

## 1 K-1 MVCモデルに基づく分散システム開発支援ツール

東京工業大学 ○山本 学 原 辰次

## 1 はじめに

ネットワーク機能を用いて、複数計算機を利用する分散システムの1つに、同レベルにある複数の独立したアプリケーションの連系により、動作するものがある。このようなシステムは、ウインドウシステムの利用により、ユーザがアプリケーションの連系を自由に変更できる柔軟性および拡張性に優れているなど多くの利点がある。しかし、そのようなアプリケーションの入出力インタフェースは、一般に非常に複雑になる。特に、ネットワーク通信のインタフェースは、通信における他アプリケーションの対応など、通信のための多くの処理が必要となる。ところが、これらの処理はアプリケーション本来の機能であるデータ処理とは関係のないものである。したがって、アプリケーションを、アプリケーション間の結合などの通信メッセージ管理部とアプリケーション本来の役割であるデータ処理部の2層に分けることにより、その開発を容易にすることができる。

本稿では、このような考えに基づいて、著者らがUNIX上で開発した、ネットワーク通信支援ツールNETlibと、非同期に発生するイベントを処理するために、Smalltalk-80のMVCモデルを用いたアプリケーション開発支援ツールMVclibについて報告する。

## 2 通信支援ツールNETlib

分散システムの開発では、通信速度、信頼性、通信メッセージの管理機能など多くのことに注意を払う必要がある。通信速度、信頼性は通常トランスポート層以下の問題であるので、ここでは、通信メッセージの管理を考える。

通信メッセージの管理は、構築すべき分散システムにより、定義されるものである。また、これらはアプリケーションの中に実際に組み込まれるために、アプリケーション開発に直接影響を与えるが、そのアプリケーションのデータ処理には関係ない。したがって、通信機能のカプセル化が必要である。著者らがC言語で開発したNETlibは、通信メッセージの管理を行な

い、通信機能をカプセル化するためのツールである。

NETlibは、1) 動的なコネクション型サービスとその結合情報管理、2) 任意のアプリケーションからの非同期なメッセージ受信、3) 受信メッセージに対する処理の実行、4) メッセージ送信を提供する。これらのサービスは、通信管理マネージャを生成することにより起動される。このマネージャは生成時において、管理に必要な専用の通信ポートと、そのポートを表す大域的に唯一のアドレスを割り当てる。

通信メッセージはNETlibによって管理されるプロトコルをTCP/IPプロトコルを用いて転送される。通信データと、そのデータの結合情報の管理は、マネージャにそのデータの使用を宣言することにより行なわれる。1つのデータは動的に、複数の送信、受信アプリケーションのデータと結合できる。この結合は明示的に切断されるまで保存される。しかし、これは、相手のアプリケーションにより切断されることもある。

データのネームは、アプリケーション内において局所的に定義される。このネームはマネージャのアドレスと組み合わせられることにより、システム内で唯一となる。

受信されたメッセージは、その型に対し定義された手続きを呼び出すことにより処理される。この手続きは、プログラマが定義できるものである。

NETlibはアプリケーションの結合情報などのシステム構成に関するサービスは提供しない。これは、構築すべきシステムに依存するからである。しかし、ネームサーバ的な結合管理アプリケーション構築のサービスを提供する。

## 3 開発支援ツールMVclib

ネットワーク通信機能は、前節で述べた通信機能のカプセル化により、データ処理部との分離を図れた。しかし、通常、アプリケーションは通信機能とともに、アプリケーションユーザによる直接入力がある。特に、ウインドウシステムを利用した入力を用いたアプリケーションでは、ウインドウ、ネットワーク両方の非同期

入力を確認する必要があり、これは、アプリケーションの複雑化を招く。

しかし、ウインドウ、ネットワークからの入力はデータ処理部にとっては入力の発生という点において共通であり、この入力をイベントとしてとらえることができる。このようなイベントを処理するアプリケーション開発法に MVC モデル [1] がある。これは、Smalltalk-80 で用いられている概念であり、データ処理、出力、入力を Model、View、Controller と呼ばれるオブジェクトに分割し、それらの依存性を DependentsFields と呼ばれるオブジェクトが管理するものである。

