

(F18p20)

## Realtime Display of Accelerator Status Using JAVA and CORBA

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

### Abstract

Recently introduction of object-oriented technologies into accelerator controls has been understood as a future promising direction. Among the object-oriented technologies, Java and CORBA have been widely accepted in all the industrial fields. We, the KEK electron/positron injector-linac, have investigated the possibility to introduce such technologies for real-time display of accelerator status. A test program developed with Java, which is act as a CORBA client, communicates with the CORBA server at an Unix workstation of the KEK-linac control system. As a result, the status of the KEK-linac was shown at a web-browser of remote PC. Experiences and discussions are given in this article.

### JAVA/CORBA による Realtime 表示の試み

#### 1 はじめに

最近、加速器制御分野では、Java[1] や cdev[2, 3] などのオブジェクト指向技術が注目を集めている。Java は platform に依存しない GUI 機能を持つため、移植性の高いアプリケーションを記述する言語として適している。また、CORBA(Common Object Request Broker Architecture) はネットワーク分散した計算機環境の標準通信プロトコルとして期待を集めている。CORBA は、OMG(Object Management Group)[4] と呼ばれる 700 以上の関連計算機会社によるグループにより標準化が行われており、さまざまな OS(Operating system) や言語 (C,C++,Java,etc.) からの接続性が保証される。

KEK 電子・陽電子入射器は、Unix 計算機をホストとし、TCP/UDP protocol (socket) を通信層に使用した制御システムで運転されている [5, 6, 7]。KEK 入射器の Man-Machine interface には、(1) PC/Windows を利用した operator's console system[8]、(2) DOS-base の touch panel system[9]、などがある。Operator's console(Windows) のアプリケーションから Unix ホストに接続するには、Winsock 関数を使って TCP/UDP protocol を話す必要があった。これは実現したが、Unix の socket 関数と Windows の Winsock では通信層の coding は別々に開発する必要があり、新しいサービスの追加や改修のたびに 2 度手間になる。また、touch panel では、FTP 社の PCTCP を導入することで Unix との通信層の source を共通に出来た。しかし、DOS 特有の 640kB のメモリ制限などから注意深い実装が必須で、アプリケーションを開発しやすい環境とはいえない。

そこで、KEK 入射器の運転情報などをリアルタイム表示するアプリケーションの開発に、Java と CORBA の両方を利用する可能性を考えた。アプリケーションを Java で記述しかつ制御システムとの接

続に CORBA を通信規約として採用すれば、PC を含むあらゆる platform で動作し、しかも同じ source を使える。さらに、いろいろな加速器 (計算機システム) に接続して共通に使う、という可能性も期待出来る。このような背景から、実際に KEK 入射器制御システムに CORBA で接続する Java のテストプログラムを作成し、その評価を試みた。本稿ではその経験を紹介し、考察を加える<sup>1</sup>。

#### 2 加速器情報のリアルタイム表示の試み

##### 2.1 概要

本稿のテストでは、一例として KEK 入射器のビーム繰り返し情報のリアルタイム表示を試みた。テストに必要な CORBA 製品として、Inprise 社の VisiBroker for C++/Java を使用した<sup>2</sup>。VisiBroker を選択したのは、Netscape で走る Java applet からの通信規約 IIOP (Internet Inter-ORB Protocol) をサポートしているからである<sup>3</sup>。

テストの全体像を図 1 に示す。アプリケーションは Java で開発した。Java クライアントを実行するには Java applet が実行できる環境であればよい。WindowsNT や Macintosh では Web-browser で、Unix では browser のほか appletviewer でも実行できる (図 1 の上半分)。VisiBroker による CORBA サーバは C++ で開発を行った (図 1 の下半分)。入射器のビー

<sup>1</sup> 紙面の制限から、Java および CORBA 自身の解説は割愛した。ご容赦願いたい。

<sup>2</sup> VisiBroker for C++ v3.0 および VisiBroker for Java V3.2。開発環境は DEC Digital Unix 4.0B。VisiBroker は [10] を参照。

