教育情報学研究　第18号（2019）
Educational Informatics Research, 2019, No.18, 11-23

ベイジアンネットワークを用いたプログラミング学習支援システムの提案

住田　智雄 *，小嶋　秀樹 *

東北大学大学院教育学研究科

要旨：2020年より小学校から始まるプログラミング教育の必修化に向けて，多くのプログラミング学習支援システムが提案されている．現在提案されているプログラミング学習支援システムは主に，学習者に対して問題演習を通した学習を支援するものか，あるいは学習者のコンピュータの操作状況を画面で監視し，その動きからプログラムの理解度を把握しようとするものが多い．しかし，学習者が記述したプログラムを直接解析し，エラーの原因や理解不十分な箇所の推測・支援を行うものはない．そこで本稿では，学習者が記述したプログラムを解析することで学習者個人のエラーの原因や理解度，理解不十分な箇所を推論し，その後の学習に役立つような情報を学習者に提示するシステムを提案する．対象とする学習者は，プログラミング学習の入門者から文系大学生などプログラムを専門としない学生，およびこれらの学生に教授するプログラムを専門としない教員である．

キーワード：プログラミング学習支援システム，ベイジアンネットワーク，Processing，Web IDE

1　背景

情報化社会において，コンピュータやインターネットを利用したサービスは必要不可欠なものとなっている．それに伴って，コンピュータをより適切に，かつ効果的に活用していくことを目的として，図1に示すように，2020年から小学校を始めとしてプログラミング教育が順次必修化される [1]．また，プログラミング能力の需要も年々高まってきている．このような時代の要請に応えるために，大学では専門家でなくともプログラミング言語の学習が行われている．しかし，慣れないプログラミング環境による学習や複雑な文法，構文の理解不足，プログラミング教育が可能な指導者の確保などさまざまな問題がある．これらの問題は，今後，小学校，中学校，高等学校で始まるプログラミング教育でも起こると予想され，その解決は急務である．

このような背景のもと，入門者や初心者を対象として図2に示すような Scratch [2] [3] を代表とするブロックを組み合わせていくビジュアルプログラミング言語によるプログラミング学習が行われている．しかし，C 言語や Java，Python といった実際に利用されているプログラミング言語の学習への円滑な移行は考慮されていない [4].

また，現在提案されているプログラミング学習支援システムはその多くが学習者に対して問題演習を通した学習を支援するものである．ほかに学習者によるコンピュータの操作を画面で監視し，その動きからプログラムの理解度を把握しようとするものがある．しかし，学習者が記述したプログラムを解析することで学習者個人のエラーの原因や理解度，理解不十分な箇所を推論し，その後の学習に役立つような情報を提示するなどの支援を行うものは少ない．

また，入門者や初心者にとってはプログラミング開発環境の構築が最初の大きな負担である．そのため，自宅のコンピュータに開発環境を構築することが難しく，プログラミング学習を諦めてしまうことが多い．さらに，Scratch のようなビジュアルプログラミング言語を使用した学習では，上述したように実際の開発で使用されるプログラミング言語の文法や構文などの必要な概念を学ぶこ

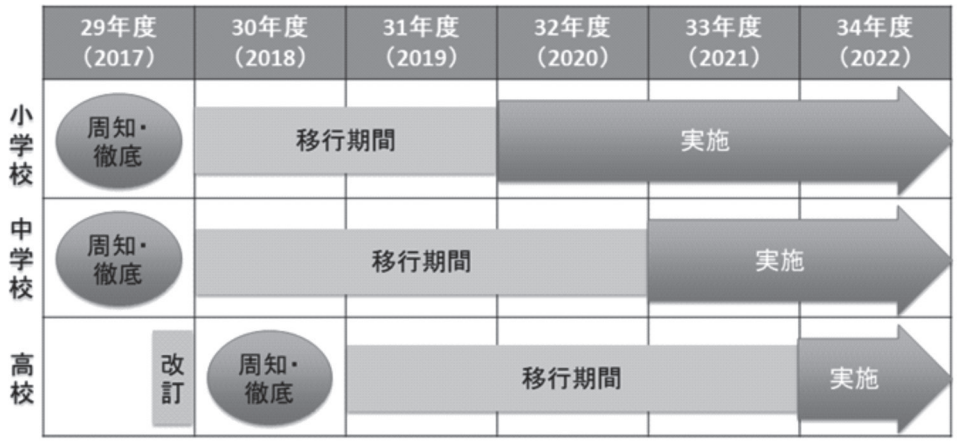


図1　学習指導要領改訂に関するスケジュール

とが難しい．そのため，Scratch ではプログラムを作成することができても，実際のプログラミング言語では作成することが難しいといった問題が生じている．また，テキスト形式のプログラミング学習では，変数名や関数名など普段使いなれていない文字列の入力を難しいと感じている入門者が多い．

