

学位論文要旨

プログラミング教育を
支援する教材に関する研究

広島大学大学院教育学研究科
教育学習科学専攻 教科教育学分野
技術・情報教育学領域

D173824 後藤 孔

第 1 章 はじめに

近年，文部科学省は Society5.0 時代の学びの実現に向けて抜本的な教育改革案 [1] を打ち出しており，その一部としてプログラミング教育に関する言及も行なっている．具体的には，発達段階に応じたプログラミング教育の充実を図るとの記述があり，まず小学校ではプログラミングの必修化，つぎに中学校ではプログラミングに関する内容の充実，そして高等学校では全ての生徒がプログラミングのほか，それと関連するネットワークや情報セキュリティなどの情報技術も学習することが求められている．また，大学では，文理を問わず全ての学生に数理・データサイエンス教育を行うことが求められている．

このような状況でプログラミングを学習する対象者の年齢層が広がると，様々な発達段階に応じた教材が必要となり，指導者の負担が大きくなる．プログラミング教育に関して，以前から様々な教材 [2-20] が開発されているが，プログラミングの指導者を積極的に支援することを目指したものはほとんど見当たらない．

そこで，本研究では学習者に加えて指導者も積極的に支援するプログラミング教材を開発する．具体的には，文字ベースのプログラミングで学習を行う際に，学習者のデバッグと指導者のデバッグ指導を援助するデバッグ支援システム [21] と，身近な技術として音声情報処理技術を活用したプログラミング教材 [22] を開発する．

第 2 章 プログラミング教育における問題点

文字入力に基づくプログラミング言語を用いたプログラミングの初学者は，作成したプログラムをコンパイルする際にさまざまなエラーメッセージに直面してプログラムが実行できず，デバッグのために苦労する場合が多い．この場合，指導者がプログラムの誤りの修正方法を指導することとなるが，その誤りが単純な文法誤りの場合でもエラー原因の特定，修正方法の説明などに一定の時間を要するため，個別の指導を行なっている間は授業が停止するという問題を抱えることとなる．このように，プログラムのデバッグは学習者だけでなく指導者にとっても負担が大きいため，デバッグ支援を充実させるとともに，その支援によって初学者のプログラム修正能力を高めていく必要がある．

ところで，作成したプログラムを動作させる際は，テストデータを入力してプログラムを実行させたり，データを出力してプログラムが意図通りに動作しているかを確認させることがある．そのような際に，入力としてデータをあらかじめプログラムに記述しておいたり，出力として文字を画面に表示させる方法などでは指導者が学習者の興味を引き出しにくい．これについては，たとえば音声情報処理技術を活用し，音声認識を入力として，音声合成を出力として採用すれば，学習者が身近な技術を使用しながらプログラミングの学習を行うことができ，指導者が学習者の意欲を高めることへつながる．また，音声情報処理技術を授業内で利用するだけでなく，プログラミングを通して技術の応用を経験させることで，学習者に開発者の視点を身につけさせることができる．

第 3 章 デバッグ支援システム

本研究では，デバッグ支援システムとして補助メッセージを提示する機能をシェルスクリプトを用いて実装しており，このシェルスクリプト内でコンパイラの呼び出しや，コン

パイラの出力メッセージの分析、補助メッセージの挿入などを行うようにしている。この手法は特定のプログラミング言語に限定するものではないが、具体例として、C言語を用いたプログラミングの学習を対象とし、コンパイラとしてGCC 5.4を用いる場合を述べる。

プログラムにおいてエラーが発生するとき、コンパイルエラーが1つのみ生じる場合と複数生じる場合がある。本研究ではエラーの個数に関わらずエラーメッセージとして最初に表示されたものに対して補助メッセージを提示し、デバッグ支援を行う。これは、1つの誤りが最初のエラーメッセージだけでなく、その後の複数のエラーメッセージに関係する場合が多いことを考慮したためで、実際のプログラミング教育においても、まずは最初のエラーメッセージに着目するように指導することが多いことと対応している。補助メッセージを付加するエラーメッセージの種類は、コンパイラが指摘した文法誤りを原因とするものと、単に文法誤りというだけでなく、文字の綴り誤りが起因してその結果文法誤りとなるものとする。

