

プログラミング授業のための可視化システムと初学者の学習分析

小林 学†

荒本 道隆‡

佐藤 一裕†

平澤 茂一†

早稲田大学†

アドソル日進‡

1. はじめに

コンピュータシステムの高度化、スマートフォンやIoTの急速な普及、並びに人工知能やビッグデータ解析の重要性拡大に伴い、ソフトウェア開発技術者の確保並びに質向上は大きな問題となっており、プログラミング教育は益々重要となってきた。文部科学省の方針においても、初等中等教育段階におけるプログラミング教育の推進に力を注いでいる。一方初等中等教育のようなプログラミングの初学者にプログラミング教育を行うためには、教育用のシステムが学習者に優しいのはもちろんのこと、教員の負担軽減が重要と考える。

そこで本研究では初学者のプログラミング教育を対象に、WEB上でプログラムの作成及び実行を可能とし、同時に学習者の詳細な編集履歴を取得・可視化するシステムについて述べる。またそこで得られた編集履歴に対してデータ分析を行ない、エラーの分類による初学者の誤りの傾向と、自動採点手法の検討を行う。

2. プログラミング編集履歴可視化システム

プログラミング初学者にとって学習を行う際に以下のような問題がある。(1)プログラム開発環境の操作が複雑で、理解するのに時間と手間がかかる。(2)学外で自学するためにプログラミング開発環境の導入が必要で敷居が高い。(3)モバイル端末では学習できない。一方授業を行う教員にとっても以下のような問題点が挙げられる。(4)学習者の進捗状況を把握するのが困難。(5)学習者がどこで躓いているかをリアルタイムに把握するのが困難。(6)採点にかなりの労度が必要。

これらを解決するために、我々は図1のような構成のプログラミング学習システム、並びに編集履歴取得・可視化システムを構築した[4, 5]。学習者はMoodleで認証を行った後、リンクをクリックすると図2のプログラミング画面に飛ぶ。この時Moodleの機能で授業番号やUserIDが「教育用WEBサーバ」に渡され、個人の特定が可能である。学習者は図2の画面で課題プログラムを記述し、「保存&実行」ボタンを押すと「教育用WEBサーバ」にプログラムが保存され、「コンパイル+実行用サーバ」においてコンパイルと実行が行われる。この結果は「教育用WEBサーバ」において保存され、図2の画面の下方に表示される。教員は図3の画面において、授業番号及び課題番号をクリックすると、各学生のその課題のプログラム及び実行結果等を一覧

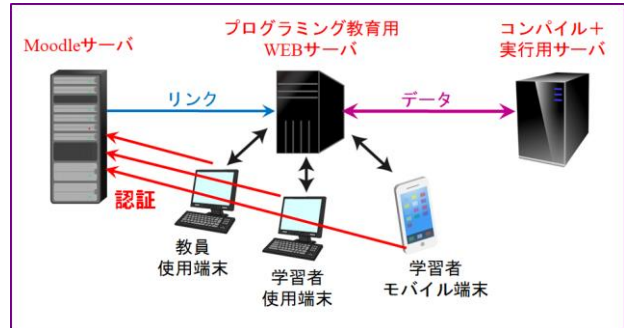


図1: 編集履歴取得・可視化システムの構成

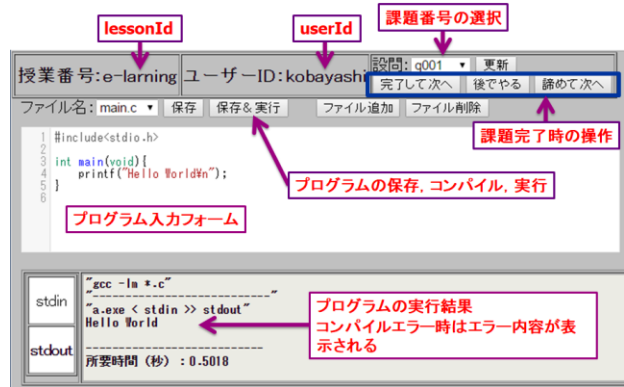


図2: 学習者のプログラミング画面

図3: 教員の編集履歴確認画面

の形式で確認することができる。これによりリアルタイムに学生の進捗状況を把握でき、また多くの学生が詰まっている個所なども把握可能である。また学生あるいは学校の個々のPC端末にプログラミング開発環境を導入する必要が無い点も大きなメリットである。

3. 学習分析

初学者を対象としたC言語の授業32回分の演習課題において取得したエラーログの分析を行った。ここで連続して同じエラーだった場合にはカウントしないものとする。図4に結果を示

Visualization System for Programming Course and Learning Analysis

Manabu Kobayashi†, Michitaka Aramoto‡, Kazuhiro Sato‡ and Shigeichi Hirasawa†

†Waseda University, ‡Ad-Sol Nissin Corp.



