1F3-3

学習者による思考及び操作を中心とした プログラミング学習支援システムの設計と開発

Design and Development of Programming Learning Support Systems Focusing on Learner's Thinking and Operations

松本 慎平 *1 Shimpei Matsumoto　石井 元規 *2 Motoki Ishii　林 雄介 *2 Yusuke Hayashi　平 嶋宗 *2 Tsukasa Hirashima

*1広島工業大学情報学部
Faculty of Applied Information Science, Hiroshima Institute of Technology

*2広島大学大学院工学研究科
Graduate School of Engineering, Hiroshima University

The authors have been working on the development of a programming learning support systems that does not require learner to write source code. The aim of these developed systems is to train factors necessary for programming other than cognitive activities related to writing. The authors assume these programming learning activities except writing source code as "thinking", and prepare a frame depending on the aim so that a teacher and a learner him/herself can understand and evaluate his/her learning level. This paper clarifies the details of these programming learning support systems.

1. はじめに

昨今，プログラミングとそれに必要な力 (Computational Thinking) は特に重要な技能として社会的に広く認識されている．大学等高等教育機関では，プログラミング技能は従来から重要な科目として位置付けられてきた．学習者のプログラミングに対する理解を支援するため，講義では様々な工夫が検討され，そして，そこでの成果は多くの学術雑誌で報告され続けている．しかしながら，プログラミングを不得手とする学習者を的確に支援できる教授法は，現在までに十分に確立されていない．その理由のひとつとして，プログラミングは，論理的思考力，言語力，発想力，数学力など，多様な技能を同時に要求している一方で，プログラミングを不得手とする学習者にはどのような技能がどれだけ不足しているのかを把握する手段が十分に整備されていないことにあると考えられる．従来プログラミングの適正が不十分と評価されてきた学習者の成熟的な学習活動を支援するためには，プログラミングに要求される知識を内的なものと外的なものとで切り分け，かつ，学習要素を分割したうえで，到達目標と逸脱の程度を客観的に評価できる粒度で教材を細分化し，理解度に応じた難易度の教材を段階的に用意する必要があると考えられる．プログラミング教育を広く普及させるためには，プログラミングを不得手とする学習者の存在を無視できない．したがって，現在プログラミングの教授に関わる者が第一段階として取り組めることとして，従来プログラミングの素養がないと判断されてきた学習者層の支援に繋がるような基盤整備を進めることであると考える．

以上背景のもとで，著者らはこれまで，"書かせない" プログラミング学習支援システムの開発を進めてきた [松本 16][石井 16]．これらは，書かせることによる認知活動以外で，プログラミングに必要とされる要素の学習を狙いとしたものである．当然，プログラムを書くことはプログラミングスキル向上において最も重要なことであるが，講義という限られた時間内に記述させる演習を行うことは学習者個々の進捗のばらつきも大きく，計画的な講義進行は期待できない．そこで，プログラムを書くという学習活動は時間外での課題とし，講義で教授した内容の確認，前回までの知識の定着度合いを確認するという目的で運用することを想定したものが，著者らのこれまでの成果である．著者らは，書かせること以外のプログラミング学習活動を "考えること" とし，学習の狙いに応じてフレームを用意することで，学習者の学習活動や理解過程を教授者及び学習者自身が評価できるようにしたいと考えている．本稿では，これらプログラミング学習支援システムの詳細を明らかにすることを目的とする．

連絡先: 松本慎平，広島工業大学情報学部知的情報システム学科，〒 731-5193 広島市佐伯区三宅 2-1-1,
E-Mail: s.matsumoto.gk@cc.it-hiroshima.ac.jp

2. 読解学習支援システム

2.1 学習方式の狙い

まず，著者らは，プログラミングの学習活動のひとつとして読解に着目した．その理由として，プログラムを書かせることは，多くの学習要素を学習者に要求する課題がある．その反面，読解は，求められる学習要素が明確であること，すなわち，ソースコードの内的な構造を適切に読解できるかどうかという狙いのもとで評価基準を設計できる．それにより，正誤の結果から，逸脱の度合いを定量的に示すことや，それを踏まえたアドバイスを学習者に自動フィードバックできると考えている．加えて，プログラムソース自体は，データ依存や制御依存の関係ネットワークにより形作られる構造でしかない．よって，外的な構造とは一切関係を持っておらず，プログラムソース自身が保持する内的な構造にのみ依存する学習課題から学習者の理解過程を把握すれば，学習者の理解を阻害する要因を究明できると考えられる．ただし，外的な構造を持たないソースコードを読解させることが学習として意味を持つことを明らかにしたうえで学習教材を開発する必要がある．そこで本研究では，外的構造の意味に頼らない，プログラムソース自身が保持する内的な構造 (データ依存関係) にのみに基づいた読解学習がプログラミング力を向上させるための必要な条件であることをまず明らかにするために，非言語的な認知理解過程を推定する手段として視線運動に着眼し，読解過程中の視線運動の推移を分析した．その結果，プログラム構造が認知過程と確かに関係していることを明らかにした [沖本 15][花房 15][花房 16].

