STUDY OF SHARABLE APPLICATIONS USING JAVA AND CORBA

Shiro Kusano*, Norihiko Kamikubota and Kazuro Furukawa
High Energy Accelerator Research Organization (KEK)
1-1 Oho, Tsukuba, Ibaraki, 305-0801, Japan
Mitsubishi Electric System and Service Co. Ltd. *
Umezono 2-8-8, Tsukuba, Ibaraki, 305-0045, Japan

Abstract

Sharing of software with object-oriented technologies is a recent trend in accelerator control. Among the object-oriented technologies, Java and CORBA have been widely accepted in all industrial fields. We have studied use of both Java and CORBA in order to develop sharable applications among various accelerators. Sample applications have been made at the KEK electron/positron injector-linac. Evaluation of the application and discussion are given in this article.

1 INTRODUCTION

Recently, the object-oriented technologies for distributed computer systems,such as Java and CORBA, have been paid special attentions in the field of accelerator controls as well as in all industrial fields. Java is a platform-independent language with the capability to create an interactive GUI (Graphical User Interface) at a web-browser. CORBA (Common Object Request Broker Architecture) has become the standard communication protocol between distributed computers over networks. The use of CORBA enables smooth communication between different languages (C, C++, Java, etc.) and different operating systems.

The feasibility studies of Java and CORBA, which have been carried out at our group [1, 2], show that using both Java and CORBA is essential to ensure high source portability over different platforms. In addition, we succeeded to develop a sharable application between DESY and KEK with the CORBA toolkit [3] (see 2.2). This experience showed that the use of standard interfaces is important to develop sharable applications over different accelerators.

In this article, we report the recent progress of our studies to enable sharable applications using Java and CORBA.

2 PRESENT STATUS OF STUDIES

2.1 Overview

The KEK injector-linac and the KEKB-rings have separate control systems [5, 6] due to historical reasons. Thus, it is difficult to develop sharable applications between them. In order to develop sharable applications, we are considering the possibility to introduce the CORBA toolkit into the two control systems. The forthcoming scheme is shown in Fig. 2. We will construct vacuum servers for each control system, then develop sharable applications (clients). We hope the development will finish by the end of this fiscal year 1999.

The present status of related developments are summarized in Fig. 1. The usefulness of Java and CORBA was demonstrated with a real-time display of the klystron status (see 2.3 and [1]). It is important for us that the C-based libraries of the existing control system can be used. The upgrade of the CORBA toolkit with recent IIOP-based[1] ORBs[2] is under development (see 2.2). The revised toolkit will provide the standard communication interfaces in the future scheme (Fig. 2, Fig. 4). In addition, the study of interoperabilities between different ORBs is also in progress (see 2.3 and 2.4).

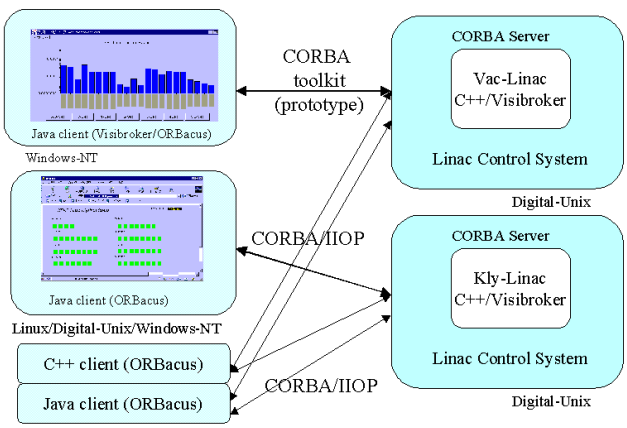

Figure 1: Current status of the studies.

2.2 The CORBA toolkit

The CORBA toolkit was developed in 1996-1997 for the DESY/KEK accelerator [3]. It provides standard interfaces for general-purpose data exchange, such as “GetValue”, “GetList”, “GetHist”, and so on. However, the ORB used for the toolkit does not have the connectivities with other

[1]Internet Inter-Operable Protocol, the standard communication protocol between different ORBs.

[2]Object Request Broker, a broker who is in charge of the information on the servers. All the clients ask server information to the broker.

