細粒度履歴を用いたプログラムの作成過程の再現環境

橋浦研究室　　1135308 髙山　源貴　　1135446 山田　善隆

1. はじめに

ソフトウェア開発は複数人によるプロジェクトを組んで実施されることが多い．プロジェクトは当初の目的を達成すると解散してしまうため，保守や改良等については，必ずしも開発を行った当人が行うとは限らない．そのような場合には，それが書かれた背景にある原作者の意図を汲み取らないと，思わぬ退行等が生じる原因となることがある．

2. 研究目的

前述のような事態を防ぐために，一般的にはドキュメントが用意されることが多い．しかし，ドキュメントには不備がある場合も多い．著者らの所属する大学内で行われているPBL[1]においても，プロジェクトの引継ぎが行われているが，そのような傾向が顕著である．

ここで，本研究での原作者の意図について述べる．例えば図１の例のようにプログラムにリファクタリング（メソッドの抽出）[2]を施し，`x+y` 部分を `sum(x, y)` のように書き換えたとする．

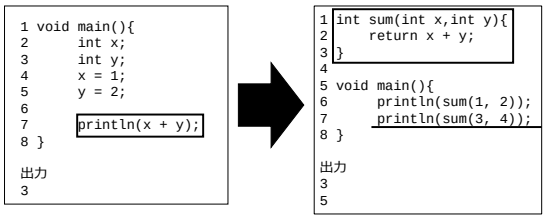

図１メソッド抽出の例

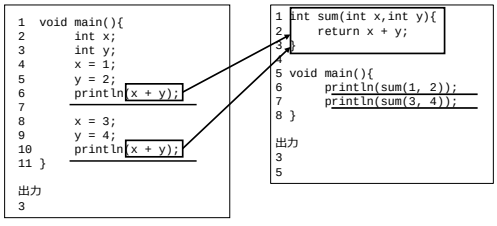

図２メソッド抽出の変更過程

このようなメソッドの抽出はプログラムから何度も再実行される典型的な処理をメソッドとして分離し，再利用を可能にするものであるが，分離されたメソッドがどのような処理から分離されたか分からない場合，うまく使いこなすことができない．

そこで，図２のような変更過程を見ることでどのような処理を分離させたのかがわかる．

本研究では，このようなプログラミング作業の過程を細粒度で自動的に記録しておき，保守や改良時にそれを再生することで，原作者ではないプログラマが，プログラムの変更の意図を推測することを支援する．

3. 提案手法

本研究では，統合開発環境（ IDE ）上で細粒度作業履歴の記録と再現を行う環境を構築する．同時に，ソフトウェアのメトリクスについても記録，可視化することより，ソフトウェアのどのような変更がどのような品質にどのような変化をもたらしたかを明確化する．

以下に，本手法を実現するための各機能について述べる．

3.1. 細粒度作業履歴の記録

本機能は，ファイルの内容が変更されたとき，変更された内容を時間ごとに記録する．

記録された細粒度作業履歴は，テキストファイルとして保存する．

3.2. 細粒度作業履歴の再現

本機能は，保存されたテキストファイルから細粒度作業履歴を時間ごとに再現し，ボタンとスライダーで操作することができる．

ボタンは，再生と停止．スライダーは，操作することで任意の時点を閲覧することができる．

3.3. 品質指標の可視化

本機能は，時間ごとのプログラムの入力量とメソッドの変更履歴を表示する．

また，選択された時間のメソッドのサイクロマチック数を表示する．

4. 実現方法

前項で述べた機能を実現するために，「細粒度作業履歴の記録ができるプラグイン」，「記録した細粒度作業履歴からプログラムの作成過程を再現するプラグイン」，「品質指標の可視化をするプラグイン」を Eclipse プラグインとして実装を行った．

5. 実験

日本工業大学工学部情報工学科に所属する 6 名を被験者とする実験を行った．まず，被験者には問題プログラムと完成時の出力結果が与えられる．この問題プログラムには予め作成過程を記録プラグインで記録しておいたメソッドが定義されており，被験者は，そのメソッドを用いてプログラムを完成させなければならない．再生プラグインを用いることができる場合，被験者は予め定義されているメソッドが作成される過程を閲覧することができる．

各被験者につき，再生プラグインを用いることができるものとできないものを 1 問ずつ実施し，プログラムの変更回数と最終結果のまとめを表 1 に示した．

6. 結果と考察

表 1 より，プラグインを使った場合でも使わなかった場合でもプログラムを完成できた被験者はいなかったが，変更回数を確認できた被験者全員がプラグインを使った場合の変更回数が少ないことが確認できた．

実験の結果から，問題 1 と問題 2 のどちらもプラグインを使った場合に変更回数が減少するという結果を確認した．これは，被験者が再生プラグインを用いて原作者の意図を読み取ることができたため，無駄な試行錯誤が減少したためであると考えられる．

7. まとめと今後の課題

本研究ではソースコードの作成過程に着目し，細粒度作業履歴の記録と再現を行う環境と品質指標の可視化を行う環境を構築し，実験の結果によって無駄な試行錯誤が減少したことから，プラグインを使用することでプログラムの変更の意図を推測することを支援できたと示唆される．

今後は，実際の PBL 等でデータの蓄積を行い，有効性を確認したい．

参考文献

[1] 粂野 文洋，辻村 泰寛，大木 幹雄，山地 秀美 ″現実の地域課題解決を対象としたソフトウェア開発 PBL の実践,″ 情処論 教育とコンピュータ，Vol.2，No.1，pp.25-40，Jun. 2016.

[2] Martin Fowler ″リファクタリング プログラミングの体質改善テクニック,″ ピアソンエディケーション，May. 2000.

[3] 柳 正純，門田 暁人，高田 義広，鳥居 宏次 ″バグ混入時のプログラミング行動の特徴を検出するツールの試作,″ 電子情報通信学会技術研究報告，Vol.94，No.334，pp.9-16，Nov. 1994.

[4] 槇原 絵理奈，藤原 賢二，井垣 宏，吉田 則裕，飯田 元 ″初学者向けプログラミング演習のための探索的プログラミング支援環境 Pockets の提案,″情報処理学会論文誌，Vol.57，No.1，pp.236-247，Jan. 2016.

[5] 松澤 芳昭，岡田 健，酒井 三四郎 ″Programming Process Visualizer:プログラミングプロセス学習を可能にするプロセス観察ツールの提案,″ 情報教育シンポジウム 2012 論文集，Vol.2012，No.4，pp.257-264，Aug. 2012.

表 1 実験結果

	問題 1			問題 2		
被験者	プラグイン	完成・未完成	変更回数	プラグイン	完成・未完成	変更回数
A	使用	未完成	33	未使用	未完成	53
B	使用	未完成	22	未使用	未完成	28
C	使用	未完成	63	未使用	未完成	-
D	未使用	未完成	164	使用	未完成	119
E	未使用	未完成	172	使用	未完成	76
F	未使用	未完成	46	使用	未完成	38