

プログラミング学習支援環境 Bit Arrow の教員支援機能の設計と試作

長島 和平^{1,a)} 堀越 将之² 長 慎也² 間辺 広樹³ 兼宗 進⁴ 並木 美太郎¹

概要: プログラミングの授業で、学習者や教員の支援を行える環境を用いて演習をすることで進行を助けることができる。次期学習指導要領では、高等学校においてプログラミングが必修となり、初学者が多い現場では学習者と教員の両方を支援する環境を扱うことが重要になる。著者らはオンラインプログラミング環境 Bit Arrow を開発している。昨年度までの実践では、学習者を支援する機能を実装し、これまで用いられてきた環境に比べて、文法エラーを発生させる割合が減り、文法エラーからの復帰も早くなったことが確認できた。一方、教員の支援についてはこれまで十分な機能が提供されていなかった。例えば、学習者が今のようなエラーに悩んでいるか、作業が止まっている学習者がいないかといったクラス全体の状況を把握したり、課題を設定、提出させたりする機能が実装されることで授業を円滑に進めることができると思われる。本稿では、Bit Arrow がプログラミング授業支援環境として教員に提供する、履修者管理や学習者の状況把握、課題等ファイルの配布、採点インタフェースといった機能の設計と、一部の試作状況について述べる。

Design and Prototype of Teacher Support for Bit Arrow

NAGASHIMA KAZUHEI^{1,a)} HORIKOSHI MASAYUKI² CHO SHINYA² MANABE HIROKI³
KANEMUNE SUSUMU⁴ NAMIKI MITARO¹

1. はじめに

次期学習指導要領では、共通教科情報でプログラミングが必修となることが決まった。現在の共通教科情報でプログラミングを扱う「情報の科学」を履修している生徒は約2割しかおらず、次期指導要領の適用と同時に多くの高校でこれまで扱ってこなかったプログラミングの授業が行われることになる。こうした現場において、学生と教員の双方をサポートする学習環境を用いることが必要になる。

現行の教科書には「Web ブラウザを用いた JavaScript の

実習」や、「表計算ソフトを用いた BASIC の実習」などが扱われている。これらの環境はプログラミング学習向けに開発された環境ではないが、多くの PC でインストールをせずに使用できることから利用されていると考えられる。プログラミング学習に適した環境をインストールするには管理者権限が必要で、作業の手間や導入にかかる費用の問題もあり容易でない。そこで、Web ベースのプログラミング学習環境の有効性が指摘されている。

こうした背景から、著者らはオンラインプログラミング学習環境 Bit Arrow を開発している。これまでは文法エラーのメッセージと場所の表示や、プログラムを短く書けるようにするライブラリの提供など学習者を支援する機能を優先して実装してきた。Bit Arrow を用いた授業で、実行時にエラーが起こる割合が少なく、エラーからの復帰が早かったことを確認できた [1]。一方で、これまでの Bit Arrow には教員の支援をする機能はほとんど実装していなかった。本論文では、Bit Arrow でサポートする教員支援

¹ 東京農工大学
Tokyo University of Agriculture and Technology
² 明星大学
Meisei University
³ 神奈川県立柏陽高等学校
Hakuyo High School
⁴ 大阪電気通信大学
Osaka Electro-Communication University
a) kazuakanapo@namikilab.tuat.ac.jp

機能についてと、本稿執筆段階で実装が行われている一部機能について述べる。

2. プログラミング教育で必要な教員支援機能

プログラミングの授業において教員は、学習者の状況を把握したり、演習課題の管理をしたりする必要がある。こうした仕事の負担を軽減するためのプログラミング演習環境や、学習支援システムなどが開発されてきた。これまでに開発されてきた環境では、課題の設定を行い、その実行状況などを元に学習者の進捗を教員に提供したり、課題を自動採点したりして教員の支援を行ってきた。

長谷川らのシステムでは、プログラムを採点し、評価項目に対する正解・不正解や提出時間から順位付けを行い、学習者の状況を表示している [2]。内藤らのシステムでは、課題の進捗の提示や、発生したエラーをランク付けして指導の支援を行っている [3]。しかし、これらのシステムでは、学習者がプログラムを作る環境はそれぞれの手にインストールさせていたり、課題プログラムをそれぞれの学習者のホームディレクトリに置かせて採点したりしており、学習者への支援はあまり重視されていない。

太田らは、タブレット端末を用いて巡回指導を支援できるツールの開発を行っている [4]。田中らは、プログラミング教育を改善するための学習記録の分析を行い、それを教員に提供している [5]。しかし、課題の提出と学生側からの質問に対してのみサポートが可能で、細かい実行履歴は見られていなかったり、授業全体を振り返るための情報提供のため、授業中にリアルタイムで苦しんでいる学習者を発見するようなサポートは欠けている。

