

## マルチベンダ CORBA 製品における C/S システムの実現

5 S-8

金子 淳一、山田 洋一、神人 将彰

NTT コミュニケーションウェア(株)

### 1 はじめに

近年、分散オブジェクト技術として CORBA が注目されている。それに伴い、ベンダ各社が CORBA 仕様に準拠した製品をリリースしている。

従来、複数の CORBA 製品が混在した環境では、他ベンダ間の ORB 接続を行なう際に、IOR (Interoperable Object Reference) を別途何らかの方法で受け渡しを行った上で通信する方法がとられていた。しかし、複数 CORBA 製品混在環境においても、単一製品環境と同様に CORBA インターフェイスのみでシステムを構築できる事が望まれる。本稿では複数の CORBA 製品混在環境で、C/S 間インターフェイスを CORBA のみで実現した事例を紹介する。

### 2 前提条件

開発システムの要求項目は以下の通りである。

- 要求 1: クライアントはサーバの位置情報を保持しない
- 要求 2: サーバは未起動の場合、クライアントからの接続要求が発生された時に起動される

これらを満たすために、CORBA 技術を採用することとした。

また、サーバ側: C++アプリケーション、クライアント側: Java アプリケーションであるため、CORBA 製品として

・サーバ: VisiBroker for C++ 3.2

・クライアント: JDK1.2

を選定した。以上の前提条件を、図 1 に示す。

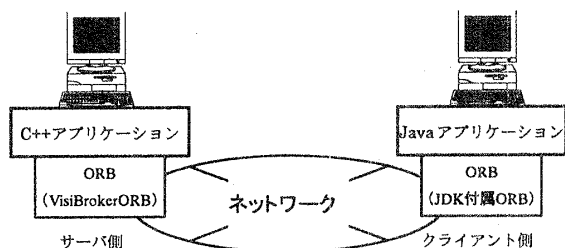


図 1 前提条件

### 3 方式検討

#### 3.1 利用可能な CORBA サービス検討

要求 1 を満たすために、CORBA サービスの 1 つであるネーミングサービスを用いることとした。

要求 2 を満たすために、インプリメントリポジトリを用いることとした。

#### 3.2 実現性の検討

先に挙げた CORBA サービスについて実現性を検討した。

・ネーミングサービスの利用について

ネーミングサービスへの接続は、ベンダ依存のプロトコルを用いるため、他ベンダ製品の ORB とネーミングサービスの接続は不可能である。

・インプリメントリポジトリ利用について

インプリメントリポジトリは、利用可能である。

以上から、ネーミングサービスの利用について、何らかの実現方式を検討する必要がある。

### 4 実現方式

#### 4.1 実現上の問題点

ネーミングサービスの他ベンダ接続の問題を解決するために、以下に示す方式を検討した。

A case of client-server system with multi vendor CORBA products.  
Junichi Kaneko, Youichi Yamada, Masaaki Kamito  
NTT Communicationware Co., Ltd.

#### 4.1.1 CosNamingFactory - JDK1.2 付属 ORB の接続

VisiBroker のネーミングサービス (CosNamingFactory) と JDK1.2 付属 ORB の接続を検討した。その手順を図 2 に示す。

手順

- 1 VisiBroker の ORB は、同一ベンダのネーミングサービスに通常の手順で登録する
- 2 JDK の ORB は、CosNamingFactory の IOR を取得し、その情報を基にネーミングサービスに接続を行う
- 3 JDK の ORB は、ネーミングサービスから得た情報を基に VisiBroker の ORB との接続を試みる

結果

- 1 JDK 付属の ORB と CosNamingFactory の接続は不可
- 2 ネーミングコンテキストを取得できないため、VisiBroker の ORB と JDK の ORB の接続は不可

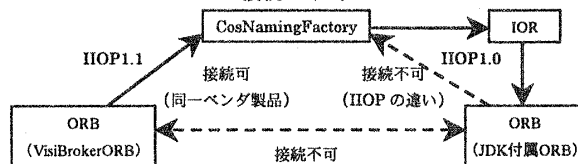


図 2 VisiBroker ネーミングサービスの利用

以上の結果から、この方式による接続は不可能であることがわかった。この原因は、通信プロトコルである IOP のバージョンが、CosNamingFactory と JDK 付属の ORB 間で異なっているためであると推測される。

