プログラム実行時情報を用いたトランザクションファンクション抽出手法

森岡　佑[†]　　谷口　考治[†]　　楠本　真二[†]　　井上　克郎[†]
英　繁雄[‡]　　前田　憲一[‡]　　津田　道夫[‡]
大阪大学大学院情報科学研究科[†]　　株式会社日立システムアンドサービス[‡]

１．はじめに

ファンクションポイント(以下，FP)[1]は，ソフトウェアの持つ機能の数をもとに，そのソフトウェアの規模を測定する手法である．FP はソフトウェアの開発費用や工数等の見積もりに利用される．効果的な FP の利用には過去に設計されたソフトウェアの FP を計測する必要がある．しかし，過去のソフトウェアの設計仕様書が存在しない場合や，最終成果物で実装されている機能が設計仕様書に反映されていない場合があり，FP の測定が困難なものになっている．本研究では，最終成果物であるプログラム自身からファンクションポイントを測定する手法の開発を目的としている．その第一歩として，本稿では，プログラムからのトランザクションファンクションファンクションの計測手法について検討する．

２．ファンクションポイント(FP)

これまでに様々な FP 計測手法が提案されているが，本研究ではビジネスアプリケーション開発で標準的に用いられている IFPUG 法[1]を対象とする．

FP 計測では，データファンクション(以下，DF)とトランザクションファンクション(以下，TF)の抽出が中心的な作業となる．DF とは，アプリケーション中にあり，ユーザが認識できる論理的な意味でのデータのまとまりのことである．DF には内部論理ファイルと外部インタフェースファイルの２種類があり，それぞれのデータ項目数とレコード種類数という二つの要素によって低・中・高の三段階の複雑さに分類される．

一方，TF とは，アプリケーションに対するデータの出入りを伴う処理のことである．TF には外部入力(EI)，外部出力(EO)，外部照会(EQ)の３種類があり，それぞれの処理が扱うデータ項目数と関連ファイル数という二つの要素によって低・中・高の三段階の複雑さに分類される．

３．TF 抽出手法

本稿では，プログラムの実行時情報からシーケンス図を描画するツール Amida[2]から得られる実行時情報を用いて，実行時に動作しているオブジェクトと呼び出されているメソッドに着目して TF を抽出する手法を提案する．具体的には，DF として捉えることができるクラス(DF クラス)のメソッド呼び出しを，トランザクションファンクションを構成するキーとなるメソッドと仮定し，このメソッド呼び出しを抽出する(図１参照)． TF は DF に対して入出力を行う処理を指すことから，DF クラスとして指定したクラスのメソッドはデータ入出力を行うメソッドである可能性が高いという仮定に基づいている．同じ DF クラスの同じメソッドが何度か呼び出された場合は最初の 1 回のみを計測する．対象プログラムの実行履歴全ての走査終了後，検出されたメソッド群を出力する．

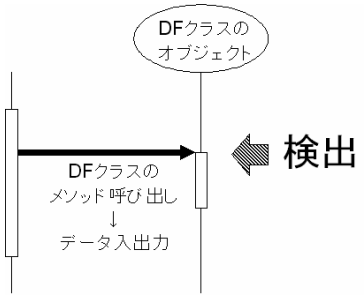

図１.検出されるメソッド

次に，検出された TF が EI，EO/EQ のどれに分類されるかを決める必要がある．そこで，入力処理，出力処理，照会処理に関係する部分文字列グループをそれぞれ指定し，実際にメソッド名に含まれている部分文字列から TF の分類を行う．

４．適用実験

4.1　計測対象

Extraction of Transactional Function Using Method Execution Trace Information
†Yu MORIOKA, †Koji TANIGUCHI, †Shinji KUSUMOTO
†Katsuro INOUE,
†Graduate School of Information Science and Technology, Osaka University
‡Shigeo HANABUSA, ‡Kenichi MAEDA, ‡Michio TSUDA
‡Hitachi Systems & Services, Ltd.

