

## 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 Practice of Programming Exercise Lesson Using Personal Adaptive Hints Suggesting and Leading System

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

### 1. はじめに

神奈川工科大学情報学部情報工学科でのプログラミングの授業は、講義形態の授業と PC を用いて問題に対するプログラムを作成する演習授業を組み合わせ構成される。講義授業では、文法やアルゴリズムの解説・例題の実行などを行い、演習授業で問題解説・プログラム解説・操作説明を行いつつ、課題問題のプログラムを作成させる。

演習型授業では、プログラミング技能の習得状況や問題の理解度に個人差があり、学生はそれぞれ個別のペースで問題を解くので、授業時間内に解答できる問題数に差が出てしまう。特に初学者の場合、どこで躓いているかに合わせてヒントを提示することにより解答できる問題数が増加すると想定される。しかし、教員にとって個々の学生の進捗状況の把握が難しく、学生の習得しているレベル・進捗に合わせて、適切なヒントを適切なタイミングで提示することは、教員の経験と技量に依存してしまう。筆者らは、プログラミング演習講義のモデル化を行い、学生の学習進度に適切してヒント情報を自動的に提示するプログラミング演習講義支援システムの研究[1]を行ってきた。プロトタイプシステムでの試用であり、学習効果の測定などが今後の課題となった。

本研究では、プログラミングの初学者向け演習授業で、受講者の解答行動のモデル化を行うと共に、進度と躓きに適切した解答のヒントの自動提示機能を提案し、システムの試作と実践について報告する。

### 2. プログラミング演習授業のモデル化

#### 2.1 対象とする C 言語のプログラミング演習

本稿では、情報工学科 1 年生を対象とする C 言語のプログラミング演習授業を扱う。1 年生を対象とするので、C 言語の初学者が多い。有限の授業時間内に所定の題数の課題に回答しなければならないので、問題毎にある程度の制限時間が設定される。図 1 の様に横軸に解答時間をとり、縦軸にその問題の完成度をとると、制限時間内に完成度 100% まで達するとプログラム完了であり、完了までの時間が短いほど進捗良好（優秀）である。制限時間内に完了できずに時間切れとなった場合は、そのまま他の問題を解答するか、他の問題の時間を使ってその問題の解答を続けるかを選択しなければならない。後者の場合、ヒントを提示することにより制限時間内での完了への底上げが期待できる。また、図 2 のようにヒント提示により未回答の受講者が完了者になることは、教育的な効果が高い。ヒントについては、受講者のつまづきの程度により 1 つではなく、何段階かのレベルで提示できるのが理想である。

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

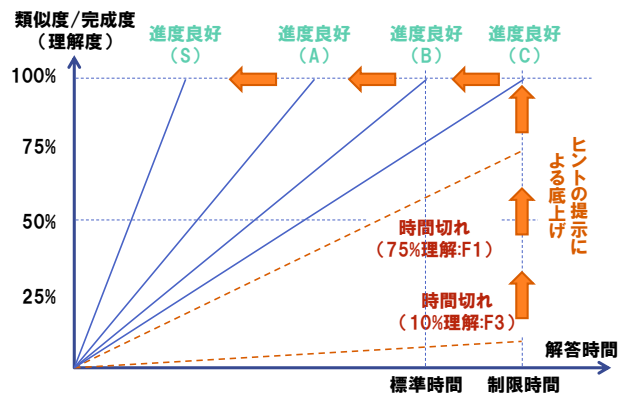


図 1 理解度や進捗の個人差とヒント提示の効果

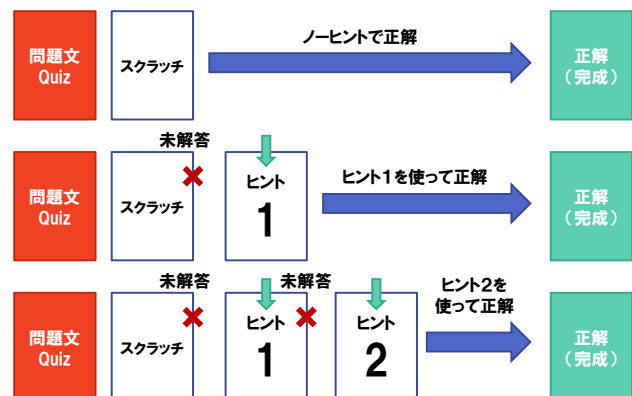


図 2 ヒントの提示による問題完成者の増加

#### 2.2 プログラム作成のモデル化

演習問題のプログラムを作成する手順をモデル化すると下記ようになる。初学者向けとして、簡単化している。

0. 問題文をよく読み、理解する
1. 必要なパーツ（文法事項・アルゴリズム等）を選ぶ
2. 選んだパーツを組み合わせる
3. 全てのパーツが動作するように設定する
4. 動作させる

このモデル化の下で、受講者がプログラミングできない理由を考えると下記ようになる。ここで、パーツとはプログラムの構文や制御構造のことである。

0. 問題文が読み取れない・誤読する・勘違いする
1. パーツを知らない
2. パーツの配置が分からない
3. パーツは配置できるが設定ができない
4. アルゴリズムを知らない
5. 何もしたくない

1.から 4.に対しては、適切なヒントを与えることで、改善が期待できる。

### 2.3 ヒントの種類

プログラミングできない理由に対して、どのようなヒントがあるかをまとめると下記ようになる。

0. 問題文を簡易な言葉で言い換え、例を多量に示す
1. パーツの働きや必要性を示す
2. パーツを選ぶ順番やパーツの働きを示す
3. パーツの設定の仕方 (パラメータの与え方) を示す
4. アルゴリズムの考え方や動作例を十分に提示する
5. 興味・関心が湧くような事例を説明する