この結果を踏まえ，次に，プログラムソースの内的構造の把握と適切な読解，すなわち，スライシングに求められる技能の訓練を狙いとし，読解の手軽さとトレース・デバッグ学習の基礎力向上に対しての有効性 [江木 09]，プログラミングの基本である順次処理の反復学習の有効性 [Okamoto 10] を踏まえた上で，外的構造の意味に頼らない，プログラムソース自身が保持する内的な構造 (データ依存関係) にのみに基づいた読解学習を容易とするための LMS(Learning Management System) を開発し，実講義での運用を達成した．

2.2 読解学習教材と期待される効果

読解学習課題は任意の規則で生成されたものであり，内的構造の把握力 (スライシングスキル)，言語仕様に関する知識，最低限の記憶力・計算力のみの技能に直結する学習課題である．よって，プログラムを不得手とする学習者の特徴を十分に把握できていないという従来の問題の解決に貢献可能な仕組みであると考えている．加えて，プログラムソース自身が保持する内的な構造にのみ依存する学習課題を題材として学習者の技量を相対的に定量化できれば，内的構造の観点においての指導が可能となるため，従来プログラミングの適性が十分でないとみなされてきた学習者層に新たな観点での教授法を提供可能になると考えている．

生成されるソースコードは数十行の短いものである．この大きな理由は，学習課題をできる限り単純にすることで，学習者の特徴分析に利用できなければならないと考えたためである．まずは，数十行程度の処理の読解を慣れさせることで一般的なプログラム読解も抵抗なく対応できるようになるのではないかと考えている．また，近年のオープンソースソフトウェアの依存度が高まっていることから，読解技能の必要性は強く認識されつつ．このような中，ソースコードを書く側も，書いたプログラムが再利用されること，読まれることを意識すべきであると著者らは考えている．多読の経験を通じて，多くのソースコードに触れさせることで，読みやすい・読みにくいソースコードが存在していることを理解させ，読みやすいソースコードを書くということを意識付けたいと考えている．本研究での狙いをまとめると，1．プログラムに慣れること，2．効率的にプログラムを読めるようになること，3．プログラムの知識を確実に憶えること，4．プログラムの読みやすさを意識すること，である．

開発したシステムは，C 言語の読解学習を対象としたものであり，ソースコードを題材とした問が自動生成され学習者に提示される．生成されるソースコードは外的な目的のもとでの処理を行うものではなく，外的な意味を持たない処理の連続である．ただし，プログラム自身が持つ構造自体には意味を持っている．著者らのこれまでの成果は，1．内的な意味のみに依存した教材の適用でプログラミングの理解の確認に有効に働くということ，2．ソースコード読解はプログラミングへの抵抗感を減らすことに寄与できること，3．ソースコード記述文の困難度水準を量的に定義できるようにすること，の 3 点に貢献可能であると考えている．

2.3 機能の詳細

開発システムは Web アプリケーションであり，Chrome など HTML5 に準拠したブラウザでの動作を推奨するものである．システムは Apache 2.4.7 でデータ入出力を行っており，問題提示には JavaScript ライブラリである jquery 1.7.2，問題データ管理には MySQL 5.6.16 を用いている．アプリケーション自体は，php 5.5.9 で開発されている．図 1 は主として教授者が利用する問題生成画面，図 2 はプログラミング学習

