

## コーディングシーケンスを用いたプログラミングスキル判定指標の検討 Study on Evaluation Indicators for Programming Skills using Coding Sequences

荒木 力樹<sup>†</sup>      納富 一宏<sup>†</sup>  
Riki ARAKI      Kazuhiro NOTOMI

### 1. はじめに

文部科学省は、GIGA スクール構想を基盤として、初等中等教育におけるプログラミング教育の充実化を企図し、2020 年度より小学校においてプログラミング教育を必修とした[1]。このような状況下において、その活用を担うデジタル人材の育成は喫緊の課題と認識されている。加えて、2025 年度からは大学入学共通テストに「情報 I」が導入される運びとなり、高等学校における情報教育の重要性は益々増大している。

このような背景の下、プログラミング教育は、学習者の論理的思考力及び問題解決能力を伸ばす上で重要な手段として位置づけられている。多くの理系の大学ではプログラミング言語による教育が行われており、演習や実習を重視している。プログラミング演習における課題は、プログラミングスキルの指導方法において、完成されたソースプログラムによる評価しか行えないという点である。現実的には文法中心の試験による評価であり教員や TA (ティーチングアシスタント) による採点時の負荷を考慮した場合、期待される IT 人材教育の水準とはやや異なる。基本的かつ小規模なプログラムを断片的に扱わざるを得ない。

演習時間は 90 分など多くの時間を割く場合が多いがレポートのチェックではソースプログラムやフローチャートクラスや関数の構成に関する説明などを中心としている。動的なチェックではテストケースを用意しておきプログラムをコンパイル・実行した際の入出力をチェックする方法などが行われる場合もある。

しかしながら、現行のプログラミングスキル評価においては、完成されたソースコードの正誤性や効率性に偏重する傾向が見られ、学習者の思考過程や試行錯誤のプロセスが十分に評価されていないという課題が存在する。事実、最終的な成果物であるソースコードのみでは、学習者がいかなる過程を経てコードを構築し、正解に至ったのかを把握することは困難である。

ソースコーディングにおける編集操作の過程の情報の確認が必要である。筆者らの先行研究では、編集時の打鍵情報を時系列で記録したデータを「コーディングシーケンス」と呼び、その分析によりプログラミングスキルを判定する手法について検討してきた[2]。

本発表では、プログラミングスキル判定に資する客観的かつ詳細な指標を確立することを目的とする。プログラミング実験を通じて収集したコーディングシーケンスデータを分析し、従来用いられてきた評価指標に加え、本研究において新たに設定した指標との比較検討を行う。将来的には、コーディングシーケンスからプログラミングスキルの自動判定を目指す。

<sup>†</sup> 神奈川工科大学情報工学科 Dept. of Information & Computer Sciences, Kanagawa Institute of Technology

### 2. プログラミングスキル判定指標の検討

#### 2.1 コーディングシーケンスデータの収集

コーディングシーケンスとは、プログラムを書き進めていく様子を表す、概念のことである。先行研究[3]において初めに定義された用語であり、現在は「ソースプログラムのコーディング時に、キーボードの打鍵操作やマウス等の編集操作によるエディタの内部バッファの状態変化やプログラムの実行、テストといった検証操作を時系列情報と共に記録したデータ配列およびその表現形式」をさす。時系列情報を基に、コーディング過程を再現することができる利点がある[2]。

コーディングシーケンスはヘッダー情報と JSON 形式のデータ構造の配列の組み合わせであり、録画データに関する録画を保持する header と内部バッファの変化情報を録画する value からなる[6]。

コーディングシーケンスを記録するオンラインエディタを図 1 に示す。このエディタでは、JSON 形式で保存されるコーディングシーケンスデータの再生が可能である。

なお、岡田ら[2]は、コーディングシーケンスデータを用いたプログラミングスキル判定に利用するための礎であるダイナミックデータの分析を行った。

橋本ら[5]は、if 文や for 文といったパートごとの分析を行うことで学習者の苦手を把握する研究をした。

また、関[6]は集計したデータ可視化しを比較するシステムを開発した。

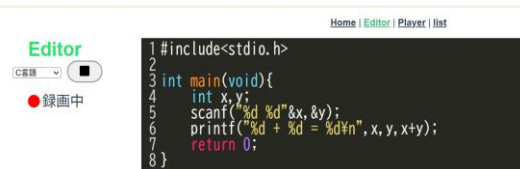


図 1. 記録用のオンラインエディタ

#### 2.2 コーディングシーケンス分析システム

筆者らは先行研究をふまえ、一定区間ごとのダイナミックデータがコーディング時の行動、思考の特徴を表していると考え、表 1 に示しているものを例に、データ項目を取り出し分析システムを用いて指標の検討を行う。その分析システムのシステム構成図を図に示す。

表 1. 集計するデータ項目の一部

No.	項目名	No.	項目名
1	10 秒ごとの追加文字数	4	停滞回数
2	打鍵速度	5	停滞から停滞までの平均間隔
3	継続時間	6	削除数

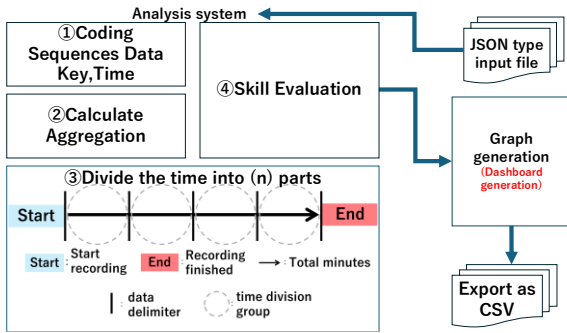


図 2. システム構成図

### 3. 実験による評価

#### 3.1 実験方法

情報系学部の学生 9 名 (3 年次 1 名, 4 年次 8 名) を実験協力者として実験を行った。実験ではエディタになれるための操作練習と, 四則演算プログラム (問題 A) と数値配列ソートプログラム (問題 B) を用いた。難易度を考慮し, 300 秒間を独力で解いてもらい, その後, 解答例を提示した。プログラミング言語は C 言語とした。

