

プログラミング学習を目的としたコーディングシーケンス共有システムの開発 Development of the coding sequence sharing system for practical programming learning

爲近 瑛太[†] 納富 一宏[†]
Yuta Tamechika Kazuhiro Notomi

1. はじめに

一般的な授業で行われるプログラミング教育だけでは、プログラミング熟練者のようなコーディングスタイルを身につけることは困難である。主な理由として、①開発環境に含まれる snippet などのコーディングに関わる機能の使用や、②コーディングスタイルの適用など、主として実技にかかわる部分を授業の評価対象にすることが難しいことが挙げられる。

上記①の開発環境に注目すると、熟練者は、IDE (統合開発環境) の補完機能や formatter, linter, デバッガなど各種ツールを駆使して開発を行う。一方、一般的なプログラミング教育では、言語の文法についての学習が主である。そのため、それに付随するツールについての説明は少ない。学生が各種ツールを使わない環境で開発を行った場合、本来なら、ツールを使うことで自動調整できるインデント幅やスペースの数などに気を使わなければならない、コーディングスタイルの習得を難しくしている。

上記②のコーディングスタイルについては、熟練者のようなコーディングスタイルというものが、人や所属する集団、時代によって変わることが挙げられる。例えば、短いコードが読みやすいと考える人ならば三項演算子は良い書き方になるだろう。しかし、三項演算子も複数回繰り返し書けば可読性は下がる。さらに言えば愚直に if 文を書いた方が読みやすいという人もいるだろう。コーディングスタイルの適用を成績として定量的に評価するのは難しい。

このような理由から一般的なプログラミング教育だけでは熟練者のようなコーディングスタイルの学習が困難だと考えざるを得ない。

これらを踏まえて、本研究ではコーディング時の打鍵情報の記録・再生を行うシステムを開発し、熟練者のコーディングを観察することで、そのノウハウを簡単に学べるシステムの改善を継続的に進めてきた^{[1],[2]}。システム構成を図 1 に示す。

本稿では、実装方法を改めることで、実用に供するレベルでの動作の安定性の確保について示すとともに、熟練度に関わらず学生同士がお互いのコーディングシーケンスを共有した場合の学習効果について、評価実験結果を基に議論する。

2. 安定性の確保

ここでは、安定性の確保について述べる。改良したシステムの動作画面を図 2 に示す。

2.1 取得するデータの変更

今までの実装手法ではキーボードの打鍵情報を時系列情報とともに記録し、再生していた。この手法は、IME の変換やコピー&ペースト等、打鍵情報以外の情報が必要な再現ができないという問題があった。今回の実装では、時系列情報と共にテキストの変化情報とカーソル位置の情報を取得するようにした。これにより IME の変換、貼り付け処理に対応できるようになった。更に保管機能による文字列の挿入やキーボードショートカットによるコメントイン、コメントアウトなどのエディタの機能を活用したコーディングの録画が可能になり、より実践的なコーディングの様子の録画ができるようになった。

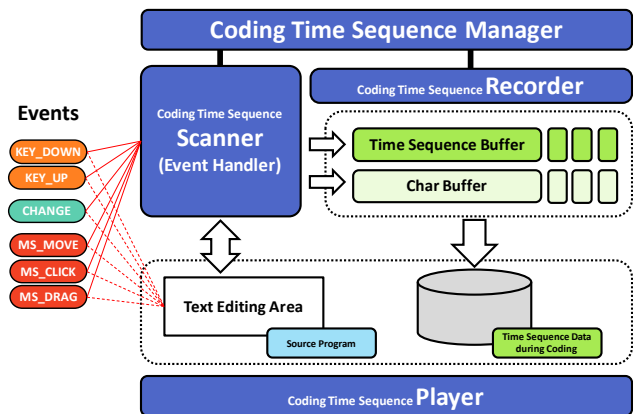


図 1. システム構成



図 2. 動作画面

2.2 プログレスバーの挙動の改善

従来のシステムでは、再生画面でのプログレスバーが、テキストが変化するタイミングで同時に動作していた。この手法だと、長時間にわたりテキストの変化が無い状態に

[†] 神奈川工科大学情報工学科 Dept. of Information and Computer Sciences, Kanagawa Institute of Technology

なった場合、録画時にプログラマがコーディング以外の作業をしていたのか、システムが動作不良を起こし固まったのか区別がつかなかった。そこで、視聴のしやすさを向上するため、なめらかに動くようにした。

