

「画像電子メール/掲示板システム」の開発(その2)

6W-5

—ユーザインタフェース機構—

松野年宏、前田靖子、石川晃、小泉正彦、本池祥子

松下電器産業株式会社

1. はじめに

本稿では、画像電子メール/掲示板システムのユーザインタフェース機構について述べる。ここでの目的は、アプリケーションの操作対象の実体と表示内容との対応を一元的に管理することにより、GUI部と機能実現部の各々の独立性を高める試みを報告することである。

1. GUIを含むプログラム開発の問題点

GUIを含むアプリケーション開発では、GUI部の開発コード量が機能実現部のコード量に匹敵する大きさとなることしばしばである。しかも、一般にGUI部はプログラムの完成後もエンドユーザの評価などに基づいて作り替えを繰り返されることが多い。このため、GUI部の開発、変更作業の効率の向上、保守性の向上はアプリケーション全体のそれに直接影響を与える重要な課題である。

最近では、GUI部の画面設計を支援するツールが使われることが多いが、表面的な画面レイアウトの作成、変更作業は簡単化することができても、コールバックにおける表示内容の管理という本質的な問題についてプログラム開発の手間を軽減するものではない。GUI部で扱うウィンドウでの表示内容と機能実現部で最終的に対象とする実体とが複雑に関係しあっているため、GUI部での表示内容管理が煩雑となる。機能実現部での実体操作による表示変化の影響を正しく反映させるためには、GUI部でアプリケーション固有の状態や実体間の関連を判断しなければならない。プログラム構造を簡潔にし、開発、保守効率を高める上では、GUI部の機能実現部からの独立性が確保されることが要求される。

画像電子メール/掲示板システムでは、GUI部での表示内容と機能実現部での操作対象との関連を一元的に管理するオブジェクト管理テーブルを導入することによって、GUI部と機能実現部のそれぞれの独立性を高めることができた。以下では、このオブジェクト管理テーブルの構造と、その導入によるGUI部での効果について述べる。

2. オブジェクト管理テーブルの構造

GUIを含むアプリケーションでは、アプリケーションが最終的に操作対象として扱うファイル、ディレクトリといった実体の内容がいくつかのウィンドウに分散して表示されている。そして各実体はアプリケーション固有の意味づけにより相互に関連しあっている。したがって、この実体間の関連に基づいて、ユーザ操作による実

体の属性変更、削除等により複数のウィンドウの表示内容は複合的に変化されなければならない。オブジェクト管理テーブルは、図1に示すようにGUI部が扱うウィンドウと機能実現部が扱う対象である実体までを、

- a) ウィンドウオブジェクト
- b) 表示オブジェクト
- c) 実オブジェクト

の3つの階層のデータを介して結びつける。

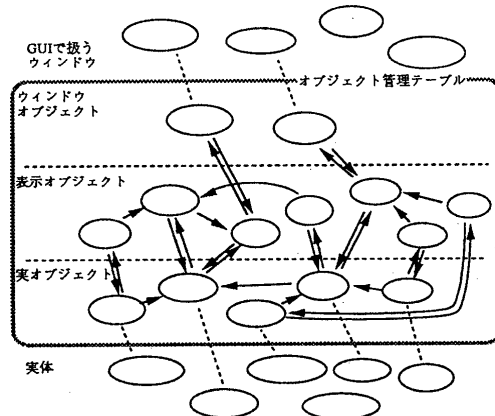


図1 オブジェクト管理テーブルの概念図

最上層のウィンドウオブジェクトはGUIで扱うウィンドウと1対1に対応している。ウィンドウオブジェクトでは、GUI部で扱うウィンドウおよびウィンドウ内のGUI部品と、その表示データを保持する表示オブジェクトの対応が管理される。

最下層の実オブジェクトは、ファイルやディレクトリといった実体に対してアプリケーションとして固有の意味を持たせる。たとえば、あるディレクトリにアプリケーションとしてファイルのフォルダの役割をさせるときには、実オブジェクトにはフォルダの種類や作成日時等の情報が保持される。また、各実オブジェクト間には、アプリケーションにおける実オブジェクトの意味にしたがってリンクが張られる。たとえば、フォルダとそれに含まれるファイルとの間には親子関係に基づいたリンクが張られる。本システムでは、電子メールまたは掲示板でユーザ間をやり取りされる情報の単位をメッセージという実オブジェクトとして扱う。さらに、このメッセージを保存するフォルダ、メッセージ中の1つの内容物である封入物といった実オブジェクトが存在する。機能実現部は

