

プログラミングにおける「より良いコード」記述のためのモデル化と コード共有プラットフォームでの妥当性検証

Modeling for Writing "Better Code" in Programming and its Validation on Code Sharing Platform

前田 新太郎^{*1}, 古池 謙人^{*2}, 東本 崇仁^{*3}
Shintaro MAEDA^{*1}, Kento KOIKE^{*2}, Takahito TOMOTO^{*2}

^{*1} 千葉工業大学大学院情報科学研究科

^{*1}Graduate School of Information and Computer Science, Chiba Institute of Technology

^{*2} 千葉工業大学附属研究所

^{*2}Research Liaison Centre, Chiba Institute of Technology

^{*3} 千葉工業大学情報変革科学部

^{*3}Faculty of Innovative Information Science, Chiba Institute of Technology

Email: front4.shintaro@gmail.com

あらまし：本稿では、プログラミングにおける「より良いコード」を記述するための学習活動を、他者のコードを取り入れることを伴う改善サイクルとしてモデル化した。これまでの提案システムを用いた実践的評価からログ分析により、モデルに基づいたコードの改善が観測された。この結果から、提案システムが「より良いコード」を記述するための学習活動を促進する可能性が示唆された。

キーワード：コード改善サイクル、学習活動のモデル化、試行錯誤、より良いコードの記述

1. はじめに

プログラミングにおいて、ただ単に動作するコードを記述するだけでなく、「より良いコード」を記述することは重要である。開発現場ではリファクタリングにより、一度完成した要求を満たすコードであっても「より良いコード」にすることが求められており、プログラミングの授業では学習者の提出した課題の要求を満たしたコードの中から、特に参考になる「より良いコード」が紹介されることも多い。そのため、プログラミング学習においても「より良いコード」を記述する活動を行うことが重要である。

著者はこれまで「より良いコード」を作るための学習支援システムを開発し(図1右部)、その評価から一定の有効性が得られたが⁽¹⁾ 具体的な学習活動のモデル化については十分に議論・分析ができていない。そこで本研究では「より良いコード」を構築するための活動を検討し、本活動に対して提案したシステムがどのような影響を与えたか分析する。

2. コード改善サイクルとその学習活動

人が何かの改善を試みる時、(A)改善のために有益な外部リソースの入手、(B)試行錯誤に基づく改善活動、の2つの段階に分けることができると著者は考える。実際に外部リソース(他者が共有した知識)から学ぶことは社会構成主義として一般的である⁽²⁾。具体的には、インターネットや書籍などで改善に活用可能である情報を調査し、その情報を元に改善活動を行うために試行錯誤するという流れとなる。このとき、(B)試行錯誤に基づく改善活動は、PDCA サイクルなどにみられるように、サイクルを回す活動になっていると考えられる。さらに、(A)で

得られた情報では改善活動が行えない場合や、さらに改善を指向する場合は、新たな外部リソースを検索することになるため、本研究では(B)を内側のサイクルとし、(A)と(B)を含むより大きな外側のサイクルという構成になっていると考える(図1左部)。

本サイクルをプログラミングの改善活動にあてはめると、(B)自身が現状所持する情報に基づき試行錯誤する内側のサイクルと、(A)必要があれば情報の獲得を目的とした外部リソース(本稿では主に他者のコードをリソースとする)を取得し、それをベースに(B)の試行錯誤を再始動するという大きな外側のサイクルからなると考える。

(A)外側のサイクルと(B)内側のサイクルは、それぞれ試行錯誤を通じたステップであるため、(B)内側のサイクルの試行錯誤で不満足を感じた学習者は(A)外側のサイクルに移行して外部リソースの取得に移ると考える。一方、そうでない学習者は(B)内側のサイクルでの試行錯誤を継続すると考えられる。

したがって、本研究では外部リソースの獲得(他者のコードの参照)を必ずしも含むとは限らない本サイクルを、「他者のコードの取り入れを伴うコード改善サイクル(以下、改善サイクル)」と定めた。

本システムの各機能と改善サイクルの各ステップ



図1 改善サイクルとシステムインターフェース

の対応関係を整理する。まず、(B)内側のサイクルの自身の範囲で試行錯誤する際には、自身の活動を評価し、より良くするための方法を検討する必要がある。そこで本システムでは、学習者のコードを評価する評価指標と、それに基づくスコア化を行う。学習者はスコアを参考にしながら、より良いコードの記述を行うことが期待される。(B)内側のサイクルでの試行錯誤に満足した学習者や、逆に行き詰まりを感じた学習者は、外部のリソースを参照し、より良い改善を目指すと考えられる。このような学習者に対しては、外部リソースとして範例となる他者のコードから学ばせることが効果的だと考える。そこで、ランキングに基づいたコード共有・制御機能が重要である。学習者は(A)外側のサイクルとして、共有された他者のコードから学び、自身のコードに取り入れる。さらに取り入れたコードを評価するため、(B)内側の試行錯誤のサイクルを再度回すことになる。さらに内側のサイクルに満足あるいは行き詰まる場合、(A)外側のサイクルに再度、取り組むと考える。

