一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
ET2020-59(2021-03)

プログラミングにおけるコーディング過程の指導支援ツールの開発

大川内 隆朗† 　平野 智紀‡

† 日本大学文理学部 〒156-8550 東京都世田谷区桜上水 3-25-40

‡ 内田洋行教育総合研究所 〒104-8282 東京都中央区新川 2-4-7

E-mail: † ohkawauchi@chs.nihon-u.ac.jp

あらまし　プログラミング教育の必修化とともに，より効率的かつ効果的なプログラミングの指導方法が求められている．プログラミング教育においては，最終成果物となるプログラム本体のみでなく，完成させるまでにどのような手順や考えでコーディングを行ったのかというプロセスが重要であると考える．しかし従来のシステムにおいては，学習者のコーディング過程を記録できるものは少ない．また記録できたとしても，その映像について，指導者が同じ時間を掛けて視聴することは，学習者の数が指導者の数をはるかに上回っている現実を考えると不可能といえる．したがって本研究では，学習者のプログラム作成時のコーディング過程を記録するとともに，学習者の手が止まったタイミングや，重要なキーワードが入力されたタイミングなど，指導を行ううえで重要な箇所を中心としたハイライト映像を効率的に視聴することができるシステムの開発を行った．

キーワード　プログラミング教育，開発環境，可視化

Development of a Support System for Coaching the Coding Process in Programming

Takaaki OHKAWAUCHI† 　Tomoki HIRANO‡

† College of Humanities and Sciences, Nihon University 3-25-40 Sakurajousui, Setagaya-ku, Tokyo, 156-8550 Japan

‡ Uchidayoko Institute for Education Research 2-4-7 Shinkawa, Chuo-ku, Tokyo, 104-8282 Japan

E-mail: † ohkawauchi@chs.nihon-u.ac.jp

Abstract　More efficient and effective programming instruction methods are required due to the compulsory of elementary school programming education. In programming education, not only the program itself as the final product, but also the coding process that what kind of procedure and idea it was from is highly important. However, most of conventional systems cannot record learners' coding process. Even if it can be recorded, it is impossible for instructors to watch all recorded video because the number of learners far exceeds the number of instructors. Therefore, in this study, we have developed a system to produce highlight video of learners' programming process automatically and it enables instructors to coach learners for their computational thinking with important points such as pausing coding or input important keywords.

Keywords　Programming Education，Development Environment，Visualization

1. はじめに

2020 年度より，日本国内の小学校におけるプログラミング教育の必修化が開始した．その内容に対する諸外国との差を見てみると，先進国の多くではプログラミング教育が独立した科目として位置付けられていることに対して，日本ではクロスカリキュラムとして設定されている[1]. すなわち, 独立科目としてではなく，算数・理科・総合的な学習など他の教科の中でプログラミング的な思考の要素を取り入れるように定められている．有識者会議によると，プログラミング的思考とは「自分が意図する一連の活動を実現するために，どのような動きの組合せが必要であり，一つ一つの動きに対応した記号を，どのように組み合わせたらいいのか，記号の組合せをどのように改善していけば，より意図した活動に近づくのか，といったことを論理的に考えていく力」であり，将来どのような進路を選択しどのような職業に就くとしても，幅広く求められる能力と説明されている[2]．これに対し，諸外国のプログラミング教育で議論される Computational Thinking[3] の概念では，課題の発見・解決能力，マネジメント，コミュニケーション能力などを含めた，より上位の概念を含めた統合的な資質としてプログラミングを捉えており，日本で必修化されたプログラミング教育は，その下位概念となる手順や実装に重点を置いたものと

考えられる．以上の点を踏まえ，国内のプログラミング教育では，実装時の手順や思考に着目した指導をどのように効果的に行うことができるかという点が，指導を行ううえでの重要な観点の一つであるといえる．

2. 従来研究

2.1. プロセスの可視化

学校におけるプログラミング教育の目的を考えたとき，それは１つの良いプログラムを完成させることではなく，プログラミング的思考を行うことのできるような人間を育成することにある．すなわち，学生が作成したプログラムに対して，教員自身が積極的に手を加えてプログラムの改善や完成を行ってはならない．教員の役割は，学生が自ら解決しプログラムを完成させることができるように上手く足場掛けを行うことにある．この指導の構図は，レポートライティングに類似していると筆者らは考えた．レポートライティングの目的も同様に，教員が積極的に手を加えて１つのレポートを良くすることではなく，レポートを１人で書くことができるように，学習者の論理的思考力や記述力を育成することにある．