まず、文法誤りに対する補助メッセージについて、デバッグ支援システムが修正案の補助メッセージを提示する方法を述べる。出力されたエラーメッセージに対して修正案が唯一と考えられるものについては、その修正案を示す補助メッセージを提示する。複数の候補が考えられる場合は、1つ目の修正案をプログラムに適用し、デバッグ支援システムで再度コンパイルを実行する。もし、この段階でエラーが解消すれば、このとき適用した修正案を補助メッセージとして提示する。一方、適用した修正案ではエラーが解消されない場合、次の段階として、考えられる他方の修正案をプログラムに適用し、再度コンパイルを実行する。この段階でエラーが解消すれば、適用した修正案を補助メッセージとして提示する。もし、この段階でもエラーが解消しない場合は補助メッセージの提示を行わない。

つぎに、綴り誤りに対する補助メッセージについて、デバッグ支援システムが修正案の補助メッセージを提示する方法を述べる。まず、デバッグ支援システムの内部において、文字列の比較に用いるファイルを作成する。このファイルには文字列のリストが並び、その項目は学習者が記述したプログラムにおける両端が空白で挟まれる文字列とC言語であらかじめ定義される関数やデータ型の名称である。そして、デバッグ支援システムがそのファイルを参照して、綴り誤りの文字列に対して類似度が高い文字列をレーベンシュタイン距離を用いて検索する。

2つの文字列の類似度を考える際によく用いられるレーベンシュタイン距離は、文字列の編集作業における1文字の削除、追加、置換を単位操作としたときに1つの文字列を他方の文字列に変更するために必要な最小の操作回数である。本研究でも同様の距離を扱うが、レーベンシュタイン距離をそのまま用いると、たとえば初学者が関数名として「func1」と記入すべきところを「funcI」と誤記入した場合に、数字の「1（イチ）」と英小文字の「I（エル）」の類似が考慮できないので、レーベンシュタイン距離における置換の対象が数字の1と英小文字のI、これに加えて数字の0と英大文字のOの場合の単位操作に対する距離を0.9に修正する。

たとえば、プログラムにおいて関数を呼び出す際に「func1」と記述すべきところを「funcI

と誤って記入した例を考える。もし、このプログラムに「func1」だけでなく「func2」も定義されている場合を考えると「func1」に対して「func1」と「func2」のレーベンシュタイン距離はいずれも1で、文字列の類似度も等しいので、通常のレーベンシュタイン距離を用いて綴りの修正案を提示すると、「func1」と「func2」を区別して取り扱うことができない。本研究では、レーベンシュタイン距離に修正を加えたものを用いることにより、「func1」への修正を優先的に提案することが可能である。

第4章 音声情報処理技術を活用したプログラミング教材

本研究では、音声情報処理を実行可能な Web Speech API を活用してプログラミング教材を開発した。本教材は文字ベースの言語の JavaScript またはビジュアルプログラミング言語の Scratch に対応している。本教材では、身近な製品が音声情報処理技術とハードウェアを融合させている点を意識して、音声情報処理のみならずハードウェアに対する計測や制御動作も実行できるプログラミング教材となっている。プログラミング言語として、Scratch を用いたプログラミングの学習を行う場合、Scratch のコマンドとして音声情報処理を実行するブロックやマイクロコントローラと通信するブロックを作成することで、音声指示により計測制御動作を実行するプログラミングが行えるようになる。そこで、本研究では Web Speech API やマイクロコントローラへの通信命令を Scratch ブロックとして独自に定義して、既存の Scratch ブロック群にそれらを追加した。

それらのブロックを使用して作成することができるプログラム (Scratch スクリプト) の例として、音声認識の結果に応じて LED を点灯・消灯するものを図1に示す。

図1のスクリプトでは、「音声認識」ブロックが実行され音声認識が完了すると「音声認識結果」ブロックへ認識結果である文字列が格納される。このブロックの値を文字列比較することで処理を分岐させ、値が「こんにちは」と一致すれば LED を点灯、「さようなら」と一致すれば LED を消灯する命令を「マイコン1」と名付けたマイクロコントローラへ出力する。

プログラミング教育において音声情報処理技術を活用することの有効性を検証するため、2018年11月に岐阜県内の高等学校にて授業実践およびアンケート調査を実施した。対象は2年生2クラス(23名と36名)の合計59名とし、実践としては JavaScript を用いる場合の方法を選択して授業を行った。アンケートの質問内容は表1に示すとおりで、