Spacco らは、保存や実行ごとにプログラムの状態を保存したり、提出時の自動採点を行ったりする Eclipse のプラグインを開発した [6]。Papancea らの CloudCoder ではキーストローク単位でログを収集して演習への取り組みの様子を示したりすることができる [7]。しかし、Eclipse のようなインストールが必要な環境は教育現場への適用が難しい場合が多く、CloudCoder も教員側でサーバを立てて動かさなければならないため、準備する負担が大きい。

加藤らの授業支援システムでは、課題登録や提出状況と進捗の閲覧などができるとともに、学習者も Web アプリケーションで実行を行うことができる [8]。Elamir らのクラウドを用いたプログラミング環境でも課題の設定とその課題への取り組みに基づく進捗の把握を行える [9]。Efopoulos らのプログラミング環境 WIPE も、コンパイルや実行を記録し、前のコードからの変更点などが教員に提供される [10]。しかし、こうした環境は自動採点を行うためにグラフィックスを用いた演習ができない場合がほとんどである。文字を出力させるだけのようなプログラムは初学者にとって興味を惹くものではなく、普通高校などで用いると学習者が苦痛を感じるおそれがある。

これらの先行研究から、教員を支援するために必要とされる機能を次に挙げる。

- クラスの作成や学習者の管理
- 学習者の状況の把握
- 課題の設定や提出と成績管理
- 採点支援

また、Bit Arrow は、プログラミング学習支援環境であるため、学習環境と学習管理システムが同じ環境になる。Bit Arrow では JavaScript, ドリトル, C を利用でき、各言語でグラフィックスを用いた出力結果を出すこともできるため、次のような機能も必要であると考えられる。

- 学習者へのファイル配布
 - グラフィックスを用いた課題の採点支援
 - クラスごとの使用言語やサポートなどの設定
- これら必要とされる機能について次に述べる。

3. 教員支援機能の設計

3.1 クラスと履修者の管理

プログラミングの授業を行うためにはクラスの管理が必要である。クラスの作成や履修者の登録を行うことで授業を始める準備ができる。現在公開されている Bit Arrow では、クラスの作成は教員の依頼を受けて開発者が行っていた。この方式では、教員がすぐにクラスを作成することができず、開発者の登録を待機しなければならない。教員は自らクラスの作成を行えるようにすべきである。また、これまでの Bit Arrow では、事前にユーザ登録を必要としていなかった。ログイン時には、クラス内で共通のクラス ID と、任意のユーザ ID を用いるが、このとき登録していないユーザ ID が入力された場合は新規ユーザとしてログインできた。教員側ではこうして登録された学習者の一覧は見ることはできたが、登録を行うことができなかった。この仕様は、準備を簡単にするために採用されたが、学習者がこれまで使っていたユーザ ID を間違えてログインするとこれまでの授業で作成したファイルがなくなったように見えてしまう問題があった。こうした点から、学習者の登録も教員側から管理できるようにする必要がある。

学習者を登録する際、学習者の名前などを同時に登録させることで、学習者がユーザ ID やパスワードを忘れたといったトラブルの対処に役立たせることができるほか、課題の採点結果などと合わせて成績管理にも役立たせることができる。

3.2 学習者の状況の把握

次に、学習者の作業の状況を把握することが必要である。学習者の状況を把握することで、現在多発しているエラーを知ることができたり、同じエラーなどで長時間悩んでいる学習者を見つけることができ、適切な指導を行うことができる。これまでの Bit Arrow では、学習者の作業履

歴のタイムラインを提供していた。提供していたタイムラインでは、学習者が実行とエラーを発生させた時間を見ることができた。また、各実行の詳細情報として、そのときのファイル名、プログラム、エラーメッセージといった情報を確認することができる。この情報から、学生がエラー続きで苦勞していることや、そのエラーの原因を把握することができる。しかし、これらの情報ではクラス全体で多く起きているエラーや、手が止まってしまっている学習者がいるかといった状況を一目で確認することは難しい。この情報と同時に今多く起きているエラーや学習者の作業状況も提供することが必要である。

クラスの学習者が現在、どのようなエラーを多く起こしているかを通知することで、クラス全体への効果的な指導を行う支援をすることができる。また、長い時間エラーに苦しんでいた、実行の手が止まっている学習者を教員に提示することで、個別に指導やアドバイスが必要な学習者を教員が把握する助けになる。

3.3 課題の管理

プログラミング演習では、教員がプログラムの説明をしながら演習を行った後で、学習した概念を用いた課題が出されることが多い。また、授業によって課題の内容は異なっており、0からプログラムを作らせることもあれば、プログラムの骨組みは教員側で提供してその一部分を埋めさせることもある。課題を設定させるにあたり、特に教員が骨組みを与えて中身を埋めさせる形式の課題では、与える骨組みを学習者に配布する必要もある。最後に、学習者は課題に取り組み、完成させたら提出しなければならない。こうした課題の設定や登録、骨組みなどを記載したファイルの配布、完成した課題の提出といった機能は Bit Arrow では提供していなかった。