以上のように，入門者や初心者を対象としたプログラミング学習支援システムには，容易な開発環境の構築，実際に使用されているプログラミング言語を用いた学習，プログラミング学習を諦めることなく継続的な学習が可能となるプログラミング学習を支援する機能が必要である．そこで本稿では，学習者のプログラミング開発環境の構築の負担軽減と学習開始までの障壁を除去することを目的として，ブラウザのみで利用可能な Web IDE（Integrated Development Environment）を開発する．さらに，プログラミング学習の入門者に多く見られる入力ミスを減らすための機能，ベイジアンネットワークを利用して入門者や初心者に多く見られるエラーとその原因，理解不足などを推論し，その後の学習に役立つ情報を提示するなどの機能を備えた Web IDE を提案する．現在，さまざまなプログラミング言語を学習することができる多くの Web IDE が開発されている [5] [6]．本稿では基本的なプログラミング言語の文法を学ぶことができ，さらに，学習者が動作結果を確認しやすいグラフィックやアニメーション，さらにサウンド機能などを簡潔に記述することができるという特徴を持ったプログラミング言語 Processing を対象言語とする [7] [8].

2　関連技術

本章では，本稿に関連する技術について説明する．

2.1　Scratch

Scratch は，アメリカの MIT メディアラボで開発されたプログラミング環境である [2] [3].

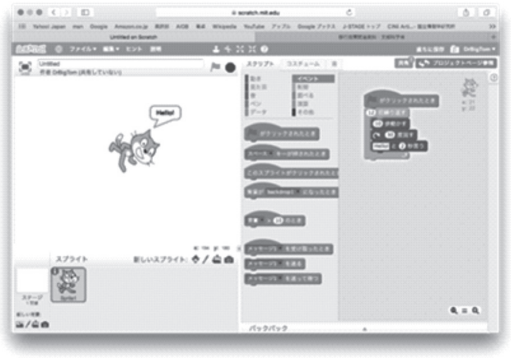


図2　Scratch

図2に示すように，あらかじめ用意されているブロックを組み合わせることで入門者でも容易にプログラムを作ることができる．また，Scratch は Web アプリケーションとして開発されているので，学習者はブラウザから Scratch のサイトにアクセスすることで使用することができる．

2.2　Processing と Processing.js

本稿で対象としたプログラミング言語である Processing は，アメリカの MIT メディアラボの Casey Reas と Benjamin Fry によってオープンソースとして開発されたプログラミング言語であり，グラフィックやアニメーション，サウンド機能などに簡単にアクセスすることが可能である [7] [8]．プログラミング初学者教育だけでなく，メディア作品やそのプロトタイプ制作など，アーティスト，デザイナー，建築家などにも使用されている．

Processing.js は Processing の文法で書かれたプログラムを JavaScript に変換してブラウザ上で実行できるようにした JavaScript ライブラリである [9]．Processing.js を用いることで，学習者にブラウザ上で Processing によるプログラミング学習環境を簡単に提供できるようになる．

2.3　学習者モデル

学習者モデルは，学習者それぞれの理解状況を学習支援システムが把握するための指標であり，学習者ごとに個別に学習者モデルを作成することができる．これにより，学習者ごとの理解状況を正確に把握し，学習者の現在の知識状態に応じた適切な指導を行うことが可能になる [10] [11] [12]．代表的なモデルとして，オーバーレイモデルやバギーモデルなどさまざまな学習者モデルが提案されている [10] [13]．

2.4　ベイジアンネットワーク

ベイジアンネットワークは，複雑な因果関係を確率で表現するグラフィカルモデルの1つである．ノード間の因果関係を非巡回有向グラフ（DAG：Directed Acyclic Graph）で表し，それぞれのノード間の依存関係を条件付き確率で表す [14] [15]．

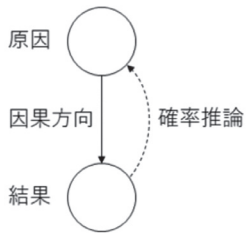


図3　ベイジアンネットワーク

ベイジアンネットワークを利用することで，図3に示すように，結果から原因を確率的に推定する確率推論が可能となる．確率推論は人工知能の分野で重要なツールの1つであり，システムの故障診断，危険を予測するシステム，医療診断のシステムなど，その予測性能の高さからさまざまな分野で応用されている．

教育の分野ではベイジアンネットワークによる学習者モデルが研究されており，学習者の知識状態を同定するための学習者モデルや学習者が問題を解決する過程を同定するためのモデルなどがある [12]．

本研究では，学習者がプログラムを実行した後にエラーが発生した場合，そのエラーのエラーメッセージを「結果」として扱い，そこからエラーの「原因」を推論するシステムを構築する予定である．

