

授業での初学者用プログラミング学習環境における学習者状況の可視化

布施 泉¹⁾, 中原 敬広²⁾

1) 北海道大学 情報基盤センター

2) 合同会社 三玄舎

ifuse@iic.hokudai.ac.jp

Visualization Method of Learner's Situation in Programming Learning Environment for Beginners in Class

Izumi FUSE¹⁾, Takahiro Nakahara²⁾

1) Information Center, Nihon Bungaku Univ.

2) Sangensha LLC.

概要

プログラミング初学者は様々なミスをしながらか学習を進める。本稿では、学習者の状況に応じ適切な支援をできるように、著者らがこれまでに構築してきた初学者に対する授業用プログラミング学習環境において、学習者状況の可視化の検討を進めた。特に、本稿では、学習者からの能動的動作である「質問」「回答」機能と、教師側での学習者把握として、時間区切りをした中での学習者状況の可視化を行う機能を中心に検討し報告する。

1 はじめに

著者らは、授業での利用を前提とした初学者用のプログラミング学習支援環境を開発し、当該環境を用いた教育実践を行っている[1][2]。当該環境の特徴は、教師が自身の授業資料を学習者の状態に合わせて提示でき、学習者は、授業資料の閲覧、プログラムの編集、実行までを一画面で行うことができる点にある。また、Moodleプラグインとして開発しており、教師は学習者のウェブブラウザ上での実行等の履歴を管理できる。学習者は自宅でも大学でも同じ学習環境でプログラミングの学習を行うことができる。

本システムを利用し、数年間、授業で用いたところ、教師としては、学習者の状態把握を、より簡便に行う機能の必要性を感じた。一方、学習者側からは、支援が必要と感じた際に、その状況を通知できるような機能があると有用であると考えた。多人数授業で、学習者にレベル差がある場合にどのように授業運営を行うかという点での支援環境の拡充である。本稿では、学習者の学習状況の可視化について焦点を当てて検討を進める。

2 プログラミング学習環境と可視化

2.1 授業での初学者用プログラミング学習環境

著者らが開発したプログラミング学習環境を

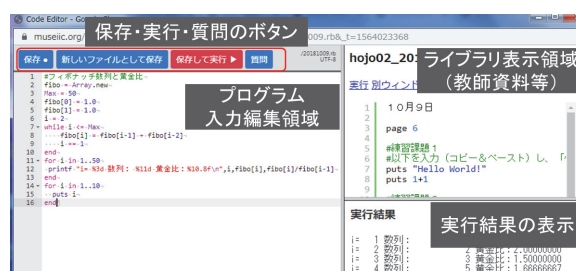


図1 プログラミング学習支援環境

図1に例示する。プログラミング言語は、Ruby, Python, JavaScriptに対応している。

本学習環境上では学習者毎にフォルダ領域が割り当てられ、その中で自由にプログラムを作ることができる。作成した各プログラムファイルをダブルクリックすることで、ファイル毎に図1のようなプログラミング支援環境が立ち上がる。本環境は1画面で学習者がプログラム作成に必要な操作ができるように工夫して構築している。画面右上は、学習者の取り込んだライブラリ内のファイルを表示する領域である。本システムでは、学習者は、公開ファイルをギャラリーと呼ばれるタブ内から取得し、自身のライブラリ領域に蓄積しておくことができる。授業では、学習者は教師から指示された授業資料を取得し、指示された資料を本領域で表示していることを想定する。画面の左半分は、プログラムの入力・編集を行う領域であ

る。右上の教師の授業資料を確認しながら、プログラムの編集作業を行うことができる。できたプログラムは、左側の上部にある「保存して実行」ボタンにより、実行することができる。実行結果は画面右下の領域に表示される。実行結果に誤りがある場合には、何度も修正しながら実行することができる。

2.2 学習者の状況可視化の必要性

大学のプログラミングの授業では、通常は一名の教師に対し、数十名程度以上の学習者で構成されることが多いと思われる。この人数比で、特に初学者を対象とした場合、学習者の進捗の差が大きくなることが多く、どの程度の学習者がどの程度の理解の度合いであるかを把握することが重要になる。しかしながら、授業を進行しながらの学習状況の把握は一般には難しい。

第一著者の授業では、授業において、毎回の作業記録を学習者に書かせ、学習者の状況把握に努めていたものの、それは授業終了時の記録であり、適切なタイミングでの十分な支援にはなりにくい状況である。

一方、本システムは、Moodleの枠組みでコース管理されており、本システムのログを組み合わせることで、機械的に学習者の状況を把握する支援を行うことができる可能性があると考えている。