課題を設定し、そのためのファイルを学習者のワークスペース内に配布できるようにすることで、他の学習支援システムに課題を設定したり、そこに提出されたファイルをダウンロードする手間をなくすことができる。さらに演習画面から提出できるようにすれば、学習者も課題を完成させた後に提出のために他のシステムへページ移動する手間を省くことができ、プログラミングに集中できる。

3.4 課題の採点

プログラミング教育の場では学習者の増加に伴い、教員側では採点や成績管理の負担も大きくなっている。自動採点を導入することで、教員の採点にかかる負担を軽減することができる。学習者も素早く正誤判定を知ることができる。これまでに存在した自動採点を行うシステムは、コンソール上に文字を出力するようなプログラム課題が対象であることがほとんどであった。しかし、Bit Arrow で利用できるドリトルや JavaScript という言語では、結果にグラ

フィックスを使ったプログラムも多く、C でも独自の実装でアニメーションを扱うことができる [11]。自由作品を作らせる課題ではこれまでに用いられてきた自動採点機能は利用できないため、異なる方法での補助が必要となる。また、文字を出力するようなプログラムであっても、出力結果がただあっていればいだけでなく、特定の構文を使うような指示がある場合もある。こうした課題でも、使用を義務づける文字列や構文などを課題の設定時に行い、採点時にその設定を考慮するような仕組みが必要である。

自動採点ができる課題については自動で採点をおこなうことによって、教員がプログラムの実行結果を見て、プログラムの中身を確認するという手間を省いて教員の負担を軽減することができる。また、自動採点ですぐに結果が返ってくることで、学習者は提出と同時に課題の成否を確認することができるため、採点を待つ時間などをなくすことができる。

3.5 クラスごとの設定

学習者のレベルに応じて、クラス毎に設定を変更したいことがある。Bit Arrow では、複数の言語を使うことができ、言語によっては特別なライブラリを用意している。学習支援として、エディタには自動インデントや全角スペースを自動的に半角に直すような機能もある。クラスごとに使える言語やその機能を制限したり、学習者にパスワード管理をさせたくないためパスワードを使用させないようにしたりすることで、教員が教えたいことやクラスのレベルに適した環境にすることができる。

クラスごとの言語や機能などの設定を行うことで、教員が教えたい内容に適した状態に環境を調整させることができる。例えば授業内で言語を1種類しか使用しない場合は、学習者が言語を選択する必要もなくなるため、自動的に授業で使用する言語を使うように設定することで、環境で必要な操作を減らし、プログラミングに集中できる。

4. 教員支援機能の実装

3で述べた支援機能の設計に基づいて、Bit Arrow に支援機能の一部を実装した。本稿執筆時点での実装状況を本章で述べる。

4.1 クラス作成と履修者登録

クラスの作成と履修者登録は、クラスを履修する学生の管理を支援し、学習者のユーザ ID などに関するトラブルを減らすことができる。

これまでは教員はクラスを作成するたびに開発者への申請を行わなければならなかったが、これは効率に欠ける。そこで、初回申請時に教員を登録し、教員権限でクラスに関わる管理を行えることとした。クラスは既に存在しているクラス ID を除き自由に作成できることとした。また、

これまでは学習者がログイン時にパスワードを必要としなかったが、設定できるようにした。履修者の登録は、ユーザ ID とパスワードを csv ファイルを用いて一括で行えるようにし、学習者の名前も登録できる。教員が一括で登録を行うことで、誤った登録を防ぐことができる。

ユーザー一覧画面では、ユーザ ID とパスワード、履修者名の情報が確認できる。パスワードは通常非表示だが、学習者が忘れた時に教員側で確認することができる。この情報を提供することで、学習者がユーザ ID やパスワードを忘れたときにも教員側で対応することができる。

4.2 学習者の状況一覧

学習者の状況一覧では、履修している学習者の今の状況を把握することを支援し、現在困っている学習者や作業が止まっている学習者を見つけることができる。また、クラス内で頻発しているエラーの情報を元にクラス全体に注意や指導を行うことができる。