図 1: 音声認識の結果を利用して LED を制御する Scratch スクリプト

表 1: アンケートの内容

| 質問番号 | 質問文 |
|------|------------------------------------|
| I | プログラミングに対する興味や関心がもてましたか？ |
| II | プログラミングに対する興味や関心が持続しましたか？ |
| III | アルゴリズムを工夫しようと思いましたか？ |
| IV | この授業の後も、プログラミングを続けてみたいと思いましたか？ |
| V | 音声情報処理技術に対する理解が深まりましたか？ |
| VI | 音声情報処理技術以外の新しい情報技術に対しても理解が深まりましたか？ |

表 2: 理系クラスのアンケート結果

| | I | II | III | IV | V | VI |
|---|------|------|------|------|------|------|
| 1 | 52.2 | 34.8 | 26.1 | 34.8 | 52.2 | 39.1 |
| 2 | 47.8 | 65.2 | 65.2 | 56.5 | 43.5 | 60.9 |
| 3 | 0.0 | 0.0 | 8.7 | 8.7 | 4.3 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

回答数 23 名, 単位: %

表 3: 文理融合クラスのアンケート結果

| | I | II | III | IV | V | VI |
|---|------|------|------|------|------|------|
| 1 | 33.3 | 27.8 | 22.2 | 36.1 | 38.9 | 19.4 |
| 2 | 66.7 | 63.9 | 55.6 | 36.1 | 55.6 | 58.3 |
| 3 | 0.0 | 8.3 | 22.2 | 25.0 | 2.8 | 22.2 |
| 4 | 0.0 | 0.0 | 0.0 | 2.8 | 2.8 | 0.0 |

回答数 36 名, 単位: %

選択式の 6 項目 (質問番号: I~VI) を設定した。選択式の項目は肯定から否定まで 4 つの選択肢 (1: とてもそう思う, 2: そう思う, 3: そう思わない, 4: 全くそう思わない) を設定した。

表 1 のアンケートを実施し, 質問項目 I から VI の回答をクラスごとに集計した結果をそれぞれ表 2 と表 3 に示す。表 2 は理系のクラスに対応しており, 全ての質問に対して肯定的な回答が多い。一方, 否定的な回答は質問項目ごとに, 2 名以下であった。表 3 は文理融合のクラスに対応しており, 表 2 と比べるとやや否定的な回答も散見される。しかし, 全体としては全ての質問に対して肯定的な回答が優勢である。

以上をまとめると, 両クラスに共通する傾向として, 全ての生徒がプログラミングに対する興味や関心をもち, 多くの生徒が音声情報処理技術に対する理解が深まったと回答している。

第 5 章 開発した教材の考察

本研究で開発した教材に関する考察を述べる。まず, 第 3 章で開発したデバッグ支援システムに関連して, 比較的バージョンが新しい GCC の出力メッセージとの比較を行う。つぎに, 第 4 章で開発した音声情報処理技術を用いた教材に関連して, 脳波情報を用いた教材の可能性を検討する。

本研究のデバッグ支援で対象としているコンパイラ GCC 5.4 は 2016 年に公開されたもので, GCC はその後も複数の版が並行して開発されている。2018 年に公開された GCC 8.1 以降では, エラーメッセージの出力内容に改良点がみられる。たとえば, 2019 年に公

```

prog.c: In function 'main':
prog.c:4:1: error: stray '\343' in program
    4 |   return 0;
      |   ^
prog.c:4:2: error: stray '\200' in program
    4 |   return 0;
      |   ^
prog.c:4:3: error: stray '\200' in program
    4 |   return 0;
      |   ^

```

図 2: 非アスキー文字が混入したときの GCC 9.2 が出力するエラーメッセージ

開された GCC 9.2 では、文法誤りのエラーに関して、非アスキー文字を含むプログラムをコンパイルすると図 2 に示すようなエラーメッセージが出力される。

図 2 のエラーメッセージでは、オリジナルのコンパイルエラーメッセージに加えて、エラー発生 の要因となったソースコードの具体的な場所が「^ (キャレット)」の記号により示されている。

この例に見られる GCC 9.2 の機能は、プログラミングの学習を重ね、ある程度プログラミングの文法を理解した学習者に対しては有用と考えられる。プログラミングの学習量が蓄積されると、エラーを修正する経験も蓄積されるため、対処方法を把握していれば自力でデバッグすることができる。しかし、プログラミングの学習経験が浅い初学者にとっては、エラーの発生箇所を特定できたとしてもエラーの修正方法までは把握できない可能性があり、その点を考慮すると GCC 9.2 の機能だけでは不十分と考えられる。

