

B-04

類似の差分の比較によるソースコードの特徴と読解時間の関係の分析 An Investigation on Relationship Source Code Characteristics and Comprehension Time Using Similar Source Code Chunks

西園 和希† 森崎 修司† 松本 健一†
Kazuki Nishizono Shuji Morisaki Ken-ichi Matsumoto

1. はじめに

保守開発や派生開発をはじめとして、既存のソフトウェアを改変したり機能追加したりして、新しいバージョンのソフトウェアを開発する形態が増えている。そのような開発では、既存部分を理解して、変更、拡張する必要があり、非常に小さな改変であっても、既存ソフトウェアの理解に時間がかかる場合が多い。

本研究では、ソフトウェア理解の 1 つであるソースコード読解を効率化することを目指す。ソースコード読解に必要な知識、ソースコードの特徴や規模から読解に必要なスキルや時間を予測できれば、適した読解者を選定する、または読解者に当該技術に関するトレーニングを行った上で読解を行うなどすることで、ソースコード読解時間の短縮、すなわち理解工数の削減が可能となり、結果として開発効率の向上が期待できる。本稿では、規模が類似しているが、読解に GUI フレームワーク/ライブラリの知識が必要なソースコード片と GUI フレームワーク/ライブラリの知識が不要なソースコード片を対象として、当該知識を有する実務者とそうでない実務者を被験者として読解時間を計測し、その結果を分析する。

2. 実験

イベントドリブン・プログラミングの経験、および Java の Applet・Swing の知識を必要とするソースコードの読解において、それらを備えた読解者とそれ以外の読解者の間で読解時間に差が生じるかどうかを確認するため、ソースコードの読解実験を行った。読解対象には GUI の知識を必要とするものと必要でないものを用意し、それぞれについて GUI の知識を持つ被験者と持たない被験者による読解時間を計測し、比較を行った。本章ではこの実験の内容について述べる。

2.1 対象ソースコード

本実験で用いる読解対象として、Java で記述されたペイントアプリケーションを用意した。その規模は、空白行を除いたコード行数が 1620 行、クラス数が 38 個で各クラス一つあたり平均 2.3 個のメソッドを持つ。

本実験では、実験用に開発したバージョン 1 のペイントアプリケーションのソースコードと、変更を加えた後のペイントアプリケーションのソースコードの 2 種類を用いる。変更前を Version1、変更後を Version2 と呼ぶ。変更は 5 種類を用意してあり、それぞれ独立に Version1 に適用さ

れる。被験者には、Version1 のソースコードと、Version1 と Version2 との差分が与えられる。差分は 5 個あり、それぞれに変更内容が記されている。

差分にはそれぞれ GUI の知識の必要・不要などの特徴が付けてある。この特徴の違いと、GUI の知識を持つ被験者および持たない被験者の読解時間との関係を調べることで、ソースコード内で用いられる技術に関する知識の所持・不所持による読解時間の差を調査する。また、差分の読解時間の比較にあたり、読解対象、すなわち差分の規模による読解難度の変化による影響を考慮するため、行数の少ない差分と多い差分の 2 グループを用意し、それぞれの中で GUI 知識の必要な差分、不要な差分を用意した²⁾。次節で、各差分の概要を述べる。

2.1.1 行数の少ない差分

小さな規模の差分として、3 つの差分を用意した。これを、A1, A2, A3 とし、詳細を表 1 の 2 行目から 4 行目に示す。表の各行が差分一つに対応する。3 つの差分はいずれも追加行数、削除行数ともに 10 行以下で、行数が少ない点で共通している。A1 はイベントリスナを 1 つ追加する。A2 は Swing コンポーネントのメソッドへ渡す引数値を変えることで、GUI の挙動を変化させる。A3 は浮動小数点数が座標の途中計算に使用されていることによって発生した演算誤差による欠陥を修正するため、計算式を変更する。A3 では座標計算が絡むものの、その計算に GUI の知識は必要ない。よってこの 3 つの差分の中で、速く読むために Swing および Applet の知識を必要とする差分は、A1 および A2 である。

