

[F18p21]

Sharing of Application-level Software with CORBA

Norihiko Kamikubota, Kazuro Furukawa, and Matthias Clausen*

High Energy Accelerator Research Organization (KEK)

1-1 Oho, Tsukuba, Ibaraki 305, Japan

Deutsches Elektronen-Synchrotron (DESY) *

Notkestrasse 85, 22603 Hamburg, Germany

Abstract

Trials to share an application-level software with CORBA have been made at DESY and KEK. The aim is to evaluate the availability of CORBA over different control systems. In order to inspect the availability, it was introduced into three different control systems at DESY and KEK. An archiver viewer, an application software developed originally for DESY-HERA cryogenic control, was modified to have CORBA-APIs, then introduced into the control system of the KEK injector linac. As a result, the archiver viewer becomes available with the data of the KEK injector linac.

CORBA を用いた上位アプリケーションソフトの共有

1 Introduction

大型加速器制御の研究分野では、汎用性のあるソフトウェアを交換し、お互いに利用しあおうという気運が高まっている。例えば LANL が開発した EPICS (Experimental Physics and Industrial Control System) が KEK を含む世界の主要な加速器施設で採用されるようになったり [1, 2], cdev[3, 4] や Java[5] など最近のオブジェクト指向技術が制御ソフトウェアの移植性を高める技術として注目を集めたりしている [6]。その一方、ある加速器用に開発されたアプリケーションソフトを他の加速器に移植するのは、一般には容易ではない。ソフト中に計算機環境に依存する部分が含まれるケースが多いからである。

CORBA (Common Object Request Broker Architecture) はネットワーク分散した計算機システム間の標準通信規約であり、主要な計算機ベンダーが参加したグループ OMG によって議論される [7]。現在、1994 年 12 月採択の CORBA 2.0 に準拠した種々の CORBA 製品が多数 (20-30) 公開されている¹。CORBA は計算機 (我々の場合加速器制御システム) の違い (ベンダー、Operating System、使用言語) を吸収し、またそれらがネットワーク分散していることを隠蔽する。このため、CORBA を通信規約とするアプリケーションソフトを用意すれば、異なる加速器制御システムで共通で使用出来ると考えられる。

DESY では、1994 年に履歴データを表示するグラフィックアプリケーション (archiver viewer) を HERA cryogenic 用に開発し、運転に使用している。このアプリケーションソフトの KEK 電子・陽電子入射器への移植を試みた。その手順は、a) 汎用の API(application interface) を持った toolkit を市販の CORBA 製品をベースに開発し、b) archiver viewer をこの CORBA toolkit にリンクする、そして c) KEK 入射器制御シ

ステムに CORBA toolkit を導入して入射器履歴データへアクセスするための toolkit server を準備、である。その結果、DESY の archiver viewer で KEK 入射器の履歴データが参照できるようになった。

本稿では、第 2 章で CORBA toolkit、第 3 章で archiver viewer、第 4 章で本研究の意義・問題点について議論する。

2 the CORBA Toolkit

2.1 Object Broker as an ORB

最初に開発ベースとする CORBA ORB (Object Request Broker) を検討した。1996 年当時、Orbix (IONA 社) が最有力候補であったが、結局 DEC 社の Object Broker が選択された。その理由は、低コストで導入出来たこと、また KEK/DESY とも制御システム基幹部は DEC 社の計算機で、DEC 社の ORB を導入するのが自然であったこと、などがあげられる。Object Broker は KEK/DESY の 3 種類の計算機に導入された (2.2 章)。Toolkit 開発途上のテストによって、少なくともこれらの機種で Object Broker のソース互換性、相互接続性が確認された。

2.2 Toolkit overview

Toolkit は DEC 社 Object Broker V2.6² をベースに開発され、

- (a) Open-VMS V6.2 (DEC Alpha250 4/266) with DEC-C and MultiNet/TCPIP
- (b) Sun OS 4.1.3 (Sparc Station 10) with acc
- (c) Digital Unix v3.2D (DEC Alpha Server 2100A) の 3 機種で動作および相互接続を確認した。開発した

² 現在の version は 2.7 / 3.0。なお、Object Broker は現在は BEA 社が扱っている。

¹ 例えば <http://www.vex.net/~ben/corba/> など。

toolkit では、表 1 に示す汎用 API を用意した。現時点では C/C++ のソースとの接続を前提としており、典型的な client program は、以下のような形になる。

```

DESY_ClientInit( \"SampleServer\" );
while( condition ) {
    DESY_GetList( \"SampleServer\", // in
        TagList, AttrList, ListCount, // in
        DataList, StatusList); // out
}
DESY_ClientTerminate( \"SampleServer\" );

