6Z－2 特定アプリケーション開発における フレームワークの評価に関する考察

藤原 晃† 今川 勝博† 楠本 真二† 井上克郎†* 大坪 稔房‡ 湯浦 克彦‡

†大阪大学大学院基礎工学研究科 ‡株式会社日立製作所ビジネスソリューション開発本部
*奈良先端科学技術大学院大学情報科学研究科

1 はじめに

ソフトウェアの開発の工数削減や品質向上において再利用の果たす役割は重要である．再利用手法のひとつとしてフレームワークを用いた開発が活発に行われている [3]．本研究では，ある特定のアプリケーション開発において，フレームワークを適用した場合の効果を評価する手法について考察する．

2 評価対象

評価対象は，ある地方自治体の窓口業務アプリケーションである．このアプリケーション開発で，従来，各画面単位の処理プログラムが部品化され，再利用することで開発が進められてきた．この手法では，開発者は画面部品を組合せ，その画面の遷移とそれに伴う内部処理を記述する必要があった．

その問題点を解消するために，フレームワークを用いた新しい再利用手法が検討されている．新しい手法では画面単位の処理プログラムの部品化に加え，画面遷移単位の処理をフレームワーク化し再利用する．この手法では，開発者は画面遷移フレームワークにその業務固有の処理を追加するだけで機能を実現することができる．地方自治体向けアプリケーション用フレームワークでは，選択，検索，更新，参照などの処理が持つ画面遷移のタイプごとにフレームワーク化がなされている．

フレームワークを用いた新しい再利用手法の従来手法に対する有用性を定量的に評価することが本研究の目的である．

3 評価手法

3.1 基本方針

評価項目として，アプリケーションの複雑さと機能量を用いる．具体的には，フレームワーク利用により，新規アプリケーション開発時に軽減された複雑さと機能量の評価と，新機能追加時に軽減された複雑さと機

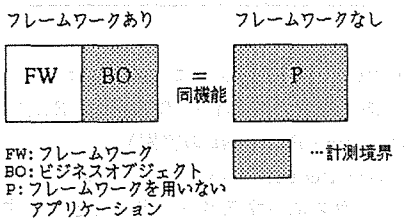

図 1: 類似アプリケーション開発時の計測方法

表 1: 類似アプリケーション開発時の機能量計測結果

	FWあり	FWなし
資格個人照会	176	526
資格世帯照会	180	526
賦課状況照会	418	671
資格取得転入	252	672

能量の評価を行なう．複雑さは Chidamber,Kemerer のメトリクス [2]（以下 C-K メトリクス）で，機能量は OOFP[1] で計測を行なう．文献 [4] では，デザインパターンの評価のために C-K メトリクスとコード行数，クラス数，メソッド数が用いられていた．

3.2 OOFP(Object Oriented Function Point)

OOFP とは，オブジェクト指向ソフトウェアの開発プロジェクトの機能量を見積る手法である．OOFP では，オブジェクトが計測対象となる．OOFP の計測は，以下の 6 ステップで行なわれる．

Step1(論理ファイルの識別):論理ファイルとはデータのまとまりで，クラスがこれに当たる．論理ファイルには内部論理ファイル (ILF, アプリケーション内部のクラス) と外部インターフェイスファイル (EIF，アプリケーション内部から参照される外部クラス) の 2 種類がある．

Step2(論理ファイルの複雑さの決定):DET(データ項目数)，RET(レコード種類数) の 2 つのパラメータを用いて決定する．単純な属性 (integer,string 等) は DET となり，クラスまたは他のクラスへ

Evaluation of a Framework to Develop Applications on a Specific Domain
†Hikaru Fujiwara, †Masahiro Imagawa, †Shinji Kusumoto
†*Katsuro Inoue, ‡Toshifusa Ootsubo and ‡Katsuhiko Yuura
†Graduate School of Engineering Science, Osaka University
‡Hitachi Ltd.
*Graduate School of Information Science, Nara Institute of Science and Technology

表 2: 新機能追加時の機能量増加計測結果

	FWあり	FWなし
F_1	176	526
$F_1 \rightarrow F_2$	75	89
$F_2 \rightarrow F_3$	327	234
$F_3 \rightarrow F_4$	165	235