2.5　Web IDE

ブラウザのみで利用できる Web IDE は，プログラミング学習の環境構築の負担軽減が可能であるため，入門者や初心者に最適である．そのため，入門者や初心者を対象としたプログラミング学習のための Web IDE が研究されている．

Processing のコードをテキストではなく，視覚的なブロックで記述するという方針を取ったプログラミング教育の研究では，Scratch と同じくブロックの色や形状で視覚的なコーディングを可能としており，可読性を高める工夫が施されている [16]．

また，現実的な環境で多数の学習者の指導を同時に行う必要性と C や Java などの実際に使用されているプログラミング言語のスキルを身につけさせたいという要請から構築された Processing を利用した Web IDE もある [17]．

3　支援環境とプログラミング教育の課題

本章では，文部科学省が行ったプログラミング教育に関するアンケートの結果，および実際に学生に指導してきた中で解決すべき課題について述べる．

3.1 支援環境の課題

文部科学省が行ったプログラミング教育の実施に向けたアンケート調査の結果，適切な教材の不足が課題である（72.4%)、指導方法の情報不足が課題である（75.4%）という結果が出ている [18]. また，予算が限られていることから，無料で使える教材を必要としている．教員のスキルが不足している場合は，機器の設定等の作業負担などの問題も指摘されている．

3.2 プログラミング教育の課題

コンピュータを使用する学習の場合，教員側の指導に関する問題もある．教員のスキルが高い場合でも，学習者の様子を十分に把握し直接指導できる人数は一度に一人と限られる．そのため，そのときに指導していない他の学習者やクラス全体の状況を十分に把握することは難しい．

また，学習者側においては，それぞれの学習者のスキルの差が主な原因となって，学習の進捗状況に差が出てくる．特に，複雑な開発環境を使用する際の操作ミス，慣れないプログラム特有の記述や命令に対する入力ミス，プログラミング言語の文法の理解度，開発環境の表示するエラーメッセージの解読などが原因として見受けられる．

以上のことから，学習者が入力したプログラムを解析し，教員に代わってそれぞれの学習者の理解度や進捗状況などを把握することで，学習者に最適なアドバイスを提示するシステムが必要であると考えられる．

4 学習者によるプログラミング学習の意識

本稿で対象としている学習者（プログラミング経験のあまりない私立文系大学の学生）にアンケートを行った．このアンケートは，プログラムの講義を履修している3 〜 4年生18名に対して，本稿の対象言語であるプログラミング言語 Processing の基本的な内容を終了し，自分で基本的なグラフィックを作成できるようになったあとに行った．

4.1 アンケート項目

アンケート項目は以下のとおりである．

- プログラムを学習したことがありますか
- プログラムの基本的ことを学習してみて，難しかったですか
- プログラムのどういうところがわかりにくかったですか（自由記述）
- （上記項目を確認後追加）エラーメッセージがわかるとやりやすいと思いますか
- アプリを使ってみて，やりやすくなりましたか
- アプリを使ってみてどうでしたか？（自由記述）

最後の2項目については後述する．

4.2 アンケート結果

4.2.1 プログラムを学習したことがありますか

図4に示すように，プログラムを学習したことのある学生は1割以下（18名中1名）だった．また，学習経験のある学生は，自分の学習したプログラミング言語の名前を憶えていなかった．

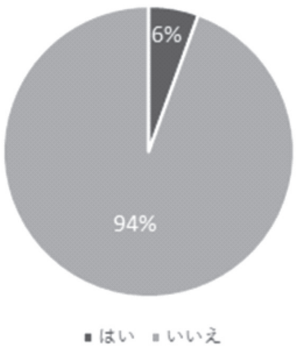


図4 プログラミング学習の有無

このことから，ほぼ全員がプログラムについて未経験であり，ある程度経験を積んだ教員でなければ教えることが非常に困難な状況であることがわかる．

4.2.2 プログラムの基本的なことを学習してみて，難しかったですか

図5に示すように，プログラムの基本的な内容を学習した後の感想では,「とても簡単」と「簡単」が1割以下,「少し難しい」と「難しい」が7割以上であった．このことから，プログラム初学者教育に適していると言われている Processing でも，多くの学生がプログラムに関して難しいと思っていることがわかる．

