オペレーティング・システム 28－4
（１９８５．９．１３）

# 分散オフィスシステムにおけるプロセス管理

山崎　幸雄　　熊野　喜一　　筒井　健作　　石井　美恵
（日本電気株式会社）

## １．はじめに

情報処理機器（パソコン，オフコン，ＷＰ等）の小型高性能化・低価格化およびローカルエリアネットワーク（以下ＬＡＮと略記）技術の進歩と共に、これらを組み合わせた応用システムとしての分散システムが徐々に普及しつつある。しかし、このようなシステムの利用にはシステムの物理構成の意識を強いているのが現状であり、今後の一層の普及を促進するための解決課題となっている。

一方、このような分散システムの有力な適用の場であるオフィスでは、統合的なオフィスシステムが研究の段階から試用の段階、実証の段階へと着実に地歩を固めつつある[1,2,3,4]。しかし、同時に、大量に導入された情報処理機器が、多くは、相互に何の関係もなく個別的にしか使用されておらず、単にオフィス作業の限定的な「機械化」のレベルのサポートに止まっている[1,2]のも現実である。さらに、オフィスにおけるサポートの質と範囲を拡大するためには、上記のような分散システム構築技術により，個別的に利用されている情報処理機器をＬＡＮを用いて有機的に接続し、ボトムアップ的に全体として統合的なオフィスシステムとして構築することが考えられる。このためには、ＬＡＮの種類や接続形態、各構成ノードの機種や搭載ＯＳ等のシステムの構成から独立した高レベルの利用環境の実現が一層強く要請されることになる。

以上のような要請に応えるべく、オフィスを対象とした分散システム環境のもとでの高レベル利用環境の検討を進めている[5]。このシステムはＬＡＮ上のノードがそれぞれ定められた機能を担う機能分散型のオフィスシステムである。ネットワーク上にユーザアプリケーション（以下ユーザＡＰと略記）の、オフィスに密着した運用環境（オペレーティングエンバイロンメント：ＯＥ）を構築することを目的としている。この運用環境の論理構造はオブジェクト指向の概念に拠っており、オブジェクト間の相互作用は，各ノード上に分散して存在するプロセス群により実現される。

本稿では、オブジェクト指向処理モデルのプロセス群による実現手法について述べる。特に、オブジェクトのプロセスへの対応付け、すなわち、オブジェクトクラスとプロセスとの対応、オブジェクト間メッセージとプロセス間メッセージとの対応、および分散環境下におけるプロセス間通信のためのプロセス管理方式について記す。

## ２．運用環境（ＯＥ）

情報処理機器がオフィス内に普及・浸透し、その主利用者が専門オペレータから一般のオフィスメンバになって来ている。しかし、従来ＯＳのサポートする機能体系およびユーザインターフェースでは低レベルすぎてオフィスには馴染まない。運用環境とは、概念的には図１に示すように、ユーザＡＰとＯＳの間にあっ

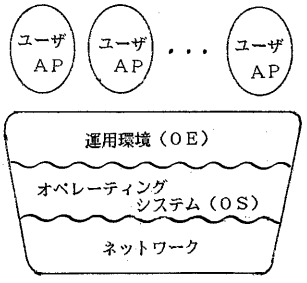


図１．運用環境（ＯＥ）

て、ユーザＡＰ運用のためのＯＳとネットワークから独立し、オフィス環境を反映した統一的かつ高レベルの環境を与えるものである。

オフィスシステムにおけるこの運用環境が具備すべき要件としては以下のものがある。

（1） ユーザインタフェース管理
- ウィンドウ管理
- グラフィックス
- ヘルプ／チュートリアル

（2） ユーザＡＰ実行管理
- ユーザＡＰ実行環境管理
- ユーザＡＰ実行制御
- 移出／移入

（3） 仮想オフィス管理
- オフィスディレクトリ管理
- メール／ファイル機能

（4） 導入／運用／ユーザＡＰ開発支援

## 3．オブジェクト指向モデル

上記のような運用環境にアプローチするに際し、オブジェク指向の概念モデルを採用した。オブジェクト指向アプローチは、実装上の問題を別にすれば、並列性の高いオフィス情報処理システムの構築上利点が多いとされている[6,7,8]。これらの利点としては以下のものが考えられる。

（1）処理とインタフェースの統一性
オブジェクトの外部仕様（メッセージインタフェース）と実装が分離されているので異機種、異ＯＳ間結合が容易であり、実装上の見通しが良い。

（2）不必要な詳細情報の隠蔽
オブジェクトの物理的な存在位置や使用資源といった詳細情報が隠される。

（3）独立性／並列性
システム全体が一様にオブジェクトの集合として記述され、他に影響を与えずオブジェクトの追加、抹消が可能である等、機能拡張や変更に強く、また高度の並列性を持つ。