一方で、本研究で開発したデバッグ支援システムが提示する補助メッセージは、エラーの修正案を示して学習者の学びを促すとともに、教員の指導負担を減らすための情報であり、学習者と指導者の支援を積極的に意図している。コンパイラの表示するメッセージは、あくまでプログラマのプログラミング効率を高めることが主な目的であり、この点が本研究との根本的な相違点である。

本研究では学習者の興味を引く技術として音声情報処理を用いた教材を開発したが、それ以外の例として脳波情報を用いたプログラミング教材も考えられる。図 3 にプロトタイプとして開発した本教材の概要を示す。

図 3 の教材では、簡易脳波計を用いて計測された脳波情報を PC で処理し、その PC に

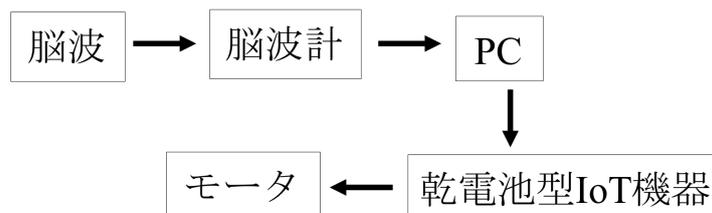


図 3: 脳波情報を用いたプログラミング教材の概要

接続される乾電池型 IoT 機器へ独自のコマンドを送信することで、最終的にモータを回転させることができる。プログラミングの学習と融合させる場合は、たとえば、PC から乾電池型 IoT 機器へのコマンド送信をプログラミングで実現することが考えられる。

第 6 章 おわりに

本研究ではプログラミング教育の支援を目指してデバッグ支援システムと音声情報処理技術を活用したプログラミング教材を開発した。このデバッグ支援システムは、学習者と指導者に対してプログラムのデバッグを支援するものである。本システムの開発においては、プログラミングの初学者がコンパイラの出力するエラーメッセージに基づいて自力でデバッグ作業を行うのが容易でないことに着目した。そして、初学者にありがちなプログラムの文法誤りに対して、具体的な修正方法を提案する補助メッセージを出力するシステムを実装した。具体的には、文法誤りによるエラーや、文字列の綴りを間違えた場合のエラーを想定し、それらの内容をシェルスクリプトで判別して、それぞれのエラーに対応する修正案を出力している。この結果は、デバッグ作業に関する学習者と指導者の双方の負担が軽減され、プログラミング教育の効果を高められることが期待される。

音声情報処理技術を活用したプログラミング教材は、近年、音声認識や音声合成の技術を用いた製品が普及していることに着目し、Web Speech API を活用して生徒が身近な技術を活用したプログラミングをできるようにしたものである。具体的には、Scratch を用いてプログラミングを行う場合は、Scratch スクリプトによって音声認識を実行し、その結果を用いて計測制御動作を実行することができるようにした。プログラミング教育において音声情報処理技術を活用することの有効性について検証した授業実践では、受講した生徒の大部分がプログラミングに関心を抱くことを確認した。この結果は、学習者がプログラムに含まれる関連技術を学び、開発者としての視点を身につけられるだけでなく、プログラミング教育を担当する指導者が学習者の意欲を高められることへつながっている。また、指導者が音声情報処理技術を授業で手軽に利用することを可能とし、指導者ごとの事情に応じた独自の教材開発を支援することにもつながっている。

参考文献

- [1] 文部科学省. Society5.0時代の教育・科学技術の在り方について, 2020. https://www5.cao.go.jp/keizai-shimon/kaigi/minutes/2019r/1113/shiryo_06.pdf (2020年1月18日最終アクセス) .
- [2] 岡本雅子・村上正行・喜多一・吉川直人. 初学者を対象とした自習中心のプログラミング教育の教材開発と評価. 情報教育シンポジウム 2010 論文集, 2010(6), pp. 87–94, 2010.
- [3] 酒井統康・長谷川元洋. Sphero を用いた小学校プログラミング学習単元の開発. 日本科学教育学会研究会研究報告, 31(8), pp. 117–122, 2017.
- [4] 佐藤慎一・飯田元・井上克郎. プログラムの依存関係解析に基づくデバッグ支援ツールの試作. 情報処理学会論文誌, 37(4), pp. 536–545, 1996.