1,2,3 のパーツに関連するヒントは、正解例となるソースコードから自動的に生成できる。学習者の学習履歴が蓄積された状態であれば、学習者レベルを設定して、それに適応したヒントをシステムが選択できる。

### 3. 個人適応型ヒント提示・誘導システムと実践

コードスニペット (code snippet) とは、コードのテンプレートを選択してから、カスタマイズすべき箇所を新たに書き換える方法であり、一部の開発環境で実現されている。初学者にとっては、便利な機能である。そこで、コードスニペットのように、初学者が 1 からコードを書くのではなく、ソースコードの一部を隠した空欄問題として導入する [2]。1 からコードを書くフルスクラッチ開発が最終目的であるが、初学者にとってはハードルが高い (図 3)。まずはソースコードの一部部分を空欄にして、そこに入る“パーツ”を考えてもらうこともヒントの提示手段であると考える。空欄補充問題であると解答の自動採点にも対応しやすくなる。

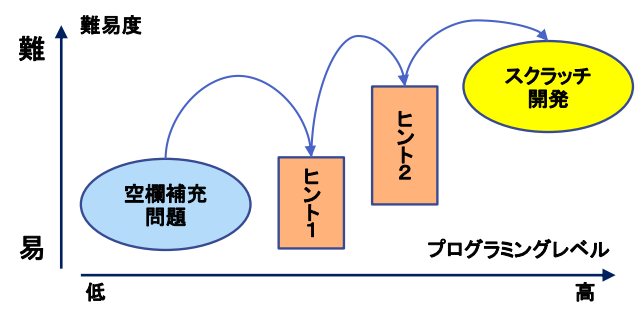


図 3 空欄補充問題からスクラッチ開発へ

また、正解例となるソースコードのファイルを指定すれば、プログラミング言語の予約語を元に空欄となる場所をシステムが自動的に設置でき、出題者の手間も軽減する。予約語に対して「何文字の単語」や「アルファベット p で始まる」など機械的に決まる事、予約語の説明となる用語をヒントとすることで、自動的にヒントを設定することができる。さらに、学習者毎の解答履歴の蓄積から習得レベルを推定し、レベルに応じたヒントを提示することで個人適応型となる。実際のプロトタイプシステムの実行画面が図 4 になる。

プロトタイプシステムを実際の演習授業の一部で試用し、ヒントの提示機能についてアンケートを行い、83 名から回答をえた (図 5)。「ヒントの提示が役に立つか」という質問に対しては「Yes」と「どちらかといえば Yes」を合

わせて 87.9%であった。「ヒントの内容が適切であるか」という問いに対しては、「Yes」と「どちらかといえば Yes」を合わせて 89.2%であり、概ね好評であった。

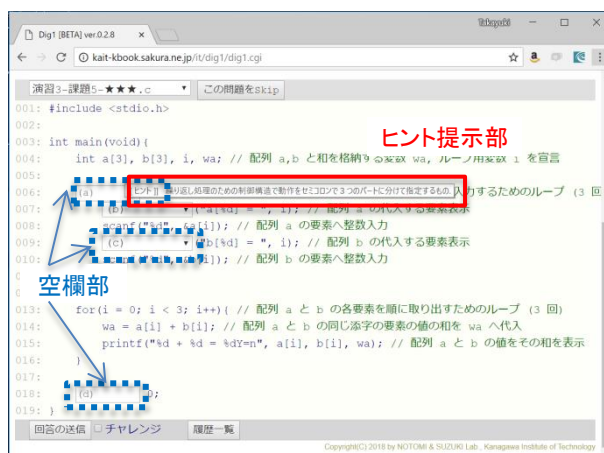
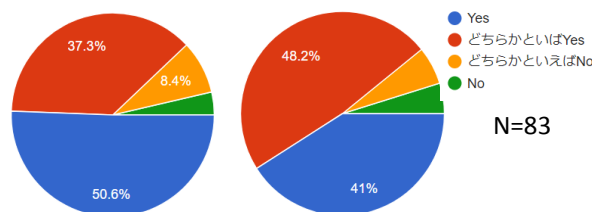


図 4 プログラム空欄穴埋め問題とヒントの提示例



ヒントは役立った ヒントは適切である

図 5 アンケート結果 (N=83)

今後、複数回の演習授業で継続的に試行する予定である。試行を重ねることで、学習履歴の蓄積量が増えて行くので、受講者レベルに適応しているかの評価を行うことができる。また、学習効果についての計測・評価についても予定している。

### 4. おわりに

本稿では、初学者に対するプログラミング演習授業で受講者のプログラミング作成とヒント提示についてモデル化し、空欄補充問題としてのプロトタイプシステムと実演習授業での実践の事例を示した。

今後の課題としては、長期間での運用による学習履歴の蓄積を行い、学習者レベルに適応したヒント提示機能の拡張と評価を行う予定である。ヒントの自動生成についてもソースコードだけでなく、コメント内での記述からも制御できるように拡張を行う。

#### 参考文献

- [1] 鈴木 孝幸, 納富 一宏, “学生の学習進度に適応したプログラミング演習講義支援システム”, 電気学会平成 29 年電子・情報・システム部門大会予稿集, GS8-3 (2017).
- [2] 野上裕二, 納富 一宏ら, “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開発と評価”, 電子情報通信学会 技術研究報告 Vol.117, No.335, 教育工学研究会 (ET), IEICE-ET2017-75, (2017).