2.1.2 行数の多い差分

大きな規模の差分として、2 つの差分を用意した。これを B1, B2 とし、詳細を表 1 の下 2 行に示す。B1, B2 共に追加行数が 100 行を超えており、行数が多い点で共通している。その内容はどちらも新たな機能をペイントアプリケーションに追加するというもので、B1 は GUI コンポーネントの状態変更の監視やキャンバス上の色取得、および特定色の変更など、その内容は多岐に渡り、変更の対象となるクラスも 7 つと多いため、読解難度は高い¹⁾³⁾。対して B2 は 9 つの GUI コンポーネントを追加しているが、変更の対象となるクラスは 1 つである。B1 は Version1 の全体構造をしっかりと把握しておく必要があるが、B2 は 1 つのクラスの詳細を把握しておけば、読むことはできる。

†奈良先端科学技術大学院大学 情報科学研究科,

表 1 差分の詳細

差分名	内容	追加行数	削除行数	説明
A1	仕様変更	6	0	マウスに関するイベントリスナを追加
A2	仕様変更	1	1	Swing コンポーネントのメソッド引数の値を変更し GUI の挙動を変更
A3	欠陥修正	2	2	座標計算部分における浮動小数点数の演算誤差による欠陥を修正
B1	機能追加	174	37	処理は多岐に渡り、変更対象のクラスは 7 つ
B2	機能追加	263	5	9 つの GUI コンポーネントを 1 つのクラスに追加

表 2 使用経験と背景知識による分類

分類名	分類方法
GN	GUI アプリケーションの開発経験なし
GS	イベントドリブン・プログラミングの仕組みは知っている
GM	GUI アプリケーションの開発経験がある

2.2 実験の流れ

被験者は実務でソフトウェア開発に携わる技術者であるが、プログラミング経験年数や開発言語などに条件は設けていない。被験者は自らの GUI の知識についてのアンケートに回答した後、読解対象であるソースコードを与えられ、まず Version1 の読解を行う。およそ全体目を通して仕組みが理解できたところで、差分の読解に移る。被験者は、事前に指示された通り、差分の適用に問題がないかどうかを確かめるという目的をもって読解する。被験者は、差分の内容を理解し、その差分を適用して問題がないか、問題があるならばそれはどのような問題をもたらすかを回答する。

2.3 取得データ

Version1 については、各クラスの読解に要した時間、差分についても各差分の読解および適用の可否の決定に要した時間を回答してもらった。また、アンケートを事前に行った。アンケートの質問項目には次が含まれる。

「GUI プログラミングに関するご経験を以下より 1 つお選びください。1、豊富な経験がある（イベントドリブン・プログラミングの経験がある）。2、多少の経験がある（イベントドリブン・プログラミングがどのようなものか知っている）。3、経験なし」

3. GUI に関する知識による被験者の分類

本実験で題材として用いた Version1 および差分は、GUI、ここでは Java の Swing、AWT ライブラリの知識を必要とするコードを多く含む。そこで、被験者を GUI の知識によって 3 つのグループに分けて比較をすることで、GUI の知識を持っていることが GUI の知識を必要とする差分の読解を助けるかを確かめる。

3 つのグループについて述べる。1 つ目のグループを、GUI アプリケーション開発の経験がない被験者を集めて作り、GN とする。2 つ目のグループを、GUI アプリケーション開発の経験はないがイベントドリブン・プログラミングの知識はある被験者の集合とし、GS と呼ぶ。最後に GUI アプリケーション開発の経験がある被験者でグループ