```

Toolkit の詳しい仕様等については、[8, 9, 10] を参照されたい。

Table 1: Toolkit の API(一部)。

Function	Description
DESY_ClientInit	Initialization
DESY_ClientTerminate	Terminate the client
DESY_GetValue	Get single any-type data
DESY_GetList	Get multiple data
DESY_GetTable	Get two-dimentional data
DESY_GetHist	Get archiver histories

2.3 Toolkit performance

代表的な toolkit 関数で、4 byte および 4 KB のデータを server から client へ転送する時間を測定した。結果を表 2 に示す。表中 “Sun to Sun” は同じマシン内に server と client がある例、“VMS to Sun” はデータ転送がネットワーク (10Mbps) を越える例である。なお、比較のため、Object Broker の簡単なテストプログラムによる測定、および socket 関数によるデータ転送の測定も示している。

表 2 によると、toolkit は Object Broker より 10-40 倍遅い。DESY_GetList の overhead は少ないが、DESY_GetList や DESY_GetHist などの overhead が大きい³。これは、データ転送部分のコードを簡単にするため常にデータ数を一定にするなどの無駄を許している部分の影響が大きいとみられ、将来的には改善の余地がある。

また、Object Broker は socket 関数よりも 50-60 倍遅い。単純な速度比較では、Object Broker は socket に全く歯が立たない。ただし、Object Broker (CORBA) は、a) 転送量制限なし (socket は 1 kB 程度)、b) 多数のデータを一度に送れる (例えば int と float を一度に)、c) IDL⁴ による通信層のコード開発・保守が楽 (platform によらず一元管理)、d) multi-thread の管理が自動 (socket でも出来るがコード開発はユーザ側の責任)、e) server redundancy (server が落ちて別ノードの server が自動 backup)、など、多くの利点がある。また、CORBA が遅いというより Object Broker が遅いのでは、という指摘もあり、近々別の ORB で調査する予定である。

³ 4KB のデータで、4.6 ms (Object Broker), 50 ms (GetList), 120 ms (GetHist.)

⁴ Interface Definition Language。CORBA 2.0 の基本機能。

Table 2: Toolkit 関数のデータ転送時間。

Function	Datasize	Sun to Sun	VMS to Sun
DESY_GetValue	4 byte	2.2 ms	12 ms
DESY_GetList	4 byte 4 kB	32 ms 50 ms	180 ms 250 ms
DESY_GetHist	4 byte 4 kB 16 kB	107 ms 120 ms 130 ms	580 ms 680 ms 800 ms
(Object Broker functions)	4 byte 4 kB	3.0 ms 4.6 ms	[at VMS] -
(socket functions)	4 byte 4 kB	0.07 ms 0.12 ms	[at Unix] -

3 Archiver viewer

3.1 Archiver viewer at DESY

DESY で 1994 年に archiver viewer を開発した際、その通信層には DEC 社の @aGlance toolkit を利用していた。この toolkit は CORBA 1.2 規約に準じた製品であったが、利用できる計算機に限られかつ DEC 社もサポートを打ち切っていた。今回開発した toolkit は古い toolkit と入れ替える形で整備され、1994 年に開発したコードには全く変更を加えずに新しい環境に移行出来た [9]。また、これまで VMS マシンでしか動作しなかった archiver viewer が別のマシン (Sparc) で使える可能性が出てきた (図 1 の左半分)。

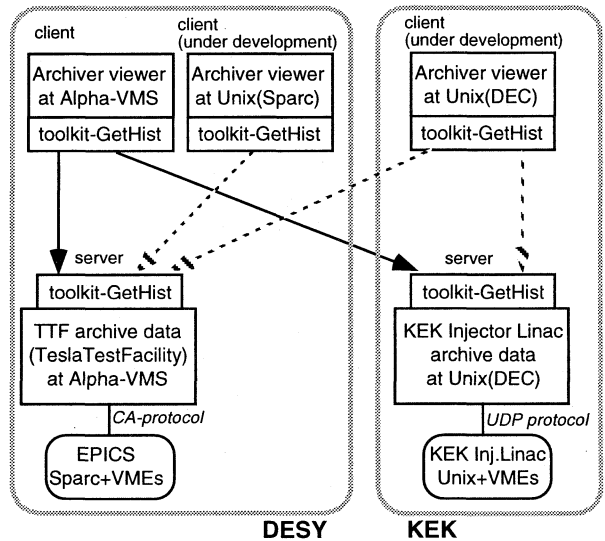


Figure 1: Relationships of the CORBA toolkit and archiver viewer(s) at DESY and KEK.

