

K-010

コーディングスタイルに着目した実践的プログラミング学習支援システムの開発：
インタフェース実装について

Development of practical programming learning support system based on coding style:
Implementation of the interface

滝本 将司† 納富 一宏†
Masashi Takimoto Kazuhiro Notomi

1. はじめに

情報化社会が進んだ近年、同時に若年人口の減少により、日本での労働人口はさらなる減少が見込まれており、今後 IT 人材の不足が懸念されている。また、新しい技術やサービスの登場により、今後さらに IT 利活用の高度化・多様化が進展すると考えられている^[1]。こうした背景から文部科学省では、2020 年度からの小学校においてもプログラミング教育を導入する方針が決定されている^[2]。このようにプログラミング教育は年々重要度を増している。

また、大学でのプログラミング教育環境では、情報系の学科がある大学の場合ほとんどがプログラミングの授業を実施している。神奈川工科大学の情報学部においても、1 週間のうち座学での学習 1 コマに加えて、プログラミング演習 1 コマの構成で授業が進められており、プログラミングの知識とコーディング技術の両方を学べるようになっている。

しかし、演習時間において学生の指導を行う場面では、ソースプログラムの作成・編集作業において、インデントの付与が適切に行われていないために、ソースコードの可読性が大幅に低下することが原因で、if 文や for 文などといった条件分岐構造を有するプログラムでは、ブロック終了を表す記号の記述忘れが発生し、結果としてエラーが生じてしまうケースが多く見られる。このように、大学生の授業におけるプログラミング教育では、文法を中心としたプログラムの命令コード記述に関する知識は十分に身に着けることが可能であるが、コーディングに関する記述表現方法や編集方法などといった実践的なプログラミングスキルを効率的に身に着けることは難しい。多くの場合、こうしたコーディングスキルに関しては、経験によって身に着くと考えられる部分が大きく、通常の授業だけでは、実践的なコーディング技術を身に着けることは難しいと考えられている。

そこで、著者らの先行研究では、操作履歴を時系列情報と共に記録することで、模範となる指導者のコーディングを学習者が学ぶことができる学習支援システムとして、コーディングの記録や再生が行えるシステム開発や模範者のコーディングをライブ配信するためのシステム開発を通して、実践的プログラミング学習支援システムに関する研究を行ってきた^{[3],[4]}。

また、複数回行われる繰り返し学習をより効率良く進めるため、模範者と学習者のコーディングから得られるコーディング規約やコーディングの順序を特徴量として扱い、比較分析結果の視覚化と学習者へのフィードバック手法について検討を行っている。

本稿では、全体のシステム設計と視覚化、さらにプログラミング学習支援を行うためのインタフェースの実装について述べる。

2. システム実装

練習用インタフェースを通してコーディングの再生と記録が行われる。ここでは、全体の流れと共に、視覚化と練習用インタフェースについて述べる。

2.1 システム設計

システム構成を図 1 に示す。模範者のコーディングデータがインタフェース上の再生エディタを通して再生され、ユーザである学習者はコーディングデータの記録を行う。コーディングの記録が終了すると、模範者と学習者のコーディングデータから特徴量の計算が行われ、グラフを表示する。

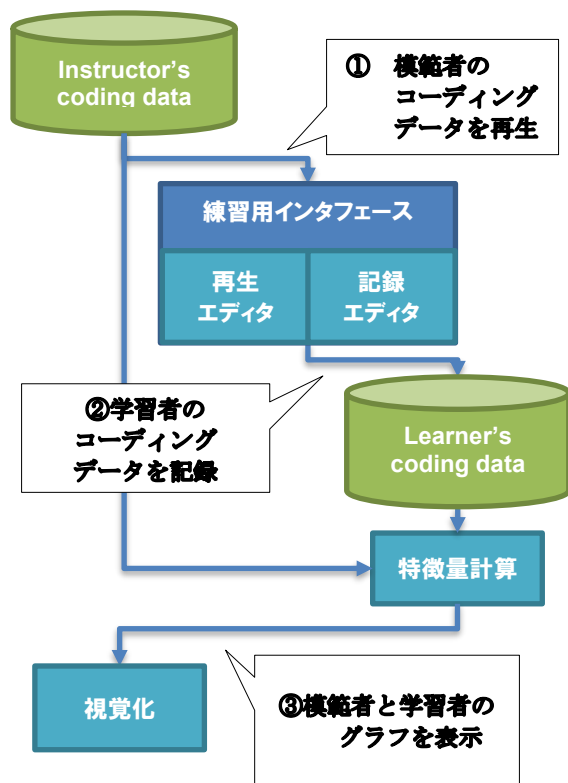


図 1 システム構成

2.2 視覚化

視覚化画面の例を図 2 に示す。なお、図 2①は模範者のコーディングデータから得られた特徴量、②は学習者の特徴量、③は模範者と学習者の類似度、④は類似度の推移をグラフ化し、上達の度合いを表したものである。視覚化は、

† 神奈川工科大学 Kanagawa Institute of Technology

記録終了後行われるため、繰り返し学習の場面で、次の記録を行う際に注意すべき部分が見えてくる。また、特徴量にはコーディングシーケンスとして、丸括弧や中括弧の記述順 2 つと、武田らの先行研究^[5]で示されていた、インデントや行の文字数のような特徴量として特に他人との差異があると考えられるもの 3 つを合わせた、計 5 つの特徴量を利用している。

