PRELIMINARY DESIGN OF RIBF CONTROL SYSTEM USING CORBA AND RTOS

T. Tanabe, T. Masuoka and M. Kase, RIKEN, Saitama, Japan

Abstract

Various tests concerning CORBA (Common Object Request Broker Architecture) based control system have been conducted at RIKEN for the preparation of RIKEN-RIBF (Radio Isotope Beam Factory) [1] controls. One of our goals is to implement all the necessary functions for the RIBF controls utilising interoperable CORBA services so that the system will not depend on a particular OS. Server object exchange using VxWorks [2] running on VME and pSOSystem [2] on Compact PCI has been successfully demonstrated without any change in the client object. Each controller has an DI/DO board as a real I/F, and VisiBroker [3] is used as an ORB. We describe the new test results as well as the preliminary design of the RIBF control system. The possibility of software sharing is also investigated.

1 INTRODUCTION

In our previous paper [4], we described the following tests, and the results are summarised 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 µs.

Further studies on CORBA based accelerator control systems have been conducted since then. We have demonstrated the feasibility of exchanging server objects without changing the upper layers of the control system. A GUI made in Visual C++ is also replaced by the one in JavaBeans made for ANKA controls [5]. In the realm of interoperability testing, ORBexpress [6] has been added and two-way operation time with varying data size has been compared with VisiBroker.

2 SERVER OBJECT EXCHANGE

One of the most touted features of the object-oriented (OO) programming is high changeability of codes thanks to "component" technology. CORBA objects are no exception. We have managed to exchange the server object from a VME (Motorola MVME 2600) running VxWorks to a compact PCI (AVAL DATA ACP-100) [7] running pSOSystem without changing anything on the client side. DIO board by the same manufacturer [8] as the corresponding board is installed for comparison. Figure 1 shows the system configuration of our tests.

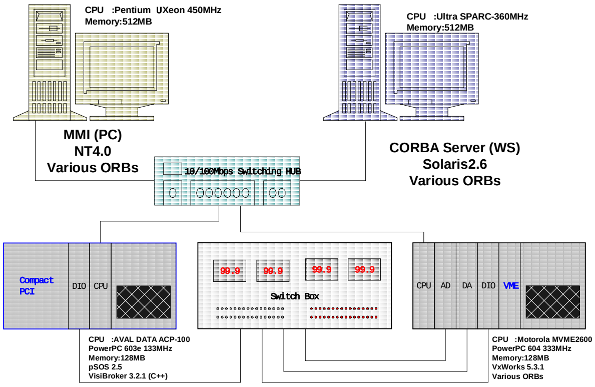

Figure 1: Test system configuration

The two-way response time (average for 100 calls *10) for both cases with corresponding I/O board is shown in Table 1. This case the ORB is VisiBroker, the client written in C++ on Windows NT4.0, and the smart agent and Static Invocation Interface (SII) are used. The measured time for each server object is shown in Table 1. The difference in performance is largely due to that of the CPU (Power PC 604@333MHz v.s. 603e@133MHz) on the board.

Table1: Comparison between two server objects.

	DI		O	
Time in msec	Ave.	Std. Dev.	Ave.	Std. Dev.
VxWorks ver. 5.3.1 (333MHz)	1.25	0.0	1.29	0.08640
pSOSystem ver. 2.5 (133 MHz)	5.176	0.05060	5.455	0.04743

3 INTEROPERABILITY TESTS

In our previous interoperability tests, the size of the data sent was fixed to 32bytes. This time the data size was varied from 32bytes to 1024bytes to measure the corresponding two-way call response time. The ratio of the CORBA operation to socket communication time for various combinations is shown in Fig. 2. One peculiar finding is that when a PC running Windows NT4.0 is used as a server, the ratio is almost constant on the contrary to our expectation.

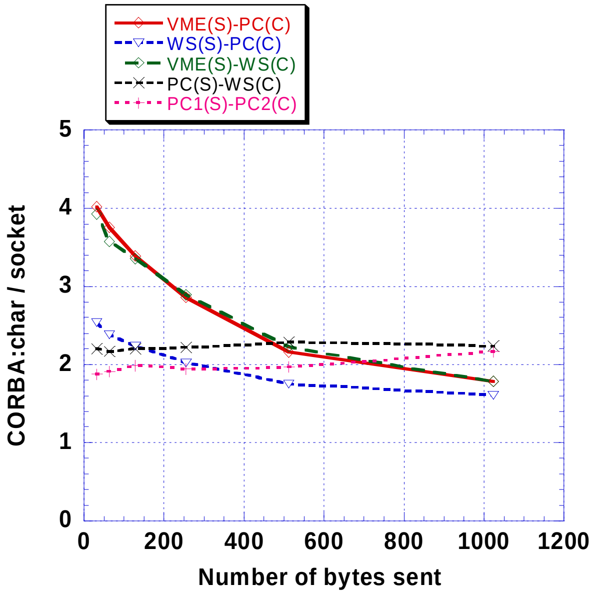

Figure 2: The ratio CORBA::char / socket for two-way calls with varying data size. What is represented in each line is shown in the legend on the upper left area.