図 4 : エラーログの分析結果

す。この授業では初学者の約 35 % のエラーは変数の宣言ミス (打ち間違い) で、18%がセミコロンの付け忘れであった。また 7%のエラーは全角文字をプログラム中に書いてしまうミスである。エラーの表記は初学者にはとても分かりにくい。ため、頻出エラーは予め FAQ のようなものを作成して注意喚起しておくとう有効と思われる。

次に得られたプログラミング編集履歴を用いて、自動採点を行う手法の検討を行った。プログラミングコンテストや、E ラーニングのみの学習システムでは、実行結果の完全マッチングで判定を行っているケースが多い[7]。一方初学者に対する通常授業においては、完全マッチングのみでは判定が厳しすぎる可能性がある。本節の残りでは、この自動採点について述べる。

まず実行結果の完全マッチングによる判定を行ったところ、正解と判定できたプログラムは全体の 45.8%であった。教員が人手で判定した正解率は約 74%であったことと比較すると、明らかに低すぎる。人手で正解にも関わらずシステムが不正解と判定したプログラムを調査したところ、スペースやタブの挿入・不挿入、全角・半角の違い等、出力に対する揺らぎが結果に大きく影響を与えていることが分かった。

この調査結果を踏まえて、全角英数字は半角に統一、タブ、カンマ、全角句読点は半角スペースに置換、=や英数字の前後にスペース挿入、と言った正規化処理を行ってから、完全マッチングを行った。その結果、正解と判定されたプログラムは 66.3%まで増やすことができた。ただしこれでも十分では無い。

正規化処理によって正しく判定できなかった例として「プレイヤー」と「プレーヤー」、「最終の」と「最後の」、「1 9 2 3」と「1923」などの違いが挙げられる。また途中結果を逐次出力しているケースなども存在した。これらは正規化することができず、どちらかは誤りと判定されることになる。そこで、ある課題の正規化した実行結果に対してワード法を用いてクラスタリングした結果を図 5 に示す。

図 5 を見ると、明らかに 2 つの大きなクラスが存在し、それぞれのクラス内部の実行結果は完全に一致していることが分かる。また内容を確認すると、「プレイヤー」と「プレーヤ

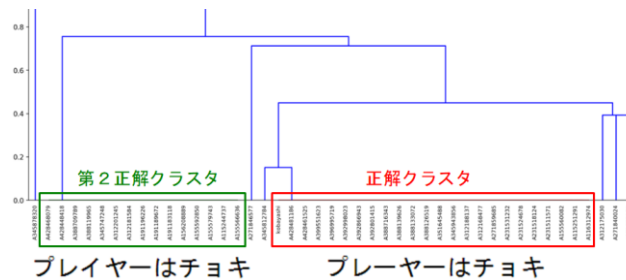


図 5 : クラスタリングの結果

ー」のみが異なっている。正解率が低く判定されている課題を調査したところ、そのような課題のほとんどに大きな 2 クラスが存在していた。そこで全ての課題に対してクラスタリングを行い、要素数が 2 番目に多いクラスを教員が手動で正誤を判定したところ、全体の正解率は 72.6%まで上昇した。これは全て手動で正解を判定した結果とほとんど変わらない正解率であり、むしろ手動での採点時に揺らぎをどこまで許容するかはばらつきの方が大きい。この方法は完全な自動採点ではないが、1 つのプログラム作成課題に対して教員が手動で判定しなければならないプログラムは高々 1 つで済むので、さほど労力を必要としない。結果的に学生全員分を手動で判定する手間と比べ、教員の労度を大きく減らすことに貢献でき、かつ採点者によるばらつきを少なくできることが分かった。

4. まとめと今後の課題

本稿ではプログラミング初学者の学習ログの分析方法について述べた。今後これらをシステムに機能として実装する予定である。

謝辞

本研究は JSPS 科研費 JP16K00491 の助成を受けたものです。

参考文献

- [1] 田口浩, 糸賀裕弥, 毛利公一, 山本哲男, 島川博光, “個々の学習者の理解状況と学習意欲に合わせたプログラミング教育支援”, 情報処理学会論文誌, Vol. 48, No. 2, pp. 958-968, 2007.
- [2] S. Fujii, K. Ohkubo, H. Tamaki, “MAX/C on Sakai - A Web-based C-Programming Course”, In Proc. of the 2nd International Conference on Computer Supported Education, pp. 196-201, 2010.
- [3] 玉木久夫, “Sakai を基盤としたプログラミング教育・学習支援システム”, 第 5 回 Ja Sakai カンファレンス, Vol. 2012 No. 3, 2012.
- [4] 小林学, 後藤正幸, 荒本道隆, 平澤茂一, “プログラミング編集履歴可視化システムとその実践”, 日本経営工学会 2015 年秋大会, 2015.
- [5] 荒本道隆, 小林学, 中澤真, 中野美知子, 後藤正幸, 平澤茂一, “編集履歴可視化システムを用いた Learning Analytics システム構成と実装”, 情報処理学会第 78 回全国大会予稿集, pp. 4-527-4-528, 2016.
- [6] paiza ラーニング : <https://paiza.jp/works>
- [7] AIZU ONLINE JUDGE : <http://judge.u-aizu.ac.jp/onlinejudge/index.jsp>