

事務処理分野へのオブジェクト指向4GL適用の試み(1)  
 -既存4GLアプリケーションへの適用-

1T-7

松永 亨\* 陳 思悦\* 橋口一生\* 森 忠夫\* 竹下 亨\*\*

\*㈱アドバンスド・システム研究所 \*\*中部大学

1. はじめに

オブジェクト指向を用いた事務処理アプリケーション作成についての一つの指針を提案し、その指針にしたがった既存アプリケーション再構築試行の結果について報告する。

ソフトウェアの部品化・再利用を促進するための鍵となる技術として、オブジェクト指向が注目されている。しかし、現在までのところ、事務処理分野のアプリケーション開発に適した高級オブジェクト指向言語が普及しておらず、オブジェクトの設計技術も確立していないため、この分野での適用例は多くない。

そこで、特に事務処理に適したオブジェクト指向化技術の検討のため、事務処理アプリケーションにおける機能ごとのモジュール分割というアプローチ[1]を導入することにより、機能単位のオブジェクト設計方法の検討を行った。これは、環境に依存した処理と業務に特有な処理とを分離した上で、大きな機能単位のモジュールとして設計し、個々のモジュールをオブジェクト指向技術を用いて実装するというものである。

さらに今回われわれは、上記の技術の実効性の検証を目的として、第4世代言語で記述された既存アプリケーションを対象として書き換えを行った。まず第4世代言語STYLEを元にしてクラス記述のための拡張をほどこしたOO4GLを想定し、アプリケーションにたいしての適用を試行した。またC++による書き換えを行った動作を確認するとともに比較材料とした[2]。

2. 機能別モジュール分割

一般にオンライン・プログラムは次のような処理から構成される。

- (1) 端末とのデータ入出力
- (2) データチェック
- (3) データ編集
- (4) データベースとのデータ入出力
- (5) 業務に特有な処理

既存のプログラムでは業務処理の途中で随時データベースへのアクセスやデータチェックなどを行うため、業務本来の処理記述と実行環境に依存した処理記述とが、

A Trial of Object-Oriented 4GL for Business Applications (1)

Toru MATSUNAGA\*, Si-Yue CHEN\*, Issei HASHIGUCHI\*, Tadao MORI\* & Toru TAKESHITA\*\*

\*Advanced System Research Organization Ltd.

\*\*Chubu University

1本のプログラムの中に複雑に絡み合ったかたちで混在している。このことがプログラムの生産性を低下させるとともに、完成したプログラムの理解を困難なものとしており、結果として保守や移植を妨げる原因ともなっている。

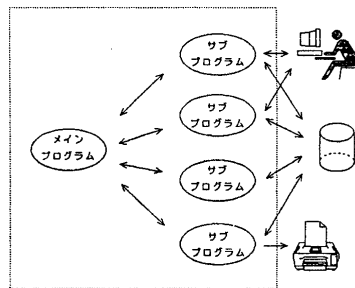


図1 既存プログラムの構成例

プログラムの中で機能ごとにモジュールを分割することにより、業務に特有な処理とデータの入出力などの環境に依存した処理とを分離することが可能になる。また、入出力を一括して処理することによりプログラム構造の単純化が期待される。これによって、保守性・移植性および部品としての再利用性の向上をはかることができる。

3. オブジェクトの設計指針

オブジェクト指向で設計を行う際には、一般に何をオブジェクトとみなすかという点でかなりの自由度があるが、ここでは2で述べたモジュール分割の方法に従い、

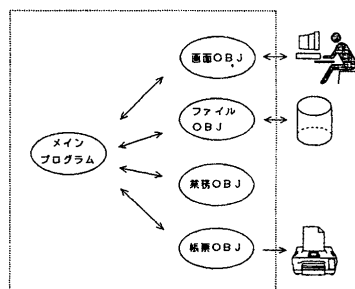


図2 オブジェクトの構成

各モジュールをオブジェクトとして実装する。つまり、端末からの入出力を行うオブジェクト、データベースとの入出力を行うオブジェクト、帳票の出力を行うオブジェクト、業務特有の処理を行うオブジェクトに分類する。この場合のプログラムの構成例を図2に示す。

メインプログラムはオブジェクトの生成/消滅および処理の制御を行う。各オブジェクトはそれぞれ特有の処理を行う。たとえば、データベースの読み書きを行うのはデータベースオブジェクトだけであり、それ以外のオブジェクトはデータベースに直接アクセスしない。

入出力に必要なデータ変数は、特定のオブジェクト内で定義される。それぞれのオブジェクト内で定義される主なデータをまとめると次のようになる。

- (1) 画面オブジェクト：画面仕様書
- (2) DBオブジェクト：DB内の必要なデータ
- (3) 業務オブジェクト：帳票仕様書（出力データ）
- (4) 帳票オブジェクト：帳票仕様書（固定データ）

各オブジェクトに要求される仕様は、プログラムごとに異なっている。各オブジェクトの設計には、まず対応する資料にしたがって、必要なデータおよびそのデータに特有な処理の分析を綿密に行わなければならない。

#### 4. アプリケーションへの適用試行

以上の考え方をを用いて、既存のアプリケーションに対する書き換えを試行した。対象としたアプリケーションは、STYLEで記述された経理システム内の一部分、貸借対象表・損益計算書を含む帳票出力処理、20帳票分である。

書き換えに用いたOO4GLはSTYLEにたいしてオブジェクト指向の機能を拡張した仮想的な言語である。クラス定義・オブジェクト生成/消滅・メッセージ交換などの文法は、C++に近いものになっている。クラスの継承については、単純継承のみを使用した。

クラス階層の設計は、画面・DB・業務・帳票の各オブジェクトごとに行った。クラスの継承機能を用いて、変数の宣言およびその変数に付随する処理に重複がないように、階層構造を定めていく。

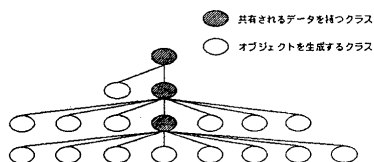


図3 業務クラス階層

クラス階層の設計にしたがって、コーディングを行った。ソースコード量の比較を表1に示す。

ソースコードのうち処理ステートメントの数は約4割

の減少がみられた。これはクラスの継承による、ソースコードの再利用の効果が大きい。

プログラム内での変数宣言の数については、処理ステートメントほどの減少はみられなかった。これはSTYLEという4GLの特質として、データベース内のデータを参照する際、データベースで定義されているフィールド名を用いて参照できるため、特に変数の宣言を必要としないからである。それに対して今回のモデルでは、

表1 STYLEに対するOO4GLのコード量

	OO4GL / STYLE
変数宣言数	79%
処理命令数	59%

データベース上の必要なデータを一括して読み込み、メモリ上で処理を行うため、対応する変数を宣言して代入を行っている。それでもそれらの変数宣言は重複がないように行われているため、結果として約2割の減少となっている。

#### 5. おわりに

アプリケーションの書き換え作業を通して、4GLのような高級事務処理向け言語についても、オブジェクト指向技術の導入により、ソースコード量の減少などの効果が期待できることを示した。

また、今回事務処理アプリケーションをオブジェクト指向化するために用いた設計方針は、保守性・移植性の向上を重視したものであり、大きな機能単位での部品化・再利用を可能にする。

#### 参考文献

- [1] 松永他 「事務処理分野における部品化のための効果的アプローチ」第43回全国大会講演論文集(5)
- [2] 陳他 「事務処理分野へのオブジェクト指向4GL適用の試み(2)」第45回全国大会講演論文集