

# 編集履歴可視化システムを用いた Learning Analytics

～ Scratch を用いた初等教育向けプログラミング教育における学習者の思考パターン分析

中澤 真<sup>†</sup> 荒本 道隆<sup>‡</sup> 後藤 正幸<sup>†††</sup> 平澤 茂一<sup>†††</sup>  
 会津大学<sup>†</sup> アドソル日進株式会社 早稲田大学<sup>†††</sup> 早稲田大学<sup>†††</sup>  
 短期大学部 創造理工学部 理工学術院 総合研究所

## 1. はじめに

初等教育にプログラミング教育を導入することに注目が集まっている[1][2]。しかし、児童がプログラムを作成する思考過程や、つまづき箇所については経験的・感覚的にしか把握されておらず、また、これまで実施されていた高等教育や成人向けのプログラミング教育と同じ教授法でよいのかどうかも明確になっていない。そこで本稿ではビジュアルプログラミング言語である「Scratch」に、学習者のプログラム作成過程を詳細に記録し、その推移を教員が確認できるようにする編集履歴可視化システムを連携させ、典型的なつまづきパターンや、学習者の思考過程の違いについて解析した結果を示す。

## 2. Scratch のための学習履歴可視化システム

「Scratch」は MIT メディアラボで開発され無償で公開されているビジュアルプログラミング環境である。タイピングに不慣れであったり、語彙力が不十分であったりする小学生でも学ぶことができるように、テキストによるコーディングではなく、命令が記されたアイコンをブロックのように積み重ねることでプログラムを簡単に作成できるところに特徴がある。バージョン 2.0 からはブラウザ上で動作する Flash ベースの Web アプリケーションとなり、多様なプラットフォームでの利用が可能になっている。

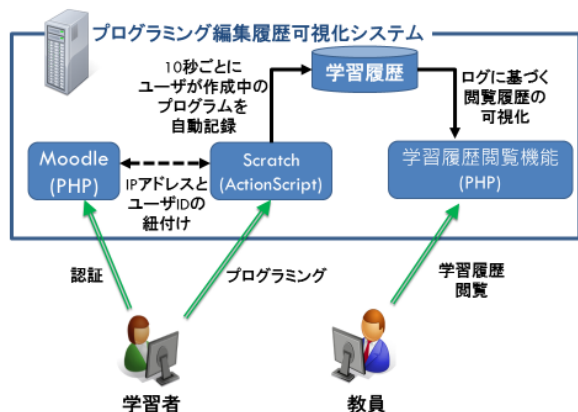


図 1: Scratch のための編集履歴可視化システム

図 1 に示したように提案システムでは、このオープンソースとして公開されている Scratch の Action Script を

修正して、10 秒ごとに学習者が編集集中のプログラム(プロジェクト)をサーバ上に自動的に保存し、さらに記録された学習履歴を可視化する機構を PHP により実現している。Scratch はプログラムを多様なイベントで実行でき、キャラクタの位置設定や画像編集などの機能もあるため、一定の時間間隔で学習履歴を記録する仕様とした。

Scratch のファイルの保存形式には JSON が用いられているため、学習者のプログラミングの変遷はこの差分をとることにより把握できる。提案システムでは学習者が削除した部分を赤色と取り消し線で、新たに追加した部分を緑色と下線で表示させ、図 2 のように視覚的に差分を把握しやすいようにしている。

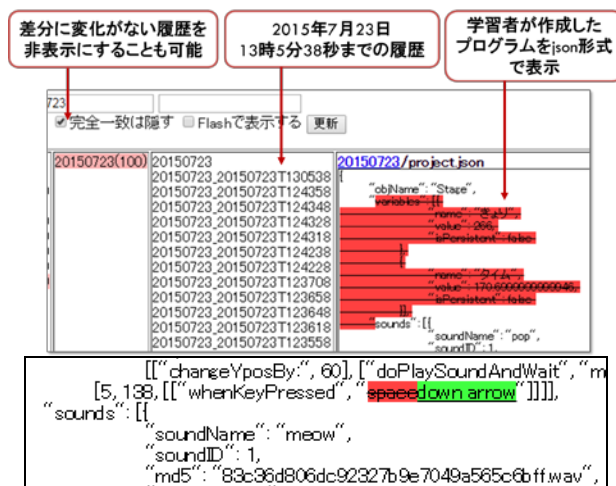


図 2: 提案システムにおける差分表示

## 3. Scratch の学習履歴を用いたつまづき分析

本システムを用いて収集した学習履歴が、学習者の思考過程や典型的なつまづきパターンを明らかにするのに有用であることを示すために、2015 年度に会津大学短期大学部で実施した小学4年生から中学生を対象とした Scratch を用いたプログラミング教育と、高校生向けの Scratch を用いた体験授業において学習履歴を収集した<sup>1</sup>。

小・中学生と高校生のつまづきパターンや思考パターンを分析すると、両者に違いがあることが明らかになった。例えば繰り返し処理に注目すると、ほとんどの高校生が繰り返しの役割を正しく把握し、ループ内に無駄のない処理を組み込むのに対し、小学生は 5 人に 1 人が省略可能な処理を無駄に加えてしまう傾向が確認できた[3]。これは年齢にある程度の開きがあると、プログラミングに

<sup>1</sup> 被験者数は高校生が 38 名、小中学生が 22 名である。

Learning Analytics via Visualization System of Edit Record  
 ~ An Analysis of Learner's Thought Patterns for  
 Elementary Programming Education using Scratch