これまでは実行とエラーの状況をタイムラインで表示させていたが、その時間内のエラーの割合や、前回実行からの経過時間などの情報は一目見ただけでは分からなかった。リアルタイムで状況を把握するためには苦勞している学習者を見つけやすいよう工夫する必要がある。新しく実装した学習者の状況を把握する画面を図 1 に示す。新しい学習者の状況把握画面では、学習者ごとに、指定された時間内の実行数とエラー数、前回実行してから今までの経過時間を表示させる。対象とする時間は、最近 30 分、60 分、90 分からはボタン一つで選択できるほか、任意の期間を選択することもできる。また、一番最近実行したファイル名も提示することで、他の学習者に比べて作業が遅れている学習者も見つけることができる。苦勞している学習者の情報を分かりやすく提供するため、エラーの発生率が高い学習者や前回の実行からの経過時間が長い学習者は表に色をつけて表示させている。表示される情報は、エラーの発生率、実行からの経過時間でソートすることもできる。図 1 で示した例では、ここ最近の実行でエラーの割合が多い順に並べ替えられており、苦勞している学習者を把握することができる。また、これまでは実行されたプログラムなどを確認できていたため、時間内の実行結果履歴をエラーが発生していたら E、エラーが出なければ R と表示させ、それをクリックすることで詳細を確認できるようにした。

4.3 課題管理

課題管理機能では、授業中に学習者に課す課題の配布支援を行う。またこの機能は、Bit Arrow の演習ページ内で使えるため、課題を設定するために他の教育支援システムなどのページに移動したり配布用ファイルをサーバ上において学習者にダウンロードさせたりする必要もなく、簡単にファイルを配布することができる。

課題を管理するためには、課題を設定し、ファイルを配布し、そして学習者側から提出する必要がある。現時点ではこのうちファイルの配布を実装した。教員として演習画面に入ったときのみ、ファイルを配布するためのメニューが表示され、今開いているファイル名と同じ名前のファイルがクラスに登録された学生全員に配布される*1。なお、同一のファイル名が既に存在している場合は、上書きされない仕組みとなっている。配布時には、現在開いているファイルの中身が表示されるが、配布前にその内容を編集することが可能で、図 2 のように課題の対象となる場所にコメント文を用いて指示を出すことができる。ファイルの配布を行えるようにしたことで、これまでは全て学習者に作らせる形でしか演習ができなかったが、プログラムの一部を配布してその授業で学習した項目を使うことに集中させたり、サンプルプログラムを配布して動作を確かめさせたりすることもできる。

提出機能については、ファイルを保存したときに学習者のワークスペースがクラウドとファイルが同期されるため、クラウドのファイルを見ることで代用している。

4.4 採点支援

採点支援の機能では、学習者が提出した課題の採点について支援を行い、教員の負担を軽減することを目的としている。またこの支援システムを活用することで、学習者への迅速なフィードバックを行えるようにし、プログラミング学習において円滑なコミュニケーションが可能になることも期待される。

この採点支援システムは現段階では C 言語のプログラミング学習を想定している。プログラムを採点するには、学習者のプログラムに教員が予め設定した入力を与えて実行し、教員が出力とソースコードを見て採点を行い、結果を学習者へと通知する、という一連の作業が必要となる。これを一人一人に対し行っていくと、教員への負担が非常に大きくなってしまふ。

今回実装した採点支援システムでは、4.3 で述べた課題管理の機能で学習者に配布した課題を、学習者が編集したあとでまとめて実行し、図 3 のような採点フォームが生成される仕掛けを用意した。この採点フォームは 1 つの課題につき 1 つの Web ページとして学習者の人数分生成され、一覧表示できる。1 つの採点フォームには学習者の名前、学習者が提出したソースプログラム、教員側で設定した入力、その出力結果、採点結果を入力するためのコメント欄が設けられている。採点結果のコメントは、そのまま学習者にフィードバックされるため、間違っている場所やヒントなどを個別に指摘することも可能である。

*1 ファイルは「プロジェクト」と呼ばれるフォルダ内にあり、もとのファイルを含んでいたフォルダと同名のフォルダに配布される。各学生が該当フォルダを作っていない場合は自動作成される。

ユーザID	エラー/実行	実行からの経過時間	今実行しているファイル	実行結果履歴
stu05	8/8(100%)	00:21:24	0512/P0512_1.c	EEEEEEEE
stu12	3/3(100%)	00:24:13	0512/P0512_4.c	EEE
stu03	21/24(87%)	00:00:18	0512/P0512_1.c	RRREEEEEEEEEEEEEEEEEEEE
stu09	14/17(82%)	00:02:05	0512/P0512_5.c	REEEEEEEEEEEEEERR
stu18	18/28(64%)	00:00:43	0512/P0512_1.c	RRRRREEREEEEEEEEEEEE
stu01	11/17(64%)	00:00:19	0512/P0512_2.c	EEEEERE
stu02	12/19(63%)	00:00:15	0512/P0512_2.c	EEEERRRRRRRR
stu10	6/10(60%)	00:00:15	0512/P0512_3.c	RRREERREER
stu06	11/17(64%)	00:00:19	0512/P0512_2.c	REEEEEEEEEERRRRRRRRRREEEEEERRR
stu04	18/31(58%)	00:00:15	0512/P0512_3.c	EREERREERREEEEEERRRRRRRREEEEE

