

視線追跡を用いたプログラミング初学者向けコーディング規約の 妥当性定量評価に関する研究

- ネスト最小化と制御構文選択の影響分析 -

A Study on Quantitative Validity Evaluation of Coding Conventions for Beginning Programmers Using Eye Tracking

- Minimizing Nesting and Analyzing the Impact of Control Syntax Selection -

渡邊 雄太^{*1}, 榎野 真道^{*1}, 松本 慎平^{*1}

Yuta WATANABE^{*1}, Masamichi MASUNO^{*1}, Shimpei MATSUMOTO^{*1}

^{*1} 広島工業大学情報学部

^{*1} Faculty of Applied Information Science, Hiroshima Institute of Technology

Email: bm22255@cc.it-hiroshima.ac.jp

あらまし : 本研究では、ソースコードの可読性ルールが実際に理解度にどのような影響を与えるかを調査することを目的とする。具体的には、ソースコードの可読性に関するルールであるネストの最小化と、制御構文である for 文および while 文が可読性や理解度に与える影響について、視線データおよび認知負荷アンケートを用いて調査を行った。実験の結果、ネストの最小化については、特定のタスクのみであるが、ネスト構造を用いたコードと複合条件のコードの正答率および注視時間に有意な差が示された。これは、Johnson らの主張とは逆の結果であった。複合条件を用いた場合、回答時間が有意に短いタスクがあった。for 文と while 文においては、コードの正答率および注視時間に有意な差は示されなかった。

キーワード : 視線追跡, プログラミング, コーディング規約, ネスト最小化

1. はじめに

ソースコードの可読性を向上させるためのコーディング規約に関する近年の研究では、様々な調査が行われている。例えば、注釈付きのソースコードの提供や、ソースコードの構造を視覚的に示すことが読解を支援することが報告されている。しかし、ソースコードの可読性を高めるためのコーディング規約の根拠については十分に言及されておらず、各規約が合理的である理由付けは行われているが、それを定量的に明らかにした取り組みは十分ではない。科学的な検証が不足していることで、規約が流行などの影響を受ける場合があり、その結果、規約が頻繁に変更されるという事態が発生する。これは開発現場に混乱をもたらす可能性がある。したがって、規約の妥当性や適切性について、科学的な根拠に基づいた評価が求められている。

そこで本研究では、ソースコードの可読性ルールが実際に理解度にどのような影響を与えるかを調査することを目的とする。本研究のリサーチクエスションは、「ネストを最小化することや for 文と while 文の適切な使い分けが、初学者のコード理解にどのような影響を与えるのかを、視線データから解明できるか?」と位置付ける。視線追跡技術を用いることで、初学者がコードを読解する際の視線の動きを記録し、どの部分に注目しているかを明らかにする。これにより、ネストの最小化が理解時間の短縮や理解度の向上に寄与する理由を定量的に捉えることができる。得られた結果は、初学者に対する適切なコーディング規約のあり方の検討に貢献できる。

2. 先行研究

学習を目的とした読解方法を分析する一つの手法として、アイトラッキング（視線運動の追跡）が広く用いられている⁽¹⁾。Rayner の研究によれば、視線移動を通じて認知過程を推論することが可能であるとされている⁽²⁾。プログラミングに関する研究分野でも、アイトラッキングを活用したさまざまな研究が行われており、具体的には学習者のスキル評価や、ソースコード特有の困難度を分析する取り組みが進められている⁽³⁾。

今回参考にした先行研究では、ネストの最小化と do-while 文の回避が開発者のコード理解にどの程度寄与するかが調査されている⁽⁴⁾。この研究では、可読性に関するルールに従ったコードであるかどうか、コードが意図された機能を正確に実行するかどうか、エラーやバグの有無などの観点から、ソースコードを4つのカテゴリに分類して分析が行われた。評価指標として、コードの理解時間、理解度、バグの発見能力、理解に対する信頼度の4つが用いられた。分析の結果、ネストの最小化はコードの理解時間を短縮し、理解への信頼度を向上させるとともに、バグを発見する能力の向上にも寄与することが示された。一方で、do-while 文の回避はコード理解度に対して有意な影響を与えないことが明らかになった。

ただし、do-while 文は初学者にとって使用頻度が低く、実際の学習や開発において触れる機会が少ない。そのため、本研究では do-while 文を対象外とし、初学者が学習する機会が多い for 文と while 文の比較に焦点を当てた。また、先行研究では、実際に開発経験を持つ大学生や現場のエンジニアを対象に調