本報告では、システムから把握できる状況の可視化とともに、学習者には、自身からの質問・回答を行うことのできる機能を検討する。学習者自身が授業終了時の記録だけではなく、授業時の状況に応じ、教師もしくは他の学習者にコメントを求める機能である。

3 学習者の状況可視化の機能

3.1 履歴状況の可視化

図2に時間毎の学習者状況の可視化例を示す。教師は管理画面から、可視化をしたい開始・終了時刻、及び、時間幅を指定し、「表示期間を設定」ボタンを押下することで表示する。図2は、60秒毎の可視化例となる。横軸が時間の流れであり、1分毎の各学習者の状況が示されている。縦軸は各学習者に対応する。

学習者の状況は、色で区別しており、各色の意味は以下の通りである。

緑：実行および実行完了（エラーなし）

赤：実行エラー

薄青：編集開始

橙：質問

黄：質問に回答

濃青：ファイル保存

紫：ギャラリー閲覧

桃：ライブラリ閲覧

緑が続いている場合には、プログラムの実行を行っている状態であり、各人は基本的に問題なく進んでいることが推察される。

図2は2018年度の授業開始時の記録を可視化したものである。この時点では、実行時にエラーログを区別しておらず、また質問機能はなかったため、一部の色は見えていない。しかし基本的な流れとして、学習者がプログラムを編集し、ライブラリを閲覧し、プログラムを実行している様子が見て取れる。状況に応じ、学習者の動向を、時間刻みを変え、より詳細を確認することができる。

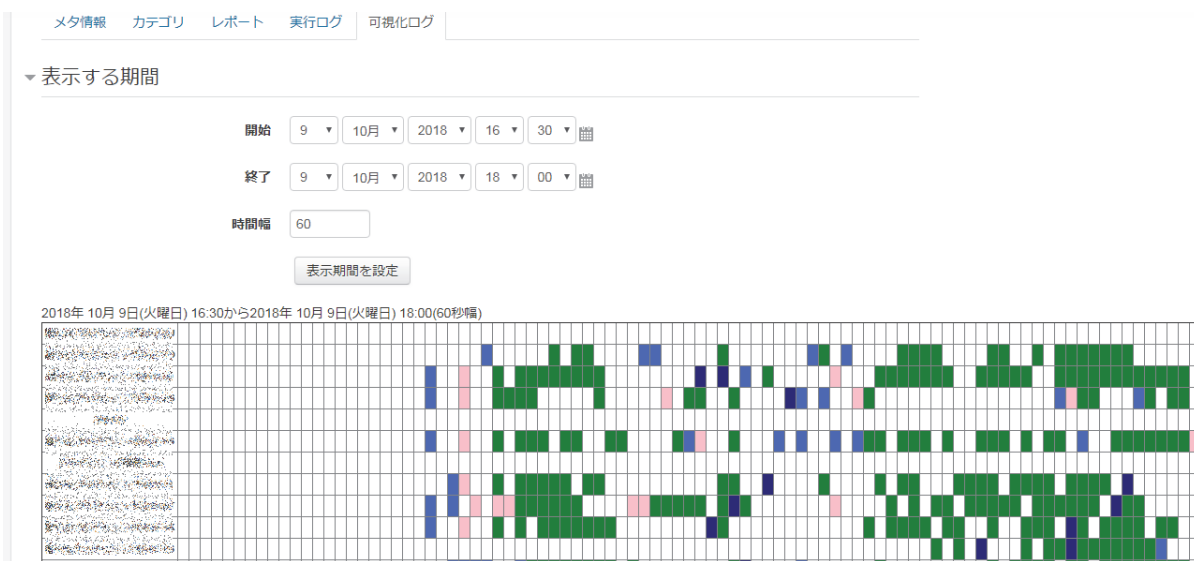





図2 学習者の指定時間における状況可視化の画面例

3.2 学習者の実行コードリスト

図 2 で、気になる状況が表出した場合には、個別の学習者のプログラムの実行状況を確認することができる。図 3 にその画面例を示す。図 3 は指定された期間における最新の実行プログラムを一覧したものである。本画面において、想定されたファイル名(課題名に対応する)でない場合には、何らかのトラブルが学習者に発生していることが懸念される。図 3 の画面右には、プログラムコードの操作が 3 種類用意されている。操作欄左のメモ帳のアイコン  を押下すると、プログラムコードが閲覧できる。中央のアイコン  は、押下することで、当該プログラムコードの実行を行い、実行結果を表示する。右のアイコン  は、押下することで当該学習者のフォルダ内を確認することができる。図 3 における左列のユーザ名のリンクをクリックすると、当該ユーザにおける表示期間内での実行履歴を確認することができる。