本システムでは、システム内に存在するすべての資源（ファイル、ディレクトリ、ユーザＡＰ等）は、その位置や形態にかかわらず、オブジェクトと呼ぶ論理的エンティティとして統一的に扱われる。オブジェクトはシステム空間でユニークな名前（オブジェクトＩＤ）を持ち、相互にメッセージを送り合うことにより相互作用を行う（図２）。

オブジェクト指向アプローチでは次の機能が基本となる。

- オブジェクト管理：オブジェクトの生成、消滅、コピー、保存等
- メッセージ管理　：オブジェクト間のメッセージ交換

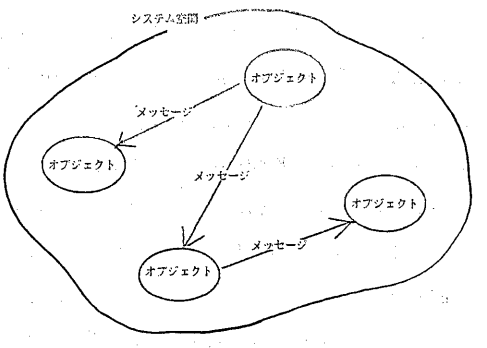


図２．オブジェクト指向モデル

## ４．実現形態

### ４．１　物理構造

上記のオブジェクトおよびそれらの相互作用は、分散化した処理環境のもとで実現される。すなわち、ネットワークの各ノード上のプロセス群によってオブジェクト管理およびメッセージ管理が実現される。

本システムの物理構造を図３に示す。ネットワーク上のノードは物理的な分散の単位であり、ユニークな名前（ノードアドレス）を持つ。各ノードには１つのスーパバイザプロセスと一般には複数のユーザプロセス（以下、単にプロセスとも称す）が存在する。スーパバイザプロセスは、ユーザプロセスの起動・停止といった実行管理機能と、ユーザプロセス間のメッセージ交換機能を提供する。各プロセスは機能分散の単位であり、システム内でユニークな名前（プロセスＩＤ）を持つ。各プロセスにはそれぞれ固有のファイル環境を持ち、１つの処理環境を構成する。このように、プロセスとそのファイル環境を結び付けた処理環境全体もプロセスＩＤで識別する。

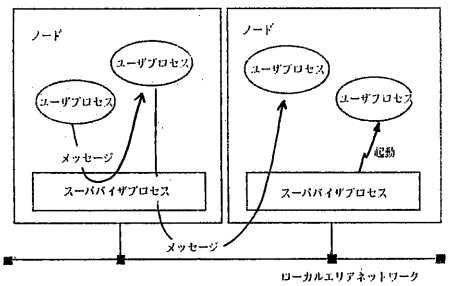


図３：物理構造

### ４．２　論理構造の物理構造へのマッピング

オブジェクトは、その動的側面（メソッド）をプロセスにより、静的側面をファイル環境内のディレクトリないしファイルにより実現される。また、オブジェクトとプロセスが一対一に対応するのではなく、一般に、複数のオブジェクトクラスに対する処理がプロセスにより実現される。プロセスがどのようなオブジェクトクラスの範囲に対して処理環境を与えるかによりそのプロセスが担う機能が定まる。

オブジェクトを識別するオブジェクトＩＤは、プロセスＩＤおよびプロセス内ＩＤによって構成される（図４）。すなわち、オブジェクトＩＤを構成するプロセスＩＤはその処理環境を識別し、プロセス内ＩＤはその処理環境内におけるファイル環境のディレクトリないしはファイルとの対応を与える。

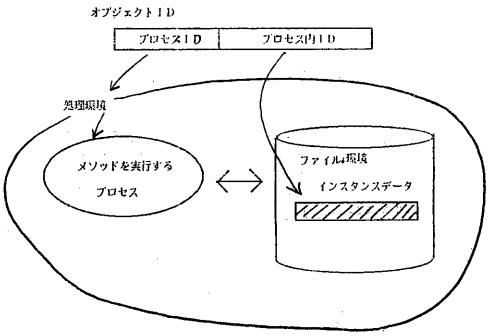


図４．オブジェクトＩＤと処理環境の対応

プロセス内ＩＤは、さらに、そのオブジェクトの属するクラスのＩＤとクラス内の一連番号とから構成される。したがって、プロセスはオブジェクトＩＤとメッセージセレクタが与えられると実行すべきメソッドを同定することができる。図５はオブジェクトＩＤおよびオブジェクト間メッセージの一般形式を示している。