#### 4.1.2 VisiBroker 付属 ORB - tnameserv の接続

次に JDK に付属するネーミングサービス (tnameserv) と VisiBroker の ORB との接続方式を検討した。その手順を図 3 に示す。

手順

- 1 VisiBroker の ORB は、tnameserv の IOR を取得し、その情報を基に自身を tnameserv に登録する
- 2 JDK の ORB は、同一ベンダのネーミングサービスに通常の手順で接続する
- 3 JDK の ORB はネーミングサービスから得た情報を基に VisiBroker の ORB と接続を試みる

結果

- 1 VisiBroker の ORB の tnameserv への登録は可能
- 2 JDK の ORB と tnameserv は同一ベンダ製品であるため、通常の手順で接続することが可能

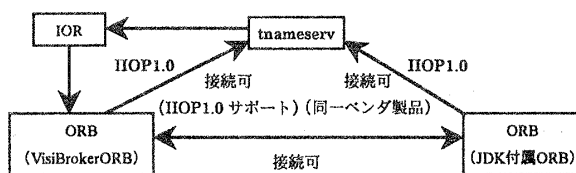


図 3 JDK のネーミングサービスの使用

以上の結果から、tnameserv を使用する方式による接続は可能であることが判明した。これは、サーバ関数は IOP1.0, 1.1 双方をサポートし、接続要求クライアントが発行した IOP のバージョンによって、サーバ側 IOP のバージョンを決定するためである。(この場合は IOP1.0 で要求を受け取ったため、IOP1.0 で tnameserv に自分を登録した)

#### 4.1.3 接続方式に関する問題

4.1.2 の方式の採用によって、他ベンダの ORB とネーミングサービスが接続可能であることを確認した。しかし

・IOR 受け渡し方法に関する問題

・tnameserv におけるネーミングコンテキストの非永続化の問題  
この 2 つの問題が残っている。

### 4.2 システム構成の検討

ネーミングサービスの取得や接続方式の問題を解決するため、図4に示すシステム構成を考案した。

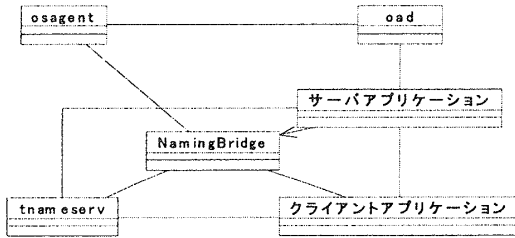


図4 システム構成図  
osagent: VisiBrokerのBOA  
oad: インプリメントリポジトリのVisiBrokerにおけるインプリメント

この構成の特徴は以下の通りである。クライアント側アプリケーションとネーミングサービスの接続のためには、tnameservを用いなければならないことを4.1.3で述べた。しかしサーバをtnameservに登録するには、サーバがそのIORを何らかの方法(CORBA以外のプロトコル)で取得しなければならない。この機能をサーバプログラムに実装すると、システムの可搬性を損なう。従って、IOR受け渡し機構を隠蔽するような機能が必要である。また、ネーミングサービスを永続的に提供する機能も必要である。これらの機能をサーバ、クライアントから隠蔽しながら実現する方式を考案した。(以下この機能をNamingBridgeと呼ぶ)

NamingBridgeの機能概要を以下に示す。

#### 4.2.1 IORの受け渡し機構の隠蔽

CORBAサービスの起動プロセスとして、NamingBridgeの起動を盛りこむ。このような構成とすることで、tnameservの起動及びIORの受け渡しがNmingServiceプロセスの中で行なわれることとなり、IOR受け渡しをサーバ、クライアント双方から隠蔽することを可能とした。CORBAサービス起動シーケンスを図5に示す。

