

オブジェクト指向に基づく、既存システムの再構成

2 J-5

稲垣 陽一

(株) コンピュータ アプリケーションズ 応用技術研究室

1 はじめに

企業におけるソフトウェア資産の蓄積にともない、既存システムの保守はシステム予算に大きな割合を占めるようになってきている。そのためシステムの保守性、拡張性は重要な問題である。しかし従来の機能的アプローチで構築されたシステムは拡張性が高いとは言えない。

一方、拡張性、再利用性の高い開発技術としてオブジェクト指向が注目されている。オブジェクト指向は自然なモデリングというトップダウンな観点から優れた設計を実現すると思われる。しかし巨大なシステムをオブジェクト指向で開発する場合、その初期コストはかなり大きなものとなることが予測される。というのはオブジェクト指向の重要なメリットの一つがクラスライブラリの再利用にあるのだが、初期段階ではそのようなクラスライブラリが充実していないからである。

そこで、既存システムをオブジェクト指向パラダイムに基づいて分析し、クラスを抽出することにより、(1)より拡張性の高いシステムへの再構成、(2)再利用可能なクラスライブラリの生成が考えられる。本論文ではこのようなりエンジニアリングをCOBOLで実装された既存の事務処理プログラムを対象として提示する。またそのためのツール支援について考察する。

2 既存システムの問題点

事務処理システムの最も顕著な特徴は扱うデータの種類が非常に豊富なことである。機能的なアプローチはシステムを機能に基づいて構造化するため、データ構造は副次的なものにとどまる。そのため同一のデータ構造がシステム中に分散してしまう傾向があり、データの局所化が非常に困難である。このようにして構築されたシステムはちょっとした変更に対しても非常に敏感で、その影響が大きく波及し、拡張性は低い。その反面、機能的アプローチはひとつの機能モジュールの周辺に多数の異なる種類のデータを結びつけることになり、個々のデータに対する処理は比較的単純であるにも関わらず、プログラムの構造は無用に複雑化し、理解容易性を低下させる。このような事情はCOBOLのような型の概念をもたない言語を実装言語として採用することにより、さらに深刻なものとなっている。

3 既存プログラムの再構成

既存システムの拡張性、理解容易性を向上させるためには、散らばったデータ構造をまとめ、機能によって結びつけられている諸データをオブジェクトに基づいた構造に再構成する必要がある。

例えば事務処理システムに典型的にみられるマスタファイルの更新プログラムは、トランザクションとマスタの2ファイルにアクセスするという単純な外見をしている。しかし一般にトランザクションは複数のデータ構造をもち、その各々についてさらに処理が存在するため、プログラム内部では種別コードと処理コードの組合せに基づくIF分岐が多数存在する。また各々のデータは他のプログラムによって生成されたものだが、プログラム間のデータを介した結合情報はプログラムのコードからはわからない¹。このようなプログラムをオブジェクトを単位とする構造に再構成することにより、IF分岐はクラス構造に吸収され、またデータと処理とのカプセル化によるモジュール性の向上が実現される。

手元にあるCOBOLのマスタ更新プログラムを例として分析してみよう。このプログラムのドキュメントおよびソースコードを最も粗い目で分析すると、図1のようなクラス構造が抽出される²。各ボックスはクラスを表している。連絡票クラスと資産マスタクラスは更新関係にあり、また連絡票クラスは物件、契約、支払日連絡票などのクラスと汎化/特化の関係にある。

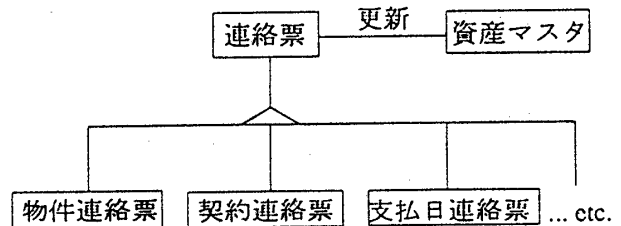


図1 クラスの構造

クラスの属性はソースコードにおけるデータ定義から得られる。ちなみにこのプログラムでは連絡票クラスとそのサブクラスは、COBOL上データフォーマットの再定義により実装されていた。また各クラスのオペレーションは、対応する種別コードおよび処理コードを