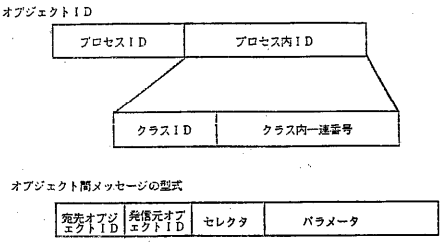


図５．オブジェクトＩＤとメッセージの形式

## 4．3　メッセージ交換

オブジェクト間のメッセージ交換は、プロセス間のメッセージ交換として実現される。オブジェクト間メッセージとプロセス間メッセージとの対応関係を図6に示す。すなわち、同一処理環境内でオブジェクトがメッセージ交換を行う場合は、オブジェクト間メッセージの形式がそのまま使用され、異処理環境内のオブジェクトの場合は、プロセス間通信用の付加情報が付けられ送信される。また、異ノード間に跨がる場合には更にノード間通信用の付加情報が付けられる。

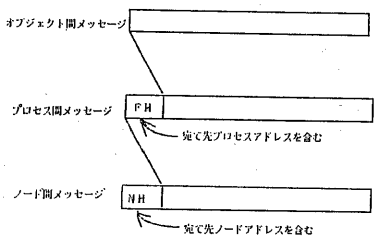


図6．メッセージの対応

分散環境のもとでのメッセージ交換を図7で概念的に示す。論理的な単位であるオブジェクトは、物理的な単位であるプロセスの中に一般に複数含まれる。また、オブジェクト間メッセージ交換はシステム内に分散するプロセス間の通信として実現される。

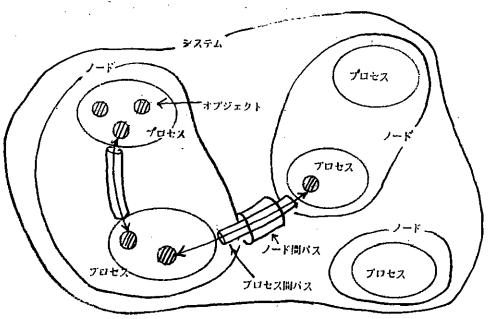


図7．分散環境におけるメッセージ交換

プロセス間通信のモデルを図8に示す。プロセスはそれぞれ専用の受信用メッセージキューを介してメッセージを受理する。また、他のプロセスへのメッセージは自ノードのスーパバイザプロセスに送信する。スーパバイザプロセスは、自ノードならびに他ノード上のプロセスに関する情報（ネットワーク定義テーブル：ＮＤＴ）を保持し、キュー内のメッセージの宛先により自ノードの場合は該当プロセスに、他ノードのプロセスであればネットワークドライバを介してそのノードのスーパバイザプロセスに転送する。

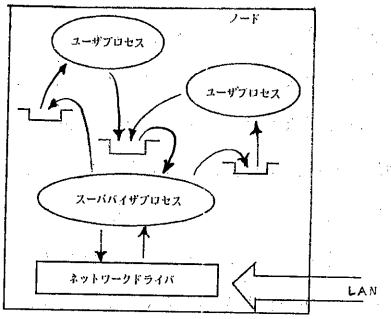


図8．メッセージ交換モデル

## 4．4　プロセス管理

プロセスは図9が示すようなクラスに分類される。

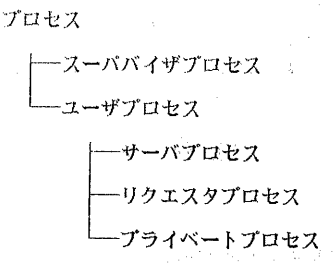


図9．プロセス体系

各プロセスクラスは次のような機能を持つ。

・スーパバイザプロセス

は各ノードに常に一個存在し、ユーザプロセスの管理およびプロセス間のメッセージ交換を行う。

・ユーザプロセス

スーパバイザプロセスの管理下にあり、種々の機能を提供する。一般に各ノードに複数存在する。

・サーバプロセス

ＮＤＴ内に定義された”システム”プロセスでありシステムの標準的な機能（サーバ機能）を提供する。各ノードに常駐する。

・リクエスタプロセス

他のプロセスに対し逐次処理要求を発する。

・プライベートプロセス

他のプロセスからの要求より起動される一時的プロセスであり、その存在は起動を行ったスーパバイザプロセスおよび起動要求を出したプロセスしか分からない。

スーパバイザによるプロセス管理の概要を述べる。

（１）　起動時処理

ノードが立ち上げられるとスーパバイザプロセスが起動され、ＮＤＴを読み込み、自ノード上に存在すべきプロセスを起動する。（ＮＤＴは、システム生成時に作成されるテーブルで、システムに標準的に存在するプロセスを定義する。ＮＤＴは、プロセスに対し、存在すべきノードのアドレスおよびそのプロセスの起動に必要な情報を保持する。）

また、起動したプロセスに関する情報を、プロセス管理テーブル（ＰＭＴ）に登録するとともに、同報メッセージとして他ノードに送る。起動されるプロセスは、そのノードがサーバの場合はサーバプロセスが（複数個、複数種類も可）、ワークステーションの場合はリクエスタプロセスであるデスクトップマネージャ（後出）である。

