

## Distributed-Object Based Design of RIBF Control System Using Java /Corba

Toshiya TANABE, Toshikatsu MASUOKA, Masayuki KASE, Motonobu TAKANO and Takeshi KATAYAMA,  
RIKEN (The Institute of Physical and Chemical Research)  
2-1, Hirosawa, Wako-shi, Saitama 351-01, Japan

### Abstract

Various R&D activities concerning Common Object Request Broker Architecture (CORBA) based control system have been undertaken at RIKEN for the preparation of Radio Isotope Beam Factory (RIBF) control system [1]. One of our goals is to implement all the necessary functions for the RIBF controls utilizing interoperable CORBA objects in combination with Java so that the system will not depend on a particular Operating System (OS) or vendor. The detailed results of our previous R&Ds have been documented elsewhere [2]. We describe the new test results as well as the designs of the subsystems on the RIBF control system. The possibility of software sharing has also been investigated.

### 1 INTRODUCTION

RIKEN-RIBF project is an expansion of the existing RIKEN accelerator research facility (RARF) [3]. The RARF control system is centralised on a small main-frame (MF) computer (Melcom M60) which controls all the aspects of machine controls through CAMAC and proprietary field-bus (Mitsubishi CIM-DIM), MMI and DB. The upper layers above CAMAC loops are currently being replaced by Experimental Physics and Industrial Control System (EPICS) [4]. However, for the new facilities such as an intermediate ring cyclotron (IRC), a superconducting ring cyclotron (SRC) and two RI beam separators (Big RIPS) and downstream, it is not necessarily good idea to adapt the same control system as the existing facility due to its obsolescence. Therefore, we plan to adapt an entirely new design for the new facility.

The followings are basic concepts of our new control system:

- Vendor independent as much as possible
- Use of object-oriented (OO) technology
- Use of commercial off-the-shelf (COTS) products as much as possible
- Portable source codes
- Software sharing with other labs
- Seamless integration with legacy systems
- Reliable hardware regardless of cost disadvantage.

As for the hardware in the field controller level, we are comfortable with VMEs running VxWorks despite of its high price. Power supply controls for the magnetic measurement of new cyclotron's sector magnets have already been undertaken using CORBA VME server objects. The GUIs can be made in JavaBeans components and so can possibly be controls logics. Another reason

which encourages us to use CORBA is the wide use of Programmable Logic Controls (PLCs) in Japanese manufacturers. Each company has its favourite type of PLCs that are often used to construct a closed system such as RF cavity controls and vacuum system. PCs are used to control these local systems. CORBA makes it easier to integrate these separate systems into the global controls. For MMI, GUIs created by JavaBeans [5] appear to be the most promising.

### 2 PREVIOUS R&D EFFORTS

CORBA is the standard distributed object architecture for an open software bus, which allows object components created by different OSs and languages to interoperate. It would make integration with legacy controls easier, which is suited for RIBF project as it is an expansion of existing facilities. Java language [5] is known to be very well thought-out objected oriented language and its most useful feature is platform independence using interpreter and Java Virtual Machine (VM) which executes the bytecode. In recent years, Java/CORBA applications for business environment have been flourishing. There are a few examples in the world, in which accelerator facilities and other large-scale experimental facilities utilize CORBA for its controls [6].

The summary of our previous R&Ds is described below:

- a) Replaced existing codes using the socket programming with those for CORBA. All the I/O boards worked without any problem with reasonable response time of the order of 1msec.
- b) Confirmed interoperability through IIOP among different ORBs and the results were satisfactory.
- c) Estimate the software overhead of various ORBs. It turned out the latency was on the order of a few to several hundreds  $\mu$ s.
- d) Managed to exchange the server object from a VME running VxWorks to a compact PCI running pSOSystem without changing anything on the client side.
- e) Preliminary software sharing using Abeans originally developed for ANKA light source.

Further studies on CORBA based accelerator control systems have been conducted since then.

### 3 SOFTWARE SHARING WITH JSI

The possibility of software sharing for the accelerator controls has been discussed for a long time. However

very few examples with the exception of EPICS and cdev [7] are found in reality. One of the reasons for the failure is that there is no standardization of data structure and communication protocol. Recent advance in XML for standard data format and CORBA/IIOP for standard inter-object communication may result in more sharing in the future. As for the GUIs, JavaBeans is one of the most promising candidates for sharable software. ANKA control system utilize this technology for their GUI and control logics as well as CORBA for communication. People at Jozef Stefan Institute (JSI) where this set of software was developed kindly allow us to examine the possibility of sharing codes in our system.

Even though JSI people provide Accelerator Corba Interface (ACI) for the connection to their Accelerator Beans (abeans) which represent control logics, it is not necessarily easy to adapt ACI for our local system. That's partly because some of our hardware have extra functionalities which are not covered by ACI. Hence writing CORBA wrapper program with our Interface Definition Language (IDL) was chosen for convenience. Establishing agreeable IDLs for accelerator controls seems to be the most difficult task for the software sharing using CORBA.

