

時系列データを用いた ブロックプログラミング初学者の陥りやすいパターン分析

布目 咲梨[†]
東京都市大学[†]

藤原賢二[‡]
東京都市大学[‡]

1 はじめに

日本では2020年度から「プログラミング的思考」を培うことを目標とした小学校におけるプログラミング教育が必修化されている。プログラミング的思考の習熟度を評価するにあたり、Wingが提唱しているコンピュテーショナル・シンキング（以降、CT）^{*1}と呼ばれる考え方がある[1]。CTによる評価手法においては、初学者の行動パターンが考慮されず、多くの研究では最終成果物のみを用いた点数化が行われている[2,3]。そのため、学校教育などの目的が同一の課題を対象とする場合、最終的なプログラムが似通り、評価結果に差異が現れにくいと考える。そこで、プログラムの開発履歴（時系列データ）を活用することで、生徒らの行動パターンを把握し、陥りやすいパターンを確認することで評価手法をより良く出来ると考える。本研究では、小・中学生を対象としたプログラミングワークショップを開催し、ブロックプログラミング言語によるプログラムの開発履歴の収集を行い、実践的な課題を行った際の初学者が陥りやすいパターンの分析を行った。

2 関連研究

ブロックプログラミングの評価手法として、Scratchのプログラム中に使用されているブロックの解析を行い、CTの習熟度を評価する

目的のシステムが León [2] と太田ら [3] によって提案されている。León らの Dr.Scratch は、7つの CT 概念の習熟度を 0 点から 3 点で算出することで評価している。安東らは、Dr.Scratch に投稿された作品群を分析し、学習者の CT スキルの習熟度予測モデルを提案している [4]。槇原らは、Scratch プログラムに対してコードメトリクスを定義し、リアルタイムで可視化可能なシステムの構築を行っている [5]。

3 時系列データの収集

3.1 履歴収集システム

本研究では、ブロックプログラミング環境のひとつである micro:bit を用いたプログラミングの時系列データを収集し、分析を行う。収集にあたっては、著者らが開発している履歴収集システムを利用する [6]。このシステムは、micro:bit の開発環境である Makecode を改良することで時系列データの収集を可能にしている。また、収集可能な情報は、ブロックの操作に関係するイベントの種類及び発生時刻、ブラウザ上で micro:bit をシミュレートできるシミュレータの操作イベントの種類及び発生時刻などがある。加えて、これらのイベントが発生した時点におけるプログラム（ブロック）の情報が XML 形式でデータベースに保存される。

3.2 収集結果

初学者が実践的なプログラミング課題を行った際に陥りやすいパターン明らかにし、時系列データの必要性を把握することを目的として、先に述べたシステムを利用して時系列データの収集を行った。収集の対象者は、著者らが所属する東京都市大学主催の「科学体験教室」

An Analysis of Pitfall Patterns to Beginners in Block Programming Using Time Series Data

[†] Nunome Emiri, Tokyo City University

[‡] Kenji Fujiwara, Tokyo City University

^{*1} 問題解決のための問題の抽象化と解決方法の自動化が基本とされている考え

に参加した小・中学生 16 名である*2。イベントは全体で 100 分あり、はじめにブロックプログラミングの基礎を学ぶための説明および基礎的なプログラミング課題をいくつか行った。その後、micro:bit 用のロボットカーである「micro:Maqueen」を、こちらが指定したように動作させる実践的なプログラミング課題を実施した。イベントは保護者の同伴が可能であったため、保護者の支援を受けて課題を行っていた生徒が多く見受けられたが、コンピュータの操作は基本的には生徒らが行っていた。また、一人での参加者に関しては、支援担当の教員又は学生が適宜支援した。

4 時系列データの分析

収集した時系列データのうち、ロボットカーを動作させる実践的プログラミング課題の時系列データの分析を行った。分析の際には、保存されたプログラムの XML ファイルおよび、XML ファイルから復元されたブロックを基にソースコードの変更の意図を推測した。そのうえで複数の参加者に横断して見受けられた変更を、初学者が陥りやすいパターンとみなした。その結果、以下のような特徴的な振る舞いが初学者に見受けられた。

1. 1 つだけあれば問題ないブロックの複数回作成
2. 不要なブロックの使用
3. 接続と接続解除を瞬間的に繰り返す

特に、「不要なブロックの使用」では、課題では使用する必要がないブロックを間違えて選択してしまい混乱している様子が見受けられた。これは、使用可能なブロックが全て一覧表示されるブロックプログラミングならではのパターンであると考えられる。

5 おわりに

本稿では、ブロックプログラミングにおける時系列データを活用することで、初学者の行動パターンを把握し、陥りやすいパターンを明らかにした。今後の課題として、より一般的な知見を得るために学校の授業などの対象者が限定されるデータに基づく分析が必要であると考えられる。

謝辞 本研究の一部は JSPS 科研費 JP19K14337 の助成を受けた。

参考文献

- [1] M.Wing, J.: Computational Thinking, *Commun. ACM*, Vol. 49, No. 3, pp. 33–35 (2006).
- [2] Moreno-León, J., Robles, G. and Román-González, M.: Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking, *Distance Educational Journal*, No. 46, pp. 1–23 (2015).
- [3] 太田剛, 加藤浩, 森本容介: コンピュータショナル・シンキング概念に基づくプログラム自動評価機能を持つ Scratch 用学習支援システムの開発, *教育システム情報学会誌*, Vol. 35, No. 2, pp. 204–214 (2018).
- [4] 安東亮汰, 伊原彰紀: コンピュータショナル・シンキング・スコアに基づく Scratch ユーザの習熟度到達予測, *情報処理学会論文誌*, Vol. 63, No. 4, pp. 928–937 (2022).
- [5] 槇原絵里奈, 米田浩崇, 小野景子: オンライン Scratch プログラミング演習支援にむけたコードメトリクス可視化ツールの提案および評価, *情報処理学会論文誌教育とコンピュータ*, Vol. 8, No. 2, pp. 37–50 (2022).
- [6] 布目咲梨, 山川祥輝, 藤原賢二: ブロックプログラミング言語による開発履歴を用いたコンピュータショナル・シンキングスキル評価手法の検討, *SES 2022 WS3*, pp. WS3–9–1–2 (2022).

*2 各回 8 名のイベントを同一内容で 2 回実施した。