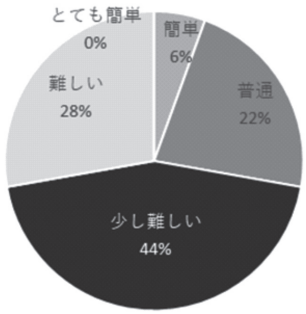


図5　プログラミング学習後の感想

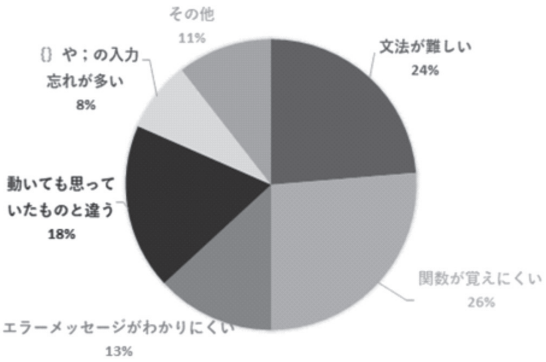


図6　プログラムのわかりにくいところ

学生が難しいと考えているところをできるだけ具体的に把握するために，自由記述で以下の質問を行った．

4.2.3　プログラムのどういうところがわかりにくかったですか（自由記述）

図6に示すように，プログラムの基本的な部分で難しいと考える学生が多いことがわかる．

「文法が難しい」という回答の中では，「変数の扱いが難しい」「if 文や for 文が難しい」といった記述が多く，特に，if 文や for 文についてはそれぞれ括弧の中に何を記述するのか，また条件式等の考え方がわかりにくいといった記述が多かった．「関数が覚えにくい」は，普段使いなれない名前（例：楕円を表す ellipse など）や括弧の中の引数の意味と引数の数についての理解に苦労している学生が多い．関数によって括弧の中の引数の数が異なり，また，同じ関数名でも引数の数が異なることが原因であると考えられる．「動いても思っていたものと違う」という意見に関しては，Processing の文法としては正しくても，図形を表現する関数を記述する順番によって結果が異なることがある．また，座標の原点を移動させる関数や図形を回転させる関数を使用した場合に，通常の座標軸が変換され新しい座標軸を構成するため座標の計算を間違えてしまい，学生が思っているような表現ができていないことが多い．「「｛｝」や「；」の入力忘れが多い」については，「｛」と「｝」，「（」と「）」の数の一致や各関数の呼び出しは常に「；」で終わるといった文法が混乱のもととなっている．また，このアンケートでは記述がなかったが，「｛」の次の行のインデントを行わない学生も見受けられ，可読性が悪くなることがエラーの原因になると考えられる．

学生が記述したエラーを含むプログラムの例を図7に示す．このプログラムは，星マークを複数個並べて表示するという自主的な課題に挑戦した学生が，完成までの過程で実際に入力したプログラムである．

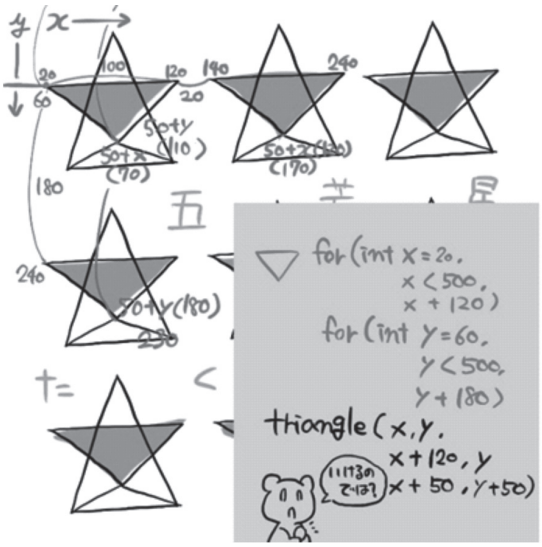
```
size(500,500);

for(int x=20 ; x<500; x +=120){
  for(int y =60 ; y <500; y +=180){
    triangle(x, y, x+100, y, x+50, y+50); //
  }}
```

図7　学生のプログラム例1

次に，このプログラムを作成した学生の思考について提出してもらった資料から一部引用する．『最初に逆三角形（桃）の数値の計算である。あくまで憶測なので想像どおりに出来るかはわからないが、なるべく単純なプログラミングにするため for ループと int を使った。早速入力である。

しかし，エラーが発生。画像そのものができなかった。変数はちゃんと入力しているのに「変数“x” は存在しません。」と Processing は私に訴えかけるのだ。私は荒れ狂ったが、よくよく確認すると for 文のお尻に；（セミコロン）がいつもの癖でついていたのだ。私はプログラム初心者であるため、詳しいことはわからないが、この；（セミコロン）はプログラムにとっての句点のようなものなのだろうか。』



ここで，エラーメッセージは「変数 “x” は存在しません。」と表示されていて，学生は何度も変数について確認している．しかし実際には，変数とは関係のない場所に不必要な「；」を記述したことが原因である．学生はエラーメッセージで指摘された場所だけを調べるが，原因がわからず，そのあとはすべての行を1行ずつ確認していく．この時点で，あきらめてしまう学生も多い．そのため，「エラーメッセージがわかりにくい」といった記述が多くなる．

4.2.4 エラーメッセージがわかるとやりやすいと思いますか

図8に示すように，エラーメッセージをわかりやすくした場合に，「とてもやりやすい」と「やりやすい」で全体の6割を超えた．上述した学生同様，エラーメッセージで苦労している学生が多いことがわかる．