手が止まっている
学習者を把握

エラーで困っている
学習者を把握

ここ最近の実行履歴
Rは実行、Eはエラー

図 1 学生の実行状況把握画面

```

C Kadai1別ページで表示
1 // C
2 #include<stdio.h>
3 int main(){
4     int i;
5     for(i=0;i<10;i++){
6         printf("hello world\n");
7     }
8 }
    
```

教師が作った課題の正解プログラム

一斉配布

プログラム

```

// C
#include<stdio.h>
int main(){
    int i;
    // 10回「hello world」と繰り返し表示せよ
    // ここまで
}
    
```

配布用に編集したプログラム

図 2 課題配布ダイアログ

meisei17pro1/hori/0414/P0414_2

プログラム

```

#include<stdio.h>
int main(void) {
    int a, b;
    printf("input a: ");
    scanf("%d", &a);
    printf("input b: ");
    scanf("%d", &b);
    printf("%d + %d = %d", a, b, a+b);
}
    
```

実行結果

入力1

15

5

出力1

input a : input b : 15 + 5 = 20

送信 正解

図 3 採点フォーム

採点システムの構造を図 4 に示す。現在の開発段階では、採点フォームを出力するコマンドが実装されている。このコマンドは指定された課題の名称（プロジェクト名およびファイル名）に対して、各学生の該当するファイルについて順番に gcc コンパイラに読み込ませ、プログラムと出力結果、採点のためのコメント欄を一覧表示した Web ページを作成する。この一覧表示においては、更新日時が新しい順でソートする機能や、採点の状態に応じてプログラムの背景色が変わる機能、教員が評価コメントを入力し各学習者へ送信するための機能などが実装されている。また、このシステムはグラフィックス表示にも対応しており、グラフィックスを利用する場合は出力にサムネイル (図 5) が表示され、アニメーションされるプログラムではこれをクリックすることで実際の動作を確認できるようになっている。

グラフィックス命令の詳細については、本シンポジウムの別の発表 [11] にて報告する予定である。現在の実装では、本来 Bit Arrow 上で実行すべき内容を gcc で実行しているため、グラフィックス命令のライブラリは gcc 上で処理可能な形で新たに作成している。このライブラリでは実際にグラフィックスを表示しているのではなく、グラフィックスを表示するための JavaScript の命令を生成し、採点フォーム内で実行できるようにしている。

5. 利用した教員による評価と考察

4 で述べた実装を行った Bit Arrow を、大学の情報科学系学部 2 年生を対象に開講されたプログラミングの授業で用いた。この授業は、Bit Arrow の C 言語に用意されたグラフィックス命令を利用しながらプログラミング演習を行っている。授業実践の詳細については本シンポジウムの別の発表 [11] で報告する。この授業を担当した教員に対し、利用した Bit Arrow のそれぞれの機能について使用した感想を聞いた。そこで得られた各機能ごとの回答についてまとめる。

