

6M-3 OZ+ : オブジェクト指向開放型分散システム¹ — オブジェクト指向型分散プログラミング言語における同報通信 —

塚本 亨治 (電総研)² 篠原 弘樹 (松下電器)³ 水谷 功 (住友電工)⁴
梶浦 広行 (シャープ)⁵ Petit Bertrand (EC 滞在研究員)⁶

1. はじめに

オブジェクト指向開放型分散システムOZ+の研究開発を行っている。OZ+ではシステム管理等のアプリケーションをオブジェクト指向型分散プログラミング言語OZで記述している [3] [4]。これらのシステムアプリケーションを作成する際、複数の応答側オブジェクトに対し、同時にメッセージを送信する同報機能が必要となる。我々はOZ+における同報通信を実現するために、Multicast RPC [5]、及び、オブジェクトのグループ化とその管理方法 [6] の研究開発をすすめ、実装を行った。本稿ではこれらについて述べた後、OZ言語における同報通信の記述形式について述べる。

2. 同報通信機能

2.1. RPC

OZ+では、次の理由から確認型のプロトコルを採用した (図1参照)。

- ・ 応答側オブジェクトが、受信したメッセージを実行中に、要求側のオブジェクトにメッセージを送信したとき生じるデッドロックを防止する。
- ・ 応答側側で送信されたメッセージが実行可能かのチェックが必要である。

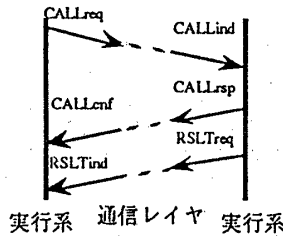


図1. 確認型RPC

2.2. グループオブジェクト

グループ化はグループオブジェクトという仮想的なオブジェクトを生成することにより実現される。一つのグループオブジェクトには複数のオブジェクトが対応しており、この対応する複数のオブジェクトは同じグループに属しているという。グループオブジェクトにメッセージを送信すると、そのグループに属する全てのオブジェクトにメッセージが送信される。

3. 同報通信の記述形式に関する問題点

通常のオブジェクト (以下、これを単一オブジェクトという) への要求とグループオブジェクトへの要求に関し、継承関係の整合性、及び返値の扱いに関する整合性を考慮した言語の記法はこれまでに提案されていない[7]。これらの整合性を実現するためにオブジェクト間通信の記述形式を次のように拡張した。

4. OZ+におけるオブジェクト間通信の記述形式

メッセージの応答側が単一のオブジェクトかグループオブジェクトかを意識しないプログラミングを可能にするために、OZ+ではオブジェクト間通信の記述として次の3通りの形式を導入した。

```
val = obj:meth(arg1, arg2, ...);      ....(1)
obj!meth(arg1, arg2, ...);          ....(2)
varArray = obj!:meth(arg1, arg2, ...); ....(3)
```

(1)を単一値型同期CALL、(2)を非同期分岐、(3)を配列値型非同期CALLという。

4.1. 単一値型同期CALL

要求側オブジェクトは全ての応答側オブジェクトにメッセージが届いたことを確認した後 (CALLcnf受信) 実行可能状態になるが、プロセスはメッセージの応答側から返値が1つ戻る (RSLTind受信) まで待ち状態となる。応答側objが単一のオブジェクトの場合はOZ+における通常のメッセージセンディングである。応答側objがグループオブジェクトの場合は、グループ化されたオブジェクトの1つから返値が戻るとプロセスは実行可能状態となる。残りのオブジェクトからの返値は要求側の実行系が破棄する (図2)。

4.2. 非同期分岐

要求側オブジェクト及びプロセスは、全ての応答側オブジェクトにメッセージが届いたことを確認した後、実行可能状態となる。この場合、CALLreqのかわりにTRNSreqパケットを送信することにより、応答側の実行系は返値の送信を行わない (図3)。