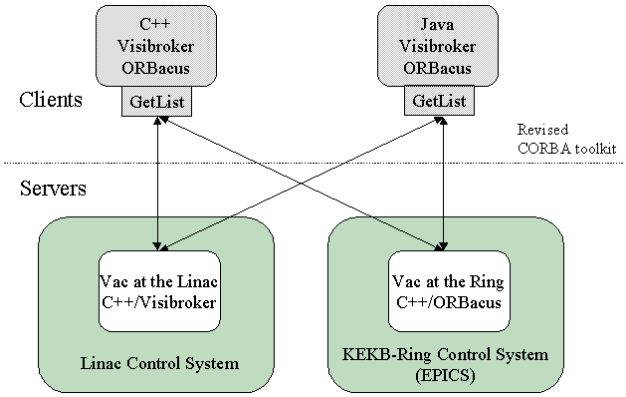

Figure 2: Forthcoming scheme of the studies.

IIOP-based ORBs. Moreover, we have received almost no update from the provider of the ORB over recent years.

We decided to introduce IIOP-based ORBs in order to develop a revised version of the toolkit. We selected VisiBroker [4] as a new ORB, however, the availabilities with free ORBs (ORBacus, Fnorb, etc.) are also in consideration. The revised toolkit will have the same interfaces as before. In addition, it will allow to use Java and Python as client-side languages.

A Java applet, which gives a status display of vacuum, and a C++ server are under development at the KEK Linac. They use a prototype of the revised CORBA toolkit (Fig. 1). The basic communications have already been successful, but it will take another couple of months to complete the full development of the toolkit.

2.3 Java and CORBA feasibility studies

We have studied the feasibility of applications which use both Java and CORBA. If an application is written in Java and communicate with the CORBA protocol, it is expected to work on any platform with the same source.

First, we developed a real-time display of klystron status at the KEK Linac (Fig. 1 and [1]). The application, which acts as a CORBA client, was developed as a Java applet. The CORBA server was written in C++, with the C-based libraries of the Linac control system. The Java applet was available on four different platforms [1], which implies high source portability of the Java language.

As a next step, we introduce a free IIOP-based ORB (ORBacus). Free ORBs are important for us, since they are expected to suppress the total cost. A Java application and a C++ client were developed with ORBacus in order to inspect the interoperability between different ORBs (Fig. 1). The result was successful.

2.4 Other studies

The use of CORBA at a communication layer enables us to introduce languages other than Java and C++. We are interested in Python, a script language with graphic features, as a client-side language. Actually Python is a standard language of the KEKB-ring control system, and several numbers of Python applications have been already developed [6].

We introduced Fnorb, a free CORBA ORB for Python, to the KEKB-ring control system. A communication test was carried out between a Python client at the KEKB-ring and a C++ server at the linac. The result with a simple interface (a “Hello” server) was successful.

We will continue the studies to include more complicated interfaces, since we want to introduce Python clients in our scheme for sharable applications (see 3.2).

3 DISCUSSION

3.1 How to bind to a specific server

One of the practical problems during the development is that the interface names of servers are always the same with the toolkit. In order to identify a specific server, two methods are possible:
(1) Use a string-type (stringified) object reference
The common rule to convert an object reference to a string is described in the CORBA specification. Fig. 3 shows an example of using a stringified object reference. The server writes the stringified object reference in a file. Though the client and the server are based on different ORBs, the client can get the object reference of the server by reading the file.
(2) Use CORBA Naming Service
Use of the CORBA Naming Service seems to be very suitable, however we are not sure how to identify interfaces with same names by Naming Service.

Among the above methods, we have used the first method in the present development. This is not an elegant way since it required a common file, but a safe solution at an early phase.

3.2 Future plan

The goal of our studies is as simple as shown in Fig. 4. The KEK linac and the KEKB-ring will prepare the CORBA servers for each device. The communication interfaces are given by the new CORBA toolkit based on IIOP-based ORBs. The applications developed for one control system can be sharable with another one, since they use common interfaces provided by the toolkit.

Most of the ORBs in the scheme are free software, thus the total cost of the ORBs is negligible. An application can be prepared with Java (normally as a Java applet), but some of them may be written in Python as well as in C++.