また実験の際の事前アンケートとして, プログラミングの経験年数, 開発経験のあるプログラミング言語, 直近でプログラミングをした日, 自分がプログラミングを得意と感じているかに関するアンケートを行った。

協力者が問題を解く過程をコーディングシーケンスとして記録し, 分析を行った。

#### 3.2 データ収集と指標の設定

実験では, コーディングシーケンスから得られたデータを基に, 10 秒間ごとの追加文字数を主要な指標として採用した。この指標は, データ再生なしにコーディングの停滞箇所を特定するうえで有効であると考えた。

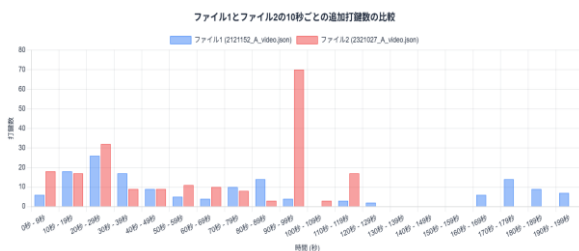


図 3. データ比較画面-10 秒ごとの追加文字数

#### 3.3 分析結果

協力者の中から代表的な 2 つのコーディングデータの比較を図 3 に示す。図 3 の x 軸は経過時間, y 軸は文字数 (打鍵数) を表し, 各棒は 10 秒間の追加打鍵数を示す。

全協力者の問題 A における平均継続時間は 233.84 秒, 平均追加文字数は 250.44 回, 平均削除数は 33.44 回であった。一方, 問題 B では平均継続時間 338.62 秒, 平均追加文字数 417.67 回, 平均削除回数 39.56 回であった。

図 3 において大きく突出した棒はコピー&ペースト操作に起因するものであり, 特に問題 A では `printf` 関数の繰り返し使用によるものである。これにより, 継続時間が他のデータと比較して短くなる結果が示された。

### 4. 考察

問題 A, B において一定間隔ごとのデータ比較を行った結果, 停滞している時間やペーストによるコーディングを詳細に観察することができた。

コーディングシーケンスを再生している中で, 停滞箇所を観察中, 何か思いついたかのように削除を繰り返した後打鍵をする協力者がいたことから, 10 秒間の追加文字数に加え, 10 秒間の削除回数と合わせて観察することでより詳細な実験結果が得られると考えられる。

今回の実験のデータ比較からは, 従来の評価指標である完成されたソースコードからは得ることができない, 学習者の試行錯誤のプロセスを観察することができた。

本研究で得られた全体の平均値は結果に示した通りであるが, データの属性 (学年, 経験, プログラミングスキルの自己評価) に偏りがあった。事前アンケートで得られた属性調査では, プログラミングを得意と感じているかという項目に, 「そう思わない」と回答した者のみであるため, 属性を考慮した追加のデータ取得を計画している。今後, プログラミングスキルのラベリングを目指し, スキルの自動判定に繋げる。

### 5. まとめ

本研究では, 学習者の実習・演習を通じた学びにおいて, 従来の完成されたソースコードの正誤やアルゴリズムの効率性に偏重する評価では十分に捉えきれない課題に着目し, プログラミングスキル判定に資する指標を確立することを目標としている。今回の実験では, コーディング時の打鍵数の違いや試行錯誤の違いなどに個人差の違いや安定性のバラつきが見られた。

今後の課題として, 停滞時間に加え, 新た集計するデータ項目を増やし, 多くのコーディングシーケンスを収集することが挙げられる。協力者に変数の定義や記号の扱いに慣れていないため試行錯誤をしているコーディングシーケンスデータが観察されたため, プログラミングスキル判定において, 「つまずき」に関する項目を加える予定である。

#### 謝辞

本研究は JSPS 科研費 JP24K15235 の助成を受けたものです。

#### 参考文献

- [1] 文部科学省; GIGA スクール構想の実現について, [https://www.mext.go.jp/a\\_menu/other/index\\_0001111.htm](https://www.mext.go.jp/a_menu/other/index_0001111.htm), (参照 2025/06/13).
- [2] 岡田竜岳, 爲近瑛太, 納富一宏: "コーディングシーケンス分析によるプログラミングスキル判定に関する考察", *バイオメディカル・ファジィ・システム学会誌*, Vol.25, No.1, pp.45-53, (2023.06).
- [3] 星野裕樹, 納富一宏, 西村広光, 示野浩士, "キーボード操作の時系列情報を活用したコーディング学習支援手法の提案", *情報科学技術フォーラム講演論文集*, Vol.14, No.3, pp.467-468 (2015).
- [4] 爲近瑛太, 段王れい子, 納富一宏: "プログラミング演習におけるコーディングシーケンス共有システムの評価", *情報処理学会第 84 回全国大会講演論文集 第 4 分冊*, 5ZH-05, pp.627-628, (2022.03).
- [5] 橋本浩規, 納富一宏: "定型コーディングフォームを用いたプログラミング学習支援システムの開発と評価", *神奈川工科大学 2024 年度修士論文* (2024.02).
- [6] 関彰太, 納富一宏: "コーディングシーケンスを用いた分析ツールの作成", *神奈川工科大学 2024 年度卒業論文* (2023.12).