A study of network-based data acquisition system

Yoshiji Yasu [a], Hirofumi Fujii [a], Eiji Inoue [a], Hideyo Kodama [a], Yasunobu Sakamoto [b]

[a] *Department of Physics, National Laboratory for High Energy Physics ,Oho 1-1, Tsukuba, Ibaraki 305, Japan*

[b] *Department of Physics, Yamagata University, kojirakawa 1-4-21, Yamagata, Yamagata 990, Japan*

Network technologies such as Java and Microsoft ActiveX may become powerful tools to construct the distributed and scalable data acquisition system(DAQ). We have developed Java interface to CAMAC and investigated the application of HORB(distributed Java) to the DAQ. We have also evaluated Microsoft OLE-automation technique(OCX) with combining the objects to Excel as the analysis process over network. The possibilities and the scalability of the network distributed DAQ system with test-bed have been also evaluated.

Key words: Distributed DAQ; Java; HORB; ActiveX; OLE; Excel;

1 Motivation and Overview

Distributed database, Graphical User Interface(GUI), platform independence are important elements for next generation of DAQ. The DAQ requires multiple computers and then the management of the database becomes complicated. WWW browser such as Netscape makes the GUI design easy. The software of the DAQ should be platform independent as possible because of the portability and the scalability. Recent distributed object technology may solve the problems? Needless to say, object oriented methodology has good characteristics such as data encapsulation, polymorphism and inheritance to build the DAQ. When we think the distributed database for the DAQ, the problem of where to put the data occurs. The answer is that the data should be placed at where the communication cost is lowest, but the problem is whether it is easy to describe the distributed database or not. The performance is also the problem. We thought we had to choose the good tool. Therefore, We have

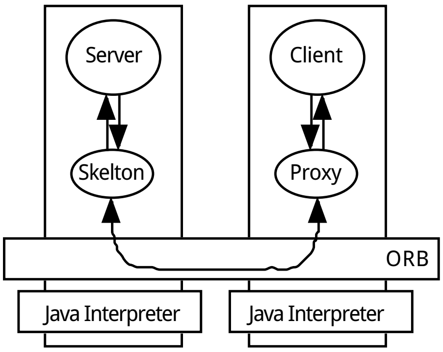

Fig. 1 HORB Architecture

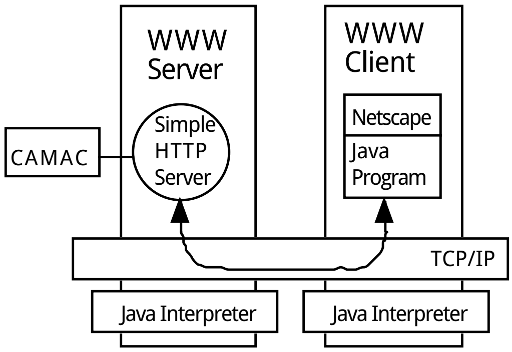

Fig. 2 Scaler monitor without HORB

investigated the feasibility of the distributed object technology on the DAQ using Java, HORB[1] and OLE[2].

Java and HORB provide powerful network computing. HORB is Java plus ORB (Object Request Broker) and parallel & distributed & persistent object oriented language. HORB architecture is shown in Fig.1. There is no IDL(Interface Definition Language of CORBA[3]). Therefore, we do not need to learn the language. The name space of HORB is similar to that of WWW. It is very easy to make remote method call(invocation) in synchronous or asynchronous way. Daemon objects can be run easily. WWW browser such as Netscape is common user interface for many DAQ users. Therefore, the browser is convenient for the operation and the monitor of the DAQ. When we make client applet with WWW browser, HORB has powerful features. First, HORB client applet can access native methods written in C language, but the Java does not. Native methods which were already made will be used in the DAQ and we may make the methods for the performance issue. The security of Java is excellent while HORB has the distributed Access Control List. HORB also provides 'invitation' mechanism that enables a server to call methods of a client, but Java does not. Because some objects in the DAQ system should be saved and/or retrieved, the persistent objects are necessary. HORB provides distributed persistent objects by simple remote object storage.