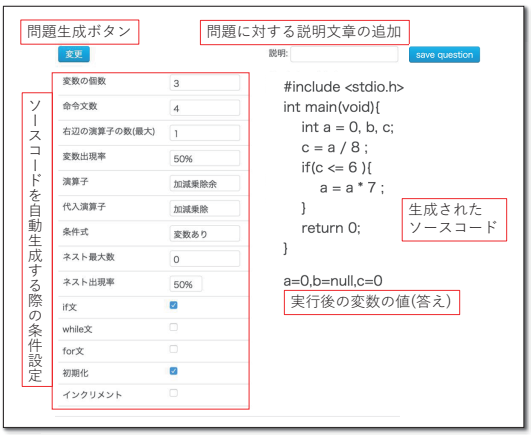

図 1: 問題生成画面

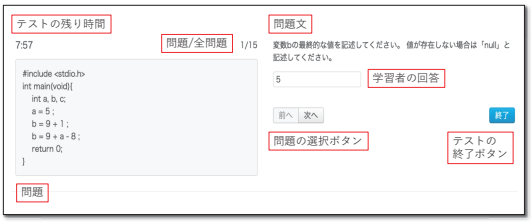

図 2: 問題提示画面

図 3: 復習用画面

者が使用するテスト問題，図 3 は復習用画面それぞれの 1 例であり，図 2，図 3 の場合は，自由記述方式の問題が提示されている．開発システムが持つ問題生成機能では，任意の規則に基づいて問題を生成できる．“変数の数”，“命令文の数”，“演算対象の変数の数”，“利用演算子 (代入演算子)”，“分岐・繰り返し利用の有無”，“条件判定文の記述に関する制約” など設定を事前に行うことで，条件に合ったソースコードを生成できる (図 1 参照)．なお，学習者の理解の度合いを確認するため，意味のない代入処理や，意味のない条件判定処理の有無を設定できるようになっている．設定情報を読み取った後，ヘッダ部が記述処理が行われ，変数の宣言記述が行われる．その後，設定された命令文数だけ本文生成処理が呼ばれ，C 言語のプログラムが形成される．制限時間を設定可能なテストは，1 問以上の複数の問題で構成される．一度受けたテストを 2 度受験できないようになっているが，一度受けたテストに含まれる問題は，問題一覧機能から何度も練習できるようになっている．

開発システムでは，自由記述方式の他に，多肢選択方式，並び替え方式，空欄補充方式が用意されており，教授設計に基づき，希望の問題形式で提示可能である．

本システムを大学 1 年生のプログラミング未経験者を対象とした実講義で適用し，学習支援を試みた [松本 16]．講義で取り上げられるプログラミング言語は C 言語である．読解学習は，15 回のうち 6 回の講義で行い，各講義時間内の 10 分間を利用しテスト形式で実施された．15 回の講義終了後，アンケートを実施し，6 段階リッカート尺度で読解学習の評価を得た．

運用の結果からは，初学者にとってプログラム読解を困難とする記述を明らかにし，プログラミング教材の作成や教授法検討の際に参考になり得る知見を得ることができた．具体的には，自動生成されたソースコードを読解教材とすることは，プログラミングの講義では通常取り扱う機会が少ない難解な仕組みに触れることがしばしばあった．このような発展的な記述は，初学者にとっては極めて高度な仕組みであり，プログラミングに対する苦手意識をさらに悪化させる可能性があることを確認した．ただし，発展的な記述は，言語仕様の限りにおいては可能なものであり，行数を削減できる場合や計算機負荷を劇的に軽減できることがあるため，場合によっては高級テクニックと称され崇められることも多い．発展的な記述を教授するのであれば，これは同時に読解を困難にする仕組みであると考えられるため，読解学習教材を提示した上で，これら弊害についても同時に触れ，その後，バグや誤解を招く可能性の高い記述であることを学習者に伝える必要があると考えられる．多様な技術水準の人に読解されることをソフトウェア開発の前提とするならば，読解が難解なソースコードを教材として活用することも有用ではないかと考えられる．

他，学習者の理解水準によって，読解学習に対するニーズが異なっていることを確認した．具体的には，上級者は，ポインタや構造体なども含み，発展的な記述の知識習得を求めていた一方で，初級者は，基本的な知識の定着度合いを確認するための教材を求めていた．このことから，上述した発展的な記述に関する指導を除くならば，上位群の期待には応えられないということを意味する．プログラミングは，熟練的な技能と教養的な技能という 2 面性を有した技能であると見なすこともできる．そうならば，プログラミングを教授する場合，プログラミングと一括りに到達目標を設定するのではなく，社会においてどのような状況で用いられるプログラミング技能なのかによって異なった授業を設定し，到達目標を掲げる必要があると考えられる．