3. 実験

実践的なコーディングスタイルを学ぶためには、他人と自分のコーディングシーケンスの比較による気づきが必要であると考えられる。

先行研究では、熟練者のコーディングシーケンスを見ることが、コーディングスタイルの学習に有効か実験をおこなった。

本実験では熟練者のコーディングシーケンスを一方から見ただけでなく、学生が相互にコーディングシーケンスを共有することが、コーディングスタイルの学習に有効か実験をした。

3.1 実験方法

学部生 10 名 (2 年次生 2 名, 3 年次生 1 名, 4 年次生 7 名) を実験協力者として実験を行った。実験では、協力者に 2 つの問題を C 言語で解いてもらう。そして、そのコーディングシーケンスを記録した。その後、記録データをランダムにシャッフルし、匿名化処理を施して配布し、システムにより再生・視聴してもらった。そして、視聴したデータについて、相互評価を含むアンケートに回答してもらった。

3.2 実験結果

協力者の 1 週間あたりのプログラミング時間を図 3 に、プログラミングを行う目的を図 4 に、また、視聴したプログラムのコメントの付け方について相手がどのような意識を持ってつけているかを聞いたので、図 5 に、また、関数の書き方についても聞いたのでその結果を図 6 に示す。

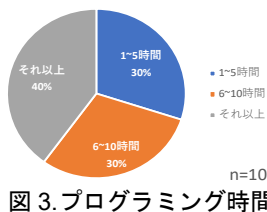


図 3. プログラミング時間

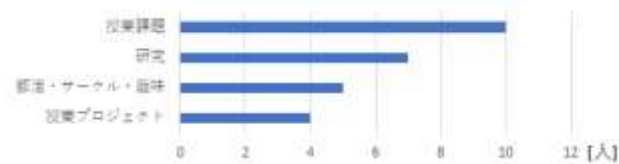


図 4. プログラミング活動

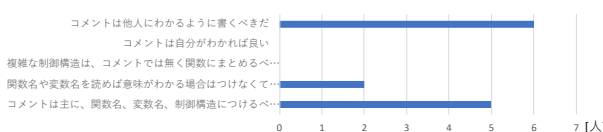


図 5. コメントの付け方について

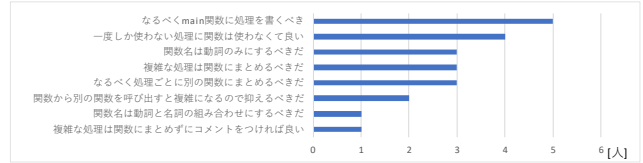


図 6. 関数の書き方について

4. 考察

協力者の特徴について見ると、図 3 より実験協力者のプログラミングを行っている時間に幅があることがわかる。このことからプログラミングにかかる時間の異なるメンバー同士で相互評価できていると考えられる。

図 4 より、すべての協力者が授業課題でプログラムを書いていることがわかる。また、趣味や授業内プロジェクトなどを行っている人が多いことから、プログラムを主体的に書いている協力者が多いことが伺える。

図 5 より、コメントは自分がわかれば良いと答えた人は 0 人であり、またコメントは他人にわかるように書くべきだと 10 名中 7 名が答えており、客観的に見て、可読性を意識して書かれたコードが多いと考えられる。

図 6 より、「関数名は動詞のみにすべきだ」と「関数は動詞と名詞の組み合わせにすべきだ」の両方に票が入っており、様々なコーディングスタイルに協力者が気付いていることがわかった。

また、他人のコードを見て良かった (参考になった) 点についてアンケートを取ったところ以下のような回答があった。

- 関数名や変数名に意味のある単語を使うと見やすく、他人が作ったものでも抵抗なく受け入れることができる。
- define を使って上限の数をコードに直接書かないようにしている。
- 変数名が見ただけで分かるようになっている。

このことから、学生同士でコーディングシーケンスを共有することは読みやすいコードの書き方について気づきを与えることができ、コーディングシーケンスの学習に寄与すると考えられる。

5. おわりに

本稿ではシステムの安定性の確保と学生同士がコーディングシーケンスを共有した場合の学習効果について述べた。

今後は学生同士がコーディングシーケンスを共有できる SNS 的特徴づけを行ったプラットフォームを構築し、実証実験を進める。

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

- 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).
- 滝本将司, 納富一宏: "実践的コーディングスタイル学習支援システムの開発 -教材提示と学生の反応-", 情報処理学会 第 82 回全国大会講演論文集 第 4 分冊, 7ZD-03, pp.415-416, (2020.03).