

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

荒本 道隆[†]
アドソル日進株式会社[†]
先端 IT 技術部

小林 学[‡]
湘南工科大学[‡]
情報工学科

中澤 真^{††}
会津大学^{††}
短期大学部

中野 美知子^{**}
早稲田大学^{**}
教務部

後藤 正幸^{**}
早稲田大学^{**}
経営システム工学科

平澤 茂一^{**}
早稲田大学^{**}
理工学術院 総合研究所

1. はじめに

プログラミング言語の学習環境において、特に多くの学習者を指導する場面では、言語の種類に関わらず様々な課題が存在する。

そこで本研究では、初学者に特化した学習環境を開発して実際の授業で使用してもらい、以下の課題に対しての有用性の確認を行う。

- ・学習環境の準備
- ・学習者の状況の把握
- ・学習者を分類するためのデータ収集

「学習環境の準備」の解決のため、HTML を使ってブラウザから利用可能な環境を作成した。そのため学習者は、PC だけでなくスマートフォンやタブレットからの学習も可能となった。

2. 対象とするプログラミング言語

本研究では、コンパイルして実行するプログラミング言語として、C/C++を対象とした[1]。学習者はブラウザ上でソースコードを編集し、編集結果を Web サーバに送付し、サーバ側でコンパイルと実行を行う。学習者がコンパイルと実行を行うたびに、ソースコード・コンパイル結果・実行結果を保存し、履歴も別途保存する。C/C++以外でも、コンパイラを使用する言語であれば、対応は容易である。

コンパイラを使用しない言語として、HTML/JavaScript も対象とした。HTML/JavaScript は学習者側のブラウザ内で実行するが、実行するたびにソースコードと実行結果を Web サーバに送付する。

キャラクタベースではない言語として、Scratch も対象とした[2]。Scratch の Flash 版は元々ブラウザを使用しているため、学習者の記

述内容を自動的に保存する機能を追加した。

また、プログラム言語以外への適用例として、英語の Writing も対象とした[3]。コンパイルや実行に相当する部分は、完全一致で正解かどうかを判定している。

3. システム構成

導入やメンテナンスを容易にするため、最も多く使われている Web サーバの Apache を使用し、サーバ側での処理は PHP で実装した。データはファイルシステム上にそのまま保存しており、データベースなどは一切使用していない。そのためシステムの導入は、Apache+PHP の実行環境に HTML と PHP ファイルをコピーするだけである。

C/C++だけは、サーバ側でコンパイルして実行するため、gcc が必要となる。また、学習者が任意のコードを実行できるため、悪意のある／なしに関わらず、他学習者や他の人に迷惑をかけるように、隔離した閉じた環境で実行する。

各履歴データは、ディレクトリ構造とファイルで管理している。それを講師がリアルタイムに参照する事で、学習者の状況を把握できる。

学習者が使用する画面は、各言語に特化するため、各言語専用の画面を作成した。ただし、Web サーバと画面のやり取りは「ファイル操作 API」を共有して開発を省力化している。

講師が使用する画面は、全ての言語で使用できる「共通履歴可視化画面」を作成した。さらに、必要に応じて言語ごとに専用の可視化画面も作成した。

4. 学習者用の画面

学習者用の画面は、その言語の学習において必要最低限の機能のみを実装し、操作方法で悩まないようにした。

C/C++用の画面は、ソースコードと標準入力を編集可能とし、標準出力が参照できる。また、バイナリデータも扱えるように、16 進数での編集モードも実装した。

Learning Analytics via Visualization System of Edit Record
～ System configuration and implementation

[†]ARAMOTO Michitaka, Ad-Sol Nissin Corp.

[‡]KOBAYASI Manabu, Shonan Institute of Technology

^{††}NAKAZAWA Makoto, The University of Aizu

^{**}NAKANO Michiko, GOTO Masayuki, HIRASAWA Shigeichi, Waseda University

HTML/JavaScript 用の画面は、1つのファイル中に HTML, JavaScript, CSS をまとめて記述し、画面右側で実行結果が確認できる。JavaScript の初学者が一番陥るのが、ブラウザのデバッグ画面を開いていないとエラーの発生に気付かないので、図1のように画面上に表示した。

Scratch 用の画面は、オープンソースの Flash 版の Scratch に対して「一定周期で保存処理を呼び出す」という機能を追加しただけで、画面デザインは変更していない。Scratch が初学者用の開発環境なので、元のままで十分に使い易い。

