

分散オブジェクト実行環境の試作

2 S - 2

岩本仁、岡下浩明、筒井彰彦、山本暢洋、村田孝志
株式会社情報技術コンソーシアム

1. 目的

オブジェクト指向開発技術の研究の一環として、我々は、CORBA[1]を使った分散オブジェクト実行環境「ITCプラットフォーム（以下、ITCPFM）」を試作してきた。オブジェクトを分析設計の基本単位としてだけではなく、プログラム実行の最少単位として捕らえ、分散システムを上流工程から下流工程にいたるまで一貫してオブジェクト指向技術で開発できるようにするためである。

分散オブジェクトの実行環境には、通信機能を実現したCORBAに加え、分散オブジェクトの集まりをシステムとして機能させるサービスが要求される。しかし、これらのサービスが標準化してあるCORBAservices[2]には次のような問題があり、標準サービスの実現が困難である。

- ◆ 個々のサービスの存在理由が不明
- ◆ サービス同士の関係が不明瞭
- ◆ 各サービスの物理アーキテクチャが未定義

我々の目的は、分散システムをCORBA上で実現するのに最低限必要なサービスを選出し、その物理アーキテクチャを決定することである。

2. ITCPFMのサービス

一般的に、「分散オブジェクトを使ったシステム」の実行環境が提供しなければならないサービスには、大きく分けて、2種類ある：

(1) 分散オブジェクトのライフサイクルを体系

的に管理するサービス

(2) 分散透過性 (distribution transparency) [3] を実現するサービス

ITCPFMは、以下に述べるサービスを実装することにより、ライフサイクルを管理するためのサービス、さらに、RM-ODP[3]の挙げた8種類の分散透過性のうち、位置透過性 (location transparency) と永続透過性 (persistence transparency) とを実現している。また、複製透過性 (replication transparency) 実現のための基本機能、メンバシップサービス [4]、も備えている。メンバシップサービス以外は、CORBAservicesに含まれており、ITCPFMでの実装はこれに準拠している。

- ◆ イベント通信サービス: 多対多のオブジェクト間でのイベント送受信のためのインタフェース Push/Pop Supplier と Push/Pop Consumer、またイベントの通信経路であるインタフェース Event Channel を提供している。
- ◆ メンバシップサービス: イベント通信サービスを利用したサービスである。複数のサーバ間でのマルチキャストによる1対多のイベント配信を実現したサービスで、分散オブジェクトのレプリケーションとライフサイクル管理に使う。グループに参加するサーバに必要なインタフェース GroupMember と、個々のグループを管理するためのインタフェース ServerGroup とを提供する。
- ◆ 永続化オブジェクトサービス: 分散オブジェクトの動的内部状態を保存するために必要な二次記憶装置とのインタフェース PO (Persistent

“Prototype of a Platform for Distributed Objects,”
H. Iwamoto, H. Okashita, A. Tsutsui, N. Yamamoto,
T. Murata; Information Technology Consortium.

Object)、POM (Persistent Object Manager)、PDS (Persistent Device Service)を提供する。これにより永続透過性が実現されている。

◆ネーミングサービス: 分散オブジェクトと論理名とを対応づけるためのインタフェース Naming Context とそれを検索するための Binding Iterator が提供してある。これにより位置透過性が実現されている。

◆ライフサイクルサービス: 分散オブジェクトの生成と削除に関わる機能を共通化している。ライフサイクルを持つ分散オブジェクトに共通したインタフェースである Life Cycle Object、分散オブジェクトを生成する機能を提供する Generic Factory、特定の Generic Factory を検索するための Factory Finder が含まれる。

3. ITCPFM の物理アーキテクチャ

ITCPFМ は、C++ライブラリ 1 本とサーバ 4 本より構成される。Windows NT 上に Orbix2.2 を使って実装した。

ITCPFМ 上で動作するシステムは、分散オブジェクトを実装したサーバの集まりである。C++ライブラリは、これらのサーバが ITCPFМ のサービスを受けるのに必要なインタフェースを実装したフレームワークライブラリ[5]である。サーバを開発するには、ライブラリ中のいくつかの C++クラスを継承により再定義し、さらに、そのサーバ固有の分散オブジェクトの実装部を追加すればよい。こうして開発されたサーバは、実行時に、ITCPFМ 中の 4 本のサーバから、先

に挙げたライフサイクル管理と分散透過性のサービスを受けることができるのである。

最後に、ITCPFМ のサービスに含まれるインタフェースの実装形態の概要を表 1 に示す。各サービスのインタフェースのうち、ITCPFМ 上で動作するシステム側に必要なものがライブラリとして、ITCPFМ 側に必要なものがサーバとして実装されている。なお、この表中のインタフェース以外にも、実際には、さらにいくつかの補助的なインタフェースが実装されている。

4. 参考文献

- [1] OMG (1996): *The Common Object Request Broker: Architecture and Specification*.
- [2] OMG (1997): *CORBA services: Common Object Services Specification*.
- [3] ISO/IEC (1995): *ITU-T X.901 | ISO/IEC 10746-1 ODP Reference Model Part 1. Overview*.
- [4] Birman, K.P., (1996): *Building Secure and Reliable Network Applications*, Manning.
- [5] Pree, W. (1995): *Design Patterns for Object-Oriented Software Development*, Addison-Wesley.

この研究は、情報処理振興事業協会 (IPA) の先進的情報技術の開発促進事業で実施されている「オブジェクト指向によるソフトウェア生産技術開発」プロジェクトの一環として実施している。

表1 主なインタフェースの実装形態の概要

サービス	ライブラリ	サーバ
イベント通信	Push/Pop Consumer, Push/PopSupplier	EventChannel
メンバシップ	GroupMember	ServerGroup
永続化オブジェクト	PO, POM, PDS	なし
ネーミング	なし	NamingContext, BindingIterator
ライフサイクル	GenericFactory, LifeCycleObject	FactoryFinder