（２）　プロセス起動・終了処理

自ノードのリクエスタプロセスないしは他ノードのスーパバイザプロセスの要求に応じて必要なユーザプロセスの起動・終了を行う。結果は、ＰＭＴに登録するとともに要求元に応答する。

（３）　プロセスのステータス管理

ユーザプロセスはサービス可（ＲＥＡＤＹ）、サービス中（ＢＵＳＹ）、サービス不可（ＤＯＷＮ）の３つの状態を取る。スーパバイザプロセスは、自ノード内の管理対象ユーザプロセスの状態変化をＰＭＴにより管理する。また、他ノードのユーザプロセスの状態については、起動要求に対する応答メッセージあるいはそのノードからの通知メッセージにより把握し、ＰＭＴに反映させる。（ＰＭＴは、各プロセスに対しその存在ノードアドレス、状態、メッセージ送信のためのアソシエーション情報を保持する。）

## ５．運用環境の実現

オフィスでのユーザＡＰを対象とした運用環境の基本機能が、今まで述べてきたサーバプロセスないしリクエスタプロセスにより実現される。

次の機能がサーバプロセスとして実現される。

・オフィス管理サーバ

システム内に一個存在して、システム環境（システムの物理構成、ソフトウェア構成）、オフィス環境（オフィスとオフィスメンバの対応）、利用環境（各オフィスおよびオフィスメンバのファイル／メール環境）の運用・管理を行う。

・ファイルサーバ

キャビネット、フォルダといった階層的な分類体系による、文書の蓄積・検索機能を提供する。

・メールサーバ

オフィスおよびオフィスメンバのメールボックスを管理し、同報・配達日指定・親展・代行受信等のメールサービス機能を提供する。

・アプリケーションサーバ
ユーザＡＰの実行環境、起動のための情報を蓄積・管理する。
・コミュニケーションサーバ
マルチネットワーク環境におけるコミュニケーション機能を提供する。

リクエスタプロセスは次の機能を実現する。

・デスクトップマネージャ
利用者（オフィスメンバ）に対応して存在し、利用者の作業環境（論理的な机）を提供する。
・アプリケーションマネージャ
あらかじめ登録された手順に従い、ＡＰの起動およびイベントの監視を行う。利用者の要求によって起動されバックグランドで処理を行う。

プライベートプロセスは、以上のような環境下で運用されるユーザＡＰに相当し、ワードプロセッサ、スプレッドシート、ビジネスグラフ、パーソナルＤＢといった机上ツールの他に、帳表処理等の業務アプリケーションも含む。

## ６．おわりに

ローカルエリアネットワーク上の複数ノード（ワークステーションおよび各種サーバに相当）から成る機能分散型のオフィスシステムの基本方式について報告した。オブジェクト指向の概念に基づくアーキテクチャを設定しており、現在、具体的な実装方式の検討を進めている。

オブジェクト指向アプローチのベースとなるオブジェクト管理およびメッセージ管理は、各ノード上に分散して存在するプロセス群により実現される。オブジェクト指向アプローチによる処理モデルを実装する場合、この処理モデル固有の制御構造によりプロセスの構造および処理方式が強く規定され、オーバヘッドの増加要因となると考えられる。反面、モジュール独立性が高まるとと部品化が容易になることで機能の変更や追加に強くなり保守性、拡張性が高まることが期待される。

## 参考文献


[1] Ellis,C. and Nutt,G.,"Computer Science and Office Information Systems", ACM Computing Surveys, 12(1), (March 1980).

[2] Hammer,M. and Sirbu,M.,"What is Office Automation?", Office Automation Conference, Georgia, (1980).

[3] 水野、渡辺、小林「オフィス情報アーキテクチャ」オフィスオートメーション、Vol.4,No.4（昭５８．１１）.

[4] 金森、山崎、宮本、今井「統合オフィスシステム”アラジン”の分散処理方式とその適用」情処学会ＬＡＮ／マルチメディアの応用と分散処理シンポジウム、（昭５９．１０）.

[5] 筒井、石井、熊野「分散オフィスシステムにおけるプロセス管理方式」情処学会全国大会（昭６０．９）.

[6] Ahlsen,M., Bjornerstedt,A., Hulten,C. and Soderlund,L. "An Architecture for Object Management in OIS," ACM Trans. on Office Info. Sys., Vol.2, No.3 (July 1984).

[7] Nierstrasz,O.M.,"An Object-Oriented System," in Office Automation Concepts and Tools, ed, D.C. Tsichritzis, Spring-Verlag Berlin Heidelberg (March 1985).

[8] Tsichritzis,D.C.,"Objectworld," in Office Automation Concepts and Tools, ed, D.C. Tsichritzis, Spring-Verlag Berlin Heidelberg (March 1985).