英語の Writing 用の画面は、画面上部で誤った英文を修正し、画面下部で検証結果を確認する。誤りの表示は、当初は文字単位だったが、実際に使うと『当たりの文字を探す』という作業となってしまうので、単語単位とした。

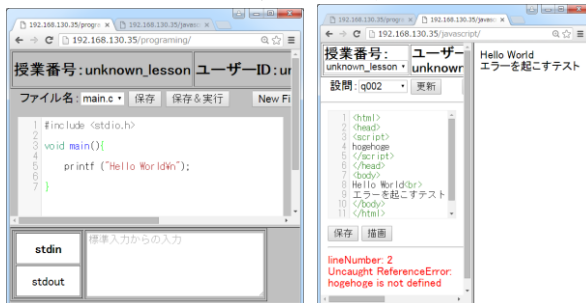


図1 C/C++とHTML/JavaScript用の画面

5. 講師用の画面

学習者が記述したソースコード・コンパイル結果・実行結果・その他のファイルの全履歴を表示する画面を作成した。

講師は、全学習者が各設問に対して「完了」「後でやる」「諦めた」とした数をリアルタイムに参照し、全体の進捗を確認できる。ただし、各状態は学習者の自己申告であり、求めている実行結果を得られていないのに「完了」としている場合もありうる。画面から、特定の学習者の各設問や、全学習者の特定の設問に対し、ソースコード・実行結果を横断的に参照できる。

Scratch は、保存形式の JSON のままでは分かり難いので、学習者と同じ Flash による表示モードも追加した。

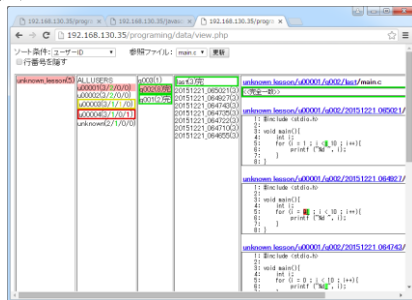


図2 講師用共通履歴可視化画面

6. 差分表示

講師用の画面では、図2で示した通り、ソースコードや実行結果の1つ前に対する差分を表示している。

ソースコードの差分を見れば、学習者がどこをどう修正したかを確認できる。実行結果の差分を見れば、結果がどう変化したかを確認できる。差分表示機能が無ければ、どこを修正したか/どこが変わったかを目視で探し出さなければならない。

7. まとめ

実際に授業で使った講師からは、「去年と比べて、授業が前倒しで進められるようになった」との感想を頂いた。

主な理由としては、学習環境のインストールの手間がかからず、各PCでのトラブルが起りにくい、が考えられる。特に学習者のPCでトラブルが発生すると、その1台の問題解決に講師が拘束され、その間は授業が止まってしまう。しかし本システムでは、トラブルが発生した場合でも別PCのブラウザからアクセスすれば学習を再開できる。また、全学習者の進捗状況を俯瞰して授業の速度を調整し、多くの学習者が同じようなミスをしていたら全体に対して指導や注意喚起が可能になり、安心して授業を進めることができる。収集したデータをリアルタイムに目視できるだけでも、十分に有用である事が確認できた。履歴データの分析機能を実装すれば、さらに有用性が高まると思われる。

課題として、学習者に高度な機能を提供するためには複雑な実装が必要になり、各プログラミング言語に特化した機能も必要になる。HTMLでの実装は限界があるので、既存の各言語専用の開発環境を拡張し、「ファイル操作API」を使ってソースコードや実行結果をWebサーバに送信した方が良さそう。

参考文献

- [1] 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”, 日本経営工学会秋季大会予稿集, pp. 8-9, 金沢, 2015年11月28日.
- [2] 中澤 真, 後藤正幸, 荒本道隆, 平澤茂一, “ビジュアルプログラミング言語「Scratch」のための学習履歴分析環境とその可能性—初等教育からのプログラミング教育に向けて—”, 日本経営工学会秋季大会予稿集, pp. 10-11, 金沢, 2015年11月28日.
- [3] 中野美知子, 荒本道隆, 吉田諭史, “プログラミング言語の学習ログ収集ソフトウェアを活用した文法矯正練習の試み”, 日本経営工学会秋季大会予稿集, pp. 12-13, 金沢, 2015年11月28日.