の参照は，1 対 1 結合は DET となり，1 対多結合 (あるクラスの集合への参照) は RET となる．

Step3(Service Request の識別):
Service Request (SR) とは，アプリケーション内のクラスに属するメソッドのことを表す．ただし，抽象メソッドは数えない．

Step4(SR の複雑さを決定):DET（データ項目数），FTR（関連ファイル数）の 2 つのパラメータを用いて決定する．変数として参照される単データ項目 (integer,string など) とグローバル変数は DET とする．メソッドに参照されるクラスは FTR とする．

Step5(複雑さを数値に変換):各論理ファイル，SR の複雑さを用いて数値を算出する．

Step6(OOFP の算出):ILF，EIF，SR から算出した数値を全クラスについて合計し，アプリケーションの OOFP 値を決定する．

4 適用実験

4.1 類似アプリケーション開発

地方自治体の窓口業務を処理するアプリケーションのうち，「資格個人照会」「資格世帯照会」「賦課状況照会」「資格取得転入」の 4 つのアプリケーションについて，フレームワークを用いて開発されたものと用いずに開発されたものの各 2 個（計 8 個）の機能量と複雑さを計測する（図 1 参照）．

計測結果を表 1 に示す．4 つの機能のうち 3 つではフレームワークを用いた場合の機能量が，用いない場合の半分以下となっている．このことから，フレームワークの利用が類似アプリケーション開発時の機能量軽減に有効であると考えられる．

4.2 新機能追加

地方自治体の窓口業務を処理するアプリケーションにおいて，機能 F_1, F_2, F_3, F_4 を次のように定める．

F_1：資格個人照会，F_2：F_1 ＋資格世帯照会，

F_3：F_2 ＋賦課状況照会，F_4：F_3 ＋資格取得転入，

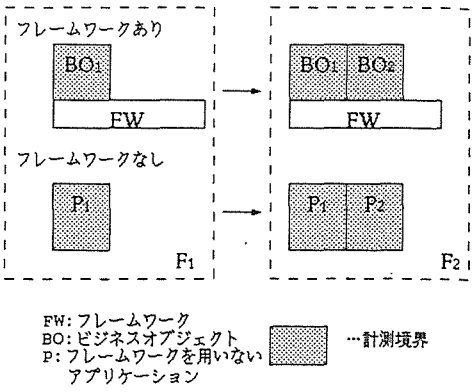

図 2: 新機能追加時の計測方法

F_1, F_2, F_3, F_4 の順に機能追加を行なうとき，フレームワークを用いて開発する場合と用いない場合とで，機能量と複雑さの増加を比較する（図 2 参照）．

計測結果を表 2 に示す．$F_1 \rightarrow F_2$ と $F_3 \rightarrow F_4$ では，フレームワークを用いた方が機能量の増加が少ない．しかし，$F_2 \rightarrow F_3$ については，フレームワークを用いた方が機能量の増加が多くなった．理由については現在検討中である．

5 まとめ

C-K メトリクス，OOFP を用いてフレームワークの効果を評価する手法を検討した．実際のアプリケーションを対象に OOFP を用いて機能量の計測と評価を行った．今後 C-K メトリクスについても計測を行ない，その計測結果を分析する予定である．

参考文献

[1] G. Caldiera, G. Antoniol, R. Fiutem and C. Lokan : *"Definition and experimental evaluation of function points for object-oriented systems"*, IEEE, Proc. of METRICS98, pp.167-178 (1998).

[2] S. R. Chidamber and C. F. Kemerer : *"A metrics suite for object-oriented design"*, IEEE Trans. on Software Eng., Vol.20, No.6, pp.476-493 (1994).

[3] I. Jacobson, M. Griss and P. Jonsson: *Software Reuse*, Addison Wesley, (1997).

[4] 増田, 坂本, 牛島 : "発展型ソフトウェア構築のためのデザインパターンの活用とその評価", コンピュータソフトウェア, vol.18, No.0, pp.4-18 (2001).