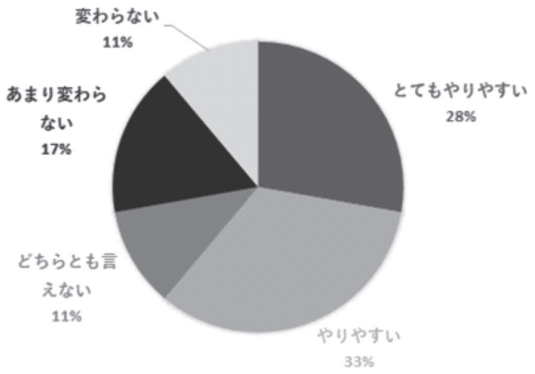


図8　エラーメッセージについて

また，「あまり変わらない」「変わらない」を選択した学生は，エラーメッセージがわかりにくいことが原因でプログラミング学習をあきらめてしまう傾向にある．入門者や初心者を対象としたプログラミング学習支援システムには，エラーメッセージに対する支援を重点的に行う必要があることがわかる．

以上のアンケートの結果から，入門者や初心者を対象としたプログラミング学習支援システムには，通常の開発環境には備わっていない機能を組み込む必要であると考えられる．

5　提案手法と基本的機能の実装

本章では，提案手法の概要について述べる．

5.1　提案手法

本稿では，プログラミング学習の入門者から文系大学生などプログラミングを専門としない学生や初心者，およびこれらの学生に教授する教員を対象として，ベイジアンネットワークを用いたプログラミング学習支援システムを提案する．

本研究では，結果から原因を確率的に推定する確率推論が可能となるベイジアンネットワークを利用する．ベイジアンネットワークを用いることで，図3に示すように，学生がプログラムを実行後にエラーが発生した場合，まずエラーメッセージを収集して分析することで，そのエラーの原因を推論することができると考えられる．また，プ

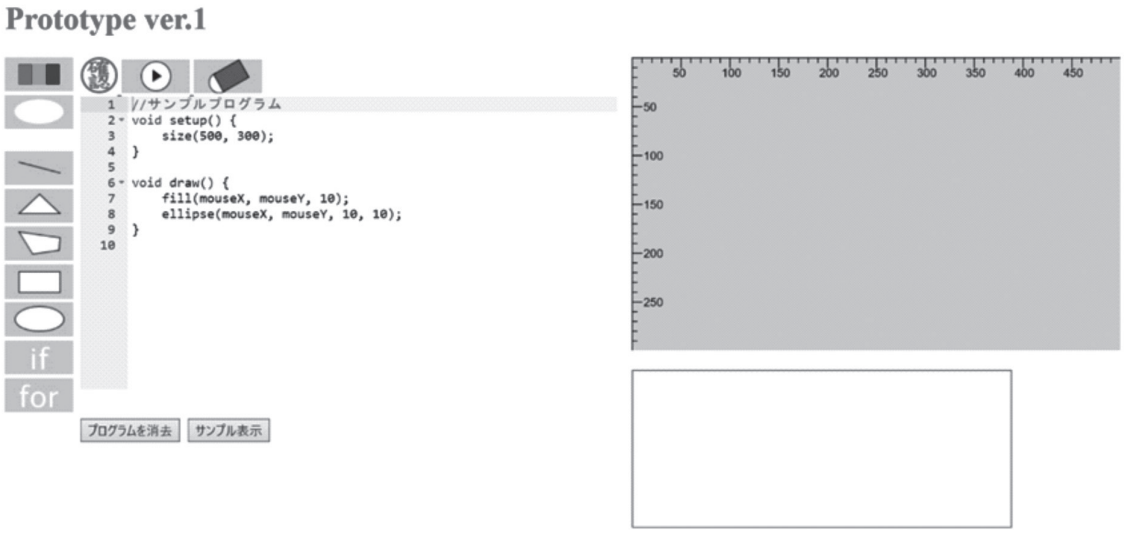


図9　Web IDE の初期画面

ログラムが動作しても思うような結果が出ていない場合の原因などの特定にもつながると考えられる．このように，これまでのプログラミング教育で利用されてきたプログラミング開発環境およびプログラミング学習支援システムでは行われていなかった学習者への支援を行うことが可能となる．

5.2　基本的機能の実装

図9に，本稿で行った基本的な機能を実装した Web IDE の初期画面を示す．提案するシステムを Web IDE として開発することで複雑な環境設定などが不必要となり，インターネットに接続されている環境であればブラウザのみで利用することができる．対象言語は Processing である．

画面左には，Processing の基本的な図形，および制御文（条件分岐の if 文と繰り返し処理の for 文）を表すボタンを配置した．図6に示したように，プログラミング学習の入門者や初心者にとって

