オブジェクト指向分散環境 OZ++ のオブジェクトマネージャの設計

6F－3

大西 雅夫(東洋情報システム*)　籠 浩昭(三菱総合研究所*)
田代 秀一(電子技術総合研究所)　西岡 利博(三菱総合研究所*)
吉屋 英二(富士ゼロックス情報システム*)　塚本 享治(電子技術総合研究所)
*情報処理振興事業協会「開放型基盤ソフトウェア研究開発評価事業」研究員

1　はじめに

OZ++は、オブジェクトの交換と共有に基づく、オブジェクト指向分散処理環境である。基本的なオブジェクトモデル、および実行意味論は、電総研で開発された分散システム OZ+[1]を基本とし、その実行性能を向上させ機能を洗練して、広く公開する予定である。現在、OZ++はその基本設計を終えて、設計の詳細化および開発を進めているところである。ここでは、OZ++ランタイムカーネルを表現する特殊なオブジェクトであるオブジェクトマネージャついて述べる。以下オブジェクトマネージャをOMと略記する。

2　OZ++の構成とOMの役割

OZ++のランタイムカーネルは、オブジェクトの実行／交換メカニズムであるエグゼキュータと、ステーションに一個あってエグゼキュータの生成／消滅管理を行なうニュークリアスからなる[2]。

OZ++では、オブジェクトの基本的な動作の実現のうち、セキュリティの検査やクラスの管理など、かなりの部分をオブジェクトとして実現する[3]ことで、オブジェクトによらないランタイムカーネルを小さくし、システムの変更／拡張を容易にする、という基本方針をとっている。

オブジェクトマネージャは言語から見た場合、エグゼキュータを抽象するオブジェクトであり、他のオブジェクトがエグゼキュータの持つ機能を利用する場合の窓口となる。また、エグゼキュータが管理オブジェクト群のサービスを利用する(アップコールの)場合にも、オブジェクトマネージャが仲介を行なう。

A Design of Object-Manager of OZ++: An Object-Oriented Distributed Systems Environment
Masao Onishi (Toyo Information Systems, Co., Ltd.*),
Hiroaki Kago (Mitsubishi Research Institute, Inc.*),
Shuuichi Tashiro (Electrotechnical Laboratory),
Toshihiro Nishioka (Mitsubishi Research Institute, Inc.*),
Eiji Yoshiya (Fuji Xerox Information Systems, Co., Ltd.*),
Michiharu Tsukamoto (Electrotechnical Laboratory)
*: Researcher of Evaluation of Open Fundamental Software Technology Project in Information-technology Promotion Agency, Japan

3　OMが表現するエグゼキュータの機能

ランタイムカーネルに持たせるのは最低限のメカニズムのみとし、ポリシーの部分(オブジェクトの状態遷移など)をオブジェクトで表現することにより、拡張性に富んだ環境が構築できる。

3.1　オブジェクトの管理

OMは、分散透過にアクセス可能なオブジェクトに対し次の処理を行なうと同時にその状態を管理する。

- オブジェクトへの仮想記憶や永続記憶領域の割り当て、初期化、解放。
- 起動時における必須オブジェクトのプリロードと起動。
- オブジェクトの永続化(永続記憶領域への記録)。
- オブジェクトのGCおよび二次記憶領域へのスワップアウト／スワップイン。

3.2　ネームディレクトリの保持

オブジェクトは何らかの文字列の名称を与えられ、それをネームディレクトリに登録されることにより、他のオブジェクトがそのオブジェクトのIDを取得できるようになっている。ネームディレクトリ自身のIDは、OMが供給する。OMはブロードキャストにより、最寄りのネームディレクトリを探索し、その結果を保持する。OMのIDは固定である。

3.3　エグゼキュータの所有者の管理

OZ++では、効率上の理由から、オブジェクトの所有者はそれが存在するエグゼキュータの所有者としている。管理／運用上の理由や、共同作業上の都合からエグゼキュータの所有者を変更できる必要があり、OMはこのためのメソッドを持つ。

