

オブジェクト指向に基づいた分散処理用ワークステーションの考察

武内春夫, 中沢克彦, 佐郷 司, 松下 温
(沖電気工業株式会社)

〔概要〕 ローカルネットワーク環境において、オブジェクト指向に基づくワークステーションの分散処理に対する有効性、問題点等を示す。

1. はじめに

本稿ではオブジェクト指向に基づくワークステーションのモデルを設定し、このモデルが分散処理環境において、どんな意味を持つか、その有効性と問題点を考察する。

オブジェクト指向システムとしては、Xerox の Smalltalk-80〔参照 1〕、CMU Hydra/C.mmp、Intel iAPX432、IBM System 38等がある。

Smalltalk-80 のオブジェクトは、メッセージを受け取ると対応するメソッドが起動され、一連の手続きが終了すると、その結果をメッセージの送り手に返す。つまり、メッセージの送信は、逐次的に行われる。並行動作の記述のためには、Process というクラスが特別に用意されている。オブジェクトを計算の単位とする計算モデルに、actor モデル〔参照 2〕がある。

オブジェクト指向の利点として次のものがある。

(1) 抽象化により、問題領域に近い概念的な見方ができる（問題領域と計算機構の間のセマンティックギャップの解消）ので、設計やプログラミングがわかり易く、生産性が高くなる。また、ユーザインタフェースの設計においても、ユーザフレンドリーなインタフェースが実現できる。これは、Xerox star ワークステーションのデスクトップインタフェースで実現されている。

(2) 定まった手続きを通してしかオブジェクトを操作できないように制限する（情報隠蔽）ことによって、オブジェクトの独立性が保証されており、より安全なシステムが構築できる。

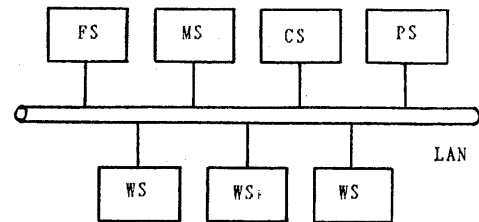
(3) メッセージ送信という統一的なメカニズムによって計算（処理）が行われるので、処理の見通しがよい。

本稿では、オブジェクト指向をネットワーク上に拡大して適用し、あるノード上のオブジェクトから他のノードにあるオブジェクトに対してメッセージ送信を行う事によって、分散処理を実現するようなモデルを考える。

2. オブジェクト指向ワークステーションのモデル

2.1 環境

ローカルエリアネットワーク（LAN）を考える。LAN 上には、複数のサーバーとワークステーションがあり、サーバーとワークステーションは共にオブジェクト指向によって動作するものとする。



FS: file server

PS: Print server

MS: mail server

CS: computing server

WS: workstation

図2.1

2.2 ワークステーションの機器構成

ローカルファイル、キーボード、ビットマップディスプレイ、マウス、音声入出力装置を備えているものとする。ビットマップディスプレイとマウスは、視覚的にオブジェクト指向のユーザインタフェースを実現する上で不可欠のものである。