したがって，プログラミング教育においてもレポートライティングにおいても，良い統合開発環境（IDE）やエディタがあることは開発効率を上げるために必要ではあるが，教育という側面を考慮すると，指導に特化したような機能が付いているとその質は向上すると考えられる．レポートライティングにおいては舘野らが，指導に特化した機能を持つソフトウェア「レポレコ」の開発を行った[4]．レポレコは，レコーディング機能を持つレポートエディタで，レポート執筆中のユーザの操作履歴をすべて記録しておくシステムである．具体的には，執筆している箇所，文字を打つペース，執筆の手が大きく止まった箇所，文章のコピーや範囲削除などで一度に内容を大きく変更した箇所について，その操作と時間を記録し，後からハイライト再生できる機能を持つ．同ソフトウェアによって，従来は完成物のレポートしか見る機会の無かった指導者が，学習者と一緒にハイライト再生を観ながら「どうしてこのとき手が止まったのか」,「なぜここで段落の順序を入れ替えたのか」といったことを質問できるようになり，成果物のレポートではなく執筆中のプロセスにより着目した指導へとつながり，その指導を受けた学生のレポート執筆能力が向上したという結果が得られている．

本研究では，上記のシステム理念をプログラミング教育用のソフトウェアにも応用できるのではないかと考え，プログラミングのコーディング過程の操作履歴をすべて記録することで，指導者がそのプロセスを効率的に振り返り，成果物では無くプロセスに着目したより効果的な指導につながると考えた．

2.2. プログラミング過程の記録

Personal Software Process (PSP) [5]の分析の枠組みなども提唱される中で，プログラミングにおけるコーディング過程を記録するシステムはいくつか見られる．例えば Matsuzawa らは，コーディング過程をグラフ化し，学生自身による振り返りを支援するためのシステムの開発を行った[6]．またコーディング過程のログを利用し講師側を支援する試みも見られる．プログラミング教育においては学習者の理解や進捗の幅が大きく，個別の学生のスキルや状況に応じた指導を行うことが重要であると指摘されており[7]，サーバ上に開発環境を実装することにより対面授業内での個別の学生の進捗状況を把握し，手が止まっているなどで遅延している学生をリアルタイムで検出し教師や TA に知らせるような取り組みも見られる[8]．

本研究では特に，従来システムに見られなかった，指導者が学習者のコーディング過程を短時間でより効率的に把握することと，遠隔でも利用できるようなシステムの開発に焦点を当てた．主な理由として，従来研究においても大学の授業内での利用を前提としたものが多いが，2020 年の小学校におけるプログラミングの必修化に伴い，今後は学習者の数が大きく増加することが予想される．そのような状況下では，対面のみで多くの学習者の状況を把握し指導することは困難であるといえる．学習者がコーディングに 30 分掛けたとして，指導者がそれと同じ時間でコーディングの様子を再生して確認する，さらにはそれをすべての学習者の数だけ行うことは実際問題として不可能だろう．したがって記録したコーディング過程を，より短い時間で把握することができるようなシステムが求められる．また大学での 1 単位の認定は，授業時間 15 時間に加え，予復習に 30 時間が目安となっており，授業時間内で利用するシステムでは限定的になってしまうため，授業外の学習の進捗や様子を把握することのできるシステムの提案が重要であるといえる．

またいずれの従来研究も，IDE の機能を充実させることによってプログラミングの支援を行うのでは無く，エディタとしての機能は最低限に，変数・条件分岐・繰り返し処理など，基本的なアルゴリズムを理解するために 1 つのファイルから構成されるプログラム課題を対象に開発および評価を行っている．1 コマから数コマ程度の少ない時間でプログラミングを学習する場合にはプログラムの動作の理解を主目的とすることが適当であると指摘されており[9]，本研究においても初めてプログラミングを学ぶような入門者に対する指導を前提にシステムの開発を行った．

3. 開発したシステム

3.1. 学習者のコーディング画面

本研究において開発したシステムは，学習者のコーディング画面と，指導者による確認画面の二つに大きく分かれる.図 1 に学習者のコーディング画面を示す.上部にソースコードの編集画面，下部にコンパイルおよび実行結果が表示されるインターフェースで，学習者から見た画面は一般的な開発環境とほぼ同等のものとなっている．バックグラウンドでは，前章で示したほかの従来研究と同様，ユーザの操作ログを自動的に取得している．具体的には，ユーザがコードの追加操作を行うと，コード全体の何文字目の位置から何の文字を挿入したかという情報を経過時間とともに逐一記録する．同様にコードの一部を削除した際には，何文字目から何文字分を削除したかという記録を行う．画面右上の実行ボタンを押すとコンパイルが行われ，その成否が時間とともに記録される．この点は従来研究でも指摘されているが，コンパイルの成否は学習者の状況や理解度に関わってくるとともに，指導者がコーディング過程を確認するうえで一つの切れ目やポイントになると考えたためである.