#### 4 POWER-SUPPLY CONTROL FOR IRC MAGNETIC MEASUREMENT

The magnetic field measurement for sector magnets of IRC has been conducted for some time. One large power supply (PS) is used for the main magnet per sector and 30 middle sized PSs are used for the trim coils. Our PS object model is used to control all the PSs through VME master-slave type field bus (Hitachi-Zosen NIO-C & NIO-S [8]). The master board and the slave board for the main magnet PS, and that for the first trim coil PS are connected by optical fibers. The rest of the trim coil PSs are connected by IEEE485 in series. GUI (shown in Fig. 1) is modified Abeans, for which 6 PSPanels can be run on one VM.

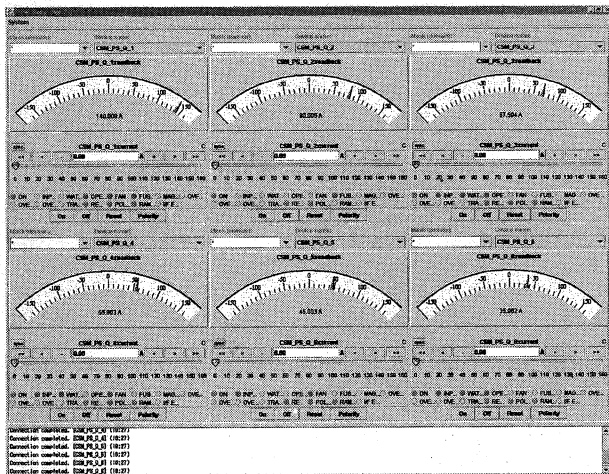


Fig. 1. IRC-PSPanel GUI

#### 5 PLC WRAPPER OBJECT

PLCs are Japanese manufacturer's favourite control tools for factory automation. There are many different kinds and they are practically incompatible among each other. Therefore one needs a mechanism which can conceal these differences in upper level software. We have developed CORBA PLC wrapper programs for the PLC by OMRON [SYSMAC CVM1/CV] and they are used for RF control, vacuum monitoring, and motor-drive controls. PLC CORBA object in VME has an interface with Abeans, and read/write the data in the memory area in the PLC through UDP using FINS command which is provided by the manufacturer.

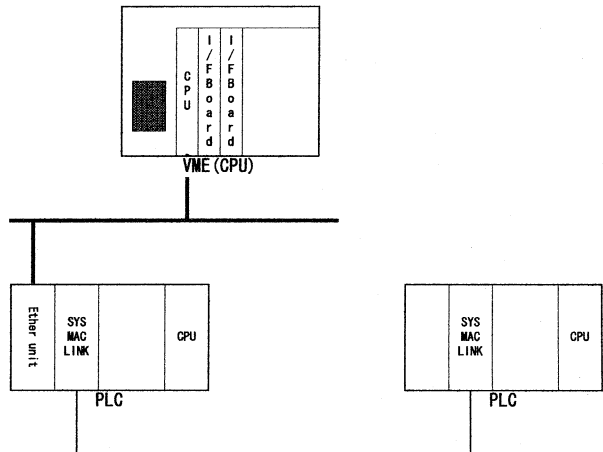


Fig. 2. PLC Wrapper system configuration.

#### 6 DBMS WITH XML

The extensible Markup Language (XML) has grown to be the standard method to put structured data in a text file. In the accelerator controls, various parameter files, log files and archiver files can be standardised in the form of XML when they are used by different programs. A standard relational database management system (RDBMS) could not only store the flat data files but also XML data files with appropriate software tools. We have started investigation using Miracle Linux with Oracle 8i in a PC as a DB and free XML parser in VxWorks.

#### 7 CORBA-EPICS SERVER

RARF controls are currently being upgraded with EPICS so a mechanism must be developed to allow for the old and the new controls to communicate with each other. CORBA is often used to integrate legacy systems with various kinds of wrapper software. However, in our case, it is easier to set up a gateway computer which acts as a translator between the two systems. The CORBA-EPICS (CE) server scheme fulfils this need.

The CORBA server used for the test is the one used for the main-coil power supply of an intermediate-stage ring cyclotron (IRC). The version of EPICS installed in a Sun Workstation (Ultra10 Model 360, Solaris 2.6) is

R3.13.2 and a portable channel access (CA) server program runs in it. The ORB is VisiBroker for C++ ver. 3.3. Sun Workshop3 C++ 4.2 was used for the compiler. Since CORBA and EPICS environments are in different network segments, some routing operation is required. Figure 2 shows the system configuration for this server.

