

オブジェクト指向に基づいた分散処理OSの設計と実現

6V-5

高橋 薫 太田 佳行 白鳥 則郎 野口 正一
(東北大学 電気通信研究所)

1. はじめに

本稿では、理解性および保守の容易性を向上させる観点から、オブジェクト指向の考え方に基づいたネットワークOS(分散処理OS)の設計法を提案する。また、その設計法に基づいたLAN向きネットワークOSの設計と実験の実現を与える。本設計法の特徴は、ネットワークOSが問題とする領域に存在する自然な対象を明確に設計上の要素として反映させることにある。従って、システム全体は従来のような実現手段としての手続き的な過程の観点からではなく、独立な対象の集まりとして表現されることになるため、理解性が向上する。

2. ネットワークOSの設計法

ネットワークOSの目的は、リソースの地理的位置などに依存しない一元的な分散リソースの操作環境をユーザやアプリケーションプログラムに提供することである。本稿では、ネットワークOSを、“分散リソース操作環境を既存のOSの上に付加機能として組み込むことによって構成したシステム”、として定義する。この時、付加機能部の移植の容易性を達成するために、既存のOSと付加機能との間に“インターフェース”を設ける。

最初に述べたように、本設計法ではオブジェクトを主体としてシステムを設計することによって、その構造を明確化する。

[オブジェクト]

外部からおよび外部へのアクションによってその振る舞い(性質)が特性づけられる独立で自律的な実体であり、問題領域中から抽出される。

オブジェクト(の集まり)はその抽象化・複雑さの度合いに応じて、論理的な階層化を行うことができる。また、オブジェクトの計算機システム上への実現に際しては、オブジェクトを“プロセス”に対応づけて考える。さて、オブジェクトはそれが属するクラスによってそれぞれ分類することができる。つまり、性質を共有するオブジェクトの集まりは“クラス”としてまとめることができる。そして、各オブジェクトはそれが属するクラスの“インスタンス”となる。従って、プロセスを次の2つに種別化する：(1)クラスプロセス---共通の性質を持つオブジェクトをまとめたものであり、役割としては、クラスに属する個々のオブジェクトをインスタンスプロセスとして生成する。(2)インスタ

ンスプロセス---クラスプロセスによってインスタンス化されるプロセスであり、そのクラスに属する個々のオブジェクトを表す。

上で述べた基本的考え方を基にして、ネットワークOS(分散リソース操作環境)の設計手順を次のように与える。

[ネットワークOSの設計手順]

(ステップI : アーキテクチャの設計)

アプリケーションプロセスから成る最上位階層から出発し、分散リソース操作環境をその抽象化・複雑さの度合いに応じて、最下位の物理層までのN-階層に階層分割する。

(ステップII : 階層内コンポーネントの設計)

- 1.各階層の要素(オブジェクト)を決定。
- 2.オブジェクト間の関係を明らかにする。
- 3.各オブジェクトに許されるアクション(自ホスト・他ホストからの)を決定する。この時、クラスおよびインスタンスプロセスの役割を分離。
- 4.各オブジェクトの実現方法を設計する。

(ステップIII : インターフェースの設計)

分散リソース操作環境の携帯性を達成する為、既存のOSとの間にインターフェースを設ける。

- 1.ステップIIで定められたオブジェクトをプログラミングするのに必要とされるプリミティブな命令(システムコール)を、実現される既存のOSには依存しないように決定する。
- 2.III-1で決定された命令を、それに対応する既存OS上の命令に展開するための機能を設計する。

3. LANを対象としたネットワークOSの設計

本節では上記の設計法に基づいてLAN向きのネットワークOS(分散リソース操作環境)を設計する。

(アーキテクチャの設計)

図1のように分散リソース操作環境を、アプリケーション層、プリミティブ層、通信層、物理層の4階層に構造化する。

(階層内コンポーネントの設計)

アプリケーション層：この層は、プリミティブ層のオブジェクト(プロセス)を活用し、ユーザに適切なアプリケーションを提供する。本設計ではアプリケーションの対象となるリソースとして