査が行われている。これに対し、本研究では大学生を対象とする。これにより、初学者向けに特化したコーディング規約が理解にどのような影響を与えるのかに焦点を当てた。

3. 実験方法

本実験の流れを以下に示す。被験者は、C 言語や Java の基本構文、およびアルゴリズムの基礎を理解している大学3年生および4年生の16名とした。

実験では、被験者にネストおよび複合条件に関するコード、ならびに for 文および while 文の使い分けに関するコードの読解をタスクとして提示した。各タスクでは、コードを実行した際の変数の値を解答させ、その正答率および解答に要した時間を計測した。

ネストおよび複合条件に関するタスクでは、ネストの最小化ルールに従ったコードと複合条件を用いたコードの2種類の問題をそれぞれ3問ずつ用意した。被験者は、①ネストの最小化ルールを適用した問題を先に解くグループと②複合条件の問題を先に解くグループの2つに分けられた。for 文と while 文の比較においても、同様の手順でタスクを提示した。各被験者は、一方のタスクに取り組んだ後、1週間以上の期間を置いた後、もう一方のタスクに取り組んだ。この手順により、ウォームアップ効果を防ぎ、課題の提示順が結果に与える影響を排除した。

本実験では、アイトラッカー Tobii Pro Nano を使用し、被験者の視線データを記録した。また、各課題における正答率および注視時間を計測し、得られた視線データと統合的に分析を行った。具体的には、各課題における被験者の正答率と視線の動きをもとに、ソースコードの読解プロセスに関するデータを収集し、視線の動きと正答率などの指標との関連を分析した。そして、ネストの最小化ルールや、for 文と while 文の使い分けがコードの理解に与える影響について調査した。

4. 結果及び考察

4.1 ネストと複合条件の比較

視線の注視時間について、Welch の t 検定 (両側) によると、特定のタスクにおいてのみではあるが、複合条件のコードはネストよりも有意に短かった ($p < .01$)。コードの理解と視線の注視に正の相関があると仮定すると、この結果は、直感的には可読性が低下したと解釈できる。Johnson らは、ネストを用いることで可読性が向上すると主張しているため⁽⁴⁾、本研究では Johnson らの主張とは逆の結果が示されたことになる。ただし、Johnson らは職業プログラマを対象としていたのに対し、本研究では非職業プログラマを被験者としている。このことを踏まえると、本研究の結果は Johnson らの主張に追加の情報を提供しており、その点で意義があると考えられる。プログラミング学習者にとっては、ネストによる明示

的な構造が可読性向上につながる可能性がある。今後、教育的観点からネストの有用性についてさらなる検討が求められる。

解答時間及び正解率については、いずれのタスクにおいても、複合条件のコードとネストのコードに有意な差は見られなかった。教育現場で扱う程度の規模のコードの場合、コード行数の短縮は限定であるため、差は示されなかったと考えられる。

認知負荷の中でも課題内在性負荷については、Welch の t 検定によると、複合条件のコードはネストのコードよりも有意に低かった ($p < .05$)。この結果は、複合条件の方が主観的な負担が少ないことを示唆している。注視時間の結果を踏まえると妥当な結果であると考えられる。

4.2 for と while 文の比較

正解率および回答時間では、for 文と while 文に有意な差は見られなかった。また、注視時間についても、for 文と while 文に有意な差は見られなかった。一方で、1回の視線移動あたりの移動距離の平均値では、for 文は while 文よりも有意に大きかった ($p < .01$)。これは、for 文の場合情報が横方向に展開されているため、視線移動が増えた可能性が示唆される。なお、認知負荷については、for 文と while 文の間に有意な差は確認されなかった。

5. おわりに

本研究では、ソースコードの可読性に関するルールであるネスト・複合条件と、制御構文である for 文・while 文の違いが可読性や理解度に与える影響を明らかにするため、視線データ、回答時間、正解率、認知負荷アンケートの結果とを関連付けて分析した。その結果、限定的ではあるが、プログラミング学習者特有の傾向を明らかにできた。

謝辞

本研究は、独立行政法人日本学術振興会科学研究費助成事業(基盤研究(C)23K02697, 22K02815)の助成を受けて実施した成果の一部である。

参考文献

- (1) E. D. Reichle, K. Rayner, A. Pollatsek, The ez reader model of eye-movement control in reading: Comparisons to other models, *Behavioral and Brain Sciences*, 26(4), pp. 445-476 (2003).
- (2) K. Rayner, Eye movements in reading and information processing: 20 years of research, *Psychological Bulletin*, 124(3), p. 372 (1998).
- (3) U. Obaidallah, M. Al Haek, P. C. H. Cheng, A survey on the usage of eye-tracking in computer programming, *ACM Computing Surveys (CSUR)*, 51(1), pp. 1-58 (2018).
- (4) J. C. Johnson, S. Lubo, N. Yedla, J. Aponte, B. Sharif, An Empirical Study Assessing Source Code Readability in Comprehension, 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 513-523 (2019).