

## 1P-5

## OZ++ 開発計画

塚本享治 濱崎陽一  
電子技術総合研究所

## 1. まえがき

筆者らは、数年前からオブジェクトの交換を基本とするオブジェクト指向型分散システムの研究開発を進めてきた<sup>1)</sup>。これまでに、OZおよびOZ+と名付けた2段階のプロジェクトを行った。第2段階のOZ+プロジェクトは、この3月に終了し、現在その評価を進めている。この評価を反映して、第3段階のプロジェクトOZ++を本年度下半期より、3年半の予定で開始する予定である。本報告では、これまでの経緯とOZ++計画の概要について述べる。

## 2. OZとOZ+について

OZとOZ+では、ネットワークを活用する分散システム上における高度な分散ソフトウェアの開発方法の確立を目指して、次の方針でシステムの研究開発を進めてきた。

- (1) 分散システム上のアプリケーションの全てをオブジェクトで構成する。
- (2) 分散システム上で相互関係を維持して、オブジェクトの転送と操作を行う。
- (3) クラスの継承と版を分散システム全体で管理する。
- (4) 名称管理と永続オブジェクトを実現して、複数サイトの複数ユーザ間でのオブジェクトの共有と交換を可能とする。

第一段階のOZでは、基本的な問題点の抽出を目的として、単一ユーザの分散環境を対象とした分散オブジェクト指向型言語システムの実現を目指した。通産省の大型プロジェクト「電子計算機相互運用データベースシステム」の一環として研究開発を行い、コネクションレス型OSIネットワークを開発し、その上に、高信頼バルクデータ転送とオブジェクト移動にともなうアドレス変更機能を備えた通信システムを実現した。分散オブジェクト指向言語システムとしては、Distributed Processの流れを汲む言語を開発し、クラスの動的ロード、分散GC、分散デバッグなどの機能をもった実行システムを実現した。

これらを用いて、複数台ロボットシステムを実現して評価した結果、オブジェクトの交換方式は分散透明性の点で優れていることが分かったが、大規模化に際しての名称とクラスの一貫性管理などの管理面での問題点が明確となった。

そこで、第二段階のOZ+では、前記の言語システムを拡張し、管理面の拡張をおこなうことによって、複数サイトの複数ユーザ間でオブジェクトの共有と交換を行うことのできる分散システムの実現を目指した。大規模化にそなえた通信システムとして、CLNPとES-IS機能、UDPとのゲイトウェイ機能を実現し、さらに分散管理に必要なグループ同報機能を実現した。言語システムとしては、前記の言語に名称空間と版指定の機能を付加し、コンパイルおよびリンク時に分散したサーバと協同してクラスを生成する実行システムを実現した。この言語を用いて、各種のサーバだけでなく、ネットワーク管理、オブジェクト管理、ユーザのアカウント管理などの分散管理システムを実現した。さらに、オブジェクトの共有と交換を活かすアプリケーションとして、ハイパテキスト機能をもったメールシステムを実現した。

現在、評価を進めているところであるが、これまででわかったこととして次のものが挙げられる。オブジェクトの交換をベースとしたシステムでは分散透明性を維持して高度なアプリケーションが実現できることが実証できた。しかし、分散システム上でのエラー処理、名称管理、継承/版管理、およびオブジェクト記憶管理に複雑な機構が必要となり、性能面において改善の余地が残された<sup>2)</sup>。

## 3. OZ++プロジェクト

第三段階のOZ++では、分散アプリケーションのベースとして使用可能なものを目指す。性能の向上と環境の整備が重要な課題となる。OZ+まではモデルに忠実な実装を行ったが、OZ++では実装方法を変えることによって効率化を図る。

## 3.1 オブジェクト交換システム

ベースのネットワークとしては、極力既存のものを使い、OZ+と同様な3ウェイのコールを行うRPCを実現す

-----  
A Plan of New Project OZ++  
Michiharu Tsukamoto and Yoichi Hamazaki  
Electrotechnical Laboratory

る。ポイントは性能向上におき、頻度の少ないグループ同報機能はアプリケーションレベルで疑似的に行うことになるであろう。クラスの種類が非常に多い場合には、スタブ生成方式では膨大なスタブが必要となるので、オブジェクトのデータ構造を解釈するOZ+と類似の方式を採用する。