ORBexpress for VxWorks has been added for the evaluation. This ORB has been chosen because of compliance of Real-time CORBA specification [9]. Unfortunately, we could take only a few data due to the memory leak problem of the ORB associated with the use of Tornado 1. The comparison for bounded character data type is given in Fig. 3.

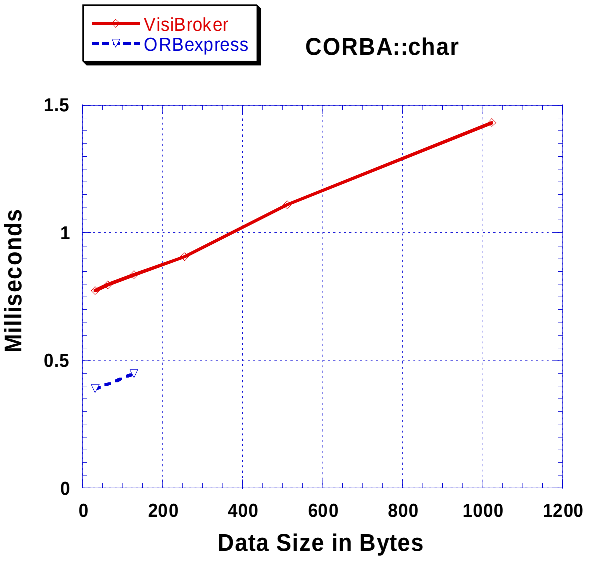

Figure 3: Response time with varying data size. The solid line represents VisiBroker and the dashed line ORBexpress.

3 RIBF CONTROL SYSTEM DESIGN POLICY

RIKEN-RIBF project is an expansion of the existing RIKEN accelerator research facility (RARF). 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 is currently being replaced by EPICS. 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 obsoleteness. 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.

One of the strong candidates satisfying above conditions is a use of CORBA middleware for the communication layer and make the system collection of distributed CORBA objects.

In terms of 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. 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 appear to be the most promising. The most difficult part is how to implement all the necessary control logic. Whether to use various CORBA services to construct the controls functions is still an open question.

Use of Object-oriented data base management system (OODBMS) was sought after for a while. However, it is not yet conclusive that OODBMS has clear advantages over relational DBMS in terms of construction cost, performance and reliability.

4 SOFTWARE SHARING WITH JSI

The possibility of software sharing for the accelerator controls has been talked about for a long time. However very few examples with the exception of EPICS [10] and cdev [11] are found in reality. One of the reasons for the failure is that there is no standardisation 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 [12] is one of the most promising candidate for sharable software. ANKA control system utilise this technology for their GUI and control logic 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.

We have managed to use a part their GUI beans to replace our GUI made by VC++(Fig.4.)

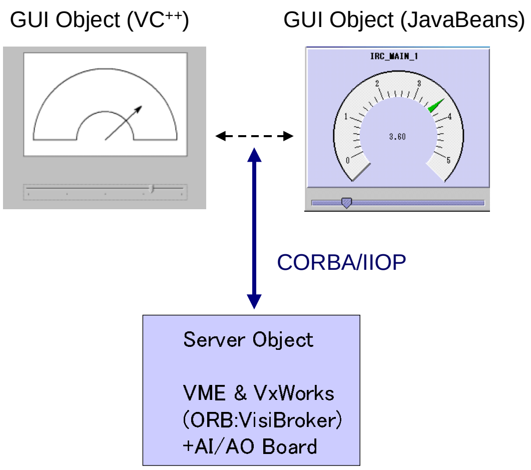

Fig.4: GUI object exchange

Even though JSI people provide Accelerator Corba Interface (ACI) for the connection to their Accelerator Beans (abeans) which represent control logics, it it not necessarily easy to adapt ACI for our local system. That's partly because some of our hardware have extra functionality which are not covered by ACI. In this test, our aim is to establish the connection between our server object and their client programs. Hence writing CORBA wrapper program with our IDLs was chosen for convenience. This approach is obviously not suitable for the purpose of software sharing. Establishing agreeable IDLs for accelerator controls seems to be the most difficult task for the software sharing.

5 CONCLUSIONS

It has been shown that replacing socket communication with CORBA layer significantly improves manageability of accelerator controls without sacrificing performance. It has been shown that a GUI and a VME/CPCI object can be separately treated. CORBA can also be useful for integrating legacy controls and stand-alone components. Interoperability between ORBacus and VisiBroker is found to be satisfactory and response times do not vary much with different combinations. Scalability for PC as a server appears to be different from others, but no reason is known , yet.

CORBA based control system is found to be promising for RIBF controls.

REFERENCES

[1] Y. Yano and et al., Proc. of PAC 97, 930.
[2] http://www.windriver.com/
[3] http://www.inprise.com/
[4] T. Tanabe, et al., Proc. of Icalepcs '99, (1999) 558
[5] http://kgb.ijs.si/KGB/
[6] http://www.ois.com/
[7] http://www.avaldata.co.jp/jpn/index.html
[8] http://elc.aska.or.jp/electro/index.html
[9] http://www.omg.org/
[10] http://epics.aps.anl.gov/asd/controls/epics/EpicsDocumentation/WWWPages/EpicsFrames.html
[11] http://www.jlab.org/cdev/
[12] http://www.javasoft.com/