また図 1 の画面上は Java でのコーディングとなっているが，コンピュータ上のコンパイラのパスや実行ファイルの指定を変更すれば，基本的には多くのプログラミング言語で本システムを動作させることができる.

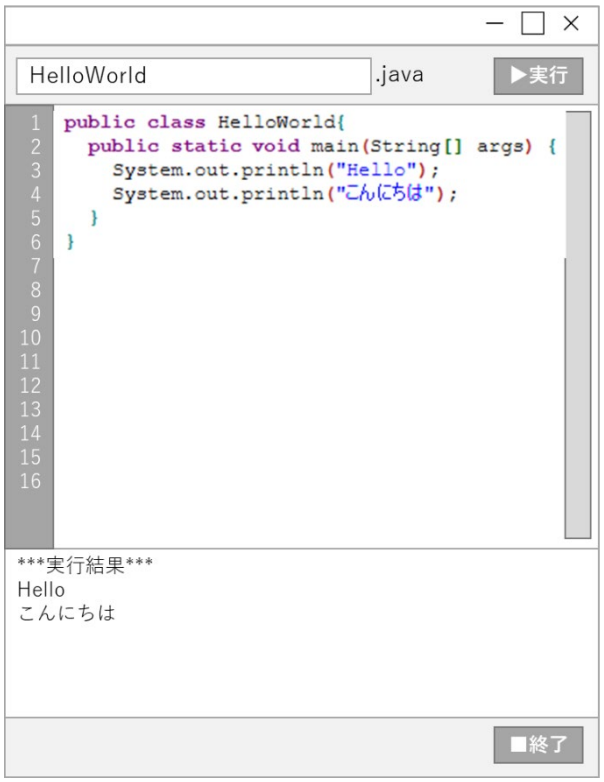

図 1. ソースコードの編集画面イメージ

3.2. 指導者による確認画面

学習者のコーディング映像については，図 2 に示す画面で指導者が後から再生し確認することが可能となっている．同画面には，指導者が効率的に確認を行うことを可能とするためのいくつかの特徴がある.

（a）コーディングのグラフ化

コーディング過程を一定時間ごとに区切って，その間のコード量の増減を画面下部にグラフ形式で表示する．棒グラフが上に伸びている部分が，その時間でのコードの増加量を示し，下に伸びている部分はコードの削除量を示している．すなわち，学習者がどのタイミングでどの程度コードを書き加えたり削除をしたりしたのかを一覧できる画面となっている.

（b）特定操作の検出

「学習者の手が 30 秒以上停止」,「コンパイルに成功」,「コンパイルに失敗」,この 3 つのタイミングついてグラフのすぐ上に吹き出しを表示することによって，把握しやすいようにしている．また 30 秒以上手が止まったタイミングについては，再生画面では「45 秒停止」といった形式で画面上に表示されるのみで，実際に 45 秒間映像が止まるようなことはない．指導者は何秒間手が止まったかを知りたいのみで，実際に手が止まっている様子を再現する必要性は低いと判断した.

（c）編集速度の最適化

本システムでは，学習者のタイピングのスピードについては指導のうえでは重要ではないと考えた．したがって再生画面では，学習者のタイピングスピードは実際の速度に関わらず，1 秒間に 5 文字程度の一定のスピードで再生されるような仕組みを採用した.

（d）キーワードの検索

プログラミング指導の際には if の条件分岐や, for／while による繰り返しなど, 特定のキーワードが利用できているかといったことに着目して指導を行いたい場合も多いと考えた．本システムでは図 2 の最下にある検索ボックスを利用することによって，特定のキーワードが現れた／増えたタイミングをグラフ上に表示し，再生ポイントを移動できるように実装した.

以上の（a）～（d）の 4 つの機能を利用し，本システムでは学習者が実際にコーディングを行った速度よりも早く再生し，また指導のうえでポイントとなりそうなタイミングをピックアップしたり検索したりする機能を実装することにより，指導者がより短い時間で学習者の様子を把握することができるシステムの開発を行った.

