

## アプリケーションフレームワーク理解のための 操作優先度メトリクスの提案

中西弘毅                      荒野高志  
NTT ソフトウェア研究所

アプリケーションフレームワークを理解する上でクラスインタフェースに関する知識は実装に関する知識よりもより重要となる。本研究では、実際に開発されたプロダクトを対象にしてクラスのインタフェースで頻繁に用いられる操作がクラスの実装で頻繁に用いられる操作と必ずしも一致しないことを明らかにし、クラスインタフェースにおける出現頻度をもとに、アプリケーションフレームワークを理解する上でのクラスライブラリにおける操作の優先度を提案した。そして、ごく一部の出現頻度の著しく高い操作を覚えればクラスインタフェースの大部分をカバーできることを確認した。

A metric for evaluating the importance of class operations  
to analyze an application framework.

Koki Nakanishi                      Takashi Arano  
NTT Software Laboratories

### Abstract

In terms of understanding an application framework, to understand class interfaces is more important than to understand class implementations. This paper shows that by measuring actual class libraries the frequently used class operations in the class interfaces are not certainly coincident with the frequently used class operations in the class implementations, and proposes a metric for evaluating the importance of class operation to analyze an application framework based on the class operation frequency in the class interfaces, and confirms that a few of frequently used class operations almost cover the class interfaces.

## 1 はじめに

オブジェクト指向技術では、再利用という観点からクラスライブラリ・フレームワークの理解が非常に重要となる [2][3]。しかし、フレームワークを構成するクラスの数が膨大になるに従って、フレームワークを理解するための労力は急激に増大し、逆に生産性を下げる原因となってしまう可能性がある。ここでフレームワークとはある特定の用途のために用意されたソフトウェアモジュールの集まりであり、抽象クラスと具象クラスから構成されたクラスライブラリとして実現されるものである。

オブジェクト指向技術では継承機能と多相化機能がクラスライブラリを構築する上で重要な役割を果たす。[4]では継承機能が実装とインタフェースの統一という二つの立場から利用されることが指摘されている。アプリケーションフレームワークの理解という観点からは、クラスインタフェースの立場がより重要となる [1]。しかし、実装のために用意された操作があるためにクラスに適用可能な操作の数は膨大となりフレームワークの理解を効率的に行なうためには、フレームワークを理解する上での優先度を与えることが望ましい。

本研究では、クラスインタフェースに現れる出現回数によってクラスライブラリ理解における操作の優先度を定義し、実際に開発されたクラスライブラリを対象に、クラスの実装を理解するという立場で有効となる知識とクラスのインタフェースを理解するという立場で有効となる知識が必ずしも一致しないことを明らかにする。

## 2 クラスの実装とインタフェースの相違点

オブジェクト指向技術では継承という概念があり、他のクラスを継承することによって、継承されたクラスとの差分のみを新たに実装すれば目的のクラスを実現することができる。このときクラスの仕様はそのクラスが継承しているクラスと新たに加えられた差分からなる。クラスのインタフェースには、新たに付け加えられた操作と継承したクラスの操作を再定義した操作が記述される。オブジェクト指向では多相化という概念があり同じ操作が適用されるクラスに応じて異なるメソッドが実行される。ここでメソッドとは実際に実行されるプログラムであり、操作とはそれにつけられた名前である。よってオブジェクト指向では多相化によって異なるプログラムが同じ名前を持つことが可能であり、多相的な操作はより頻繁にクラスのインタフェースに現れる。例えば、ファイルというクラスの Print という操作がこのクラスを継承する ASCII ファイルクラス、テキストファイルクラス、ポストスクリプトファイルクラスでそれぞれ Print という操作を実行するメソッドが定義され、実行時に Print という操作を受けたオブジェクトがどのクラスに属するかに応じて適切なメソッドが実行される。この様に多相化はクラスのインターフェースを統一する上で有効であり、多相的な操作はアプリケーションフレームワークを理解する上で重要であると考えられる。