- 関数名がわかりにくい
- 関数名の入力が困難
- 関数の引数の数，および引数の意味がわかりにくい

といった意見に対応するために，描画したい図形を表す模様をボタンに表示し，カーソルをボタン上に合わせることで，関数の引数の数および意味を表示するようにした．

図10に基本的な図形を選択した場合の画面を示す．描画したい図形を表すボタンを選択することで画面上でその図形に対応する関数名，引数の意味，引数の数を視覚的に確認することができる．ボタンをクリックすると，右側のエディタ部分に描画したい関数名などが入力される．

図11に，四角形と for 文を選択した場合に入力される書式を示す．

エディタ部分は，通常の開発環境と同じように利用することができる．また図12に示すように，初期画面にはサンプルプログラムとして基本的な図形を描画するアニメーションプログラムが入力されている．学習者は，サンプルプログラムを利用してプログラムを学習することができる．また，サンプルプログラムのパターンを増やすことで，より学習しやすい環境を構築することもできる．

画面右側には，学習者が入力したプログラムの実行結果が表示される．グラフィックの場合，中学数学で扱う座標系と異なり，y 軸の向きが逆になる．入門者や初心者にとっては初めての考え方であることが多く，そのため混乱することが多い．本研究では，グラフィックの座標を表示するようにした．

図10　基本的な図形(四角形)を選択した画面

```
// 四角形の書式
quad( x1 , y1 , x2 , y2 , x3 , y3 , x4 , y4 );

// for文の書式
for (   ;   ;   ) { // 左から初期化；条件式；後処理

}
```

図11　ボタン選択で入力される四角形とfor文の書式

図12　サンプルプログラムと実行結果例

6　評価

前章で作成したアプリの試作版を学生に使用してもらい，以下のアンケートを行った．

6.1　アプリを使ってみて，やりやすくなりましたか

図13にアプリ使用後の評価について示す．

「とてもやりやすい」と「やりやすい」が6割以上（18名中11名）を占めた．このことから一定の評価を得ることができたと考えられる．

「あまり変わらない」と「変わらない」と答えた学生の中には，従来のエラーメッセージがわかりにくいと考えている学生が含まれている．後述するように，今回利用している Processing.js のコンパイラは，文法や構文などの厳密なチェックを行わない．また，プログラムの中にエラーが存在しても，エラーメッセージを表示する場合と表示しない場合がある．そのため，アンケートの結果から，この課題を改善することでさらに有用なツールになると考えられる．

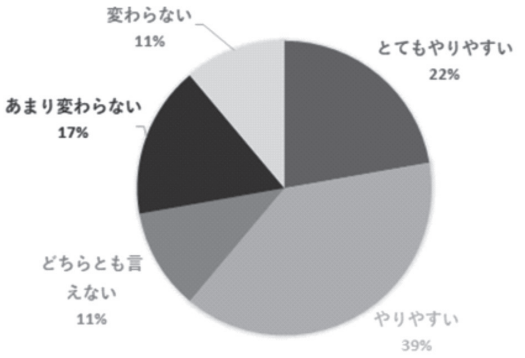


図13　使用後の評価

6.2　アプリを使ってみてどうでしたか？（自由記述）

図14にアプリ使用後の感想の結果を示す．「図形の意味と入力がわかりやすい」や「引数がわかりやすい」など，図6で示した学生が考えているプログラムのわかりにくいところが改善されていることがわかる．ただし，今回の実装では基本的な図形のみのため，同じ関数でも異なる引数の数を必要とするものや複雑な図形を描画する関数などの実装は行っていない．また「「 ｛｛ 」の入力がわかりやすい」は，「 ｛ 」と「｝ 」の組み合わせの数が一致していないことが原因のエラーで苦労する学生が多い．ここでは，図11の for 文の例で示したように for 文を入力するボタンをクリックすることで for 文の基本的な部分はすべて入力するようにしたため，「 ｛｛ 」に関するエラーが大幅に減少した結果だと考えられる．特に for 文や if 文のネスト（入れ子）にも対応したことが学生には評価が高かった．

