

リエンジニアリング・ツール  
— 分離抽出による業務処理の紹介 —

7P-8

和泉 忠尚 近藤 稔 志賀 浩  
(株) アドバンスド・システム研究所

## 1 はじめに

効率的なシステム開発を考える上で、既存システムを再利用する方法がより効率的であると考えられるが、それを実現しているケースはなかなか見つからない。既存システムのロジックを移植できればその開発効率は大幅に向上することはあきらかである。実際に今日の新規システム開発と言えどもその多くは、同じことの繰り返しでありその中に組み込まれるロジックはさして新しいものはそう多くはない。

同じロジックの繰り返しと言えども単純に部品化できるわけではない。オペレーティング・システムとのインターフェースを持たなければならない入出力処理は、ハードウェアやDB/DC(DataBase/Data Communication)製品が変わると移植できなくなってしまう。したがって、ハードウェアや環境によって影響される入出力処理部分を分離することによって、残された部分がそのプログラムの純粋な業務処理部分となり、移植性を高めることができる。当研究所において、先に入出力処理部と業務処理部を分離する考えに基づいたアプリケーション・モデルを提案したが(情報処理学会第・回全国大会論文参照)、そのモデルに基づいて、既存システムからその実行環境に依存しない、すなわち業務処理部を自動分離抽出(以降、分離抽出ツールと呼ぶ)する1つの方式について検証したのでここに紹介する。

## 2 既存システムからの業務処理部分の分離抽出

構造化、モジュール化が考慮されているシステムでは、その入出力処理と業務処理とは、多くの場合別モジュールに分割されており入出力処理は業務処理から外部ルーチンとして呼び出される。そこで、あるモジュールが業務処理中心のモジュールであるかどうかを見極め、そうであればそのモジュールの入出力処理と判断された部分を分離する。すなわち業務処理に当たる部分から入出力処理を呼び出している部分を削除することによって業務部だけを抽出することができる。

当研究所において、既存システムの入出力処理と業務処理の分離をコンピュータで自動化する方式として、業務処理から呼び出している外部サブルーチンのうちどれが入出力処理用サブルーチンであるかまた、その前後処理としてどのような変数名が使用されているのかを分離抽出の判断条件として与えることにより分離抽出の実現を試みた。ここではこの判断条件として与える情報を業務処理抽出条件と呼ぶ。これを手がかりに分離抽出ツールは入出力処理および関連する処理を削除する。

## 2.1 分離抽出の判断条件

COBOLで作成されたシステムについて業務処理本来の処理ロジックであるか、入出力処理ロジックであるかの判断は、以下の情報を基に行う。

- (1) 入出力処理を行っている外部サブルーチン名  
入出力処理を行うサブルーチン名をCALL文で呼び出すということは、すなわち入出力処

理を行っていることを意味するので削除するための手がかりとなる。

## (2) 入出力処理に関連する変数名

入出力処理の作業域として使用されている変数名は、入出力処理に関連するステートメントとして削除するための手がかりとなる。

## 2.2 業務処理の抽出

次に、与えられた判断条件を基にいかに入出力処理を識別するかを考える。

## (1) 外部インターフェース

入出力処理をCALLする前後では、主に変数に値を代入したり、戻り値を基にエラー処理を行っている場合が多いので、抽出の手がかりとして、与えられた外部サブルーチン名、および、入出力処理に関連する変数名を使用している部分が分離の対象であると判断できる。

## (2) データ構造

入出力処理モジュールに渡されるパラメータは、各メンバーをまとめている上位レベルの変数名であることが多い。したがって、その前後処理を分離するためにはパラメータとして渡された変数の下位レベルの変数もその対象とする必要がある。判断条件として与えられた変数名をプログラムのDATA DIVISIONからサーチして分離対象の変数として取り出す。