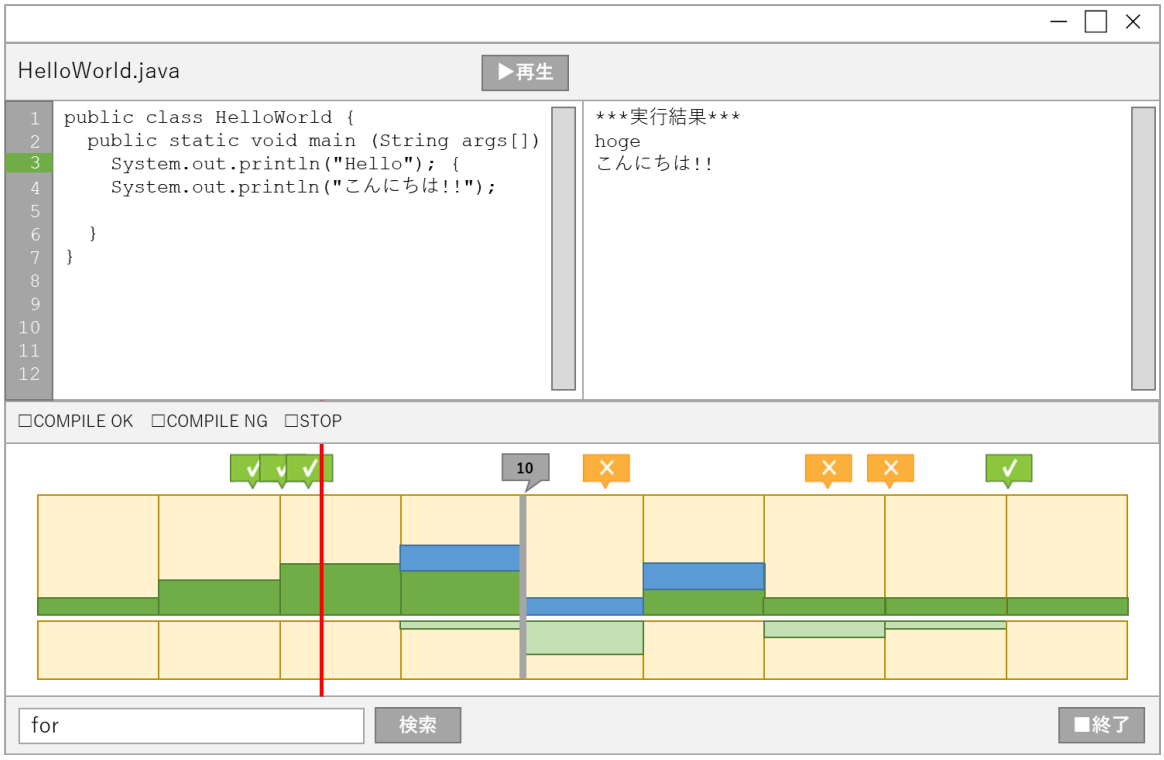

図 2. 記録されたコーディング過程の確認画面イメージ

4. まとめ

本研究では，プログラミングにおける学習者のコーディングプロセスを記録し，指導者が効率的に振り返ったり検索を行ったりするためのシステムの開発を行った．システムの枠組みは完成した一方で，次の二点の課題を残している．一点目は，指導者からのフィードバックに関するシステム的な支援を実装できていない点である．対面でなくても，本システムを通して，今度は指導者の指摘を学習者に視覚的に伝えるような仕組みを考案する必要がある．二点目として，本システムを実際のプログラミング教育を行っている講座等で利用し，教授者および学習者の双方の意見を基にシステムの評価を行っていく必要がある．

謝　辞

本研究は，JSPS 科研費 20K12097 の助成を受けて実施した成果の一部である．

文　献

[1] 赤堀侃司，“プログラミング教育に関する現状と今後の展開，” 教育テスト研究センター年報，Vol.3, pp.11-18, Jul.2018.

[2] 文部科学省，“小学校プログラミング教育の手引（第三版），” 参照 Feb. 14, 2021. https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf

[3] J. M. Wing, “Computational Thinking,” Communications of the ACM, Vol.49(3), pp.33-35, Mar.2006.

[4] 舘野泰一，大川内隆朗，平野智紀，中原淳，“レポート執筆プロセスの可視化システム「レポレコ」の開発：チューターによる正課課程外の指導場面に着目して，” 日本教育工学会論文誌，Vol.37(3), pp.241-254, Nov.2013.

[5] W. S. Humphrey, “Introduction to the Personal Software Process,” Addison-Wesley Professional, Dec.1996.

[6] Y. Matsuzawa, K. Okada and S. Sakai, “Programming Process Visualizer: A Proposal of the Tool for Students to Observe Their Programming Process,” Innovation and Technology in Computer Science Education, pp.46-51, Jul.2013.

[7] 加藤利康，石川孝，“プログラミング演習のための授業支援システムにおける学習状況把握機能の実現，” 情報処理学会論文誌，Vol.55(8), pp.1918-1939, Aug.2014.

[8] 井垣宏，齊藤俊，井上亮文，中村亮太，楠本真二，“プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案，” 情報処理学会論文誌，Vol.54(1), pp.330-339, Jan.2013.

[9] 布施泉，“大学の一般教育としてのプログラミング教育，” システム制御情報学会論文誌，Vol.62(7), pp.266-271, Jul.2018.