

2U-7

伝票処理アプリケーション構築環境における オブジェクト指向フレームワークの適用

津田和幸 工藤道治 横内寛文

日本アイ・ビー・エム（株） 東京基礎研究所

1.はじめに

オフィスでは、多量のデータが伝票(Form)という形態を用いて表現され、様々なユーザの手によって処理されている。我々は、オフィスにおける伝票処理の重要性に注目し、伝票処理アプリケーション構築を容易にするツールについて研究してきた。具体的には、アプリケーションモデルの提案をおこない^[1]、それに基づいた部品群および開発ツール作成した^[2]。その結果、アプリケーション開発期間の短縮を可能にするとともに、少ない教育で多くのユーザが開発に参加できるような環境が提供できた^[3]。

しかしながら、システムの拡張性、特に新たな部品の提供、といった面で問題を残している。こうした問題の解決を図るため、システムへのオブジェクト指向フレームワークの適用に付いて考察を行っている。この場合エンドユーザは、フレームワークのオブジェクトを部品を用いてインスタンス化することによりプログラムを作成する。

本稿では、伝票処理アプリケーションのための、フレームワークについて述べた後に、フレームワーク上での部品組み合わせの仕組みについて述べ、まとめをおこなう。

2. 伝票処理アプリケーションフレームワーク

オブジェクト指向フレームワークは、類似したアプリケーションの構築を簡単にする目的で提案された手法で、ウィンドウシステムのクラスライブラリ

などに用いられている。クラスライブラリには、抽象クラスが提供されており、作成しようとするアプリケーション独自のコードを仮想関数を用いて具体化する。

以上の特長から、部品作成が手順化でき、その部品特有の部分の実装に専念できる事は容易に想像できる。

2.1.伝票処理アプリケーション

ここで述べる伝票処理アプリケーションとは、送られてきた複数の伝票に対し、なんらかの処理をおこない、次の人に送るまでの一連の処理を手順化したものをいう。具体的な例としては、“住所を変更した際、複数の届け出用紙を起票し、データのコピーを行い、適切な届け先へ送る”というものや、“1ヶ月の購買要求を部門毎に集計し、集計表を各担当者に送る”などがあげられる。

我々はこうしたアプリケーションを伝票に対する基本操作と、それらの関係を記述した手順という2つでモデル化し、エンドユーザがプログラミングする環境を構築した。フレームワークにおいても、複数の伝票と基本操作を結び付ける処理オブジェクトと、処理オブジェクト間を結び付ける手順オブジェクトを用いる。

2.2.処理オブジェクト

処理オブジェクトは、伝票のリストと基本操作を結び付けるオブジェクトであり、伝票リストと基本操作を用いてインスタンス化される。

処理オブジェクトの代表的な仮想関数は“Execute”である。これは、次に述べる手順オブジェクトから呼び出される。これが呼び出されると、処理オブジェクトは、伝票リストから適切な伝票オブジェクトを順に取り出し、基本操作オブジェクトに渡していく。

処理クラスのサブクラスには以下の2つがある：

ユーザ選択処理：実行時に伝票リストを表示し、そこから選択。

条件選択処理：実行時に渡された伝票リストから条件に合致するものだけを基本操作オブジェクトにわたす。

ユーザ選択処理オブジェクトをインスタンス化するC++のコードの例を図1に示す。これによって、フォルダ内の伝票のリストを一覧表示させ、そこで選んだ伝票のコピーを別のフォルダに保管する処理を定義している。

```
Formlist * flist;
FormBox * sourceBox;
FormBox * targetBox;
...
UserSelectFuncObj * fl = new UserSelectFuncObj( new
CopyFunc( targetBox ), FormList(sourceBox));
```

図1 C++によるインスタンス化

2.3.手順オブジェクト

手順オブジェクトは作成された複数の処理オブジェクトを結び付けそれらの間の実行順序を制御するとともに、エラー（例外）処理を行う。また、実際に実行した処理オブジェクトを登録しUNDOなども実現するオブジェクトである。

3.オブジェクトインスタンス化ユーザインタフェース

前節で述べたアプリケーションフレームワークにそって、伝票処理アプリケーションを生成するためにエンドユーザは、処理オブジェクトおよび手順オブジェクトのインスタンス化をおこなう。これは、図1に示した様なC++のコードを記述することでも実現できる。しかし、エンドユーザが自分の処

理手順を実現するには簡単なユーザインタフェースが必要不可欠である。そこで、エンドユーザには、パネルに必要な値を埋めることでオブジェクトのインスタンス化を行わせる。

これを実現するために、システムはクラスに対応したパネルを用意する。パネルもクラス階層にしたがっており、サブクラスでは、その差分のみのパネルを用意すればよい。

図2は、前節で説明した抽象クラスのパネルである。パネル上のチェックボックスがこのクラス独自のインタフェースである。実行時どちらかをチェックすることでサブクラスのパネル領域に対応するものが表示される。また、このオブジェクトをインスタンス化するために必要な引数のクラスに対応した領域も用意されており、それぞれ、基本操作、伝票リストをインスタンス化するためのパネルが表示される。

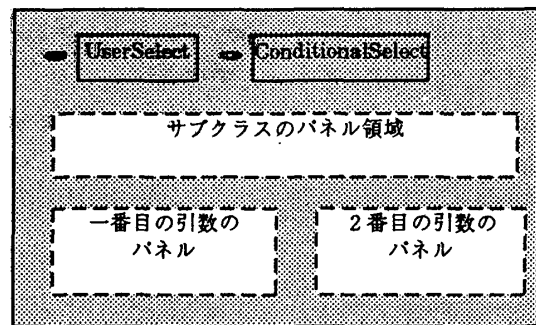


図2インスタンス化のためのパネル

4.まとめ

本稿では、既に作成した伝票処理システム開発環境をより柔軟なシステムにするための、オブジェクト指向フレームワークの適用について述べた。現在システムの実装をすすめている。

[参考文献]

- ¹⁾ 津田他, 伝票を用いたオフィス業務作成のための枠組み, 情報処理48回全国大会, 1994.
- ²⁾ 工藤他, 伝票を用いたオフィス業務の機能要素の分析と構成法, 情報処理48回全国大会, 1994.
- ³⁾ 工藤他, 伝票を用いたグループワーク構築環境の実装と業務構築における評価, 情報処理51回全国大会, 1995.