提案手法を用いることで，どの程度トランザクションファンクションを抽出できるかを評価するために，Java で記述された酒屋問題プログラムを対象に適用実験を行った．仕様書から，酒屋問題のプログラムには 7 個のトランザクションファンクションが含まれることが分かっている(表１参照)．このプログラムに対して，全てのトランザクションが含まれるテストデータを作成し，プログラムを実行して，実行履歴を得た．

表１．トランザクションファンクション

項番	タイプ	機能名
#1	EI	積荷の情報登録
#2	EO	在庫不足確認
#3	EI	在庫不足情報更新（削除）
#4	EO	在庫確認
#5	EO	在庫情報更新と出庫指示票出力
#6	EI	在庫不足情報作成
#7	EQ	在庫不足票出力

4.2 実験結果

上記のテストデータに対して，EI(入力)として識別するためにメソッド名に含まれるべき文字列として，“append”，“set”，“Repair”，“remove”，“decrease”，“add” の 6 個を，EO,EQ(出力)として識別するためにメソッド名に含まれるべき文字列として，“display”，“check”，“stockCount”，“Out” の 4 個をそれぞれ指定し，DF クラスのメソド呼び出しを計測すると，最終的に 13 個のメソッドが残った．各トランザクションと 13 個のメソッドとの対応をまとめたものを表 2 に示す．

表２．トランザクションとメソッドの対応

トランザクション名	クラス名	メソッド名
積荷の新規登録	BrandList	appendNode
	DataBase	addContainer
	SakeDBList	appendNode
在庫不足確認	StockLackList	checkStockLack
在庫不足情報更新	StockLackList	removeNode
在庫確認	DataBase	stockCount
	SakeDBList	stockCount
在庫情報更新と出庫指示票出力	Container	decreaseBrandCount
	Brand	brandCountRepair
	BrandList	removeNode
	DataBase	getOutIndicate
在庫不足情報作成	StockLackList	appendNode
在庫不足票出力	StockLack	display

５．考察

今回行った実験では，「在庫情報更新と出庫指示票出力(EO)」の TF に対応するメソッド 4 個のうち 3 個が，メソッド名から EI として識別されている．これは，設計仕様書からの FP 計測において「在庫情報更新と出庫指示票出力」の TF の主たる目的は出庫指示票出力であり，在庫情報更新は付随的な処理と判断されたためである．しかし，実験において在庫情報更新を行うメソッドも抽出したため，このような矛盾が生じた．

適用実験によって，TF によっては複数のキーとなるメソッドで構成されるものがあることがわかった．したがって，複数のメソッドで構成される TF を 1 つの TF として判断する必要がある．これは，実行履歴中のある一定の範囲で複数のキーとなるメソッドを抽出した場合に，抽出されたメソッド群をグループ化し，1 つグループにつき 1 つの TF を割り当てるといった処理を行うことで解決することができると考えている．グループ化する範囲は，対象プログラムのユーザからシステムへの何らかの入力があったときに必ず呼び出されるクラス(GUI クラスなど)を境界として利用することを想定している．

また，今回のアルゴリズムで用いた，メソッド名のキーワードによる TF の種類の分類では EO と EQ を分類することが困難である．これらを分類するには，出力されるデータがどのような処理を受けて出てきたのかを知る必要があるので，対象プログラムのソースコードの静的解析などにより，データ依存関係を把握することによって分類が可能になると考えられる．

６．まとめと今後の課題

本稿では，プログラム実行時情報を用いて，DF クラスのメソッド呼び出しを検出することによってトランザクションファンクションを抽出する手法を提案した．考察で述べた，複数のキーメソッドで構成される TF の識別方法や EO と EQ の分類アルゴリズムの考察，およびトランザクションとメソッドの対応の自動化が今後の課題である．

文献

[1]IFPUG: “Function Point Counting Practices Manual, Release 4.2”, International Function Point Users Group(2005).

[2] 谷口他: Java プログラムの実行履歴に基づくシーケンス図の作成，ソフトウェア工学の基礎ワークショップ(FOSE2004)，pp.5-15，11 月 2004.