

ソフトウェア機能規模の自動計測の試み



研究ノート

楠本 真二*

Automatic Measurement of Software Functional Size

Key Words : Software Metrics, Function Point, Estimation

1. はじめに

近年、ソフトウェアはますます大規模化、複雑化、多様化してきている一方で、開発期間の短縮化も求められるようになってきている。したがって、高品質なソフトウェアを効率良く開発するために、開発計画の下で開発プロセスの全工程を系統づけて管理する必要性が高まってきている。

ソフトウェアプロジェクトの計画の際に重要な情報となる開発工数は、通常、ソフトウェアの規模を基にして予測される。このソフトウェアの規模を計測する手法として、ファンクションポイント(FP)法[1]が用いられている。

FP法は、ソフトウェアの機能的な規模を評価する手法として、1987年にAlbrechtによって提案された。現在、これをベースにIFPUG法[2]が考案され、広く利用されている。FP法は、要求仕様書や設計仕様書等からソフトウェアの機能要件だけを抽出して定量的にソフトウェアの規模を計測する手法である。

FPを基にして精度の高い見積りを行うためには、そのソフトウェア開発組織で過去に開発されたソフトウェアのFPや開発工数の実績値が基礎データとして充実していることが条件となる。基礎データが充実していない場合、過去のプロジェクトに対してFPを計測する必要がある。しかしこれを従来のよ

うに仕様書を基にして人の手によって計測する場合、(1)余分なコストがかかる、(2)仕様書が残存しない場合計測できない、(3)開発途中で発生した仕様変更が仕様書に反映されていない場合正確なFPを計測できない、といった問題が発生する。したがって、開発の終了したソフトウェアから自動でFPを計測する手法の開発は、効率よく過去のデータを蓄積する上で非常に有用である。本稿では、特定の構成で開発されたWebアプリケーションのソースコードからFPを自動計測する手法について紹介する。

2. IFPUG法

IFPUG法では、計測する機能をデータファンクションとトランザクションファンクションの2種類に分類する。そして、この2種類の機能をアプリケーションから抽出し、抽出された各機能に対してその機能の複雑さを基にしてFPを計算し、それらを合計することで、アプリケーション全体のFP(未調整FP)を計測する。以下に、IFPUG法によるFP計測の主要プロセスであるデータファンクションとトランザクションファンクションの計測方法を紹介する。

2.1 データファンクションの計測

データファンクション(DF)とは、ユーザが認識できる論理的なデータのまとまりである。DFは、内部論理ファイル(ILF)と外部インターフェイスファイル(EIF)に分けられる。ILFは、計測対象のアプリケーション内で更新されるデータの集合であり、EIFは、計測対象のアプリケーションによってデータが更新されることがなく、参照のみされるデータの集合を意味する。DFを構成するデータ項目数、レコード種類数により複雑度の重み付けがされる。



*Shinji KUSUMOTO

1965年9月生
大阪大学大学院基礎工学研究科博士前期課程修了(1990年)
現在、大阪大学大学院 情報科学研究科 コンピュータサイエンス専攻 教授
博士(工学) ソフトウェア工学
TEL : 06-6850-6605
FAX : 06-6850-6609
E-mail : kusumoto@ist.osaka-u.ac.jp

2.2 トランザクションファンクションの計測

アプリケーションに対するデータの出入りを伴う処理をトランザクションファンクション(TF)という。TFは、外部入力(EI)、外部出力(EO)、外部照会(EQ)に分類される。EIは、ユーザからのデータ入力によってDFの更新を行うことを主目的とする処理である。EOは、ユーザへのデータ出力を主目的とする処理のうち、出力データに何らかの処理をされた(計算や条件判断など何らかの加工がされた)データを表す。TFで処理されるデータ項目数やファイル数によって複雑度の重み付けがされる。

3. 提案手法

3.1 基本方針

FPをソースコードから計測する際、もっとも重要となるのは、機能がソースコード上にどのような形で実装されているかを決定することである。本研究では、適用対象を“MVCモデルで開発されたSQLを使用するWebアプリケーション”に限定した上で、DFやTFを以下のように定義する。

DF: データベース上のテーブル

TF: 画面遷移時に実行されるSQLの集合

DFはシステムによって使用されるデータの論理的集合である為、データベースのテーブルを対応させている。また、TFはユーザがシステムに対して行う入出力処理である為、Webアプリケーションにおいては画面遷移時に実行されるデータベースアクセス(SQL)の集合を対応させている。

3.2 計測手法

以下の手順でFPを計測する。

Step 1: 各画面遷移で実行されるSQLを抽出する
(図1の1.)

MVCモデルを構成しているフレームワークの設定ファイル(Strutsではstruts-config.xml)を解析して、画面遷移時に初めに呼び出されるメソッドを取得する。そして、このメソッドを起点としてソースコードを解析し、各画面遷移で実行されるSQLを収集する。

Step 2: SQLからDFを識別する(図1の2.)