次に，学生からの改善点の要望について述べる．「変数関係を工夫」については，変数の考え方や使い方が難しいと考えている学生が多く，特に関

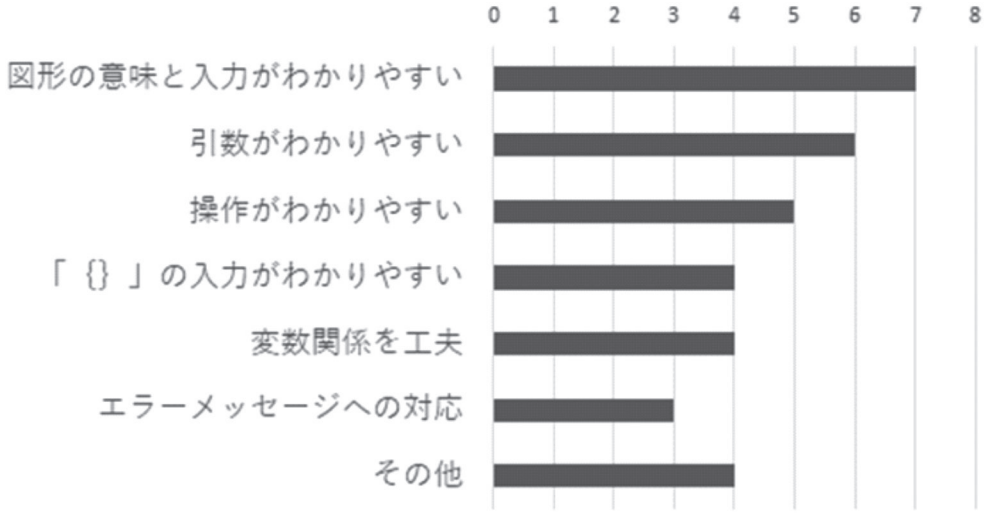


図14　アプリ使用後の感想

数を利用するプログラムの場合，変数のスコープの理解不足によるエラーも多い．また if 文や for 文と関連して、「動いても思っていたものと違う」という学生の感想から，関係演算子を利用した変数の扱いにも工夫が必要である．図15に for 文を使った変数の勘違いの例として学生のプログラムを示す．

図15に示したこの学生のプログラム例2は，小さい円を10度ごとに回転させながら360度描画するという課題である．図16に示すように左が学生の実行例，右が正解例である．

```
int x =90;

size(500, 500);
translate(width/2, height/2);

for(int i =0; i <=360; i=i+10) {
  ellipse(100,0, 10,10);
  rotate(PI/36);
}
```

図15　学生のプログラム例2

学生は図16(右)のように正しく動作すると思っていても，実際には図16 (左) のように動作してしまう．このように学生のエラーに対応するだけでなく学生が思ったとおりの結果が出ない場合のサポートや，図15のプログラム例2で示したように，未使用の変数の存在 (一行目の「int x = 90;」) といったサポートを行う機能も必要と考えられる．

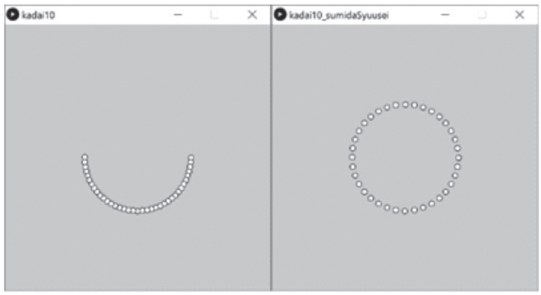


図16　誤った実行例(左)と正解例(右)

7　まとめと今後の課題

本稿では，2020年度より始まるプログラミング学習の必修化に向けて，入門者や初心者を対象としたプログラミング学習支援システムについて提案した．また，提案したシステムの基本的な機能を実装した．今回は，従来の開発環境にはない学習者向けの機能として，Processing でプログラミング学習を行うときに必要となる基本的な関数の概要と説明や制御文の概要と説明を実装した．

次に，本稿で実装したシステムを，プログラムを専門としない私立文系大学の学生18名に実際に使用してもらい, アンケートによる評価を行った．プログラムの入門者や初心者にとって，プログラム初学者教育に適していると言われる Processing でも難しいと考える学習者が多いことがわかった．そこで，実際に学生が難しいと考えている内容を確認し，学生が作成したプログラムとの比較検討を行った．図7に示した学生が作成したプログラムの完成版と実行例を図17に示す．

また，本稿で作成したシステムの使用に対するアンケートの結果から，一定の評価を得ることができた．しかし，以下に示すような改善点も見つかった．

今後の課題として，エラーメッセージへの対策が挙げられる．本稿で使用している Processing.js のコンパイラは，文法や構文の厳密なチェックを行わない．また，プログラムの中にエラーが存在しても，エラーメッセージを表示する場合と表示しない場合がある．たとえば，正しくは size ( ) という名前の関数を間違えて sze と入力してしまった場合，「ReferenceError: 'sze' is not dened」といったエラーメッセージが表示されるが, draw( ) 関数や ellipse ( ) 関数の名前を入力ミスしてもエラーメッセージは表示されず，何も実行しないまま処理が終了してしまう．このような場合，入門者や初心者では何がエラーなのかわからず，プログラム学習自体が先に進まない．その解決策として，学習者が入力したプログラムを実行時にサーバに一度送信したあと Processing のコンパイラでチェックし，その結果を再び学習者のブラウザに表示するといった対策が考えられる．