ユーザ名	ファイル名	日付	操作
181023	-	-	
181023	新規ファイル 1.rb	2018年 10月 9日(火曜日) 17:54	  
181023	kadai1.rb	2018年 10月 9日(火曜日) 17:59	  
181023	181023.rb	2018年 10月 23日(火曜日) 17:46	  

図 3 学習者の実行プログラム一覧の画面



ファイル名	日付	操作
181023.rb	2018年 10月 23日(火曜日) 17:46	  
181023.rb	2018年 10月 23日(火曜日) 17:46	  
181023.rb	2018年 10月 23日(火曜日) 17:42	  
181023.rb	2018年 10月 23日(火曜日) 17:42	  
181023.rb	2018年 10月 23日(火曜日) 17:42	  

図 4 あるユーザの実行プログラム一覧画面

```

181023.rb
実行 < >

1  a, r, n = 100_0000_0000, 0.5, 500
2  t = a*(1+r)**n
3  printf "500年後の人口は %d 人",t
4
5  puts 2 == 2
6  puts 3 > 1
7  puts -1 > 0
8  puts "a" > "b"
9  puts "こんにちは" == "こんにちは"
10 puts 2 < 1
11
12
13
14 #条件分岐 --if修飾子--
15 h, w = 1.60, 100
16 bmi = w / h ** 2
17 puts "やせ" if bmi < 20
18
19 #条件分岐 --if文--
20 h, w = 1.70, 66.0 #値を変えて実行してみる
21 bmi = w / h ** 2
22 if bmi < 20
23   puts "やせ" #字下げ (プログラムを見易く
24 else
25   puts "普通あるいは肥満"
26 end
  
```

図 5 あるユーザのファイルの履歴確認

図 4 で、問題が生じているプログラムファイルがある場合には、当該ファイル名のリンクをクリックすることで、図 5 のようなファイル内容を確認することができる。上部にファイル名が表示され、その下に、「実行」リンク、<>で当該ファイルの前後の実行履歴を一覧できる。各実行時のプログラムファイルを教師も確認し、実行することができるため、問題が起きた学習者の状況をトレースすることが可能である。図 5 のプログラムは、全角空白が原因のエラーがあったことが確認できた。

3.3 質問・回答等の学習者の能動的行動の可視化

前節では、全体状況から個別の実行ログを確認して問題を把握可能な機能について説明した。しかし、多数の学習者がいる授業では、授業時に個別の問題解決の状況を一名の教師が確認する余裕がない状況もあると思われる。その場合、補助者である TA 等に確認を依頼することはあり得るものの、それ以外の解として、学習者自身の質問行動を支援する機能を検討した。質問ボタンとそれに対する回答を行う機能である。本機能のポイントは以下の 3 点である。

- 各プログラムファイルと質問フォームを紐づける (図 6 参照)。
- 質問者自身が、教師のみか、他の学習者を含めてかを選択して質問できる (図 6 参照)。
- 他の学習者宛での質問であれば、他の学習者が確認し、コメントも可能である (図 7 参照)。



図 6 質問フォームへの記載例

図 6 は、実際の質問画面例である。図 1 に記載したプログラム実行画面上部に、「質問」ボタンを配置しており、当該ボタンを押下した場合に、図 6 の表示が出る。各プログラムで押下した質問フォームであるため、質問とプログラムの紐づけが可能である。

質問フォームは、質問内容、相手（「教師に」「みんなに」を選択）、座席（記載は任意）の 3 項目から構成される。座席は、質問者が授業時には着席の位置を記載することにより、教師に自分の位置の情報を教え、支援を受けやすくする機能である。

質問を送信すると、質問タブにその内容が表示される。また、Moodle トップのメッセージで、質問が届いたことを確認できるようにした。図 7 は管理者における質問タブでの質問リストの表示例である。質問とともにプログラムが紐づけられており、実際にコードを実行 ▶ して確認し、結果をアイコン 🗨️ からのコメントも可能である。



図 7 質問タブにおける質問リスト例



図 8 各質問に対するコメント表示例

図 7 には、他にも目のアイコンとゴミ箱のアイコンがある。目のアイコンは、当該質問の表示・非表示の制御、ゴミ箱は、当該質問の削除に対応している。

図 8 は、コメントアイコンをクリックした際の例である。「みんなへ」質問をした場合には、教師もしくは他の学習者のコメントがその他の学習者にも閲覧できることがわかる。このような質問・回答の表示を制御することで、学習者の理解度の向上を図り、スキルの差を縮めることに寄与できないかを検討している。なお、図 7, 8 では実名が表示されているが、実際の授業では、学習者の実名は見えなくすることを想定している。

