

# 初学者向けプログラミング演習支援システムにおける 学習進度適応型ヒント提示機能の実装 Implementation of Learning Progress Adaptive Hint Suggestion Function in Programming Exercises Support System for Beginners

鈴木 孝幸<sup>†</sup>      納富 一宏<sup>†</sup>  
Takayuki Suzuki    Kazuhiro Nootomi

## 1. はじめに

筆者らの所属する神奈川工科大学情報学部情報工学科では、1年生の初学者プログラミング教育にC言語で行っている。講義授業と演習型授業を組み合わせるが、演習型授業では実際にプログラムを作成・実行する必要がある。初学者向けのプログラミング言語の演習型授業では、学生の習得状況や進捗状況の個人差が大きく、教員にとって個別の学生の進捗状況の把握は難しい。また、演習時間中に学生の習得しているレベル・進捗に合わせて問題のヒントを提示することは有効であるが、適切なヒントを適切なタイミングで提示することは教員の経験に依存している。本研究では、先行研究として筆者らが提案している学生の解答行動のモデル化と学生の進捗と躓きに適応した解答のヒントの自動提示機能[1][2]の演習支援システムへの実装を行った。演習問題であるので、正答が事前に用意されているので、正答となるソースコードからヒントを生成する。ヒントの提示は、問題の解答を始めてからの経過時間と問題の難易度に従い、段階的に提示する。提案モデルとヒント提示のタイミングなどの評価のため、受講生に試作システムを試用してもらった。適応型ヒントの提示による学習効果が高いかなどの評価を行ったので報告する。

## 2. プログラミング演習支援システムの概要

本研究でのプログラミング演習支援システムの概要を示す。本システムは、大学で使用しているCMS (Course Management System) に組み込んで実装する。

### 2.1 ソースコード提示方法

C言語のプログラミング演習支援については、ソースコードの一部を隠して穴埋め問題として出題するツールが多い[3][4]。穴埋め問題の場合、自動採点にも対応しやすい。また、穴埋めの部分を選択肢にするなど難易度の調整も可能となる。初学者の導入として穴埋め問題は最適であるが、C言語のプログラミングの場合、ソースコードを1から記述することを目的でもあるので、穴埋め問題だけでは不十分である。そこで、初心者への絵画の練習法で薄く表示されたサンプルをなぞって完成させる方法があるが、それをプログラミングに適用して、ソースコードが薄い色で表示されて、学習者がそれにしたがってタイプしていったプログラムを完成させる手法を採用する。ソースコードのすべてが一度に見えるようになるのではなく、穴埋め問題のように(演習問題として)重要な部分は、時間の経過とともに表示されていくようにする。

## 2.2 教員と学習者の操作

教員は、演習問題のソースコードを用意する。ソースコードを元に、穴埋め問題の自動生成手法[1]により、文法事項を分析して、①関数定義、②変数宣言、③関数コール、④制御構造などをキーワードにする出題対象を決定して、ブランクソースとフィラーを作成する。ブランクソースが問題の穴になる部分を指定し、フィラーがそこに入るべき解となる。このブランクソースとフィラー群を格納して、学生が演習を受ける際に、自動的に出題する(図1)。

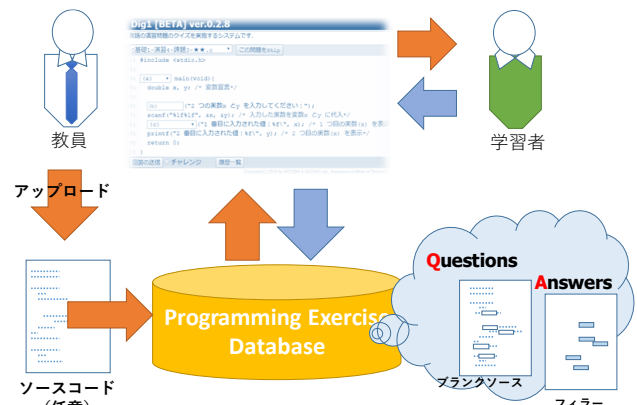


図1 教員と学習者の操作

### 2.3 ヒントの提示による学習者への効果

演習授業では、難易度が違う複数題の問題が出題される。授業時間は有限の時間なので、難易度を基に1題あたりの制限時間を設定できる。この制限時間内でソースプログラムを完成させた場合、問題を理解していると判断できるが、時間切れになってしまう場合は、何らかの理解が不十分で

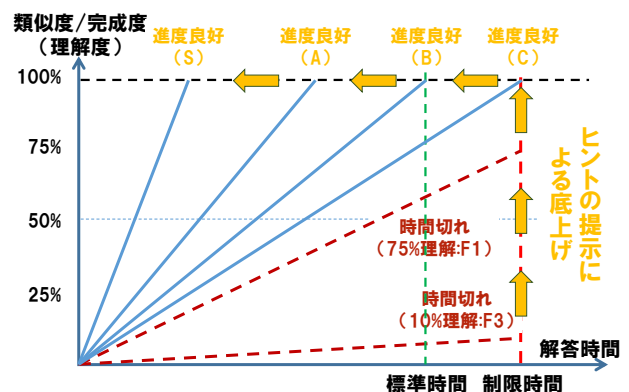


図2 理解度と進捗の個人差とヒント提示の効果

<sup>†</sup> 神奈川工科大学, Kanagawa Institute of Technology

あると想像できる。この部分を足りない部分をヒントとして補ってあげれば、理解度を上げたことになり制限時間内にプログラムを完成させることになる。また、制限時間内に完成させている場合でも、完成時間が短くなるので、成功体験を経験させることができる(図2)。