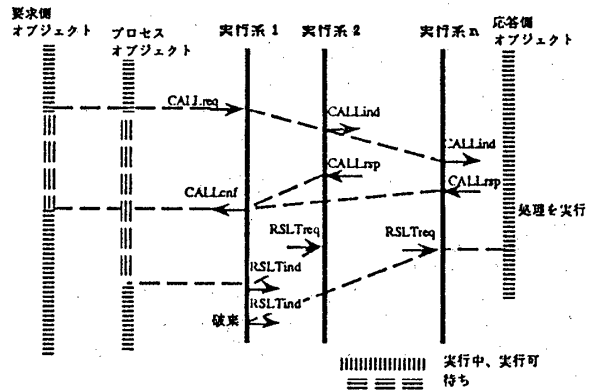


図2. 単一値型同期CALL

4.3. 配列値型非同期CALL

要求側オブジェクト及びプロセスは、応答側オブジェクトにメッセージが届いたことを確認した後 (CALLcnf受信)、実行可能状態となる。このとき、要求側の実行系はクラスvarArrayのインスタンス (以下varArrayオブジェクトと略記) を生成し、これを返値として要求側オブジェクトに返す (クラスvarArrayについては後述)。応答側オブジェクトからの返値はこのvarArrayオブジェクトに格納される。応答側からの返値を受け取るためには、

¹OZ+ : Object Oriented Open Distributed System — Programming Language Specification for Multicast Communication in OZ+ —

²Michiharu TUKAMOTO (Electrotechnical Laboratory)

³Hiroki SHINOHARA (MATSUSHITA Electric Industrial Co., Ltd.)

⁴Isao MIZUTANI (SUMITOMO Electric Industries, Ltd.)

⁵Hiroyuki KAZIURA (SHARP Corporation)

⁶Petit BERTRAND (EC Guest Researcher)

アプリケーション上で

```
val = varArray.get;
```

と記述する。(4)の記述を繰り返すことにより、複数の返値を受け取ることが可能となる(図4)。

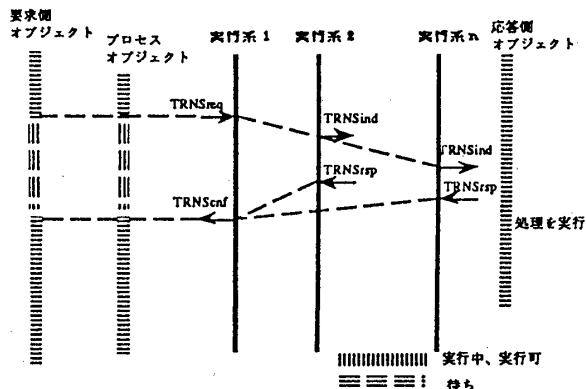


図3.非同期分岐

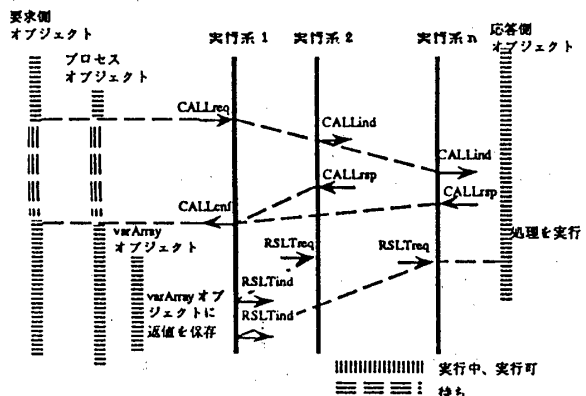


図4.配列値型非同期CALL

5. 実現方法

5.1. クラス varArray

クラス varArray は配列を内部に持つリスト型のクラスで、この配列内に応答側からの返値を格納する(図5)。インスタンス変数 callID は識別子で、実行系が割り当てる。

```
class varArray;
var next,array,callID,.....;
method get;
.....
end varArray;
```