また，本稿では基本的な図形および制御構文の実装のみで学生によるアンケート評価を行ったが，Processing にはベジェ曲線や図形の回転，座標軸の移動といったグラフィックの描画に必要な

```
size(500, 500);
background(#52475D);

for(int x=20 ; x<500; x +=120){
  for(int y =60 ; y <500; y +=180){
    fill(#F0E56D);
    stroke(#F0E56D);
    strokeWeight(2);
    triangle(x, y, x+100, y, x+50, y+50); //pink
    triangle(x+50, y-35, x+10, y+70, x+90, y+70); //blue

    stroke(#52475D);
    strokeWeight(4);
    fill(#52475D);
    triangle(x+10, y+70, x+90, y+70, x+50, y+50); //orange
  }}
```

図17　学生が作成したプログラムと実行結果(完成版)

関数が他にも多数用意されている．図17に示した学生のプログラムからもわかるように，初心者でもさまざまな関数を使うことがわかる．そのため，基本的な図形だけではなく，Processing に用意されている多くの関数を実装する必要がある．そのためには改めて GUI の検討も行う予定である．

さらに，ベイジアンネットワークを利用した学習者のエラーの原因の推論や理解不足の箇所の特定についても実装する．また，学習者のその後の学習について有益な情報を提示するシステムも実装する予定である．

参考文献

[1] 文部科学省 . 教育課程部会（第110回）配布資料 一 覧：http://www.mext.go.jp/b menu/shingi/chukyo/chukyo3/004/siryo/attach/1418184.htm.

[2] Scratch. https://scratch.mit.edu/.

[3] 鷲崎弘宣 , 齋藤大輔 , 坂本一憲 . 新学習指導要領準拠 Scratch でたのしく学ぶプログラミング的思考 . マイナビ , 2019.

[4] 朝日翔太 . 初等教育におけるテキスト型プログラミング言語 python によるプログラミング教育の効果検証 . 日本教育工学会第34回全国大会講演集 , pp.231-232, 2018.

[5] Max Goldman, Greg Little, and Robert C Miller. Realtime collaborative coding in a web ide. *In Proceedings of the 24th annual ACM symposium on User interface software and technology,* pp. 155-164, 2011.

[6] Vu Nguyen, Hai H Dang, Kha NDo, and Thu D Tran. Learningand practicing object-oriented programming using a collaborative web-based ide. *In Frontiers in Education Conference,* pp. 1-9, 2014.

[7] Processing. https://processing.org/.

[8] DanielShiffman（著）, 尼岡利崇（訳）. 初めての Processing 第2版 .O'REILLY, 2018.

[9] Processing.js. http://processingjs.org/.

[10] 溝口理一郎 . 誤りを科学する－学習者モデルの構築－ . 人工知能学会誌 , Vol.10,No.3, pp.348-353, 1995.

[11] 溝口理一郎 , 角所収 . 知的 cai における学習者モデル . 情報処理 , Vol.29, No.1, pp. 1275-1281, 1995.

[12] 岡本敏雄 , 香山瑞恵（共著）. 人工知能と教育工学 . Ohmsha, 2008.

[13] J.S.Brown and R.B.Burton. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science,* Vol.2, No.2, pp.155-192, 1995.

[14] 田中和之 . ベイジアンネットワークの確率的推論の数理 . コロナ社 , 2015.

[15] 安田宗樹 , 片岡駿 , 田中和之 . 確率的グラフィカルモデル－ベイジアンネットワークとその周辺－ . オペレーションズ・リサーチ ,Vol.58, No.4, pp.191-197, 2013.

[16] 栗原あずさ , 佐々木晃 , 脇田建 , 細部博史 . Processing アプリケーション開発のための視

覚的ドメイン特化言語の実装 . 日本ソフトウェア科学会第31回大会講演論文集（PPL6-2）, 2014.
[17] 三浦元喜 . Processing web ide を用いたプログラミング基礎教育の試み . 情報処理学会情報教育シンポジウム予稿集（SSS2013）, pp.225-231, 2013.
[18] 文部科学省 . 教育委員会等における小学校プログラミング教育に関する取組状況等：http://www.mext.go.jp/a menu/shotou/zyouhou/detail/1406307.htm.

Proposal of programming learning support system with Bayesian network

Tomoo SUMIDA*,　Hideki KOJIMA*

* Graduate School of Education, Tohoku University

ABSTRACT

From 2020, many programming learning support systems have been proposed for compulsory programming education starting from elementary school. Currently, the proposed programming learning support system mainly supports learners through problem exercises or monitors the learner's computer operation status on the screen. There are few systems that directly analyze the program written by the learner and infer and support the cause of the error or the insufficient understanding.

In this paper, we propose a system that analyzes the program written by the learner and infers the cause of the learner's error and lack of understanding. In addition, we propose a system that teaches learners useful information for subsequent learning.The target learners are beginners in programming learning, liberal arts students who do not specialize in programs, and teachers who teach programs to these students.

Key words: programming learning support system, Bayesian network, Processing, Web IDE