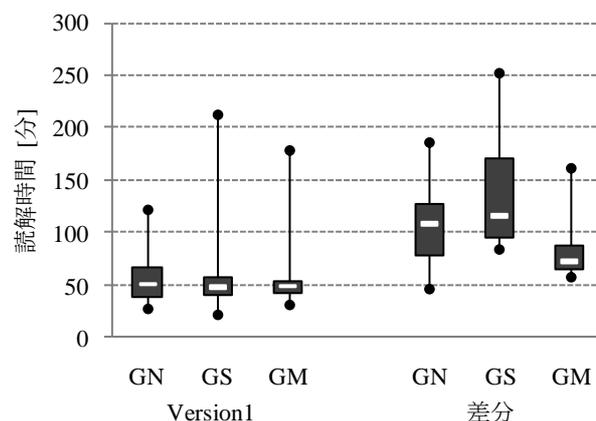


図 1 GUI 知識別の読解時間分布の比較

を作り、GM とする。以上の分類を表 2 に示す。表の各行が 1 グループを表し、3 行で 3 つのグループの分類名および分類方法を示す。

4. 実験結果

4.1 Version1 および差分ファイル差分の読解時間分布

表 2 に示す 3 つのグループ別に、用意した実験の題材に対する合計読解時間の分布を調べることで、GUI の知識の程度が読解時間に与える影響を調査する。

GUI の知識によるグループ GN、GS、GM について、Version1 および差分の読解時間の分布を比較したグラフを図 1 に示す。縦軸に読解時間を取り、左 3 本のグラフが Version1、右 3 本のグラフが差分全体の読解時間の分布を示し、それぞれ左側が GN、中央が GS、右側が GM となっている。まず、箱の上端は、そのグループに属する被験者の読解時間について、75 パーセンタイルに位置する値を示す。同じく箱の下端は 25 パーセンタイルを示し、箱の中の白抜き線は中央値を指す。箱から飛び出た線分の上端は 95 パーセンタイル、下端は 5 パーセンタイルを示し、上位 5 パーセントと下位 5 パーセントは外れ値として取り除いてある。

Version1 は、各グループ間で読解時間の中央値に違いはなかった。差分では、GN の読解時間の中央値が GM の読解時間の中央値の 1.49 倍であった。また、Version1、差分とともに、読解時間の 25 パーセンタイルから 75 パーセンタイルの分布範囲が最も狭いのは GM であった。

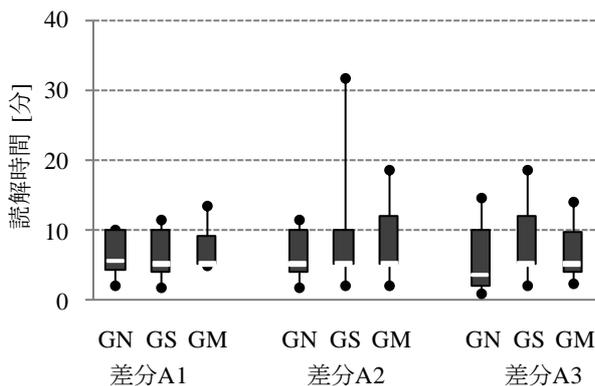


図2 GUI知識別の小さな差分の読解時間分布の比較

4.2 行数の少ない差分による実験の結果

行数の少ない差分のグループに含まれる A1, A2, A3 について、GN, GS, GM の 3 つの被験者グループ毎の読解時間の分布を調べた。結果を図 2 に示す。縦軸に読解時間を取り、グラフは左から 3 本ずつ差分 A1, A2, A3 の読解時間の分布を示す。各差分のグラフは、左から GN, GS, GM に属する被験者で分けられている。箱や線分の指す値の意味は図 1 と同じである。

A1, A2, A3 とともに、GUI に関する豊富な知識を持つ被験者のグループである GM の読解時間の中央値が、そうではない被験者のグループである GN, GS のそれに比べて特に小さいという傾向は見られなかった。25 パーセントから 75 パーセントの分布範囲を示す箱の位置に関しても、GM とそれ以外で大きな差はみられなかった。

4.3 行数の多い差分による実験の結果

行数の多い差分のグループに含まれる B1, B2 それぞれについて、GN, GS, GM の 3 つの被験者グループ毎の読解時間の分布を調べた。結果を図 3 に示す。縦軸が読解時間で、左 3 本のグラフが B1 の読解時間の分布を、右 3 本のグラフが B2 の読解時間の分布を表す。箱や線分の指す値の意味は図 1 と同じである。

B1, B2 共に、読解時間の中央値が最も小さいのは、GUI に関して豊富な知識を持つ被験者グループである GM であった。その傾向が顕著に出たのは B1 で、GN の読解時間の中央値が GM のその 2 倍となった。B2 では 1.4 倍であった。B1 では 95 パーセントの被験者の読解時間も GM が最も短く、GN の 0.57 倍、GS の 0.49 倍であった。

4. 考察

GUI の知識がある被験者の方が、差分全体を速く読める傾向がみられた。その理由として、与えられた差分の中に GUI の知識を要するものがいくつか含まれており、GUI の知識がある被験者はそうでない被験者に比べてそれらを短い時間で読むことができたため、差分全体の読解時間に差が生じたと考えられる。この場合は当然、GUI の知識をそれほど必要としない差分であれば、その読解時間に GUI の知識による違いは生じない。次に、GUI の知識