Step 1で収集したSQLからDFを抽出する。DFはテーブルに相当するため、SQLで参照されているテーブルを抽出すればよい。しかし、システムの

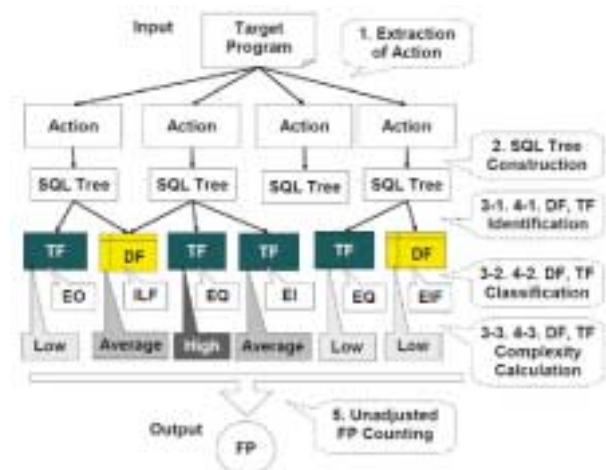


図1 提案手法の概要

実装上の理由のみで作られているようなテーブルは、ユーザに認識されないためDFとはみなされない。このようなテーブルは排除する必要がある。次に、以下のルールでDFをILF、EIFに分類する。
RuleD 1: データベースを更新するSQL(insertやupdate, deleteなど)でアクセスされている場合、ILFと判定する。

RuleD 2: RuleD 1を満たさない場合、EIFと判定する。

Step 3: SQLからTFを識別する

(図1の3-1~3-3.)

Step 1で抽出した各画面遷移のSQL集合をTFとして抽出する。次に、以下のルールでTFをEI, EO, EQに分類する。

RuleT 1: データベースを更新するSQLを一つでも持つ場合、EIと判定する。

RuleT 2: RuleT 1を満たさず、データベースから取得したデータに算術演算がなされていた場合、EOと判定する。

RuleT 3: RuleT 1, RuleT 2を満たさない場合、EQと判定する。

Step 4: 未調整FPの計算

(図1の4-1~4-3., 5.)

機能種別と複雑度から各機能のFPを計算し、それらを合計することでシステム全体の未調整FPを得る。なお、DFとTFの複雑度特定については文献[3]を参照されたい。

4. 評価実験

提案手法の有効性を評価するために、ある Web アプリケーションに対して提案手法を適用した結果について述べる。

提案手法を適用して得られたファンクションポイントと、設計仕様書からファンクションポイント計測の熟練者が計測したファンクションポイントと比較することによって提案手法の有効性を評価した。対象とした Web アプリケーションは、学生のグループが作成した図書管理システムである。

図書管理システムは、Struts を用いて開発された Web アプリケーションである。図書の貸し出しや管理する図書の追加などの機能を持っている。

表1に計測結果を示す。提案手法によって計測されたデータファンクション全てにおいて、熟練者が計測したデータファンクションと1対1対応していた。ツールによる結果も、熟練者による結果も11個のトランザクションファンクションを抽出した。ツールで抽出された各トランザクションファンクションは、分類、複雑さが異なるものが存在するが、熟練者によって計測されたトランザクションファンクションと1対1対応していた。

ログイン処理は熟練者による計測では外部出力に分類されていたが、ツールでは外部入力に分類された。この原因は、ツールでは新規ユーザを登録する SQL 文が含まれていたため、ログイン処理中に実行される SQL 文をまとめると外部入力に分類されてしまうためである。熟練者は、照会処理中にテーブル更新が存在する場合は外部出力と判断していた。外部入力との違いは、その機能の主目的が何かを考え、主目的がテーブルの更新ならば外部入力、出力にすることが主目的ならば外部出力として判断を

している。現在の分類条件の検討、改良を行うことが今後の課題の1つである。

表1 FP計測結果

		ツール	熟練者
DF	ILF	4	4
	EIF	0	0
TF	EIF	6	5
	EQ	0	1
	EQ	5	5
FP		65	68

5. まとめ

本研究では、Web アプリケーションの画面遷移時の処理に着目してファンクションポイントを計測する手法を提案した。

今後の課題として、より多くの Web アプリケーションに提案手法を適用すること、適用対象のソースコードの範囲を広げること、DFの自動分類手法の検討等があげられる。

参考文献

[1] International Function Point Users Group: Function Point Counting Practices Manual, Release 4.2.1 (2005).

[2] A. J. Albrecht: "Function point analysis", Encyclopedia of Software Engineering, 1, pp. 518-524 (1994).

[3] 赤池, 楠本, 英, 芝元: "ソースコードからのファンクションポイント計測とその適用", 電子情報通信学会技術研究報告, 107, 392, pp. 97-102 (2007).