2.3 オブジェクトモデル

Smalltalk-80のオブジェクトをベースとして、これを拡張する。Smalltalk-80のオブジェクトは次の性質をもつ。

- ・ オブジェクトは、共通のクラスにまとめられる。あるクラスに属するオブジェクトをそのクラスのインスタンスと呼ぶ。

- ・ クラス間にインヘリタンスの関係によって階層関係が定義される。インヘリトされるクラスの属性は、メソッド（手続き）、インスタンス変数、クラス変数である。

- ・ オブジェクトに対してメッセージ送信を行い、その結果を利用することによって、処理または計算が遂行される。

オブジェクトを次のように拡張する。

[1] 並行動作の導入

非同期のメッセージ送信を考える。

非同期メッセージ送信後、その結果を適当な時点で待つことができる。

非同期メッセージの受信者が、その結果を、メッセージの送信者に返すかどうかは自由である。

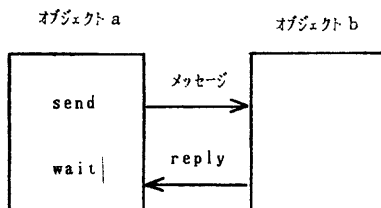


図 2.2

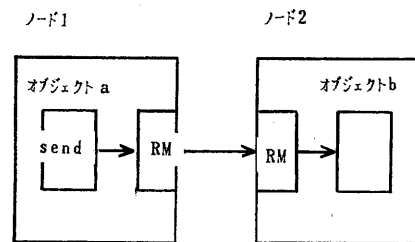
[2] リモートオブジェクトへのメッセージ送信

ネットワークワイドなリモートメッセージ送信（同期または非同期）を導入する。

メッセージの送信者はリモートメッセージ送信を、ローカルなメッセージ送信と区別する必要がないものとする（transparencyの保証）。

オブジェクト（クラス及びインスタンス）は、ネットワーク上で一意な識別子を持つものとする。

オブジェクトがどこに存在するかは各ノード毎にグローバルテーブルの中にもつ。グローバルテーブルの中にある時には、ネットワーク上の各ノードに問い合わせしてから、そのノード位置と識別子をグローバルテーブル中におくものとする。



RM: リモート通信機構

図 2.3

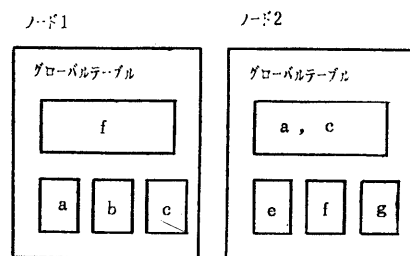


図 2.4 a, b, c, e, f, g: オブジェクト

[3] オブジェクトのマイグレーション

一般にオブジェクトは他のノードに移動させる事が可能である。

[4] オブジェクトの複製

一般にオブジェクトの複製を、他のノード上に作ることができる。

但し、複製は同じクラスに属する異なったインスタンスとして扱われる。一方を変化させても他方は変化しない。

[5] リモートメッセージ送信における引数

引数としてはオブジェクトの識別子が渡される。メッセージの受信者が、引数のオブジェクトを参照するためには、渡された識別子を使って参照するか、または、そのオブジェクトをマイグレートさせるか、する必要がある。

3. オブジェクト指向の分散処理に対する適用

分散処理の諸相に先程定義したオブジェクトモデルを適用してみる。

3. 1 ファイルシステム

ワークステーションから、ファイルサーバへのアクセスをローカルファイルへのアクセスと同じように行える (transparent)。

ファイルサーバを、ローカルファイルの自然な延長として用いる事ができる。

3. 2 リモートプロシジャコール

コンピューティングサーバの計算能力の利用が、ローカルなオブジェクトを利用するのと全く同じようにメッセージ送信によって行える。

計算の結果は、(同期メッセージを使用して)メッセージ送信と同期して、または、非同期メッセージを利用してメッセージ送信とは非同期に受け取ることができる。

3. 3 リモートリソースアクセス

リモートのプリンタやディスクをオブジェクトとして、メッセージ送信によって、ローカルなプリンタやディスクと同じように統一的にアクセスできる。

3. 4 データベース

トランザクションをメッセージ送信によって送る。データベースシステムはもともと、オブジェクト指向的なシステムであり、データベースは、データとアクセス方法が一体となって定義されている。