3.4　その他

エグゼキュータのシャットダウンや様々な統計情報の提供などを行なう。

4　アップコールの処理のために持つ機能

OZ++言語のメソッド呼び出しは、分散する資源に対するアクセスを簡潔に表現する手段である。従って、ネットワーク上に分散するバージョンやクラスの管理をオブジェクトで実装し、それをランタイムカーネル側がアップコールすれば、ランタイムカーネルで煩雑なネットワークプログラミングを行なわずに済ますことができる。その一括窓口がOMである。

4.1　コンフィギュレーション管理

OZ++はクラスのバージョンシステムを持つ[4]。バージョンの決定はオブジェクトの実行時に行なわれる。オブジェクトがインスタンシエートされたクラスおよびそのクラスが継承するすべてのクラスのバージョン番号の組を「コンフィギュレーション」と呼ぶ。OMはコンパイル時に決定されるクラスのIDからそれに対するコンフィギュレーションへの写像を保持し、管理する。これは、エグゼキュータがオブジェクトの生成やメソッド起動を実行する際に利用される。

4.2　クラス探索関連機能

OZ++ではクラスとその各バージョンの管理もオブジェクトで行なっており、ネットワーク上に分散されている[4]。従って、オブジェクトの生成やメソッド起動の際にOMは、クラスを高速に探索し、そのクラスに関する情報やメソッドの実行コードをエグゼキュータに対して供給する必要がある。クラスの探索にはブロードキャストが使われる。

5　OMの実現方式

OMとランタイムカーネルとの切り分けは、ポリシーとメカニズムとの分離を基本方針とした。例えばオブジェクトの状態管理はOMとエグゼキュータとで2重化しているが、エグゼキュータ側ではオブジェクトの実行に最低限必要な状態のみを管理し、その遷移はOM側のさらに詳細な状態管理の中でコントロールされる。また、カスタマイザビリティも考慮してOMのクラスやメソッドの構成を決定している。OMのクラス構成の概略を図1に示す。OMの中でもエグゼキュータ内ランタイムルーチンの呼び出しやアップコール処理はExecutorというクラスに集中化されている。

エグゼキュータからのアップコールは、各要求ごとに存在するOMのデーモンプロセスによって処理される。デーモンプロセスは、エグゼキュータが「要求キューが空でない」という条件をシグナルするまで待つ。シグナルされたプロセスは、その延長で要求された処理を行ない、アップコールの結果を待っているプロセスに対し、シグナルを送る。

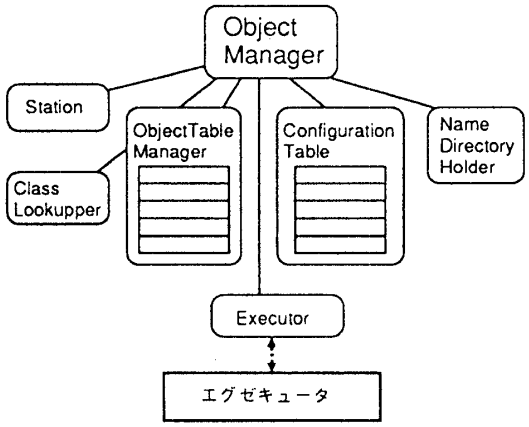

図 1: OMの構成

6　まとめ

現在開発を進めているオブジェクト指向分散環境OZ++のオブジェクトマネージャの機能および実現方式について述べた。

ランタイムカーネルを表現するオブジェクトを設け、オブジェクトで実装されている管理機構とランタイムカーネルとのやりとりをそこに集中させることにより、一層拡張性に富む分散処理環境が実現できる。

本研究において熱心な討論を頂いた、藤野 晃延(富士ゼロックス情報システム)、千葉 滋(東京大学理学部)両氏に感謝する。

本研究は、情報処理振興事業協会「開放型基盤ソフトウェア研究開発評価事業」の一環として行われたものである。

参考文献

[1] 塚本他：「オブジェクト指向開放型分散システム OZ+の研究開発」、電総研彙報、vol. 56、No. 9、Sep. 1992

[2] 塚本他: 「オブジェクト指向分散環境 OZ++ の基本設計」、SWoPP93、Aug. 1993

[3] 籠他: 「オブジェクト指向分散環境 OZ++ のオブジェクト管理機構の概要」、情報処理学会第47回全国大会、Oct. 1993

[4] 吉屋他: 「オブジェクト指向分散環境 OZ++ のクラス管理方式」、情報処理学会第47回全国大会、Oct. 1993