¹このような情報はJCL(Job Control Language)で記述される。

²記法はOMTを参考とした。参考文献[2]を参照されたい。

固定し、プログラムにおける本体部分（多くの場合メインループ）でパスを追跡することにより取り出すことができる。たとえば物件連絡票クラスには新規、追加、更新など6種の処理が存在する。このようなカプセル化を実行した後、汎化階層中の各サブクラスにおいて類似パターンを持つコードをスーパークラスにもたせること、つまり逆継承することで継承機構が利用可能となる。

上の例では主として意味的な側面から非常に粗い目で再構成を行なった。分析のレベルをより詳細化し、類似構造をもつデータを見つけることにより、さらにクラスが抽出される。このような同型性はCOBOLコード上、データ定義部分だけではなく、再定義、代入、CALLパラメタなどによって示される。上例のプログラムではキー項目や日付データなどが見い出される。

このような再構成を順次、プログラムに適用していくことにより、対象となっている既存システムのクラスとそれらの構造を抽出することが可能である。こうして得られる情報はそれ自体、保守業務や再構築時のドキュメントとして役立つが、さらにこれをオブジェクト指向言語にマッピングすることにより、再構築のためのクラスライブラリを生成することができる。

4 ツールによる支援

上記のような再構成を行なう場合、以下の4点に関してツール支援が望まれる。

1. クラスおよびその構造の発見支援：
データの隣接関係、再定義関係、構造の同型性からプログラム/システム中に分散する同型データを発見し、またクラス構造の案を提示する。
2. オペレーション抽出支援：
種別コードなどを指定した上で抽象実行を行ない、各クラスに所属するオペレーションを抽出する。
3. 継承機構の利用支援：
コードの共通部分を発見し、繰り返しコードの最小化を支援する。
4. 言語変換：
オブジェクト指向言語への自動変換を行なう。

このような機能をもつ再構築支援ツールを研究開発する計画である。このツールの基盤としては、リバースエンジニアリングの研究過程で開発された「保守支援システム PLASMA」を利用する。PLASMAは入力のCOBOLプログラムをprologのデータベース表現に自動変換する機能をもっている。変換された情報はそこから元のCOBOLプログラムを逆生成することが可能なほど豊富である。そこでPLASMAによって生成され

るデータベース上のアプリケーションとして、上記の機能を実現する（図2参照）。

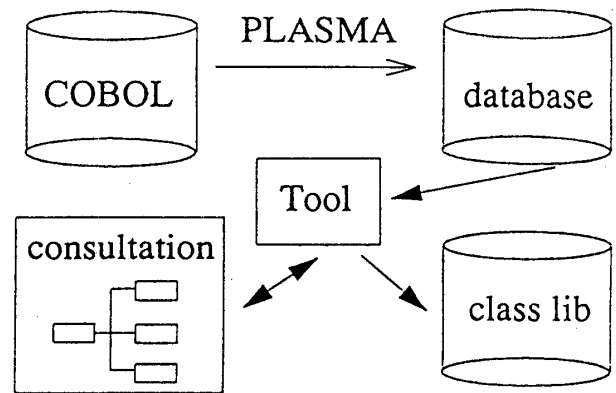


図2 ツールの構成

5 今後の課題

現在、COBOLで実装された既存の事務処理システムを対象として再構成のケーススタディを行なっている。今後は以下の点に留意しながら、オブジェクト指向に基づく既存システムの再構成手法およびその支援ツールの研究を行なう。

- 抽出されるクラス部品とその構造は、システムに対して妥当なものであるか。またその品質（保守性、拡張性、理解容易性）はどの程度向上するか。
- 再利用可能なクラスライブラリは生成できるか。
- 機械化支援が可能な部分とその実装。

謝辞

本研究に対し貴重な御指摘を頂いた、秋田昌幸室長、城市優氏、萩原高行氏、森小世樹氏に感謝します。

参考文献

- [1] Meyer, B.: オブジェクト指向入門, 二木厚吉監訳, アスキー出版局, 1990.
- [2] Rumbaugh, J. et al.: Object-Oriented Modeling And Design, Prentice-Hall, 1991.
- [3] 萩原高行, 稲垣陽一: 保守支援システム PLASMA, 第12回品質管理シンポジウム発表報文集, PP.213-220 (1992年9月)
- [4] 広本 治, 城市 優: 事務処理システムにオブジェクト指向を適用する, 日経インテリジェントシステム別冊, PP.66-81 (1992年春号)