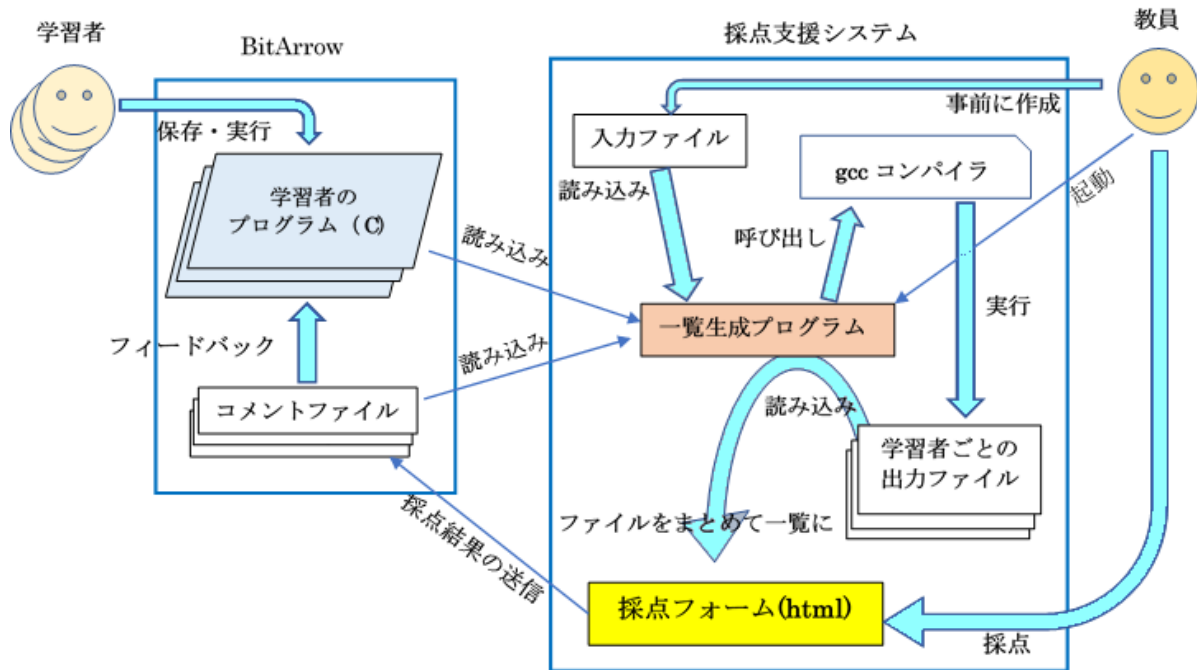


図 4 採点システムの構造

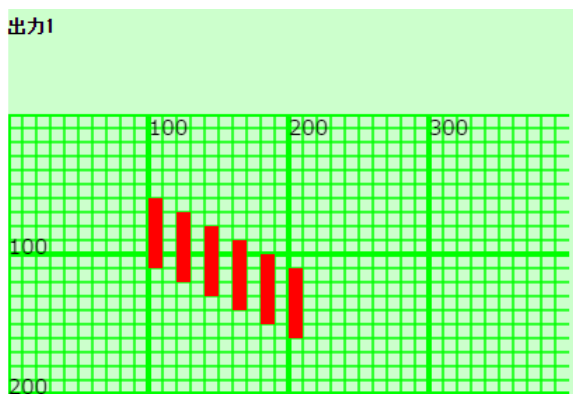


図 5 グラフィックスのサムネイル

5.1 履修者登録とクラス管理

初回授業の前には教員がクラスを作成したのち、受講者45名のユーザIDとパスワードを一括登録した。また、履修が遅れた受講者は後から追加で登録が行われた。使用した教員からは次のような意見が得られた。

- 履修者の一括登録は簡単に扱うことができた
- 後から履修者を追加したときに二重登録が防がれているか分からずに不安だった
- 履修者の管理なので、課題の進捗管理も行いたい

履修登録については、実装上二重登録は防いでいるが、登録完了を伝える画面で教員に与える情報が十分でなかった。すでに登録されているユーザと新たに登録したユーザの情報を示すことで分かりやすい表示をさせる必要がある。履修者の管理では、名前、ユーザID、パスワードの表示のみを行っていたが、成績をつけるときにはそれと同時に課題の進捗状況も提供することで管理が楽になると考え

られる。現在は課題の進捗状況をまとめて管理していないため、ユーザごとにこれまでの課題の達成状況を管理する必要がある。

5.2 学習者の状況把握

学習者の状況把握では、指定した時間の中での学習者のエラー率や前回実行からの経過時間を提示した。この機能についての教員からの評価は次のとおりである。

- エラーで困っている、作業が止まっている学習者の把握はできなかった。手動で見るのではなく、作業が止まっている学習者を通知してほしい
- 取り組んでいる課題がファイル名で見られるため、クラス全体の進捗は把握できた
- 今作業しているファイルの情報だけでなく、課題の完了度合いも一覧したい

学習者の状況把握は、任意の時間をしていてその期間内の学習者の学習状況を表示するが、ページの自動更新はされないため、確認するときはその都度教員側でボタンを押したりページの更新をかけたりする必要があった。過去の授業時の学習状況を確認しているときを除き、一定の間隔ごとに学習状況を更新して、そのときに手が止まっていたりエラーが多く出ている学習者を教員側に通知することで常に最新の状況を提供することができる。クラス全体の進捗では、課題の採点結果を確認できるようにすることで、クラス全体が今日の課題を行っているか以外にもどの課題で多くの学生が間違えているかのような情報を提供できる。

5.3 課題管理とファイルの配布

授業では毎回4つ前後の課題が課されており、プログラムの一部を実装する課題が多い。こうした課題のファイルは各授業が始まる前に学習者が実装する部分を取り除いた状態で一括配布されている。学習者は、演習中のファイルは各自で自由に作成し、課題に取り組むときは配布されたファイルを編集した。課題管理と配布機能の教員による評価を次に示す。

- 課題配布機能は便利に使えたが、正解と配布用のプログラムを別に管理するのが面倒であった
- プログラムの一部を実装する課題で、変更しなくて良い部分を書き換える学習者が多くいたため、書き換えてはいけない箇所を指定したい

課題の配布機能は利用頻度が高く、中でもこの授業では穴埋め問題の方式で課題を出題していたが、変更を指示した場所以外を編集されることが多かった。このため、課題プログラムのうち編集可能な部分を教員側で設定し、それ以外の部分は編集不可能にして配布する工夫が必要である。