図5. クラス varArray

5.2. varArray オブジェクトの管理

4.3 で生成される varArray オブジェクトは配列値型非同期CALLの要求側オブジェクトのセルに、varArray オブジェクトをリストにしてつなぐことにより管理する(図6)。以下、アプリケーションプログラムで val = obj;!meth; と記述し、これを実行したときの実行系の動きについて述べる(図6参照)。

- (1) 実行系は varArray オブジェクトを生成し、要求側のオブジェクトにつなぐ。
- (2) 実行系は Multicast RPC に対して CALLreq を行う。このとき生成した varArray オブジェクトの callID もいっしょに送る。
- (3) 応答側から CALLenf が返ってくる。実行系は CALLenf

の情報から応答側のオブジェクトの個数わかるので、その個数ぶんだけ配列を varArray オブジェクトに生成し、varArray オブジェクトをリターン値としてアプリケーションに返す。要求側オブジェクトとプロセスを実行可能状態にする。

(4) 応答側から RSLTind (返値) が返ってくると実行系は返値を varArray オブジェクトの配列に格納する。

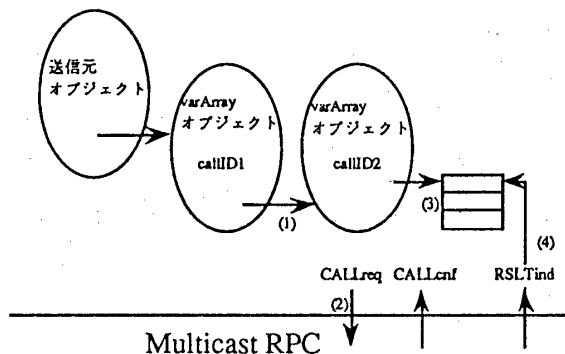


図6. varArray オブジェクトの管理

要求側のオブジェクトに varArray オブジェクトが2つ以上リストになってつながっている場合、上記の(3),(4)の処理をどの varArray オブジェクトに対して行えばいいかを判断するために CALLenf、RSLTind の情報の中に(2)で送信した callID を入れておく。応答側の全てのオブジェクトからの返値を格納した varArray オブジェクトはリストから外される。しかし、何かの事情で応答側から返値が戻ってこないとき、varArray オブジェクトは永久にリストから外せなくなる。そこで、実行系は varArray リストに対し特別なガベージコレクション処理を行い、このような varArray オブジェクトをリストから外す。

6. まとめ

OZ+における同報通信の記述方法について述べた。現在、実現中であり、この記述に従ったアプリケーション等は別の機会に報告する予定である。

なお、本研究は通産省大型プロジェクト「電子計算機相互運用データベースシステム」の一環として行われている。

【参考文献】

- [1] M.TSUKAMOTO et al.,The Architecture of Object-Oriented Open Distributed System: OZ, Interoperable Information Systems ISIIS '88,Ohmsha,PP153-166(1988,11)
- [2] 塚本他、OZ+:オブジェクト指向開放型分散システム-オブジェクトの分散管理-,情報処理学会1990年代の分散処理シンポジウム論文集、PP57-64(1990,11)
- [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-OZ+システム管理の基本設計-,情報処理学会第41回全国大会、6Q-3(1990,9)
- [4] 塚本他、OZ+:オブジェクト指向開放型分散システム-OZ+ステーション管理の実現-,情報処理学会第43回全国大会、1L-10(1991,10)
- [5] 塚本他、OZ+:オブジェクト指向開放型分散システム-OZ+におけるMulticast RPC-,情報処理学会第42回全国大会、3G-9(1991,3)
- [6] 塚本他、OZ+:オブジェクト指向開放型分散システム-グループオブジェクトの管理-,情報処理学会第43回全国大会、1L-8(1991,10)
- [7] E.C.Cooper: Programming Language Support for Multicast Communication in Distributed Systems,Proc of 10th.ICDCS(1990)