<sup>†</sup> Makoto Nakazawa, University of Aizu  
<sup>‡</sup> Michitaka Aramoto, Ad-Sol Nissin Corp.  
<sup>†††</sup> Masayuki Goto, Waseda University  
<sup>†††</sup> Shigeichi Hirasawa, Waseda University

おけるつまずき方や考え方が異なっていることを示唆している。

次に、小・中学生のみに着目して分析すると、プログラムの完成までに異なるアプローチタイプがあることが明らかになった。例えば、図3に示したようなスペースキーを押したときにキャラクタをジャンプさせるプログラムを考えさせる場合、1. 座標の概念(X軸・Y軸)、2. 座標の変量、3. 負の値、4. タイマーの4つの要素を理解する必要がある。教員側では、この順序でそれぞれの処理を追加していくことを想定したが、表1に示したように、想定通りのアプローチをした学習者は約3分の1であり、多くの学習者は異なる順序でプログラムを組み上げていることが明らかになった。

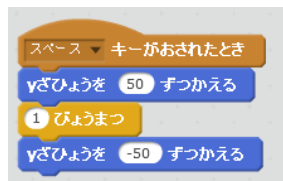


図3: 課題としてプログラムの完成形

表1: プログラムの解決順序パターン的人数分布

パターンNo.	座標の概念	座標の変量	負の値	タイマー	人数
No.1	1	2	3	4	8
No.2	1	2	4	3	8
No.3	1	4	2	3	1
No.4	1	3	2	4	1
No.5	1	2	未完成	3	4

これは上方に移動させたキャラクタを着地のために下方へと移動させるために、座標で負の数を使う概念が学習者にとって難しかったことが一因で、編集履歴の時間推移などから確認することができる。実際、パターン2の負の数の処理を最後に追加した学習者の群のプログラム完成までの平均所要時間は、パターン1の群の時間より1分以上長くなっている。また、座標の概念には相対的に位置を指定するものと、絶対的な基準で位置を指定するものがあり、この区別がうまくできず完成までに時間のかかるケースがあることも判明した。授業における説明では、相対的に座標を変化させる方法を説明したが、4名が絶対的な座標指定をする方法でプログラムを構成しようとし、内2名が作成時間の増加要因となっていた。ただし、残りの2名は十分な理解の上でこの機能を選択したようで、プログラム完成までの所要時間が最も早かった上位2名であった。

学習履歴からは個別の学習者のつまずきパターンも検出することが可能であり、完成までの時間が長くかかっている学習者は、先に述べた負の数の概念でつまずいているケースと、それ以前の段階のプログラムの処理の流れやループなどが十分に理解できず、とりあえず手当たり次第に試しているケースがあることがわかった。

また、X座標とY座標の方向についての概念がうまく理解できず、Y座標の指定をするべき処理をX座標に関する処理を組み込んでしまい、長い時間この解決に時間がかかってしまう学習者も2名いた。このような座標

の概念の誤りは、高校生ではまったく見られなかったつまずきパターンである。

なお、今回の実証実験では、算数や国語の好き嫌いや、PC、インターネット、ゲームなどとの親近感、利用頻度などをアンケートで尋ねたが、これらの要素と思考プロセスのパターンや理解度との間に明確な相関は示されなかった。また、プログラム作成に要した時間の逆数を理解度とした場合、小学校の学年の違いは必ずしも理解度に影響しなかった。

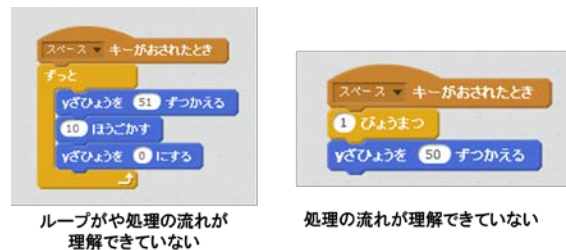


図4: 個別のつまずきパターンの抽出

#### 4. まとめと今後の課題

今回提案したシステムを用いることにより、学習者のつまずきパターンを効率的に抽出することが可能なり、また授業中にリアルタイムに学習者の理解状況や思考のプロセスなども把握することが可能になった。実際、これらの学習履歴を解析することにより、年齢に開きがある場合、思考プロセスやつまずき箇所には差異が生じる可能性が高いことが示され、同じ単元のプログラミング教育であっても、各学年に適した教授方法や教材をそれぞれ準備する必要性が明らかになった。

このScratchでは一般的なプログラミング言語と異なり、シンタックスエラーは存在しないため、学習者が取り組むのは論理エラーの解消のみとなる。このため、アルゴリズム的な考え方を鍛えるのに非常に適している。一方で、学習履歴の解析としては比較的扱いやすいエラーメッセージを利用することができないため、ソースコードの構造的な変遷や、その差分のみを用いなければならない。これらをそのまま視覚化するだけでなく、論理エラーの原因や、学習者の思考の推移を視覚的に表現できるモデルを今後考える必要がある。さらに、解析結果を初等教育向けプログラミング教育の具体的な教授法の構成に活かす方策についても検討しなければならない。

#### 謝辞

本研究の一部は、会津大学短期大学部競争的研究費および独立行政法人日本学術振興会学術研究助成基金助成金基盤研究(C) 26350299の助成による。

#### 参考文献

[1] 石戸奈々子, プログラミング学習の広がり, 情報処理 vol.55 No.11, 1294-1297, 2014.  
 [2] 辰己丈夫, 久野靖, 情報教育とICT活用教育, 情報処理 vol.56 No.4, 337-341, 2015.  
 [3] 中澤真, 荒本道隆, 後藤正幸, 平澤茂一, “ビジュアルプログラミング言語「Scratch」のための学習履歴分析環境とその可能性,” 日本経営工学会2015年秋季大会, 2015.