<sup>3</sup> IIOP 自体は CORBA の規格でもり、他社の CORBA 製品でもサポートされている。VisiBroker が注目されるのは、Netscape が VisiBroker for Java の通信 class を採用したからである。

ム繰り返し情報を得るには、入射器制御システムから提供されている C 言語ベースの関数をリンクすることになる。また、クライアントとサーバ間の通信 interface は、CORBA の IDL<sup>4</sup> で記述する。実際の通信は IIOP protocol で行われるが、ユーザーがそれを意識することはない。

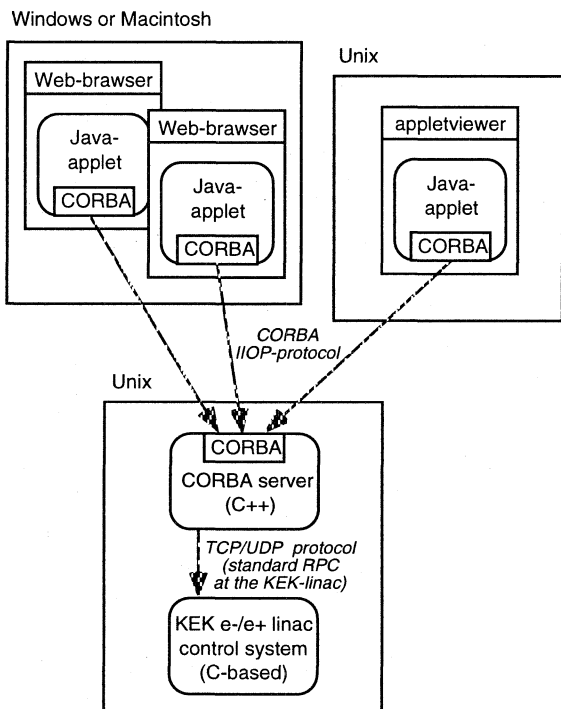


Figure 1: 分散システム図

## 2.2 実装

**IDL file** まずはじめに、クライアントとサーバ通信層の interface を決める IDL file を作成する。

```
//intlk.idl
interface Intlk {
    string sts(in string name);
};
```

intlk.idl から、クライアントとサーバの通信層のコードは自動的に生成される。

**入射器制御サービスの実装** CORBA サーバ (C++ 環境) では、ビーム繰り返し情報を得る入射器制御システムの C 関数 intlk() を、以下のように実装する。

```
virtual char *
intlk_impl::sts(const char * name){
    int rtn;
    char data[256];
    //interloack service in KEK-Linac control
    rtn = intlk("STS", name, data);
    return data;
}
```

<sup>4</sup> Interface Definition Language. 通信層の interface を記述する言語で、CORBA の基本機能の一つ。

**CORBA サーバ側の実装** CORBA サーバ側で以下の実装が必要である。

```
main() {
    Intlk_var intlkImpl = new intlk_impl();
    CORBA_String_var ior =
        orb->object_to_string(intlkImpl);
}
```

**html の実装** Java applet を download させるための html script は以下のようになる。

```
//intlk.html
<applet code=testApplet.class>
<param name=ior value=IOR:012020200.....>
</applet>
```

**Java クライアント側の実装** クライアント (Java) 側の実装を示す。

```
org.omg.CORBA.Object orbObject =
    orb.string_to_object(url);
intlk = INTLKHelper.narrow(orbObject);
Value = intlk.sts("FREQ");
```

## 2.3 結果

テストプログラムを実行すると入射器のビーム繰り返し情報が表示され (図 2)、定められた一定時間毎 (現在は 10 秒) に更新される。

表 1 に、いろいろな platform でテストした結果を示す。表中にある none は、その platform には install していないことを表す。多くの platform で動作することが確認できた。

Table 1: 各 platform とブラウザの動作結果

Platform	Netscape4.0.x	Netscape4.5
Digital Unix	OK	OK
WindowsNT/95	OK	none
MacOS8	OK	none
Linux	OK	OK