3. カード方式を用いた学習支援システム

3.1 学習方式の狙い

著者らは，プログラミングをいくつかの活動に分割した後，一部の活動を非本質的認知負荷として間接的に減らし，“考えること” に認知資源を集中させる仕組みとして “カード演習方式” を定義し，この方式を採用した学習支援システムの開発を行った (図 4 参照)．この “カード演習方式” は，図 5 に示すとおり，ヴィジュアルプログラミングと似た仕組みである．問題文とプログラムコード 1 行分が書かれたカードが提示され，学習者は問題文の処理に合うようにカードを選択し，並び替える操作によってプログラムを組み上げ，回答する演習となっている．なお，コード 1 行分とは，1 処理分のことを指す．選択肢のカード群は，回答に用いるカードと回答に必要ないダミーカードで構成されている．また，並び替えるプログラムは，全体ではなく，部分的となっている．ここでは，タイプミス等による文法エラー多発によって，より学習が不十分だと考えられる行間の活動に注目したものである，すなわち，行間の活動，

図 4: カード方式による学習支援システムの動作例

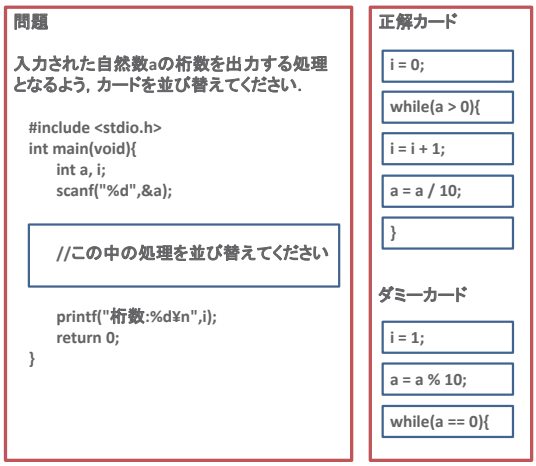

図 5: カード方式

特にアルゴリズム等に集中できる演習だと考えている．

カード方式に基づいた学習支援システムは，主に 4 つの特徴を有している．1 つ目は，プログラムを直接打つ必要がないという点である．タイプミスに起因する文法エラー等が発生しないため，文法やタイピングに関する認知負荷が減らせていると思われる．2 つ目は，作るプログラムが選択肢によって固定されているため，良いデザインパターンを学ばせられる点である．3 つ目は，各問題の採点時に正誤判定やヒント等，即時のフィードバックを返せる点である．4 つ目は，正誤判定のみならず，回答に至るまでのカードの動き等をログとして残せる点である．

3.2 プログラミング学習支援システム

開発したプログラミング学習支援システムは，カード演習方式を採用している．開発言語は Ruby で，データベースは MySQL を用いている．サーバ上で動作するシステムであり，Web ブラウザからアクセスできるようになっている．開発したシステムは，カード演習方式の演習を行う機能だけでなく，LMS (Learning Management System, 学習管理システム) としての機能も一部持ち合わせている．ユーザは学生，先生，管理者の 3 種類があり，それぞれ行える機能が異なる．

提案システムは，サーバ上で動作するシステムであり，Web ブラウザからアクセスできる．提案システムの実装の詳細は以下のとおりである．

- 基本言語：Ruby 2.1.2
- Web アプリケーションフレームワーク：Ruby on Rails 4.2.4

- OS：Ubuntu 15.04
- Web サーバソフトウェア：Apache 2.4.10
- データベース管理システム：MySQL 5.6.28
- CSS フレームワーク：UIkit 2.23.0
- JavaScript ライブラリ：jQuery 1.11.3

演習は，演習形式と試験形式の 2 種類がある．演習形式では，間違えた場合に何度でも同じ問題に再挑戦可能となっている．一方の試験形式では，各問題の回答は 1 度きりである．また，分単位で時間制限を設けることもできる．問題作成では，タイトル，レベル，問題文，コードのカード群 (ダミーを含む)，回答時のフィードバック (ヒントや出題の意図等) の要素を登録できる．コードのカード群は正解の順番を指定する必要があるが，順序不同 (順番が異なっていても正解) の場合にも対応している．演習時は，上側に問題文，中央に各種ボタン，右側にカード群の選択肢欄，左側に回答欄が表示される．問題文の「//この中の処理を並び替えてください」部分が問われているプログラムの部分であり，回答欄と対応している．また，回答欄の行数は，回答に必要なカードの枚数を表している．

カードは，マウスのドラッグ＆ドロップ操作で動かすことができ，右側から左側にカードを移動し，プログラムを組み立てていく．回答ボタンを押すと回答することができ，演習形式の場合は，正誤判定とヒントや出題の意図等のフィードバックが表示される．試験形式の場合は，何も表示されない．そして，演習形式の場合は正解のとき，試験形式の場合は正誤にかかわらず，自動的に次の問題へと遷移する．また，前の問題ボタンや次の問題ボタンを押すことで，意図的に問題を遷移することも可能である．なお，既に解き終わった問題へは遷移できない．全ての問題を解き終わると，演習自体が終了し，演習結果が表示される．なお，演習形式の場合，全問正解にならない限りは演習が終了しない．しかし，どうしても解けない問題がある場合は，パスボタンを押すことで，その問題をパスすることができる．このとき，回答結果は当然不正解となる．