GUI部からこの実オブジェクトに対する処理を指示される。

中間層の表示オブジェクトは、ウィンドウオブジェクトと実オブジェクトの関係を管理する。GUI部のウィンドウに表示されるべき表示データはこの表示オブジェクトで保持される。各表示オブジェクト間には表示データの関係に基づいてリンクが張られる。さらに、表示オブジェクトには、それと関連するウィンドウおよび実オブジェクトへのリンク情報も保持される。表示オブジェクトには少なくともウィンドウオブジェクトに対応したものが存在する。また、1つのウィンドウで複数の実オブジェクトに関係した表示がある場合には、それらウィンドウの各部分の表示に対応した表示オブジェクトが存在することになる。たとえば、あるウィンドウで1つのフォルダの内容のリストを表示している場合、そのリストの各1行はそれぞれ、フォルダ内内容物である実オブジェクトの属性の一部を表している。このような場合、フォルダ内の内容物それぞれに実オブジェクトが対応し、そのリストでの1行分の表示を行うための表示オブジェクトも存在する。このようなウィンドウのGUI部品に対応する表示オブジェクトと、フォルダを表示しているウィンドウ全体に対応する表示オブジェクトとの間には表示上の親子関係に基づいたリンクが張られる。

上記のウィンドウオブジェクト、表示オブジェクトおよび実オブジェクトの3層からなるネットワーク状の関連をたどることにより、ユーザ操作の結果変化すべきウィンドウのGUI部品とその表示内容を導き出すことができる。すなわち、GUI部からの指示によりある実オブジェクトの属性が変化するとき、それと関連するすべての実オブジェクトを表示しているウィンドウの表示が変更されることがわかる。

### 3. GUI部と機能実現部のインタフェース

オブジェクト管理テーブルを用いることにより、GUI部の各コールバックルーチンは、GUI部の内部処理と、機能実現部のデータアクセスに直接的または間接的に関わる次の2種類のAPI群の一連の呼び出しにより構成される。

- a) 機能実現API群
  - ・メッセージの移動、複写、削除
  - ・メッセージ内容の表示
  - ・メッセージの送信
  - ・その他、アプリケーションの基本機能
- b) オブジェクト管理テーブルアクセスAPI群
  - ・ウィンドウオブジェクトの登録
  - ・表示データの取り出し
  - ・実オブジェクトの同定
  - ・ウィンドウオブジェクトの削除
  - ・表示変更ウィンドウの問い合わせ

ここに、機能実現APIは、メッセージの保存、移動、複写などのアプリケーションの基本的な機能を実現するためのライブラリ群である。一方、オブジェクト管理テーブルアクセスAPIは、オブジェクト管理テーブルをGUI部からアクセスするためのライブラリ群である。

たとえば、ウィンドウの新規作成を伴わないコールバックでは、次のような一連の処理となる(図2参照)。

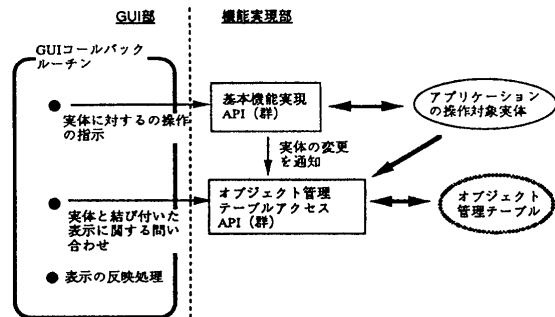


図2 GUI部と機能本体部とのインタフェース

- 1) GUI部品の識別子を引数として実オブジェクト特定のAPIを呼び出し、操作対象の実オブジェクト識別子を得る。
- 2) 1)で得た実オブジェクトの識別子を引数として機能実現APIを呼び出し、実体操作を指示する。
- 3) 表示変更ウィンドウ問い合わせAPIの呼び出しにより実体操作に付随して表示を変更すべきウィンドウの識別子を得る。
- 4) 3)で得たウィンドウ識別子を引数として表示データの取り出しAPIを呼び出し、表示データを獲得する。
- 5) GUI部で扱うウィンドウへ表示データを反映させる。

### 4. オブジェクト管理テーブルの効果

オブジェクト管理テーブルは、GUI部と機能実現部との相互依存部分を吸収する役割を果たす。GUI部のコールバックの多くは機能実現APIの呼び出しによる実体操作、オブジェクト管理テーブルアクセスAPIによる変更表示内容の獲得、そしてウィンドウへの表示データの反映、という単純な定型的な構造とすることができる。画像電子メール/掲示板システムでは、電子メールと掲示板という異なる2種類の実体を扱っているにもかかわらず、その扱いの違いはすべてオブジェクト管理テーブルにより吸収されてしまうため、GUI部では両者を区別して扱う必要がなくなっている。

### 5. おわりに

オブジェクト管理テーブルの基本構造は他のアプリケーションにも適用することができるので、多くのアプリケーションに適用し評価を得たいと考えている。