## 3 考察

### 3.1 通信速度

クライアントとサーバ間の通信時間 (往復) を測定するために、別途簡単なクライアントプログラム (Java) を作成した。転送データ量は、行き (帰り) は 5 (256) バイトである。結果は 50-60 ms であった。この時間は、クライアントが CORBA サーバのある Unix マシンで走っても、別の計算機 (Linux, Pentium@133MHz) で走っても同じであった。この結果、クライアント (Java applet) が KEK 内の

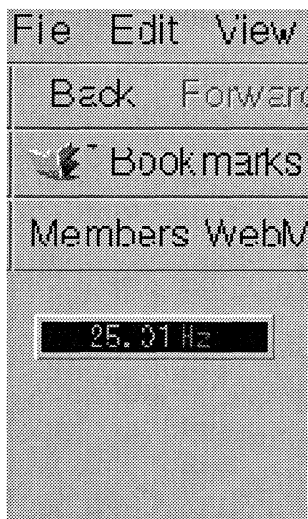


Figure 2: Netscape 上でのビーム繰返しの表示

計算機で走るとすれば、計算上は 0.1 秒のリアルタイム性が可能ということになる<sup>5</sup>。

### 3.2 実装時に経験した問題点

Java もしくは web-browser のバージョンの違いで正常に動作しないことがあった。当初 JDK1.1.6<sup>6</sup> の class を使用したが、Netscape 側の Java 環境での class が JDK1.0.x 相当であったため動かない GUI 機能があった。解決方法として、テストでは JDK1.0.x の class を使用した。また、KEK 入射器制御システムは C 言語の関数しか提供していないため、C++ が前提の CORBA サーバへの実装時にリンクエラーが頻発した。この問題は、C 関数の定義をヘッダファイルに記述することで解決した。

### 3.3 Windows での問題点

Microsoft 社は、Windows 環境の分散システムの通信規約には、CORBA でなく自社の DCOM/ActiveX を推奨している<sup>7</sup>。CORBA と DCOM との通信は現在 OMG で検討中であるが、実現していない。これに関しては将来に期待するしかない。

## 4 謝辞

本稿のテストプログラム開発には、DESY の Gennadiy Obukhov 氏の JOI(Java Operator Interface[11]) のソースコードを参考にしました。ここに感謝します。

<sup>5</sup> 多数のクライアントが同時に 10Hz でデータ更新するなら、それに見合うネットワーク容量やサーバ側 CPU 能力について考慮が必要である。

<sup>6</sup> Visibroker for Java は、JDK1.1.5 以降で動作。

<sup>7</sup> KEK 入射器でも、Visual Basic のアプリケーションとデータベース Microsoft-SQL との通信に DCOM を使っている。

## References

- [1] <http://java.sun.com>
- [2] J. Chen, G. Heyes, W. Akers, D. Wu and W.A. Watson III, Proc. 1995 Int'l Conf. on Accelerator and Large Experimental Physics Control Systems (ICALPECS'95), Chicago, Oct.-Nov.1995, FERMILAB-CONF96/069, p.97
- [3] <http://www.jlab.org/cdev/>
- [4] <http://www.corba.org> or <http://www.omg.org>
- [5] N.Kamikubota, K.Furukawa, K.Nakahara and I.Abe, Nucl. Instr. Meth. A352(1994)131
- [6] 上窪田紀彦、他、第 20 回ライナック研究会、大阪、1995 年 9 月、Proc. of the 20th. Linear Accelerator Meeting in Japan, p.209
- [7] N. Kamikubota, K. Furukawa, K. Nakahara, I. Abe and A. Shirakawa, Proc. Int'l Conf. on Accelerator and Large Experimental Physics Control Systems (ICALPECS'95), Chicago, October 1995, FERMI-LAB Conf-96/069 p.1052
- [8] K.Nakahara, I.Abe, K.Furukawa and N.Kamikubota, Nucl. Instr. Meth. A293(1990)446
- [9] 上窪田紀彦、他、第 17 回ライナック研究会、仙台、1992 年 9 月、Proc. of the 17th Linear Accelerator Meeting in Japan, p.276
- [10] <http://www.ogis-ri.co.jp/otc/products/visigen/>
- [11] private communication, or check: <http://www.kryo.desy.de/~obukhov/JOI/>