3.3 実験内容

提案システムにより文法等の認知負荷を減らし，認知資源をアルゴリズム等に集中できるか検証するため，実験を行った．実際に開講されたプログラミングの授業にて行い，被験者は主に学部 1 年生とした．実験内容としては，授業期間を通してシステムを用いた問題演習を行い，最後の授業にて認知負荷に関するアンケートを行った．出題する問題の言語は，授業でも用いている C 言語とした．アンケートでは，一般的なコーディング演習とカード演習方式各々の認知負荷について，6 段階リッカート尺度での主観的評価を要求した．そして本研究の目的は，学習者の各活動における認知負荷を測定することにより，カード演習方式が実際に認知資源をそれらの活動に集中できるかを検証することとする．なお，認知負荷の測定方法としては，人間が認知負荷を適切に評定できるという報告 [Gopher 84] を基に，学習者自身に認知負荷を数段階で評価してもらうこととする．

アンケートでは 98 名から回答を得た．そのうち，適当に回答したと思われる 10 名分のデータを除き，88 名分のデータで分析を行った．その結果，カード演習方式は確かに，学習者の認知負荷を減らせていることが示唆された．また，コーディング演習とカード演習方式それぞれにおいて，アルゴリズムとその他の活動の認知負荷量を比較した結果，カード演習方式は，文法，タイピング，デバッグ等の認知負荷を減らし，アルゴリズム等の活動に認知資源を集中できていたことを確認した．

4. おわりに

本稿では，“考えること” に主眼を置いたプログラミング学習支援システムの詳細を明らかにした．

謝辞

本研究は，独立行政法人日本学術振興会科学研究費助成事業 (基盤研究 (C)16K01147，26350296) の助成を受けて実施した成果の一部である．ここに記して謝意を表します．

参考文献

[松本 16] 松本慎平, 山岸秀一, 加島智子, 林雄介, 平嶋宗, 書かせること以外のプログラミング指導から見えてきたこと, The 30th Annual Conference of the Japanese Society for Artificial Intelligence, 2016, 1C5-OS-13b-2 (2016).

[石井 16] 石井元規, 岩井健吾, 松本慎平, 平嶋宗, 林雄介, プログラムを書かせないプログラミング学習支援の可能性検証－命令を組み立てることによるプログラム作成方式－, 2015 年度教育システム情報学会学生研究発表会中国地区講演論文集, P02, pp.115-116 (2016).

[沖本 15] 沖本恒輝，松本慎平，山岸秀一，加島智子，プログラミング読解過程中の視線運動のプログラム構造に基づく解析，電気学会 電子・情報・システム部門大会，P06，pp.129-130 (2015).

[花房 15] 花房亮, 沖本恒輝, 松本慎平, 山岸秀一, 林雄介, 平嶋宗, 確率モデルによるプログラム読解中の視線運動の分析, 第 66 回電気・情報関連学会中国支部連合大会講演論文集, 27-11 (2015).

[花房 16] 花房亮, 松本慎平, 平嶋宗, 林雄介, 初学者のソースコード読解における視線運動とデータ依存構造の関係分析, 人工知能学会研究会資料 SIG-ALST-B504-02, pp.5-10 (2016).

[江木 09] 江木鶴子, 竹内章, プログラミング初心者に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, Vol.32, No.4, pp.369-381 (2009).

[Okamoto 10] M. Okamoto, K. Terakawa, M. Murakami, K. Ikeda, M. Mori, T. Uehara and H. Kita, Computer Programming Course Materials for Self-Learning Novices, Proc. of World Conference on Educational Multimedia, Hypermedia and Telecommunications, Vol.2010, No.1, pp.2855-2861 (2010).

[Boswell 11] D. Boswell and T. Foucher, The Art of Readable Code (Theory in Practice), O'Reilly Media (2011).

[石井 17] 石井元規, 松本慎平, 林雄介, 平嶋宗, カード方式を用いたプログラミング学習支援システムの認知負荷の観点による評価, , 人工知能学会研究会資料, SIG-ALST-B506 (2017).

[Gopher 84] D. Gopher, R. Braune, On the psychophysics of workload: Why bother with subjective measures?, Human Factors: The Journal of the Human Factors and Ergonomics Society, Vol.26, No.5, pp.519-532 (1984).