
	DNS	BIND
	NFS	Samba/Netatalk
	DHCP	dhcpcd
	CORBA	MICO 2.3.9/ JacORB 1.4.1
Others	GNU Make	

VME

Without the second phase project having multiple synchrotrons, the use of VME and real-time operating system (RTOS) could have been avoided since controls for cyclotrons did not require particularly fast response among separated components. However, our main interface boards for PSs (NIO system) are based in VME and the driver software for VxWorks [7] was already available then. Therefore, the original design with VME&RTOS was maintained.

Motorola's VME-CPU board (MVME2400) with 512MB RAM is chosen for the field controller. The model with 256MB or 128MB RAM is no longer in production and our board support package (BSP) for Tornado 2.0 can support only up to 256MB of RAM. VME64x crate (HS-SY013) has 15 slots and hot-swappable redundant power-supplies for easy maintenance. It also has alarm function for errors, and 5V35A, +12V4A, -12V1A; 3.3V25A of DC output can be used.

All the equipment-CORBA-objects are running in the VME including a poller program. There is only one type of program is running in all the VME and difference in terms of configurations and type of objects to be activated are determined from the DB data. Since VisiBroker 4.1c for Tornado does not support "wstring" type, the Japanese

characters cannot be handled in the same way as Latin alphabets. Therefore, Unicode escape sequence was used to send Japanese via CORBA when necessary. The features of RT-CORBA such as "thread pool" and "server declared CORBA priority" are employed to control the various task priorities. PropertyResourceBundle class in Java is used to switch the languages.

NIO I/F Card

NIO system comprises of the NIO-C master board in the VME crate and NIO-S slave boards which locate in the PSs.[8] They can be connected with each other by serial communication of 2Mbps either with SI fibre optical (F/O) cable in a star topology using branch boards or RS485 balanced serial cable in series. NIO-S has its CPU with a firmware running so it can be operated independently from the master, if necessary. This concept has been carried over from the CAMAC based I/O system that our existing facility utilizes. The firmware of NIO-S can also be downloaded from the host computer in order to ensure the easy maintenance and the flexibility of modification.

Application server and RDBMS

Linux is used for the AS and the RDBMS. For the AS, Redhat Linux Advance Server 2.1 was chosen for its support availability. Oracle 9i for Linux is installed in the Miracle Linux which is produced by a subsidiary of the Oracle, Inc. and the integrated support for both DB and OS are available. One big problem for Oracle for Linux is that the security patches cannot be obtained without rather expensive support contract which nullifies the cost advantage of Linux system.

SOFTWARE FEATURES

PS controls

The PS CORBA objects on any VME can be accessed and controlled by a CORBA client which in our case is PPanel program made by the JSI and modified by the Cosylab. There are two types of screens; one is for monitoring, and the other for set-up and operating multiple PSs at once. Individual PPanel can be opened only through either of the above-mentioned screens. Each VME has many PS objects and it is inefficient to inquire the interoperable naming service (INS) for the interoperable object reference (IOR) of those every time PPanel is activated. There top screens for the PS control collect all the IORs at the beginning so that each individual PPanel can immediately know where the object is. The setup information for each PS is sent to the PS object in terms of XML generated from the data from the RDBMS through JDBC. Figure 2 shows the PS monitoring screen running on Mac OS-X (10.2.6). The blue arrows (triangles) indicate the value intended to set in the PS and green ones show the read-back values. If the discrepancy between the both exceeds a certain limit, the box turns to red. By clicking a check box,

corresponding PPanel will emerge. UML Sequence diagram of PS Summary operation is given in Fig. 3.



Fig.2: PS Summary Screen.