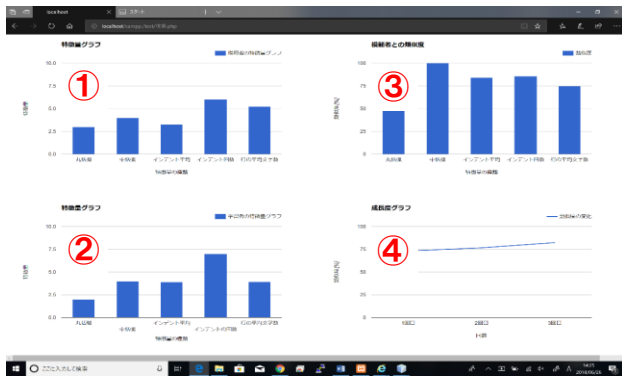


図 2 視覚化画面

2.3 練習用インタフェース

本稿で述べているプログラミング学習支援システムでは、学習者はコーディングの記録を行いながらであるため、再生側のコーディングを見逃してしまう場面が多くなるのが想定される。学習効率低下の要因をなるべく減らす方法としてインタフェース部分についていくつか述べる。垂直分割ウィンドウを図 3 とし、水平分割ウィンドウを図 4 として以下に示す。

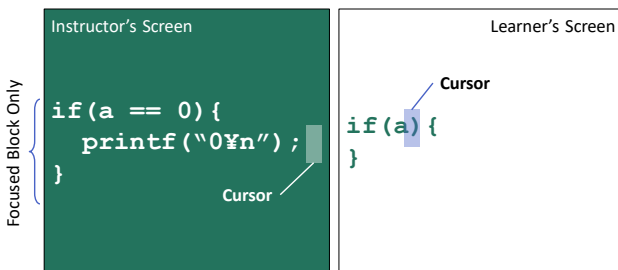


図 3 垂直分割ウィンドウ

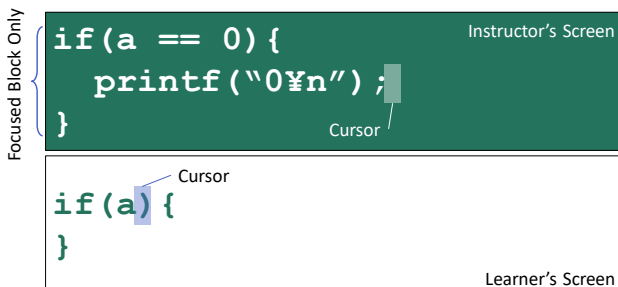


図 4 水平分割ウィンドウ

2.3.1 垂直分割ウィンドウ

図 3 では、左右にエディタを並べた形をとることで、模範者のコーディングにおけるブロック内の空行の使い方などを把握しやすくなる。

2.3.2 水平分割ウィンドウ

図 4 では上下にエディタを並べることで、模範者のコーディングに現れるスペースの個数やインデントなどが把握しやすくなる。

3. インタフェース設計における課題

男性 2 名に垂直分割ウィンドウのインタフェースに対するユーザテストを行ってもらった結果、いくつか改善点が見つかった。1 つ目は記録を行いながらコーディングの再生を見る場面では、再生が早かったため、コーディングの記録を行っている間に再生側が先へ進んでしまっていた。そのため、最初から再生を繰り返すことが多くなり不便であるようだった。また、記録を一旦止め、最初から最後まで再生を一通り見てから記録を行う場面では、余裕をもってコーディングの再生を観察することができていた。したがって、ある程度の再生速度がある場合の方が観察する時間を短縮することができ、スムーズに学習を進められる。そのため、再生速度については調整可能にする必要があることが分かった。2 つ目は、途中で一時停止を挟まずに再生を続けてしまうと、再生の見逃しにより、模範者のコードの書き順を意識し、記録を行うことが困難になってしまう場面が見受けられたことである。そのため、コーディングの再生を部分的に区切って行う必要があった。

4. まとめ

本稿では、コーディングスタイルの学習を目的としたインタフェース開発について述べた。今後は 3 で述べたユーザテストの結果より得られた改善点をもとに、コーディング学習に適した利便性の良いインタフェース開発を考慮し、システムの完成を目指す。

謝辞

本研究は JSPS 科研費 JP18K11587 の助成を受けたものです。

参考文献

- [1] 経済産業省：IT 人材の最新動向と将来推計に関する調査結果を取りまとめました (METI/経済産業省)，
<http://www.meti.go.jp/press/2016/06/20160610002/20160610002.html> (2018/6/26)
- [2] 文部科学省：小学校プログラミング教育の手引（第一版）：文部科学省，
http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm (2018/6/26)
- [3] 星野裕樹, 納富一宏, 西村広光, 示野浩士: "プログラミングにおけるコーディングスタイルの学習 ～コード記述の特徴解析手法の提案～", 電子情報通信学会 技術研究報告 Vol.115 No.351, 教育工学研究会(ET), IEICE-SITE2015-46, ET2015-74 (2015-12), pp.31-36, (2015.12).
- [4] 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", 情報処理学会 第 14 回情報科学技術フォーラム(FIT2015)講演論文集, 第 3 分冊, K-009, pp.467-468, (2015.09)
- [5] 武田隆之, 牛窓朋義, 山内寛己, 門田暁人, 松本健一: "コーディングスタイルの特徴量とソースコード盗用との関係の分析", 情報処理学会研究報告 Vol.2009 No.6 Page.ROMBUNNO.SE-167,8 (2010.04.15).