## 2.4 演習問題提示アルゴリズム

学習者には、問題文と入力できるエディット部が表示されて、エディット部にソースコードを入力していくことになる。エディット部には、学習者が入力するコード以外に、状況にあわせてソースコードの部分が表示色を変えて表示されることがある。学習者の画面例を図3に示す。

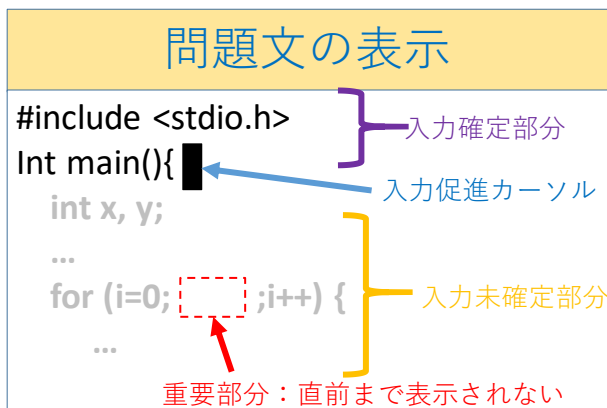


図3 学習者の画面例

学習者へのソースコードの部分の表示は、下記のアルゴリズムで行う。学習者の習熟レベル( $L_s$ )に合わせて、空白ソースから、表示しない(穴となる)場所を選択しておく( $P_i$ )。習熟レベルは、学習者の学習履歴としての正解率などから設定する。

1. 問題開始時刻  $t_0$ からの経過時間を測定して  $t_i$  とする
2.  $t_i$  が、制限時間  $T$  よりも十分に小さい場合(例えば、 $1/2 * T$  など)、空白ソースで、 $P_i$  以外の場所を順次、別の色(薄い色)で表示して、学習者に入力を促す
  - (ア) 表示する順番は、その時の入力用のカーソルがある場所の近くから、決めていく
  - (イ) 入力用のカーソルの位置の近くに表示できるコードがなければ、離れた場所を選択して表示していく
3. 残り時間 ( $T-t_i$ ) が制限時間の  $1/5$  以下(猶予時間と呼ぶ)になったら、表示していない  $P_i$  も順次、別の色で表示していき、学習者に入力を促す

途中で、学習者からの入力が一定時間なかった場合( $T_b$ )は、プロンプトの色が変わり点滅の速度を早くして警告をする。また、習熟レベルに応じた表示しない場所の選択については、選択する個数で調整する。つまり、レベルが低い場合は、選択箇所を少なくすることで対応する。

## 2.5 現状について

論文執筆時では、実装が終わり、パラメータの調整を行っている。また、試作システムを実際の演習授業で使用してもらうことを計画している。パラメータについては、少数の被験者のデータをサンプルとして設定するが、それが適正であるかの評価を実施中である。

## 3. 今後の計画

### 3.1 学習者の打鍵履歴の活用

学習者の打鍵履歴の記録が可能なので、その履歴を元に、コードの入力状況を再生して振り返る機能の実装を計画している。実際のコード入力の順番からプログラム作成時の自身の問題点などに気がつくかもしれない。必ずしも、プログラムの先頭から入力していくのが最善ではない。逆に、その打鍵履歴から、自動的にアドバイスを生成する手法の開発も検討中である。演習授業の場合には成績をつける関係から個人で行う場合が多い。そのためコードレビューなどを行う機会がない。自学自習のために有効であると考えられる。

個人でプログラミングを行うのではなく、何らかのアプリケーションと一緒に(自動で)ペアプログラミングを行うなどすると初学者のプログラミングレベルを上げられると思われる。

### 3.2 演習授業中の困難者の発見

学習者の入力がない状態が一定時間続くことが検知できる。その場合は学習者が困難をかかえていることである。そうした状況を教員に通知する仕組みの導入も演習授業中には有効であると考えられる。CMSと連携して実現することが可能である。発見した困難者に対して、教員がどのような対応するのが良いかは、難しい問題である。

## 4. おわりに

本論文では、プログラミング初心者を対象としたプログラミング演習支援システムにおける学習者の進捗に応じたヒント提示機能の実現について述べた。

演習授業で実施を計画している。その実施結果から、設定したパラメータの評価や、継続的な使用による学習効果などを評価して予定である。

### 参考文献

- [1] 鈴木 孝幸, 野上 裕二, 納富 一宏, “個人適応型ヒント提示・誘導方式によるプログラミング演習授業の支援”, 電子情報通信学会技術研究報告, Vol.118, No.351, pp.9-14 (2018).
- [2] 鈴木 孝幸, 納富 一宏, “個人適応型ヒント提示・誘導システムを用いたプログラミング演習授業の実践”, 第17回情報科学技術フォーラム(FIT2018)予稿集, 第3分冊, pp.337-339[K-012] (2018).
- [3] 掛下 哲郎, 柳田 竣, 大田 恵介, “穴埋め問題を用いたプログラミング教育支援ツール pgtracerの開発と評価”, 情報処理学会論文誌教育とコンピュータ, Vol.2, No.2, pp.20-36 (2016).
- [4] 野上 裕二, 鈴木 孝幸, 納富 一宏, “プログラム空欄補充問題を用いたオンライン学習システムの運用と評価”, 第17回情報科学技術フォーラム(FIT2018)講演論文集, 第3分冊, pp.335-336 [K-011] (2018).