5.4 採点支援

提出機能を保存時のクラウドとの同期で代用していることから、採点時には学習者の最新のファイルを読み込み、教員に提供している。採点時には全学習者の該当ファイルが更新日時で並べ替えられ、採点済みか、採点後に中身が変更されたかといった情報を元に、未採点の課題を教員が採点する。なお、この授業では課題に提出期限は設けられておらず、授業の期間が終わるまでに提出することが求められている。また、授業の前半を新しい概念や今日の課題の説明、後半を課題に取り組む時間としている。採点支援の評価を次に示す。

- 一覧で見られることが便利で、採点にかかる時間を短縮できた
- 未着手か返答済みかどうかの区別が付きにくい
- 採点フォームは、エラーメッセージや不要な printf 出力が多過ぎると見にくかったため、出力部分を折りたためると良い

各課題ごとにプログラムと出力結果が一覧で見られるが、一部課題の状態が分かりにくいことや、出力によって採点フォームが見にくくなる問題点が挙げられたため、見やすくするための工夫を行う必要がある。

現在は採点時に常に最新のファイルを見ることから、完成前に試行錯誤している段階のプログラムも教員側で閲覧される。これにより、採点中に教員が頻発しているエラーや考え方の間違いなどに気づき、口頭でクラス全体に注意喚起をしたり、評価コメントで間違っている箇所を指摘するといった指導につながっている。

現在の採点は4.4で述べた通り、採点フォームを出力するコマンドが実装されている。それを教員が起動すること

で、その時点での各学習者のファイルの中身を取得してHTMLページが出力され、そのページ上で採点を行う仕組みとなっている。このコマンドの起動は教員が、学習者がプログラムを書いた頃合いを見計らって行っている。すると、採点結果を受け取る時間が学習者によってまちまちで、タイミングが良ければすぐに採点される人もいれば、長い時間待たされる人もいる。また、コマンドを起動した後でプログラムを書き換えた学習者は、以前のプログラムに対する評価コメントが送られてくるなど、フィードバックが役立つケースもある。この点は、課題を提出することができるようになれば改善できると考えられる。

6. 今後の拡張について

現在の実装は、4で述べた通りであり、3で挙げたすべての機能が網羅されていない。また、5でも実際に支援機能を利用した教員からいくつかの指摘を受けた。それぞれの支援項目について今後の拡張予定をまとめる。

6.1 クラスと履修者管理

クラスと履修者の管理では、教員がクラスを作成することと、そのクラスの履修者を登録することは実装した。しかし現在は、教員が一度登録した学習者の情報を変更できない。例えばパスワードを変更させたいことや、誤って登録した情報を変更・削除したいこともあるため、学習者の情報を教員が変更できる機能の実装を行う予定である。また、学習者の人数が多い授業や、大学の講義などではアシスタントが付くことがある。現在は、ユーザを教員と学習者に分けているが、教員ができることのうち一部の権限を持ったアシスタントのユーザも必要になると考えられる。また、このアシスタントに与える権限は授業やそのアシスタントによっても異なると考えられるため、権限の設定も行える必要がある。教員の評価からは、履修者の登録時に表示される登録情報を表示することや、ユーザ一覧ではIDとパスワード、名前に加えて課題の進捗状況も閲覧できるようにする。

6.2 学習者の状況把握

学習者の状況把握では、指定した時間中の学習者の実行でエラーが発生した割合や前回の実行からの経過時間を提供している。しかし現在の実装では、各実行の間隔が分からず、教員からも通知がこないことは不便で、課題の進捗管理もしたいと指摘された。指定された時間内でも、実行の間隔がどれだけ空いていたかという情報も提供し、自動更新でエラー率が高い学生や手が止まっている学生を通知することが必要がある。

6.3 課題の管理

課題の管理では、ファイルの配布を実装した。現在この機能を用いている授業では課題の提出期限が設けられていないため、課題の設定や学習者による提出を実装してなくても運用ができています。しかし、授業によっては提出期限が厳しく定められていることもあるほか、今の実装では採点作業で未完成のファイルを見なければなりません。こうした点からも、課題を設定し、学習者はそのファイルを提出できるような実装が必要であると考えられる。また、提出期限や、後述する採点機能で用いるテストケースと望まれる実行結果などを同時に設定することが考えられる。

6.4 採点支援

採点機能では、学習者の最新のファイルを元に、実行結果を提示し教員が採点を行うインタフェースを実装した。また、評価コメントを学習者にフィードバックすることもできる。一方で、現在の採点インタフェースでは、教員がコマンドを起動したタイミングで採点フォームが作成されるため、これ以降で学習者がプログラムを更新しても採点フォームには反映されない。そのため作成中のプログラムに対するフィードバックが送られてしまうことがあった。そこで、学習者のプログラムの更新がリアルタイムに採点フォームへと反映されるようにシステムを改善する必要がある。