4 CONCLUSION

We developed two Java applets, which show status displays of the klystrons and the vacuum system, and C++ servers of the KEK Linac. In the examples, the Java applets

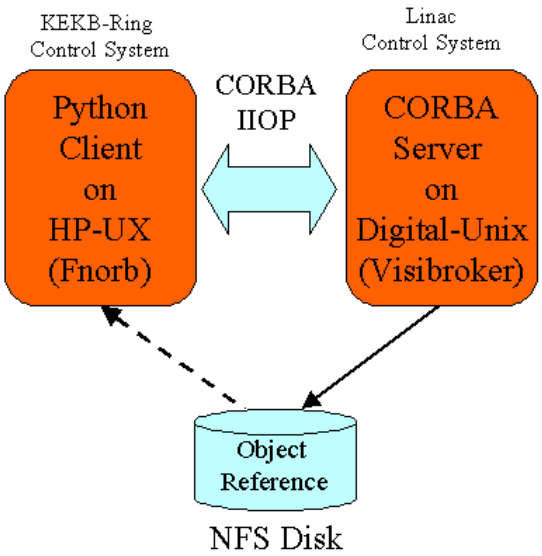

Figure 3: How to get an object reference by a file.

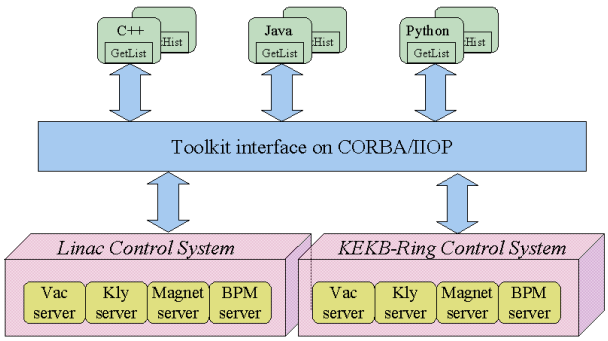

Figure 4: Goal of our studies in the future.

communicate with an existing control system by using the CORBA protocol.

We confirmed interoperability between two different ORBs (VisiBroker and ORBacus). However, we need to further investigate the common scheme to get an object reference of servers.

We have a plan to have sharable applications between different control systems, the KEKB-ring and the KEK Linac. We will install a CORBA environment in the KEKB-ring control system. In this case, we will use Python (in addition to Java) as a client-side language.

5 ACKNOWLEDGMENTS

The authors acknowledge to Prof. K. Nakahara and Mr. K. Asai for kindly supervising our work. We thank Dr. N. Yamamoto for the help at developing a Python client application. The preparation of the development environment for Java is supported in part by a Grant in Aid of Scientific Research, Ministry of Education, Science, Sports and Culture, Japan.

6 REFERENCES

[1] S. Kusano, N. Kamikubota and K. Furukawa, Proc. of the Second International Workshops on Personal Computers and Particle Accelerator Controls (PCaPAC'99), Tsukuba, Japan, Jan.12-15,1999, KEK-Proceedings 98-14 (CD-ROM); KEK-Preprint 98-238

[2] G. Obukhov, M. Clausen and N. Kamikubota, Proc. of the Second International Workshops on Personal Computers and Particle Accelerator Controls (PCaPAC'99), Tsukuba, Japan, Jan.12-15,1999, KEK-Proceedings 98-14 (CD-ROM)

[3] N. Kamikubota, M. Clausen and K. Furukawa, Proc. of the 1997 International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'97), Beijing, P.R.China, Nov.3-7, 1997, Science Press (1998), p.351-353; KEK-Preprint 97-198

[4] VisiBroker for C++ V3.3, and VisiBroker for Java V3.3, Inprise Corporation, CA U.S.A.

[5] N. Kamikubota, K. Furukawa, K. Nakahara and I. Abe, Nucl. Instr. Meth. A352(1994)131

[6] T. Katoh, A. Akiyama, S. Araki, M. Kaji, T. Kawaguchi, T. Kawamoto, I. Komada, K. Kudo, T. Naito, J. Odagiri, M. Takagi, S. Yoshida and N. Yamamoto, Proc. of the 1997 International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'97), Beijing, P.R.China, Nov.3-7, 1997, Science Press (1998)