3 考察

本可視化機能は、実装をして間もないため、2019 年 10 月から授業実践で、実際の評価を行う予定である。本章では、どのような状況が予想できるかについて考察する。

3.1 学習者集団の状況の可視化

参考文献[1]では、Moodle の履歴から今回の開発機能である各時間でのシステム利用状況を描写している。しかし、これは履歴ログから手作業で整理して行う必要があったため、リアルタイムでの描写は無理であった。今回、図 2 に示したような図は、授業時にもリアルタイムで行うことが可能である。図 2 では、上から 2 番目の学習者の状況が他の学習者と比較して特異で

あることが一目にわかる。具体的には、ライブラリの閲覧を示す（ピンクの）区分がない。これは、当該学習者が、本システムでのライブラリの表示ができていないことを表す。この日はまだ履修者が確定していないため、当該学習者は履修の希望が弱く話を聞いていないか、または指示された内容が理解できず、困っている状況が推察される。

今回の可視化機能を用い、他の学習者と明らかに異なる動作をしている学習者の状況を確認することが可能と思われる。さらに、実行時のエラーログを採集し、今回から表示が可能になったことから、明らかにエラーを頻発させ、問題を起こしている学生の抽出も可能となると考えられる。そのような学習者のプログラムの実行ログ等を授業補助者とともに確認し、対応していくことが有効ではないかと考えている。

3.2 質問・回答機能について

第一著者の授業では、これまででも、本システムを用いて、学習者に質問をすることについて、促してきた。具体的には、ファイルを公開する機能を用い、質問がある場合には、ファイルを公開してもらえれば対応する旨の連絡をしていた。しかしながら、実際に質問をする学習者は少ない。

理由は2点あると考えている。第一は、プログラムファイルを質問するために、公開という手間がかかることである。第二には、自分のプログラムは同じクラスといえど、不特定の学習者に見せるということの心理的な壁があることが予想される。

今回の機能は、この2点の問題に対し、一定程度の解決をもたらすと考えている。まず、プログラムファイルを編集・実行する画面で、直接質問ボタンから質問ができることである。これにより、エラーが出続けた場合には、実行画面ではなく、質問ボタンで質問することができ、質問への余計な負荷や手間はかからないものと考えている。第二に、質問者は全体への学生ではなく、教師のみに質問をすることができる。また全員に対して質問をする場合にも、学習者のユーザ名等の表示がないことで、より気兼ねな投稿がしやすいものと考えている。

また、教師の視点では、「みんな」に対する質問に対し、回答をした学生には、その寄与の度合いによって評価をすることも考えられる。さらに

は、教師のみへの質問が多い場合、同じような質問に対し、教師が代わりに同様の質問・回答を学習者全体に提示することで、他の困っている学習者に対しても、まとめて支援をすることができるようになると考えている。

4 まとめ

本稿では、授業での利用を前提とした初学者用プログラミング学習環境において、学習者の状況の可視化について検討を進めた。本稿では、特に、学習者の本システム利用ログの可視化と、学習者の質問・回答の機能について、報告した。

学習者の利用ログの可視化については、各学習者のシステム利用状況を色分けして表示する機能を追加した。指定した時間帯で、指定した時間幅でその利用状況が色分けして可視化される。授業でのリアルタイムの利用により、他の学習者と差異のある学習者をより早く抽出できることが期待される。実際に気になる学習者がいた場合には、個別の実行ログを確認して状況の詳細を確認することも可能である。また、実際の学習者の下に行き、支援を行うことも可能であると考えている。

学習者による質問・回答機能については、これまでの本システムを用いた質問のしにくさを回避していると考えている。質問者の希望により、回答を他の学習者に委ねることも可能とした。本質問・回答システムの有効な利用方法については、実際の授業を踏まえ、学習者の状況に応じて、検討をさらに進めて行く予定である。

これらの機能は2019年10月からの授業において評価を進めていく。

参考文献

- [1] 布施泉, 中原敬広, 岡部成玄, "プログラムの相互利用と相互評価が可能な初学者用プログラミング授業支援環境の構築", 教育システム情報学会誌 Vol35, No.2, pp.221-pp.226, 2018.
- [2] 布施泉, 中原敬広, 岡部成玄, "授業での利用を前提とした初学者用プログラミング学習環境の開発", 大学 ICT 推進協議会年次大会, FE22, 2016.