Microsoft ActiveX also provided similar functionality. OLE is special protocol of Microsoft while Java is open and can run on many platforms. ActiveX is the umbrella name which consists of Active Controls and Active Documents. OLE is an implementation of the Component Object Model(COM). OCX(OLE Custom Controls) can run in COM/ActiveX architecture. Powerful analysis/display program, e.g., Excel can be combined to the OCX easily. This feature is quite attractive for end-users.

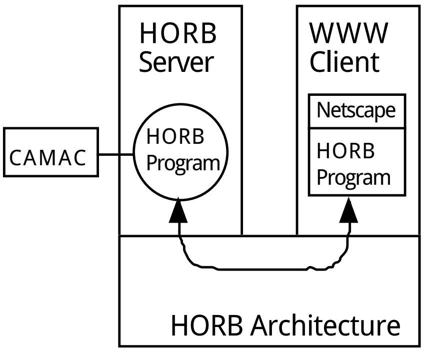

Fig. 3 Scaler monitor with HORB

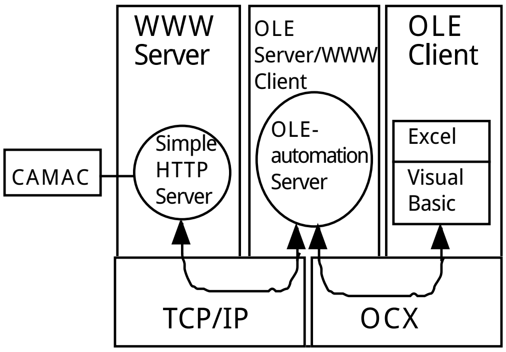

Fig. 4 Excel with OCX

2 Development and Investigation

We provided PC/Linux with CAMAC system and some workstations. The PC is Gateway2000(P5-120)/Linux2.0.0 with gcc2.7.2. A CAMAC Scaler was in the CAMAC Crate with CC7x00 Crate Controller[4]. KEK standard CAMAC library for the Crate Controller was already developed.

We have developed Java interface to the CAMAC library[5]. The user interfaces have been slightly modified to make it fit in object oriented way. The version of the Java is 1.0.2.

We have investigated “remote method call” for distributed DAQ by using HORB1.3.b1 and Netscape3.0. There are two examples. One is that a server program using Java without HORB read data from CAMAC Scaler and then Netscape displayed the data by being browsed. Another is that HORB applet read data from CAMAC Scaler on remote computer and then Netscape displayed the data. The Fig.2 shows Scaler monitor without HORB and Fig.3 shows Scaler monitor with HORB.

We also have investigated the possibility to use the Excel for the remote data analysis. OLE supports the automation which supplies public data and public method accessible by the client. We have made an example server using the automation. The object was remotely called on another computer which Microsoft Excel run. The client program is written in Visual Basic in the Excel. The server also read data from CAMAC Scaler via a simple WWW server and then the Excel displayed the data. Fig.4 shows the Excel with the OCX server.

We have made a network-based small test system which consists of two data sources and one destination. The destination distributes the data to analyzers, which consist of Excel and Netcape with HORB program. A server program written in Java language as the source takes Scaler data from a CAMAC Crate. The program sends the data to the destination via a special TCP/IP port. The program on the destination is also written in Java. Another server program written in Java as another source generates data on a workstation and sends the data to the destination in same way. Then, the program on the destination sends the data to two analyzers alternatively. One of the analyzers is Excel on one computer and another one is Netscape with HORB applet. The data with HTML header were sent to two analyzers in same way.

3 Conclusion

The distributed database has been written easily by using the daemon objects of HORB. The "synchronized" statement of Java was powerful for exclusive process. HORB applet programming was good. The "remote method call" of HORB made the remote program easy. Excel with OCX was useful. Java with HORB and Excel with OCX have made the DAQ design easy. However, the investigations still remains. From performance issue, the investigation of those tools is not enough. From real-time issue such as the schedulability, those tools on real-time OS including micro-kernel based OS is not evaluated yet.

References

[1] http://ring.etl.go.jp/openlab/horb/

[2] http://www.microsoft.com/oledev/

[3] http://www.corba.org/

[4] TOYO Corporation, User's Guide of CC7000 & CC7700 Crate Controller for CC/ISA & CC/PCI

[5] Yasunobu Sakamoto and Yoshiji Yasu, Beginner's Guide to CAMAC Data Acquisition using Java language(to be published).