一方実装では、クラスインタフェースにリストされた操作の実現方法が書かれる。そこではより計算機に近いデータを扱う操作が頻繁に用いられると考えられる。

この様に、実装で頻繁に現れる操作とクラスインタフェースに頻繁に現れる操作はその目的が異なることがわかる。

### 3 クラスインタフェースにおける操作の出現頻度

この章では、クラスインタフェースに現れる操作の出現頻度を計測する。オブジェクト指向プログラミング言語では多相化を支援する機能が備わっている。多相化とは一つの操作に対して、それが適用されるオブジェクトの属するクラスに応じて異なるメソッドが実行されることである。これによって継承したクラスは、継承されたクラスにある操作を再定義することができる。特に抽象クラスを継承する場合、抽象クラスにある操作はこれから作るクラスに応じて再定義される場合が多い。よって多くのクラスに適用可能な抽象的な操作は幾つものクラスで再定義される場合が多い。

InterViews3.1 というオブジェクト指向言語 C++ で書かれたクラスライブラリに対してクラスインタフェースに現れる操作の出現頻度を計測する。C++ ではクラスインタフェースは\*.h (ヘッダーファイル) に記述される。また C++ では構築子は各クラスが独自に持つ操作であり多相化されることがないので計測対象から除いた。図1は InterViews3.1 というクラスライブラリに対してクラスインタフェースに現れる操作の出現頻度とその大きさの順にソーティングしたときの順位の対をプロットしたものである。この図を見てわかるように、小数の極度に出現回数の多い操作と、大部分の出現回数の極度に小さい操作から構成されていることがわかる。この結果より出現回数の大きい順に操作に優先度をつけて覚えた場合とそうでない場合では操作に出合う確率に大きな差がでることがわかる。これによってごく一部の出現頻度の著しく高い操作を覚えればクラスインタフェース大部分をカバーできることがわかる。

### 4 実装における出現頻度

この章では、クラスの実装の中で現れる操作の出現頻度を InterViews3.1 クラスライブラリを対象に計測し、各操作の実装における出現頻度を分析する。C++ では実装は\*.C ファイルに記述される。図2はクラスライブラリ InterViews3.1 の操作を実装における出現回数の大きい順にソーティングして順序付け、その順位と実装における出現回数をプロットしたものである。この図から、クラスインタフェースに現れる操作の出現回数の計測結果と同様に実装の中に現れる操作も小数の頻繁に出現する操作と多数の出現回数が少ない操作からなることがわかる。

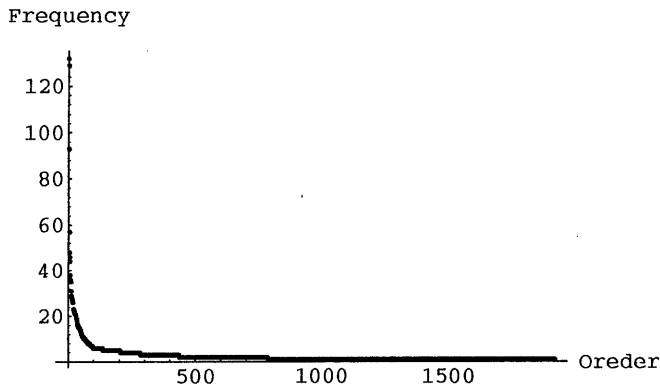


図 1: クラスインタフェースにおける操作の出現頻度

## 5 計測結果

この章では、幾つかの良く知られたクラスライブラリを対象に、実装における操作の出現頻度とクラスインタフェースにおける操作の出現頻度の相関を調べる。本研究で計測対象としたクラスライブラリは

1. IntrerViews3.1
2. ET++3.01
3. VisualC++ Microsoft fundation class library(以下 MFC と記す。)
4. Borland C++ ObjectWindows library(以下 OWL と記す。)
5. NIHCL3.0

以上、4個のクラスライブラリである。まず、InterViews3.1における操作のクラスインタフェースにおける出現頻度と実装における出現頻度の関係を計測した結果を図3に示す。この図を見るとわかるように、一部の出現頻度の高い操作が特異値となり、それらの点の有無が、相関係数の値に大きな影響を与えてしまうことが考えられる。そこで、この様な、影響を除くため対数値に変換した上で相関係数を計る。次の表は先に示したクラスライブラリに対して、操作のクラスインタフェースにおける出現頻度を対数変換