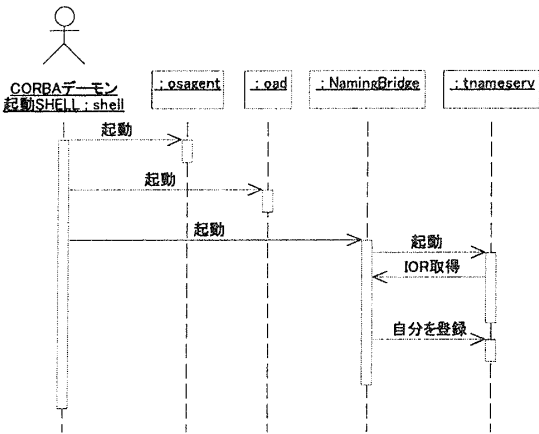


図5 CORBAサービス起動シーケンス

#### 4.2.2 ネーミングコンテキストの非永続化問題の克服

本システムで用いるJDK付属のネーミングサービス、tnameservは、動作時のみネーミングコンテキストを保持し、終了と同時にネーミングコンテキスト情報を破棄してしまう。しかし図6に示すように、NamingBridgeへの事前要求フェーズを設け、クライアントがtnameservに対してネーミングサービスを取得する前に、必ずサーバが登録されていることを保証する機構とする。このような構成によって、永続的なネーミングサービスの提供を実現した。

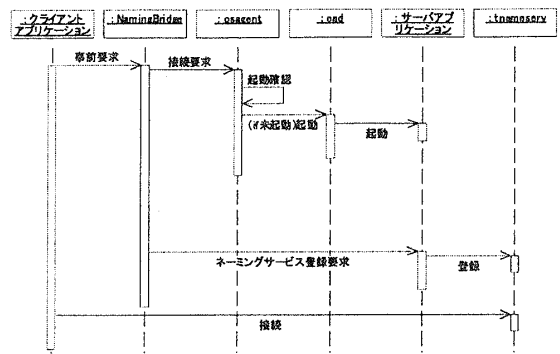


図6 ネーミングコンテキスト永続化シーケンス

### 4.3 実現システム

システム全体のコラボレーション図と、処理の流れを示す。

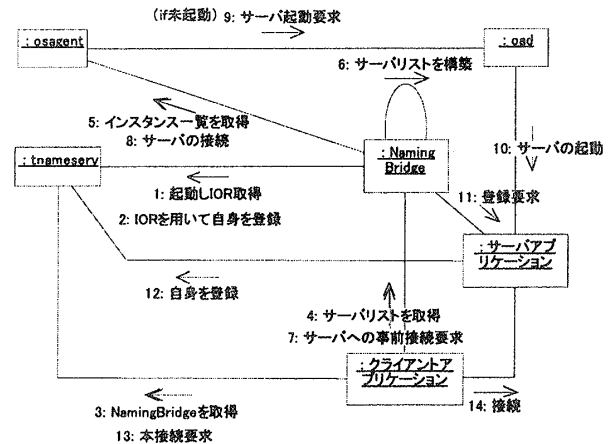


図7 実現システムコラボレーション図

CORBAサービスの起動(図5、図7参照)

- 1 osagent、oad、NamingBridgeをそれぞれ起動する
- 2 NamingBridgeは、内部処理としてtnameservの起動を行なう

クライアントの処理(図6、図7参照)

- 1 tnameservへNamingBridge取得を要求する(図7-3)
- 2 NamingBridgeにサーバリストの取得を要求する(図7-4~6)
- 3 クライアントはサーバへの接続要求を、事前にNamingBridgeに対して行なう(図7-7~10)
- 4 NamingBridgeはサーバの起動、未起動に関わらず、サーバをtnameservへ登録させる(図7-11~12)
- 5 クライアントはtnameservに本接続要求を発生し、サーバと本接続する(図7-13~14)

### 5 まとめ

今回考案したNamingBridge機能により

- ・永続的なネーミングサービスの提供
  - ・CORBAプロトコルのみを用いたシステム構成
  - ・アプリケーション内部のベンダ依存部分の局所化
- を実現することができた。しかし抜本的には、CORBA仕様にネーミングサービスとの接続に関する規定を設けることが望まれる。

VisiBrokerはInprise Corporationの商標です  
osagentはInprise Corporationの商標です  
oadはInprise Corporationの商標です  
JDKはSun Microsystems社の商標です

#### [参考文献]

- [1] 中村信孝他 "CORBA仕様ORB製品を用いた相互接続方法と性能評価" NTT R&D Vol47 No9 pp.915-920,1998

- [2] Robert Orfali, Dan Harkey 著 "Java&CORBA C/S プログラミング",日経 BP 社
- [3] Java プラットフォーム 1.2 API仕様  
<http://java.sun.com/products/jdk/1.2/ja/docs/ja/api/index.html>
- [4] VisiBroker for C++ programmer's guide
- [5] R.Orfali, D.Harkey, J.Edwards 著 "インスタントCORBA" トッパン