

システム統合に於けるプログラム再利用の新しい方法<sup>1</sup>

3L-5

秋山 義博<sup>2</sup>伊藤 隆<sup>3</sup>

金沢工業大学情報工学科

日本アイ・ビー・エム(株)SIE研究所

## はじめに

新しい目的ソフトウェアを開発するシステム統合においては、既に開発されているソフトウェアを再利用する技術が、その開発の生産性と品質の維持向上の為に重要である。この再利用技術として、ソフトウェア部品(ソースコード・セグメント)又はその設計情報、ジェネレータ部品コンパイル、アーキテクチャ(ドメインモデル)等[以下では、簡単に部品と呼ぶ]が提案されているが、これらが、目的ソフトウェアの機能要件に該当する部品を検索し再利用する割合を増やすことは、一般的には難しいと考えられる。機能、インターフェース、実行環境等が微妙にずれている為に使えない部品が起こるからである。この論文では、この類の問題を検討した後、簡単に新しいソフトウェアの再利用方式を提案する。

## 問題記述

ここで、統合対象の既存システム機能を同様に提供し、更に、それらを融合してできる新機能も提供するような新システムを開発するためのソフトウェア再利用を考える。この場合、新機能に含まれるビジネスルールは、それが既存システム機能にある場合には、新旧内容が同等でなければならない。従来のサービス機能を実現するソフトウェア・コンポーネントが含まれることは大体において予想出来るが、それらを融合して新機能を実現する為に、既存ソフトウェアに実現されている小機能を細かく組み合わせ、不足分は追加して再利用を計ることを考えるが、このように、将来システムへの統合を予想して現部品を開発することは難しく、また、その様な部品を想定して新システムの設計を考えることも一般には自明でも簡単でもない。

## 解決案のモデル的検討と新しい再利用法

部品を利用する難易度を考える。n番目の部品の機能を $F_n$ 、インターフェースを $I_n$ として(但し、修正困難度をここでは考えない)、この難易度は、それらのパラメータで計った機能量 $|F_n|$ (ファンクションポイントで計っても良い)とインターフェース量 $|I_n|$ (パラメータの数)の積 $|F_n| \times |I_n|$ に比例するという簡単な第1近似モデルで考える。d, eは、それぞれ機能とインターフェース・パラメータ変数の間の内部干渉の程度の強さを表し、経験的にd, e>1であり、ほぼ定数であると考えられる。 $|F_n|$ と $|I_n|$ の評価を、幾つかの代表的部品について検討すると、先ず、ソフトウェア部品については、インターフェースの陽的記述も無いことから、その全ソースコードが検討対象になる。同等レベルの機能を持つサブルーチンの場合には、インターフェースの記述、定義が与えられているので、 $|I_n|$ 値は相対的に小さくなる。従って、ソフトウェア部品の形式的仕様記述を与えることで見掛け上の再利用難易度改善がなされていることは良く知られている。更に、 $|F_n|$ は、そのインターフェースのパラメータの数とそれぞれのパラメータが取りうる値の可能な組みの平均数の積で与えられるだろうから、 $W=|I_n|$ ,  $Z=|I_n|$ とおくと、 $|F_n| \sim W \times Z$ の形になり、難易度の最終的な形は、 $\sim W^d \times Z^e$ となる。従って、WとZが最小値を取る場合に、難易度が最も低くなる。以上の議論から、難易度は、実質的には2パラメータ関数であるから、WとZが出来るだけ同時に最小値を取るような部品を見つけて再利用することを考えれば良い事が解る。

以上述べたようなWとZを最少にする部品を次のように定義する。先ず、目的ソフトウェアの設計を行い、それに含まれるサブプログラムとサブルーチンの新機能名、及びその新インターフェースを定義する。同様に、再利用検討対象ソフトウェアについて、サブプログラムとサブルーチンの旧機能名とその呼出しの為に旧インターフェースを同定し、且つ、新機能と新インターフェースに含まれると思われるビジネスルールを念頭に於いて、旧機能名と旧パラメータに対して適切

<sup>1</sup> A New Approach for Software Reuse in System Integration  
<sup>2</sup> Yoshihiro Akiyama, Information and Computer Engineering, Kanazawa Institute of Technology, 7-1 Ohgigaoka Nonoichi-machi Ishikawa 921 Japan  
<sup>3</sup> Takashi Itoh, Systems Laboratory IBM Japan Ltd., 1-1 Nakase Mihana-ku, Chiba Chiba 261 Japan

な前提値又は制限値を決めると、これが単純なビジネスルールの表現になっていて、改めてこれを現部品として定義する。もし大局変数が使用されている場合には、その値を適当に与えるように呼出側で設定する。このように、機能名と呼出しインターフェースで使われているパラメータを利用して、利用し易い(アプリケーション的な処理ルールとして純度の高い)機能を取り出し、これらを簡単に新機能に組み込むことが可能になる。このような部品を、動的機能サイズ(DFS)と呼ぶことがある。そこでは、完全に前提値を与えた変数はW値に寄与しないこと、及び、制限値を与えた変数についてはZ値が減少すること、から、全体として現部品の利用困難度がかなり減少する事が解る。しかも、このような現部品の作成は従来の方式に比べてより簡単であり、新機能に組み込む場合にも、現部品はアミックアクションとして利用出来るので、設計/実現に於いても作業は容易になる。

再利用全体の工程は概略次のようになる：1)統合する基本システムを決める、2)新システムの全体設計を行なう、3)新旧システムの部品とDFSを求める、4)新部品、DFSを旧部品、DFSを利用してそれぞれ実現する。この方式は、他の手法、アプローチとの相性も良いと考えられる。例えば、この方式による部品の定義は、R. C. Linger等が提唱しているCleanroom Process Modelと非常に良く合い、特にテスト設計に良い。又、システム統合に於いて必要な最下層部に位置する既存コンポーネントの一つの例である。下図は、IBM AIXwindows OSF/Motifを用いたテキストエディタとグラフ描画プログラム[APIは従来型であり、READコマンドを実行して画面入力/操作情報を読み取る]の二つのソフトウェアを統合し、テキストウィンドウとグラフィックウィンドウがその統合APIを通して連動するように設計したプロトタイプである。基本的なテキストウィンドウ操作とグラフィックウィンドウ操作は、すべて旧ソフトウェアより部品又はDFSの形式で抜き出したものである。又、連動制御は新たに付け加えたもので、これによりユーザによる操作が非常に簡単になった。

#### まとめ

ソフトウェア再利用の難しさの度合を準定量的(定性的)に考え、それに基づいて、最も単純と考えられる場合の再利用方式を述べた。この方式によれば、再利用したいソフトウェアの設計的/機能的仕様は殆ど考え無くて済むと同時に、目的ソフトウェアの設計を、旧ソフトウェア機能に制限されずに行なうことが出来る。

#### 参考文献

- C. W. Krueger, Software Reuse, ACM Computing Survey, Vol.24, No.2, June 1992
- R. C. Linger, Cleanroom Process Model, IEEE Software Vol.11, No.2, March 1994
- IBM XWindow Programming Guide, IBM Austin ITSC, Texas, GG24-3382-02, Feb. 1992
- Systematic Reuse Issue, IEEE Software, Vol.11, No.5, Spring 1994