Design and Implementation of a Network OS Based on the concept of an Object

Kaoru Takahashi, Yoshiyuki Ohta, Norio Shiratori and Shoichi Noguchi

Tohoku University

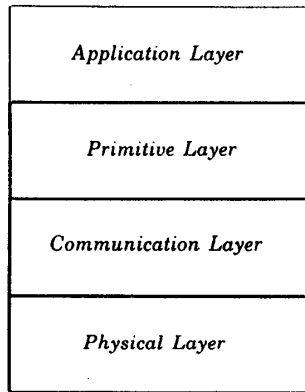


図1 Architecture of distributed resource access environment

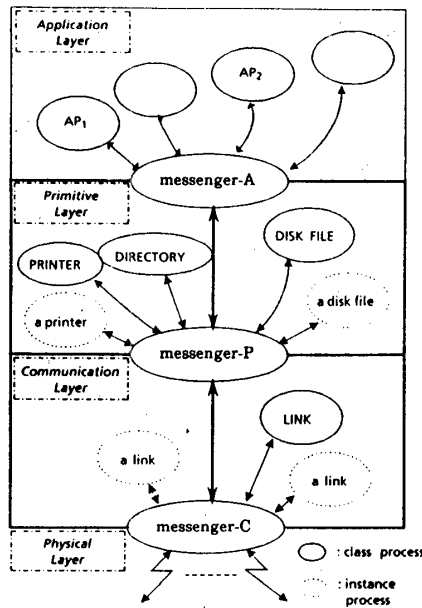


図2 Whole structure of distributed resource access environment

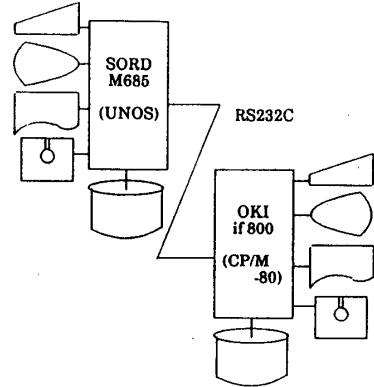


図3 Configuration of experimental system

は、ファイルを想定する。本設計では、以下に示すようなコマンド(そのコマンドを実行するプロセス)を与える。

- (1) ncp --- ファイル内容のコピー
- (2) ncat --- ファイル内容の表示
- (3) nls --- ファイルの一覧
- (4) nch --- ファイルのパーミッションの変更
- (5) nrm --- ファイルの削除

プリミティブ層: この層は、アプリケーションプロセスによって使用されるリソースを表すオブジェクトから成り、アプリケーションプロセスに対してリソースの内部の表現構造や地理的位置を意識させないアクセス環境を提供する。具体的なオブジェクトはファイルであり、この中で特にディスクファイルを考える。ディスクファイルに関連してディスクファイル・クラスプロセスとディスクファイル・インスタンスプロセスがこの層に組み込まれる。また、ファイルリソースの情報(ファイル位置、状態、所有者名等)を表すオブジェクトとしてディレクトリが必要であり、ディレクトリ・プロセスも組み込まれる。

通信層: この層の役割は、プリミティブ層のプロセスが他のホストのプリミティブ層のプロセスと通信を行う際の通信環境を提供することにある。本設計では、OSI参照モデルとの整合性を踏まえ、このような通信は“バーチャルサーキット”の考え方に基づくこととし、バーチャルサーキットを抽象化したオブジェクトである“リンク”によって通信層を構成する。

具体的なプロセスも含めた分散リソース操作環境の全体構造を図2に示す。

```
>nls takahashi
[ file name || user name || place || date || perm || status ]
test      takahashi  OKI   86/4/30  r w
sfileA    takahashi  SORD  86/4/30  r . r
sfileB    takahashi  OKI   86/5/1   r w r w
sfileC    takahashi  SORD  86/5/1   r w r w
>nrm
(Usage)   nrm < file name >
          nrm < user name ><< file name >
>nrm sfileC
>ncat test
An object is a self-contained and autonomous entity whose
behaviour is characterized by the actions it undergoes and
those which it requires of other objects
```

図4 Execution examples of applications

4. ネットワークOSのインプリメンテーション

前節で与えた分散リソース操作環境の論理設計を具体的に実験システム上にインプリメントした結果を述べる。本実現においてはプログラミング言語としてはCを用いた。実験システムの構成を図3に示す。本実験システムは論理設計の有効性を確かめることを主目的としている為、実現は図中のSORD M685上にのみ行った。他方、OKI if-800は搭載しているOSの制約から、マニュアル操作によって実験の上で必要な種々の処理を行った。

以上のような実現形態の下で、実際にアプリケーションを駆動した結果の例を図4に示す。この例はユーザが、ファイルの一覧、ファイルの削除、ファイルの表示、を行った結果を示している。

5. むすび

本設計法の特徴は以下の点に要約される。(1) 個々の対象の明確な反映、(2) 対象の抽象化、(3) 理解性の向上、(4) 移植性の向上。