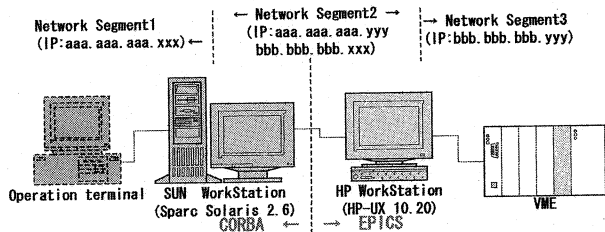


Fig. 3. CORBA-EPICS server system configuration.

The operation from EPICS to CORBA is initiated by a CA client. It specifies the records to process in the CE server and a modified portable CA server program passes data to the CORBA client program in it. This CORBA client connects CORBA servers in the CORBA domain. A sample program (excas) has been modified to accommodate the following functions in the CA server:

From CORBA to EPICS, the CORBA client has to know the records to be accessed in the CA server in the EPICS domain. After the CORBA server program receives the call from the CORBA client, it calls a CA client in the CE server to connect to CA servers in the EPICS domain. Schematic diagrams of both operations are given in Figs. 4 (a) and (b).

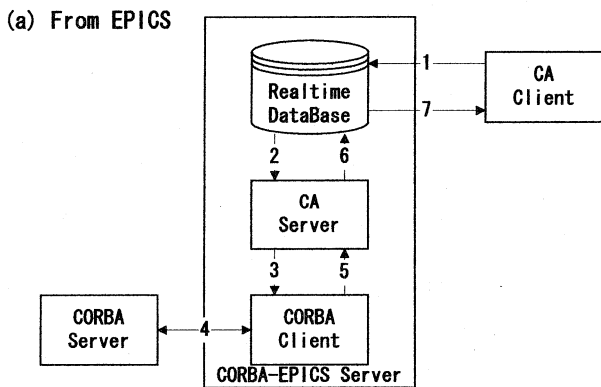


Fig. 4 (a): Schematic diagram of EPICS to CORBA operation. 1. The client accesses a record (put/get) 2. The record accessed is referred to by the CA server program (put/get). 3. The data set in the record are delivered to the CORBA client program (put/get). 4. Received data are delivered to the CORBA server (put/get). The data from the CORBA server returns to the CORBA client (get). 5. & 6. A reply from the CORBA server is reflected in the record (get). 7. The CA client gets a result from the record (get).

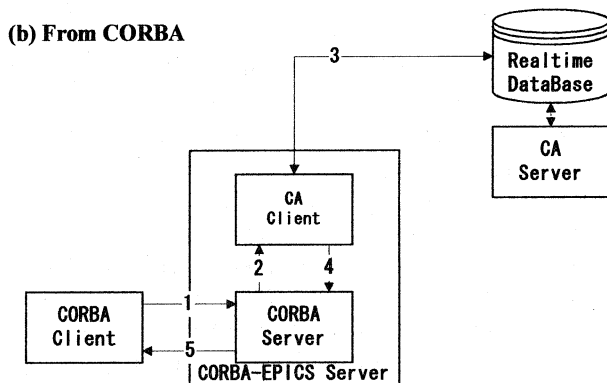


Fig. 4 (b): Schematic diagram of CORBA to EPICS operation. 1. The record to be accessed is specified in the CORBA server (put/get). 2. Received data are delivered to the CA client (put/get). 3. Data are put to or get from the specified record (put/get). 4. Acquired data are delivered to the CORBA server (get). 5. The data are delivered to the CORBA client via CORBA (get).

There are a few aspects to be modified if the CE server is used in real controls. First, data consistencies have to be monitored and maintained in some ways if "put" operations from both sides are permitted. Second, in this test, it was assumed that one record contained one piece of data. It is necessary to modify the CORBA part of programs if multiple data such as arrays are used for the record.

### 8 RT-CORBA TESTS

Even though the OS used in the controller is a real-time OS, the CORBA used so far does not have any real-time feature such scheduling mechanisms. Recently the Real-time CORBA specification has been completed by OMG [9], and it supports the real-time features. We have started testing an ORBexpress [10] which is the first implementation of RT-CORBA specs. The results of the test will be reported elsewhere.

### REFERENCES

- [1] <http://ribfweb1.riken.go.jp/cont/home.html>
- [2] <http://accelconf.web.cern.ch/accelconf/e00/PAPERS/TUP4B15.pdf>
- [3] <http://www.rarf.riken.go.jp/rarf/index.html>
- [4] <http://epics.aps.anl.gov/asd/controls/epics/EpicsDocumentation/WWWPages/EpicsFrames.html>
- [5] <http://www.javasoft.com/>
- [6] <http://kgb.ijs.si/KGB/>
- [7] <http://www.jlab.org/cdev/>
- [8] <http://elc.asaka.or.jp/electro/index.html>
- [9] <http://www.omg.org/>
- [10] <http://www.ois.com/>