3.2 Archiver viewer at KEK

Object Broker および toolkit は 1996-1997 年に KEK 入射器でも導入され、つづいて入射器の履歴データ

にアクセスする toolkit server が開発された。この結果、DESY で走る archiver viewer で KEK 入射器の履歴情報が表示できるようになった (図 1、図 2)。さらに archiver viewer 自身を KEK へ移植するべく準備しているが、必要となる graphic package が KEK 側に無く、現在解決策を模索している。

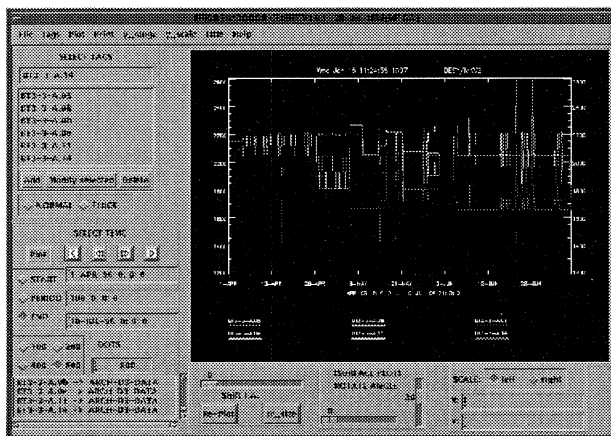


Figure 2: Archiver viewer running at DESY with histories of six steering magnets of the KEK-Linac.

4 Discussion

CORBA の導入によって、DESY で開発された上位アプリケーションが KEK でも利用可能になることが実証された。このような上位アプリケーションの複数の研究所間での共有・交換は、cdev[3, 4] を利用した例があるが、そう多くの成功例があるわけではないと思われ、本研究の意義は大きい。Archiver viewer を KEK 入射器での実際の運転に使うには、3.2 で指摘した graphic package の問題とともに、HERA cryogenic に特化した部分⁵ を改修する必要がある。さらに移植性について一般化の議論を進めれば、cdev や Java などのオブジェクト指向技術との組み合わせが望ましいと思われる [6]。

KEK 入射器と Ring (HER/LER) は、歴史的事情から別々の加速器制御システムで運転されている。今回の研究をもとに、一つの上位アプリケーションが、CORBA を介して入射器・リング両方の制御システムとデータを交換するような応用⁶ が考えられる⁷。

現時点で、toolkit は同時に複数の server に接続できない。この問題は、toolkit の開発時間が限られていたために削った機能であり、早晚改修されるであろう。また、Object Broker は、最新の version でも CORBA 2.0 に準拠していない部分があるため、他の ORB への移行を検討している。

⁵ 例えば、archiver viewer の 3 次元表示では縦軸の単位は必ず「(超伝導電磁石) 温度」になってしまう。

⁶ 例えば入射器エンドからリングのトランスポートへの電磁石や真空を 1 つの表示 (1 つのアプリケーション) にする。

⁷ 入射器・Ring の相互通信に CORBA を利用する可能性は、1995 年の [11, 12] で指摘されている。

5 Acknowledgments

この研究の大部分は、著者の一人 (n.k.) が 1996 年に 10 ヶ月間 DESY に滞在した時に行われました。技術的サポートのみならず、多岐に渡って助けていただいた DESY cryogenic control の人たちに感謝します。渡航費・滞在費は「平成 7 年度海外への中核的研究拠点への派遣研究者」(JSPS Fellowship for Research at Center of Excellence Abroad) として日本学術振興会から支給されました。ここに感謝します。

References

- [1] M.E. Thuot, M. Clausen, L.R. Dalesio, T. Kato, M.E. Kraimer, R. Mueller, H. Shoae and W.A. Watson, Proc. of the XVIII Int'l Linear Accelerator Conference (LINAC96), Geneva, August 1996, p.611
- [2] <http://www.aps.anl.gov/xfd/SoftDist/>
- [3] 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 (ICALEPCS'95), Chicago, Oct.-Nov.1995, FERMILAB-CONF96/069, p.97
- [4] <http://www.jlab.org/cdev/>
- [5] <http://java.sun.com>
- [6] <http://www.sls.psi.ch/SOSH98/summary2.html>
- [7] <http://www.corba.org> or <http://www.omg.org>
- [8] http://www-linac.kek.jp/~kami/report/desy96/corba_desy.html
- [9] N. Kamikubota, M. Clausen and K. Furukawa, Proc. Int'l Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'97), Beijing, P.R.China, November 1997, in press; KEK Preprint 97-198(Nov.97)
- [10] http://www.sls.psi.ch/SOSH98/nk/status_toolkit.html
- [11] K. Furukawa, N. Kamikubota, K. Nakahara, I. Abe and A. Shirakawa, Proc. 1995 Int'l Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'95), Chicago, Oct.-Nov.1995, FERMILAB-CONF96/069, p.863
- [12] 古川和朗、上窪田紀彦、阿部勇、白川明広、中原和夫、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.221