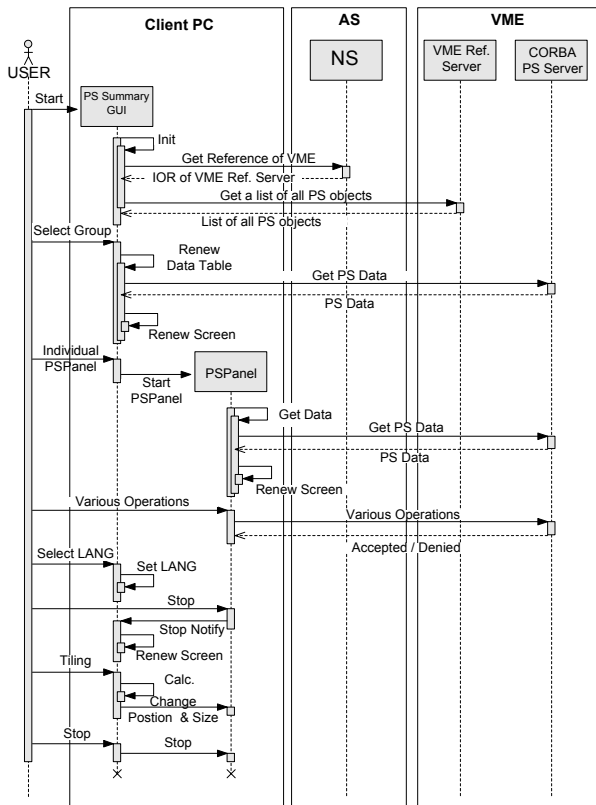


Fig. 3: Sequence Diagram for PS Summary operation.

Alarm Handler and Logging Function

The program for the polling runs in the VME and it sends the data to the logging object in the AS. This object writes the contents both to the shared memory and to the DB. Alarm Handler (AH) object will read the contents of the shared memory and report errors when the conditions are met. Figure 4 shows the sequence diagram of the alarm-operations.

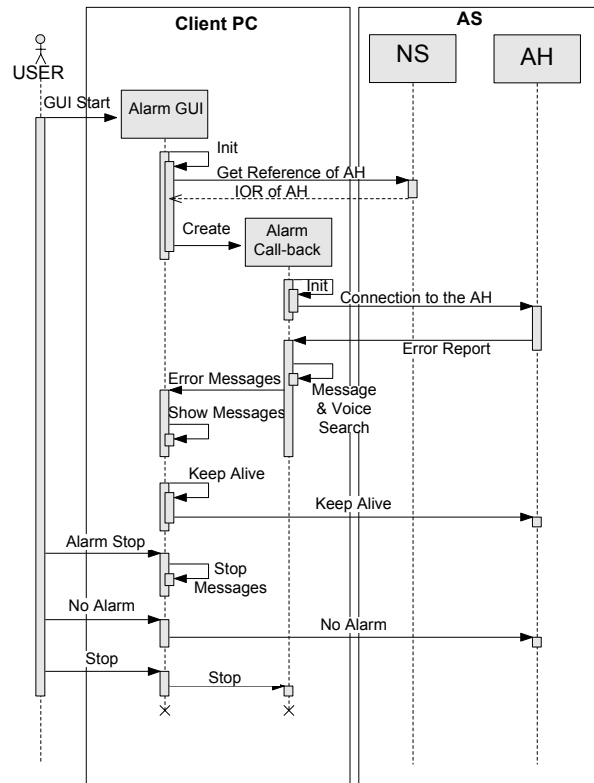


Fig. 4: Sequence diagram for the alarm-operation.

RF Monitor Panel

RF monitor screen is also made of Abeans. The GUI program communicates with PLC wrapper object [9] which read/write the data in the memory area of the PLC (OMRON SYSMAC CVM1/CV) through UDP with FINS commands that are provided by the manufacturer.

SUMMARY

World's first "heterogeneous" RT-CORBA based accelerator control system has been implemented for the new cyclotrons in the RIBF project at RIKEN. The RDBMS is closely tied to the system in order to manage the setup parameters of equipment, various data and operation logging. The integration with the existing system is carried out using CORBA-EPICS gateway server.

REFERENCES

- [1] <http://ribfweb1.riken.go.jp/>
- [2] <http://www.cosylab.com/>
- [3] T. Tanabe and T. Masuoka, RIKEN Accel. Prog. Rep. 34 (2001) 320
- [4] <http://www.aps.anl.gov/epics/>
- [5] <http://hikwww1.fzk.de/anka/>
- [6] <http://www.llnl.gov/nif/nif.html>
- [7] <http://www.windriver.com/>
- [8] <http://www.ndssf.co.jp/>
- [9] <http://accelconf.web.cern.ch/AccelConf/ica01/papers/THAT003.pdf>