また、現在はプログラムを実行する際に gcc コンパイラを利用している。しかし、Bit Arrow の C は、gcc コンパイラを利用せずに実行している [11]。また、Bit Arrow では C だけでなく JavaScript とドリトルも使用できる。採点時に gcc コンパイラを利用すると、他言語へのシステムの流用が不可能になってしまうため、gcc コンパイラを使わないで採点支援を行えるシステムに修正する必要がある。その他にも自動で判定が可能な課題では出力結果を元に自動採点する機能、採点結果をデータベース上に保管し成績管理を行える機能の実装を行う。

5 の実践では、C でもグラフィックスを用いた授業が行われており、ドリトル、JavaScript でもグラフィックスを利用した作品制作を行うことも考えられる。そのため、クラス内で作品を公開できる機能も今後実装する。

7. まとめ

本稿では、オンラインプログラミング学習支援環境 Bit Arrow の教員支援機能の現在の実装について述べた。本稿執筆時の実装の進捗として、教員権限によるクラスの作成と履修者登録、学習者の状況把握、ファイルの配布、採点支援機能の実装について報告した。

これらの機能を実装した Bit Arrow を用いた大学の情報科学系学部の 2 年生向けの授業では、ファイルの配布や採点支援機能を毎週活用し、採点時間の短縮などに役立って

いることが分かった。一方で、採点時に学習者の最新ファイルと教員が採点しているファイルの内容がリアルタイムに反映されていないという問題も見つかった。

今後は教員を支援する機能の拡張として、クラスへのアシスタント役の追加や課題の提出、教員から学習者へリアルタイムにフィードバックできる機能などを実装する。

謝辞 本研究は JSPS 科研費 17K00989 の助成を受けたものです。

参考文献

- [1] 長島 和平, 長 慎也, 間辺 広樹, 兼宗 進, 並木美太郎: Web ブラウザを用いたプログラミング学習支援環境 Bit Arrow の設計と評価, 情報処理学会研究報告コンピュータと教育 (CE), Vol. 2017-CE-138, No. 2, pp. 1-8 (2017).
- [2] 長谷川 伸, 松田 承一, 高野 辰之, 宮川 治: プログラミング入門教育を対象としたリアルタイム授業支援システム, 情報処理学会論文誌, Vol. 52, No. 12, pp. 3135-3149 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110008719872/>) (2011).
- [3] 内藤 広志, 齊藤 隆: プログラミング演習のための進捗モニタリングシステム, 情報処理学会研究報告コンピュータと教育 (CE), Vol. 2008, No. 13, pp. 33-40 (2008).
- [4] 太田 翔也, 富永 浩之: プログラミング演習における補助者の巡回指導のためのタブレット PC 上の支援ツール-小コンテスト形式の初級 C 演習での実践におけるツールの操作ログの分析-, 情報処理学会研究報告コンピュータと教育 (CE), Vol. 2016-CE-137, No. 1, pp. 1-8 (2016).
- [5] 田中 良樹, 松澤 芳昭, 木谷 友哉, 酒井 三四郎: Hanabi: プログラミング教育改善のための横断的フィルタ機能を有するダッシュボード, 情報処理学会研究報告コンピュータと教育 (CE), Vol. 2017-CE-139, No. 23, pp. 1-8 (2017).
- [6] Spacco, J., Hovemeyer, D. and Pugh, W.: An Eclipse-based Course Project Snapshot and Submission System, *Proceedings of the 2004 OOPSLA Workshop on Eclipse Technology eXchange*, eclipse '04, New York, NY, USA, ACM, pp. 52-56 (online), DOI: 10.1145/1066129.1066140 (2004).
- [7] Papancea, A., Spacco, J. and Hovemeyer, D.: An Open Platform for Managing Short Programming Exercises, *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, ICER '13, New York, NY, USA, ACM, pp. 47-52 (online), DOI: 10.1145/2493394.2493401 (2013).
- [8] 加藤 利康, 石川 孝: 授業支援システムにおけるプログラミング演習のための学習状況把握支援機能の設計と評価, 情報処理学会研究報告コンピュータと教育 (CE), Vol. 2012-CE-113, No. 6, pp. 1-8 (2012).
- [9] Elamir, A. M., Jailani, N. and Bakar, M. A.: Framework and Architecture for Programming Education Environment as a Cloud Computing Service, *Procedia Technology*, Vol. 11, pp. 1299-1308 (2013).
- [10] Efopoulos, V., Dagdilelis, V., Evangelidis, G. and Satratzemi, M.: WIPE: A Programming Environment for Novices, *SIGCSE Bull.*, Vol. 37, No. 3, pp. 113-117 (online), DOI: 10.1145/1151954.1067479 (2005).
- [11] 長 慎也, 長島 和平, 堀越 将之, 間辺 広樹, 兼宗 進, 並木美太郎: オンラインプログラミング環境 Bit Arrow を用いた C 言語プログラミングの授業実践, 情報教育シンポジウム 2017 論文集, Vol. 2017 (2017, 予定).