### 3. 2 分散オブジェクト指向言語

C++などをベースとすることが考えられるが、それらはクラス間のインタフェースの制約が緩いため、オブジェクト相互のモジュラリティを侵すプログラムが容易にかけしてしまう。これでは、交換されたオブジェクトの実行時に障害を起こす可能性がある。そこで、OZ+と同様にインタフェース制約のきつい言語とするが、可能な限りC++のサブセット化を図る。OZ+ではバイトコードで実行していたが、OZ++では複数機種のネイティブコード（符号化/復号化で使用する情報を含む）を同時に生成し、実行機種に応じて必要なクラスが選択するものとする。受動的なオブジェクトは異機種間でも自由に交換できるが、機種に依存するコンテキストを含む能動的なオブジェクトの交換は同機種間に制限する。特殊な場合を除くと交換されるオブジェクトの大半が受動的なものだからである。

### 3. 3 システム管理

ホスト計算機をネットワークで接続して、1つのマルチプロセッサとして統一的に見せるために、各ホストにはホスト管理のためのシステムを搭載する。このシステムはサイトにまたがった管理範囲を考慮したものであり、オブジェクトの共有と交換をベースにしたものである。次に述べるオブジェクト管理機能を必要とするが、過度にオブジェクト管理機能を使うとシステムの立上げ時や異常時への対応が難しくなる。システム全体の整合性と各ホストの独立性を両立させる工夫が必要である。

### 3. 4 オブジェクト管理

オブジェクトの管理は、名称、クラス、および永続オブジェクトの管理からなる。OZ+では、名称空間内の管理は行っているが、名称空間同士の関係や空間名と実体の写像などに関しては単純なものに限定している。OZ++では、名称空間を単位とした空間群の管理など、名称管理に関して再帰性をもたせる必要がある。クラスの管理に関しても同様であり、クラスサーバ群の管理が必要になる。永続オブジェクトに関しては、そのオブジェクト群が永続オブジェクト群しか参照しない範囲に限定しているが、これはかなりきつい制約である。OZ++では、ディレクトリによって動的に写像を決める余地を増やすことにより、その制約を緩める。

分散システムをまたがったオブジェクト管理の性能向

上を図るためには、オブジェクト通信ですべてを行うOZ+の方式にかわってファイル単位でのキャッシュなどの方法を採用する必要がある。一貫性の保証をどのレベルで実現するかが重要な課題である。

### 3. 5 利用者環境

利用者環境に関しては、OZ+ではシステムの機能を確認する程度のアドホックなものしか実現していない。実用化にあたっては、その整備が欠かせない。OZ++では、前節までに述べたOZ++システムの実現と活用に関連の深い部分に限定してクラスライブラリの整備を行う。また、分散システムに跨ってクラスの参照を行う関係上、高度なブラウザの実現も不可欠である。さらに、OZ+で実現したハイパertext機能、遠隔アクセス機能などをさらに高度化した統合的な分散アプリケーションの実現を通して、統合的な利用者環境整備を行う。

### 4. むすび

OZ++計画においては、OZ+までに開発したオブジェクトの共有と交換を行う方式を基本とした分散システムの実用化を目標としている。そのため、各部分においては、奇をてらうことなく、既知の技術を採用する予定である。なお、この計画は、一般公募で集った数社の企業と協力して進める予定である。

### 参考文献

- 1) 塚本、棟上: OZ: オブジェクト指向開放型分散システムアーキテクチャ -- アーキテクチャとその実現方法、情処学会マルチメディア通信と分散処理研究会、87-DPS-34-7 (1987,7)
- 2) M.Tsukamoto et al.: The Architecture of OZ: Object-Oriented Open Distributed System, Proc of ISIIS'88 (1988,11)
- 3) 塚本他: OZ: オブジェクト指向開放型分散システムアーキテクチャ -- オブジェクト指向型分散プログラミング言語とその実装、情処学会プログラミング言語研究会、89-PL-21-4 (1989,7)
- 4) 塚本他: オブジェクト指向開放型分散システムOZ -- OZの実現とOZ+開発計画、大型プロジェクト「電子計算機相互運用データベースシステムの研究開発」中間成果発表会講演予稿集(1989,12)
- 5) 塚本他: OZ: オブジェクト指向開放型分散システムアーキテクチャ -- オブジェクトの分散管理、情処学会シンポジウム「1990年代の分散処理」(1990, 11)
- 6) 塚本他: オブジェクト指向開放型分散システムOZ+の開発、大型プロジェクト「電子計算機相互運用データベースシステムの研究開発」成果発表会講演予稿集(1991,11)
- 7) 濱崎、塚本: OZ+のオブジェクト通信機構の評価、情処第45回全大、1P-04 (1992,10)