

## オブジェクト指向開発におけるメンテナンス時の局所性の測定

## 4H-2

上原 智, 池田寛治, 木下 誠一, 船生 幸宏

NTTデータ通信株式会社

## 1. はじめに

オブジェクト指向による開発は、カプセル化（情報隠蔽）により、メンテナンス性に優れているとされている。今回、我々はオブジェクト指向を用いたソフトウェア開発を通して、オブジェクト指向による開発がどの程度メンテナンス性に優れているか測定を試みた。具体的には、オブジェクト指向で開発したプログラムに対して仕様変更が発生した場合に、どの程度のクラスにその影響が波及するかそのクラス数を計測した。その結果と考察を示す。

## 2. 適用事例概要

今回開発したソフトウェアは、ある大型機のシステムのメンテナンス支援を目的とするもので、そのシステムの機能追加・変更を行う際、設計ドキュメントの内容を容易に参照・検索できる機能を備えたものである。今回、このソフトウェアをオブジェクト指向で開発することにより、オブジェクト指向による開発は本当に優れているのか、特にメンテナンス性に着目して調査した。

開発スケジュールを図1に示す。まず業務内容の調査を約3週間、分析・設計を約2週間行った後、C++による実装に入った。実装段階においては、約1ヶ月間でソフトウェアを完成させ、その後、ユーザによる仕様変更を1ヶ月毎に取り入れて計2回メンテナンスを行い、約3ヶ月間で開発は終了した。尚、方法論としてはオブジェクト指向方法論OMT[1]を採用した。

最終的にソフトウェアの規模は約8.9Kステップとなり、2回の仕様変更要求による変更項目数は、合計18項目となった。

ソースコード以外の生成物として主なものを挙げれば、OMT法におけるオブジェクト図、イ

ベントトレース図、イベントフロー図、データフロー図である。仕様変更があった場合には、これらの設計ドキュメントの変更もソースコードと同時に終わった。

尚、開発はMVCモデル[2]に準拠して進められ、モデル（Model）、ビュー（View）、コントローラ（Controller）に分けたクラス構成とした。

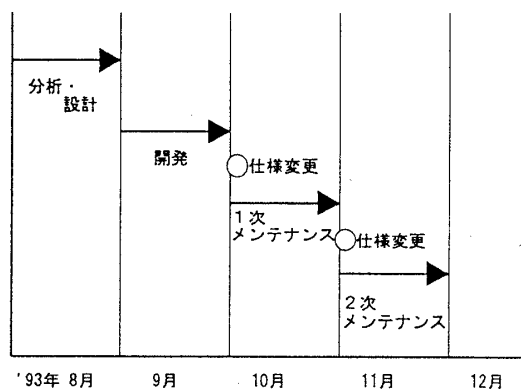


図1 開発スケジュール

## 3. メンテナンス性の測定方法

仕様変更の全18項目に対して、それぞれ設計ドキュメントとコードの修正を行った。その際、修正したコードにコメント文を追加することによって、それぞれの仕様変更に対してどれだけのクラスが変更・追加されたかという情報を正確に記録することができた。そして、これらのデータをもとにメンテナンス性の測定を試みた。

ここで着目したのは、それぞれの仕様変更に対して変更・追加されたクラスの数である。なぜならば、変更・追加を受けたクラス数が、少なければ、それだけ仕様変更によるプログラムへの影響が局所化されているということであり、メンテナンス性は優れていると考えたからである。

## 4. 結果

それぞれの仕様変更に対する追加・変更クラ

ス数の全体のクラス数に対する割合をグラフにしたものを図2に示す。なお、全クラス数は56個である。

1つの仕様変更に対して、変更・追加されたクラス数の、全体のクラス数に占める割合は、最大でも13%以内に留まっている。

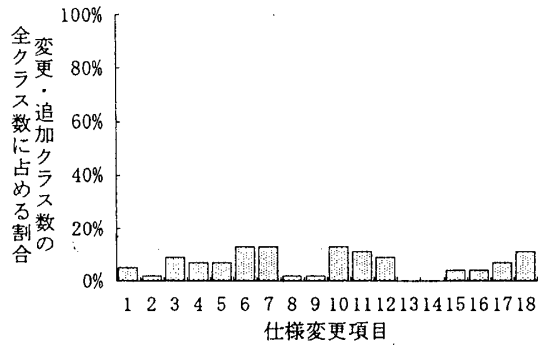


図2 追加・変更クラスの全クラス数に占める割合

次にクラスをモデルを構成するクラスと、ビュー（ユーザインターフェイス）を構成するクラスとに大きく分類し、それぞれのクラスについて変更・追加状況を調べてみた。図3は、図2をクラスの分類毎に色分けし、割合の大きい順に並べ換えたものである。

これより、一部の例外を除いてモデルクラスに変更・追加のあった仕様変更では、全体の変更・追加クラス数が多いという傾向が見られた。

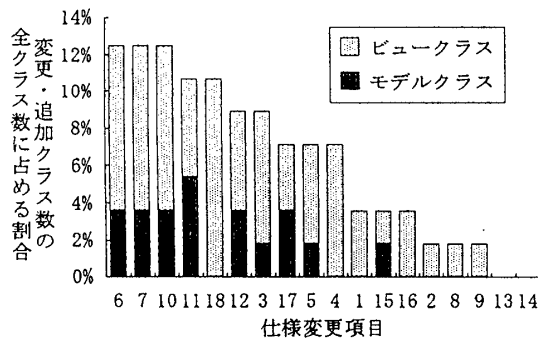


図3 追加・変更クラスの内訳

5. 考察

1つの仕様変更に対して最大でも13%以内のクラスへの影響に留まっており、オブジェクト指向開発におけるメンテナンス時の局所性は優れていると評価できる。また、ビュークラスとモデルクラスとに大きく分類した場合に、モデルクラスに

影響のあった仕様変更は、全体の追加・変更クラス数も多いということから、オブジェクト指向開発におけるモデルを構築する工程、すなわち分析工程の重要性があらためて浮き彫りにされたといえる。

今回のソフトウェアを開発するにあたっては、ユーザである業務担当者へのインタビューにより業務要件の習得を行ったが、実装段階に入った後に一部誤解等によるモデリングのミスが発覚し、オブジェクトモデルレベルでの変更が幾つか入った。また、オブジェクトモデルを変更するに伴い、オブジェクトモデルを表現するモデルクラスのみならず、ビュークラスにも変更が必要となった。このことから、オブジェクト指向開発では、分析段階においてモデルを強固に構築し、仕様変更によるモデルクラスへの影響を極力押さえることがメンテナンスコスト削減への第一歩であると言える。

尚、オブジェクト指向開発においては、設計・実装の一貫性によりプログラムとドキュメントとの対応が明確になされる。このため、プログラムとドキュメントのメンテナンスを同時に行う際にもスムーズに行うことができた。このことから、オブジェクト指向開発は、ドキュメントのメンテナンスに関しても優れていると言えるのではないだろうか。

6. おわりに

オブジェクト指向によるソフトウェア開発の有効性が各所で議論されているが、研究レベルだけではなく、今回のような現場レベルでの地道な評価の積み重ねも重要であると言えるのではないだろうか。我々は今後もこのようなオブジェクト指向開発の有効性の検証を行っていく予定である。

参考文献

[1] J.ランボー他著（羽生田 栄一 監訳）：  
「オブジェクト指向方法論OMT」, トッパン,  
jul, 1992.  
[2] 青木 淳著：「オブジェクト指向システム分析  
設計入門」, ソフト・リサーチ・センター,  
oct, 1993