3. 妥当性検証

3.1 概要

提案する改善サイクルが、本システムの学習活動として実際に機能していたかを検証するため、過去の実践的評価($n=49$)を用いて分析した。まず(B)について、評価指標に基づくスコアの前後差を比較することで軸の範囲内で改善されたか検証する。また必ずしも(A)に移行するとは限らないため、(B)のみ遂行した学習者と(B)と(A)を行き来した学習者を抽出し、そのスコア差についても調査した。(A)について、他者のコードを閲覧した学習者を対象にログ分析を行い具体的な改善によるコードの変化を調査した。

3.2 結果

まず、(B)を評価するため、各問題で学習者が他者のコードを閲覧した回数を調査した(表1)。結果から、問題1と2では多くの学習者が他者のコードを閲覧していたが、問題3では閲覧した学習者が少ないことがわかった。特に、問題3は水たまりという回避行動が要求されるため難易度が高い。しかし、問題3は他者のコードの閲覧率が低い結果となった。これは、問題の難易度が高くなると、他者のコードから学ぶよりも自身で試行錯誤する割合が増えることが示唆された。つまり、(B)にとどまる学習者が存在し、(B)と(A)を行き来する学習者も存在することがわかった。さらに、問題の難易度が上がると、(B)のみで終える学習者が増えることが示唆された。

(B)のみの学習者と(B)と(A)を繰り返した学習者にスコア差が存在する可能性がある。そこで、それぞれを群に分け、収穫スコアの変動を調査した(表2)。効果量を算出した結果からまず、全ての問題において事前と事後で得点が向上していた。これにより、改善サイクルの(B)のみを行った学習者と(B)と(A)を繰り返した学習者、どちらもスコアを向上させ

表1 他者のコードを閲覧した学習者の数の比較

問題	閲覧した	閲覧しなかった
1(四角形)	34人	15人
2(メッシュ)	34人	15人
3(水たまり)	24人	25人

表2 他者のコードを閲覧したA群と閲覧しなかったB群における収穫スコアの変動

問題	群	事前の平均	事後の平均	p値	Cohen's d	効果量の目安
1	A	189.4	899.4	<0.005	1.81	大
	B	62.7	548.0	<0.005	1.95	大
2	A	142.4	576.5	<0.005	1.65	大
	B	16.0	286.7	<0.005	1.18	大
3	A	149.2	658.8	<0.005	1.67	大
	B	89.6	356.0	<0.005	0.94	大



図2 実際に学習者が記述したコードの変化

ることがわかった。その一方で、A群の事前ではB群と比較して低い傾向にある。これは、継続的な(B)の活動に良し悪しが存在することを示唆している。特に悪い部分としては、自身の持つリソースでは改善が見込めないにも関わらず、非効率な試行錯誤に陥ってしまう可能性がある。今後として、この仮説の検証と支援方法の検討があげられる。

最後に、(B)と(A)の改善サイクルによるコードの変化を調査した(図2)。元のコードでは、ロボットを前進させ、種を植える行動が記述されているが、他者のコードから収穫機能を学び、実際に自身のコードにアレンジした上で取り入れていることがわかる。これは、(B)と(A)の実際の活動として、有意義な機能レベルの他者のコードから学び、自身のコードを改善していたことが示唆された。なお、ログ分析の結果から、4名の学習者が同じ活動をしていた。

4. おわりに

本研究では、これまで提案してきたコード共有プラットフォームの学習活動をモデル化しそれらの活動が実際に行われていたかを評価した。評価から改善サイクルの一部、満たすことが示唆された。今後の課題として、定量的な分析・評価があげられる。

謝辞

本研究の一部は JSPS 科研費 JP22K12322, JP21H03565 の助成による。

参考文献

- (1) 前田新太郎, 茂木誠拓, 古池謙人, 東本崇仁: “仮想ロボットプログラミングを用いたコード共有プラットフォームの開発と評価”, 教育システム情報学会誌, Vol.40, No.3, pp. 240-245 (2023)
- (2) Vygotsky, Lev S: “Thought and language”, MIT press (2012)