- [5] 江木鶴子・竹内章. プログラミング初心者にはトレースを指導するデバッグ支援システムの開発と評価. 日本教育工学会論文誌, 32(4), pp. 369–381, 2009.
- [6] 兼宗進・中谷多哉子・御手洗理英・福井眞吾・久野靖. 初中等教育におけるオブジェクト指向プログラミングの実践と評価. 情報処理学会論文誌, 44(13), pp. 58–71, 2003.
- [7] 間辺広樹・長島和平・長慎也・並木美太郎・兼宗進. 高等学校における複数言語によるプログラミング教育の提案～情報システムの理解を目標としたドリトル, JavaScript, PHP の連携～. 研究報告コンピュータと教育 (CE), 133(3), pp. 1–10, 2016.
- [8] 大見嘉弘・滑川敬章・永井保夫. 情報系高校におけるセンサを利用したプログラミング教育の実践. 研究報告コンピュータと教育 (CE), 114(5), pp. 1–7, 2012.
- [9] 滑川敬章・落合秀也・山内正人・高岡詠子・中山雅哉・江崎浩・砂原秀樹. 情報系高校における環境情報を計測・可視化する実用的なプログラミング教育の実践. 研究報告コンピュータと教育 (CE), 116(16), pp. 1–8, 2012.
- [10] 村田育也・広田高. 5パズルを用いたプログラミング教材開発と高等学校情報科における授業実践. 北海道教育大学紀要教育科学編, 63(1), pp. 201–209, 2012.
- [11] 木下崇・鎌田敏之・本多満正. Scratch の Mesh 機能を用いた双方向性のプログラミング教材の開発～中学校技術科のネットワークを用いたコンテンツ制作の導入として～. 愛知教育大学技術教育研究, 6, pp. 7–12, 2018.
- [12] 秋山大翼・木下崇・本多満正. ネットワークを用いた双方向性のあるプログラミングの教材開発. 愛知教育大学研究報告芸術・保健体育・家政・技術科学・創作編, 67(2), pp. 15–19, 2018.
- [13] 鎌田敏之・本多満正・木下崇・秋山大翼. ネットワークを用いた双方向性のあるプログラミングの授業へ向けた教員支援の試みとその評価. 愛知教育大学研究報告芸術・保健体育・家政・技術科学・創作編, 67(1), pp. 31–36, 2018.
- [14] 西ヶ谷浩史・紅林秀治・青木浩幸・保福やよい・原久太郎・久野靖・兼宗進. ITクラフトマンシッププロジェクト～中学生によるネットワークプログラミング～. 情報処理学会研究報告コンピュータと教育 (CE), 16, pp. 173–180, 2006.
- [15] 川井勝登・荻窪光慈・山本利一. ネットワークを利用した双方向性のあるコンテンツのプログラミングに関する指導過程の提案～反転学習で活用する学習コンテンツの開発と授業実践～. 埼玉大学教育学部附属教育実践総合センター紀要, 17, pp. 77–84, 2019.
- [16] 滑川敬章・落合秀也・山内正人・高岡詠子・中山雅哉・江崎浩・砂原秀樹. 情報系高校における環境情報を計測・可視化する実用的なプログラミング教育の実践. 研究報告コンピュータと教育 (CE), 2012-CE-116(16), pp. 1–8, 2012.

- [17] 間辺広樹・長島和平・長慎也・並木美太郎・兼宗進. 高等学校における複数言語によるプログラミング教育の提案～情報システムの理解を目標としたドリトル, JavaScript, PHP の連携～. 研究報告コンピュータと教育 (CE), (3), pp. 1–10, 2016.
- [18] 中原久志・村長廉太郎. 双方向性コミュニケーションを行うブロック型プログラミングアプリケーションの開発. 大分大学教育学部研究紀要, 40(1), pp. 69–79, 2018.
- [19] 島袋舞子・兼宗進. ドリトル言語における Leap Motion 対応と教育的利用の可能性. 情報教育シンポジウム 2014 論文集, 2, pp. 239–243, 2014.
- [20] 鈴木聡. 視覚的プログラミング言語 Scratch を活用した計測制御教材の開発. 木更津工業高等専門学校紀要, 46, pp. 21–26, 2013.
- [21] 後藤孔・藤中透. プログラミング教育におけるデバッグ支援. システム制御情報学会論文誌, 32(6), pp. 249–255, 2019.
- [22] 後藤孔・藤中透. プログラミング教育における音声情報処理技術の活用. 日本情報科教育学会誌, 11(1), pp. 15–22, 2018.