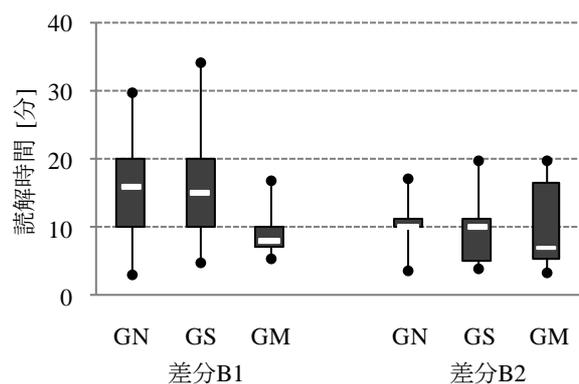


図3 GUI知識別の大きな差分の読解時間分布の比較

を有する被験者が速く読むことができるソースコードの特徴について考察する。

小さなサイズの差分について読解時間の比較を行ったところ、GUI の知識の必要・不要に関わらず、GUI の知識を持った被験者が特別に速く読めるという傾向は見られなかった。

しかし、大きなサイズの差分について同じく比較を行ったところ、GUI の知識が必要でかつ複雑な処理を含む差分 B1 において、GUI の知識を有する被験者が速く読める傾向がみられた。この点について詳しく分析する。GUI の知識が必要で、変更箇所が Version1 の広範囲に分散している B1 では、GUI の知識を持っている人であれば速く読むことができた。対して、GUI の知識が必要ではあるが、ただ単にコンポーネントを追加していくだけであり、Version1 に関してもそのクラスさえ理解しておけば問題ない B2 では、GUI の知識を持っている被験者の方が多少は速く読めるものの、GN, GS との違いは B1 ほど顕著ではない。

すなわち、GUI の知識を持った読解者がそうでない読解者に比べて特別に速く読むことのできる差分とは、GUI の知識を必要とし、かつ規模が大きく、その変更範囲が広く複雑な差分であるといえる。そして、この条件に当てはまるソースコードのレビューを、GUI の豊富な知識を有する技術者に割り当てることによって、レビュー工数の削減につなげることができると期待される。

5. まとめ

GUI の知識が必要となるソースコードの読解の際、当該知識の有無が読解時間に影響を与えるかを確認することを目的とし、実務者を対象とした実験を実施した。読解対象のソフトウェアは本実験のために作成したペイントアプリケーションであり、Version 1 と機能追加や変更を実施した Version2 を用いた。被験者には Version1 のソースコード、及び、Version 2 と Version 1 の差分 5 つが与えられる。5 つの差分は規模によって大きく 2 種類に分けられ、それぞれの中で GUI フレームワーク/ライブラリの知識を必要とするものと不要なものに分けられる。また、あらかじめ被験者にはアンケートによって GUI フレームワーク/ライブラリの知識の有無を 3 段階で回答してもらった。

実験の結果、GUI フレームワーク/ライブラリの知識を必要とする差分の読解では、GUI アプリケーション開発の経験があると回答した被験者と、そうでない被験者の間に差はなかった。GUI の知識を必要とするものの他に、読解の分量が多いこと、内容が単純な記述の繰り返しでないこと、という 2 つの条件が加わったソースコードの読解において、GUI の知識を有する被験者の読解時間が、そうでない被験者のそれを大きく下回った。

また本稿では、読解に GUI の知識を必要とする差分と、そうでない差分の 2 種類があることを前提に実験を行ったが、この 2 つの違いを主観的ではなく客観的な指標に基づいて定義する必要がある。定義方法として、差分における GUI に関するライブラリの参照回数を数え、GUI の知識が必要か否かの判断に使うなどが考えられる。

謝辞

本研究の一部は、文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われた。また、本研究の一部は、文部科学省科学研究補助費（若手 B:課題番号 21700033）による助成を受けた。本論文の分析対象データの一部は IBM Academic Initiative Program (<http://www.ibm.com/developerworks/university/academicinitiative/>)の支援によって収集された。

参考文献

- 1) HC Benestad, B Anda and E Arisholm: Understanding cost drivers of software evolution: a quantitative and qualitative investigation of change effort in two evolving software systems, *Empirical Software Engineering*, Vol.15, No.2, pp.166-203 (2010)
- 2) TL Graves, AF Karr, JS Marron and HP Siy: Predicting fault incidence using software change history, *IEEE Transactions on Software Engineering*, Vol.26, No.7, pp.653-661 (2000)
- 3) AE Hassan: Predicting faults using the complexity of code changes, *Proceeding of the IEEE 31st International Conference on Software Engineering*, pp.78-88 (2009)