Smalltalk-80 では、Controller、View はウインドウ入出力に対するオブジェクトであったが、ネットワーク通信入出力を扱う機能を持つ Controller、View を定義することにより、データ処理を行なう Model に対し、入出力インタフェースの透明化を図ることができる (Fig.1)。

この考えに基づき、著者らは、MVC モデル環境を提供するツール MVClib を C 言語で開発した。これは、C 言語で記述されているが、オブジェクト指向的であり、クラス階層を持つ [2]。Fig.2 にクラス階層を示す。

CoreModelClass、CoreViewClass、CoreCtrlClass は、Model、View、Controller の最上位クラスであり、これらのクラスが MVC の動作に必要な機能を持つ。

Controller、View のクラスは大きく PrivateClass と GeneralClass に分けられる。前者は、特定の Model のクラスのインスタンスに対する処理をするクラスであり、後者は Model を限定しないクラスである。

WidgetCtrlClass、WidgetViewClass は Xwindow システムの Xtoolkit Widget を用いてウインドウ入出力を処理する。また、NetCtrlClass、NetViewClass は NE-Tlib を用いて、ネットワーク入出力を処理する。TransViewClass は同一プロセス内の 2 つの Model 間の通信機能を持つ。

これらのオブジェクトは、プログラマが設定するユーザーデータ領域を持つ。ユーザーデータの管理はプログラマの責任であり、オブジェクトに管理メソッドを定義することにより行なう。Model、View、Controller 間のユーザーデータの依存性を DependentsFields が管理する。これは Smalltalk-80 における MVC と異なる点であるが、これにより、新たなクラスを定義せずに MVC の環境を利用できる。また、各オブジェクトの動作は

プログラマは全く意識する必要がない。

ウインドウ、ネットワークの入出力の管理は、管理クラスの定義により行なっているため、新たなインタフェース機能の追加も、それを管理するクラスを定義することにより容易に行なえる。

#### 4 おわりに

本稿では、分散システムのアプリケーション開発に MVC モデルを導入することにより、その開発が容易になることを述べ、そのような環境を開発するための支援ツール MVClib を紹介した。現在は、Fig.2 で示したクラスだけであるが、その利用環境に合わせたクラスを定義することにより、より利用し易いものとなる。著者らは、現在このような環境で制御系 CAD を開発中である。

#### 参考文献

- [1] 春木良且: オブジェクト指向への招待、啓学出版、1989
- [2] D.A.Young、川手恭輔訳: XToolkit プログラミング、トッパン、1990
- [3] M.Sloman、J.Kramer 齊藤忠夫監訳: 分散システムと計算機ネットワーク、丸善出版、1988
- [4] SunTechJournal(日本語版)、vol3、日本サンマイクロスシステムズ、1990

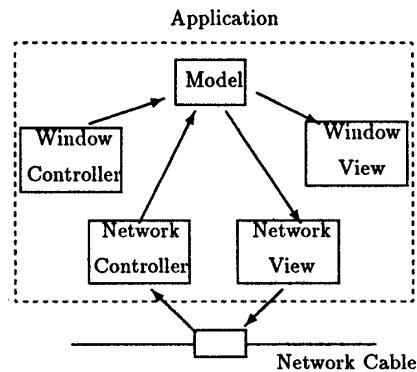


Fig.1 MVC モデル

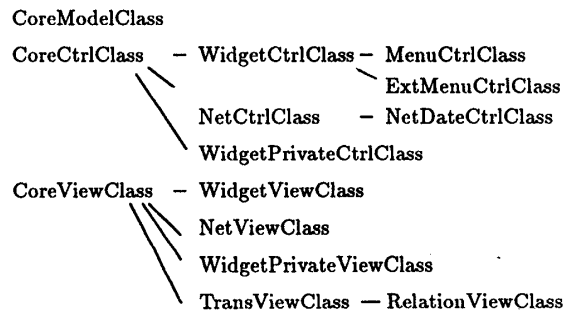


Fig.2 クラス階層