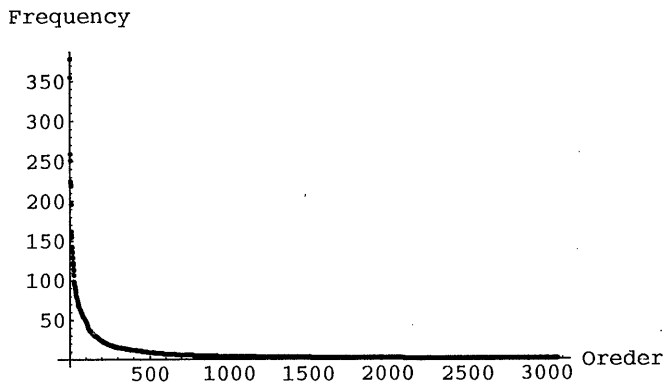


図 2: 実装における操作の出現頻度

したものと実装の中に現れる出現頻度を対数変換したものの相関を示したものである。ここで、相関係数とは二つの変量がグラフの直線の周りに集まっている度合を示す値であり、最大1（完全に直線上にのっている場合）を取り、1に近い程直線の周りに集まっていることを意味する。

クラスライブラリ名	相関係数
ET++3	0.416533
InterViews3.1	0.536232
MFC	0.292202
OWL	0.321177
NIHCL3.0	0.341706

この結果より、操作のクラスインターフェースの記述に現れる頻度と実装の中に現れる頻度は必ずしも一致しないことがわかる。

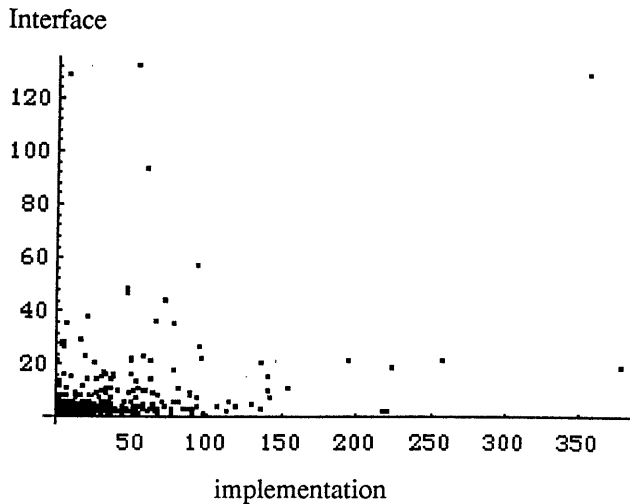


図3： 実装とインタフェースにおける出現頻度の関係

## 6 結果

本研究では、アプリケーションフレームワークの理解のための操作優先度としてクラスのインタフェースに現れる出現頻度を用いることを提案した。そしてそしてこの操作優先度に関して以下の二点が明らかになった。

- 実際にクラスライブラリに対してクラスのインタフェースに現れる出現頻度を計測した結果、一部小数の操作が著しく高い出現頻度を持ち大部分の操作は稀にしか出現しない。これによってごく一部の出現頻度の著しく高い操作を覚えればクラスインタフェース大部分をカバーできる。
- クラスインタフェースの記述に頻繁に現れる操作と実装の中に頻繁に現れる操作は必ずしも一致しない。

## 参考文献

- [1] 中西弘毅、荒野高志、今瀬真、”クラスライブラリーインタフェースの成熟度評価方法” 電子情報学会論文誌 D - I Vol.J77-D-I, No.9, pp.628-634, Sep, 1994
- [2] 荒野高志、青野博志、藤崎智宏 “オブジェクト指向フレームワーク再利用性の一実験” 情報処理学会第49回全国大会
- [3] Wirifs-Brock,R.J.and Johnson,R.E,”A Survey of Current Research in Object-Oriented Design”, Communication of the ACM, Vol.33, No.9, pp.104-124, 1990.
- [4] W.R.Lalonde,D.A.Thomas, and J.R.Pugh,”An Exemplar-Based Smalltalk, OOPSLA