## (3) 入出力関連ステートメント

実際のREAD/WRITE, OPEN/CLOSE文、または、DB/DC製品を使用している場合などは、その固有なステートメントを分離の対象とする。

## 2.3 分離抽出の具体的なケース

各セクション、命令ごとにその分離抽出処理方法を具体的なケースで示す。

## (1) DATA DIVISION 内データの扱いについて。

## (a) ケース1

XDATA-01が入出力処理関連の変数名とする。

```
処理前
000100
000110 01 XDATA=01 PIC X(08).
000120
```

```
処理後
000100
000120
```

## (b) ケース 2 (下位レベル削除)

XDATA-01, XDATA-02が入出力処理関連の変数名とする。

## 処理前

```
000100
000110 01 WK-AREA-01.
000120 03 XDATA-01.
000130 05 DATA-01 PIC X(08).
000140 03 XDATA-02.
000150 05 DATA-02 PIC X(08).
000160 01 ....
```

## 処理後

```
000100
000160 01 ....
```

## (2) PROCEDURE DIVISION 内命令の扱い

## (a) ケース 3 MOVE 文 (命令の一部削除)

XDATA-01, XDATA-02 が入出力処理関連の変数名とする。

## 処理前

```
000110 MOVE DATA-01 TO XDATA-01
000120 DATA-02
000130 XDATA-02.
000140
```

## 処理後

```
000110 MOVE DATA-01 TO DATA-02.
000140
```

対象行が削除されると同時にピリオドがDATA-01の直後に移動される。

## (b) ケース 4 IF 文

XDATA-01が入出力処理関連の変数名とする。

## 処理前

```
000090
000100 IF XDATA-01 = ZERO
000110 MOVE DATA-03 TO DATA-04
000120 ELSE
000130 MOVE DATA-01 TO DATA-02.
000140
```

## 処理後

```
000090
000140
```

## (c) ケース 5 (NEXT SENTENCE への書き換え)

IF 文中に削除対象の命令が存在する場合、該当の命令の部分スペースに置き換える。ただしその結果IF文そのものが文法エラーになるケースが考えられるが、その場合は、次の例に示すようにする。

XDATA-01が入出力処理関連の変数名とする。

## 処理前

```
000100 IF DATA-01 = ZERO
000110 MOVE SPACE TO XDATA-01
000120 ELSE
000130 MOVE DATA-01 TO DATA-02.
```

## 処理後

```
000100 IF DATA-01 = ZERO
000110 NEXT SENTENCE
000120 ELSE
000130 MOVE DATA-01 TO DATA-02.
```

COBOL II は、CONTINUE 文となる。

## (d) ケース 6 CALL 文

SUBMODが入出力処理を行う外部サブルーチン名とする。

## 処理前

```
000090
000100 CALL 'SUBMOD' USING DATA-A
000110 DATA-B.
000120
```

## 処理後

```
000090
000120
```

同様に、入出力処理関連の変数名または、入出力処理を行う外部サブルーチン名を含む各命令について分離する対象行を決定する。

## 3 おわりに

この方式によれば、業務処理抽出条件を与えらることにより、入出力処理と業務処理を分離することが可能となる。このことは開発効率向上のための業務処理部分分離抽出の1ステップであると確認する。

本報告からは外してはいるが、現在、業務処理部の明確化、分離抽出後の検証方法、および、業務処理抽出条件の自動抽出方法について研究中である。

今後の課題としては、分離抽出処理対象のシステムによってはこのツールの効果が必ずしも高くないケースがあるかも知れないので、業務システム別にどれだけの効果的が得られるかを検討することと、さらに分離抽出アルゴリズムの高度化については引き続き研究をする予定である。

## 参考文献:

- [EJC90] Elliot J. Chikofsky: Reverse Engineering and Design Recovery, IEEE Software, January, 1990
- [BVR90] Basili, V.R.: Viewing Maintenance as Reuse-Oriented Software Development, IEEE Software, Jan. 1990, pp. 19-25
- [L. Clnd89] L. Cleveland, "A Program understanding support environment", IBM SYSTEMS JOURNAL, Vol. 28, No. 2, 1989, pp. 324-344