3. 5 ジョブ転送

一連の手続き (メッセージ送信の並び) をデータとしてメッセージ送信を行う。

受信者は、一連の手続きを解釈実行し、その結果を、送信者に返す。

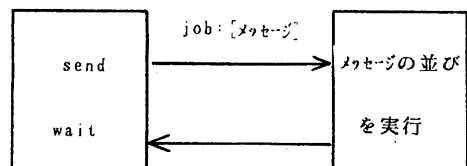


図3.1

3.6 security

オブジェクトの独立性は保たれているからsecurityは高い。

メッセージ自身の保護のためには、暗号化等の手段を講じる必要がある。

3.7 分散セマフォ

複数ノードにまたがって相互排除を実現するために、分散セマフォを用いる。

これを用いて、複数ノードからの要求を受け付ける待ち行列を実現する事ができる。

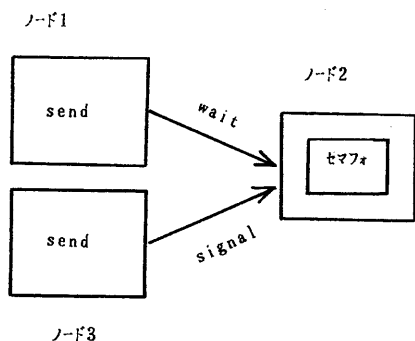


図3.2

3.8 マルチサイト・トランザクション

2フェーズコミットメントの手続きを用いて、複数ノードにまたがるトランザクションを実行できる。

トランザクションのアトミック性の保証が必要。

3.9 電子メール

電子メールシステムは、クライアント、メール、ポスト、メールボックスというオブジェクトの間のメッセージ送信によって処理が行われるようにモデル化することができる。

クライアント: ワークステーションのユーザ。

ポスト: メールの送信相手。

メールボックス: メールの受信者のためにメールが格納される場所。

メールサーバの機能は、ポスト及びメールボックスというオブジェクトの中に含まれている。

ユーザインタフェースの上では、ビットマップディスプレイ上に、メール、ポスト、及びメールボックスのアイコンが表示され、クライアントはユーザ自身である。アイコンを選択してコマンドを送ることによりメッセージ送信が行われる。

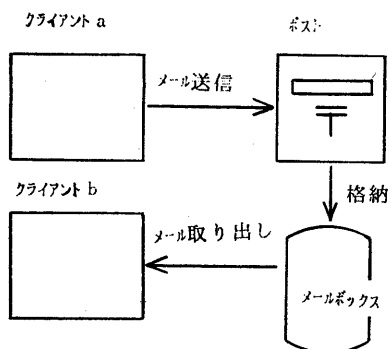


図3.3

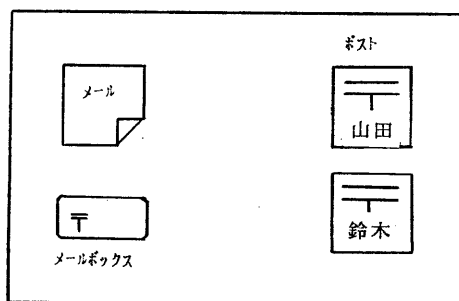


図3.4

3. 10

ユーザは、どのワークステーションからでもログインできる。ユーザ自身をオブジェクトとして見れば、ユーザがワークステーション間をマイグレートしたと見なせる。ユーザがアクセス権を持っているオブジェクトは、どのワークステーションからでもアクセスできる。

4. まとめ

オブジェクト指向を分散処理に適用する利点としては、

- ・ ユーザインタフェースが "対象に対して処理をする" という形であり、直観的でわかりやすいインタフェースである。
- ・ プログラミングが、問題領域の概念レベルで行える。

問題点としては、

- ・ 長大なデータの一括通信の手段が必要。メッセージ通信では細切れで効率が悪い。
- ・ トランザクション処理の機構（トランザクションのアトミック性の保証）が別に必要。
- ・ オブジェクトに対するアクセスの制限が必要である。

今後の課題として

- ・ オブジェクト指向言語を使用して分散処理のアプリケーションを実際に記述し、評価する必要がある。

参考文献

- (1) Goldberg, Adele. Smalltalk-80: The Language and Its Implementation.
- (2) Hewitt, Carl. "Control Structure as Patterns of Passing Messages." In Artificial Intelligence: An MIT Perspective Vol. II.
- (3) Schroeder, M. D. Experience with Grapevine. CSL-83-12 Aug. 1983 Xerox PARC
- (4) Peterson, J.L. and Silberschatz, A. Operating System Concepts. Addison-Wesley.