

学習システムを用いたソースコーディングにおける
「つまずき」原因推定の検討
Experiment of cause estimation for 'stumbling' in source coding
with coding form learning system

橋本 浩規[†] 納富 一宏[†]
Kohki Hashimoto Kazuhiro Notomi

1. はじめに

筆者らは C プログラミング学習者をエンジニアレベルへ引き上げることを目的に、定型コーディングフォームを用いたプログラミング学習支援システムの研究を行っている。これまでに、学習者のプログラムの組み立てを支援する仕組みとして、プログラムをいくつかの構成要素（パート）に分け、各構成要素の記述場所を学習者に提示する定型コーディングフォームを設計した [1][3]。また、この仕組みを用いたシステムの各機能を検討した上で、プロトタイプ作成と評価を行った。さらに、各パートにおける関数定義や制御構造などに特化したフォームについても検討し、プログラミングを行いやすくなるかの評価を行った[2][3]。

先行研究では、アテンションとして記述内容の説明やコーディング時のポイントを提示し、インストラクションとして記述すべき文法のサンプルコードを提示している。空欄補充問題を利用した研究としては、野上らの、任意のソースコードをもとに決められた出題対象（例：関数定義、変数宣言、制御構造）に空欄を作成し、問題として出題するシステムの提案もある[4]。これらはフォームデータ内の同じヒントを全ての学習者に対して、コーディング状況を問わず提示している。プログラミング学習では、与えられたヒントや解説が、学習者が本当に知りたい内容と一致しているとは限らない。そのため、各学習者の「つまずき」原因を推定した上で、それに基づいた内容を提供することが望ましい。そこで本稿では、コーディング時に収集できる情報をもとに、つまずき原因の推定について、また各つまずきに応じたヒントの提示内容について検討する。そして、つまずき原因の選定や各ヒントに対するヒントが適切かどうかを調べる実験を行い、その結果について報告する。

2. 学習者のつまずき

2.1 つまずきの種類

学習者がコーディングする際、何をどのように記述したら良いかわからず、その先へ進めなくなる現象のことを、本稿では「つまずき」と定義する。学習者のつまずきとして考えられる原因の種類を表 1 に示す。

表 1 中の「段階」とは、コーディングを行う際のフェーズを示す。作成するプログラムの仕様を理解し、コーディングを行うフェーズを「作成時」、コーディングを一旦終えてコンパイルや実行を行うフェーズを「実行時」と表現している。本稿では、タイプ A～C を対象に検討を行う。

表 1 つまずきの種類

タイプ	原因	段階
A	何を記述するパートかわからない	作成時
B	構文や制御構造の書き方がわからない	
C	問題を解くアルゴリズムがわからない	
D	エラーの解消法がわからない	実行時
E	間違った実行結果になる	

2.2 つまずきの原因推定方法

原因推定への最初のステップとして、コーディングフォームの仕組みを活用する方法を考える。コーディングフォームは、それぞれの入力欄ごとにどのような内容が入るかについてのデータを保有している。そのため、ある入力欄でつまずいた場合、どのパートのどの要素を記述しようとしてつまずいているのかを推定可能である。推定の流れとクラス図を図 1 に示す。ヒントのタイプの推定については、タイプ A のヒントは最初から学習者に提示することで、推定の対象としない。タイプ B と C の推定については、構文に従って記述されているかや、入力・削除の状況といった行動の分析が必要であると考える。

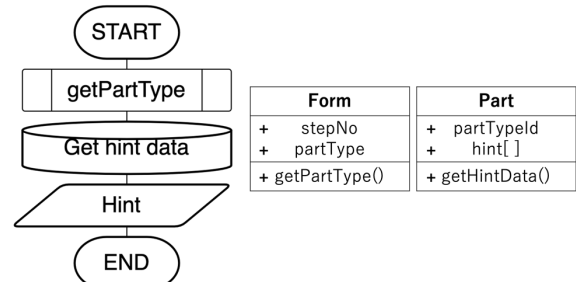


図 1 推定の流れとクラス図

2.3 原因に応じたヒント提示

つまずきの各タイプに応じて、適切な種類のヒントの提示を目指す。各タイプの提示内容について表 2 に示す。

表 2 各つまずきに対するヒント提示内容

タイプ	ヒント提示内容
A	何を書くパートなのかの説明文
B	構文や制御構造の説明
C	各パートの解説文または動画による説明

3. 学習支援システム

3.1 ヒント提示・フォーム画面

ここまで述べたヒント提示の仕組みを取り入れたヒント提示・フォーム画面は図 2 のようになる。画面左側には、

[†] 神奈川工科大学 Kanagawa Institute of Technology

現在コーディングしているパートに応じてつまずきの原因とそれに対応するヒントが表示される。画面右側には、各パートのコーディングを行うための入力欄が表示される。

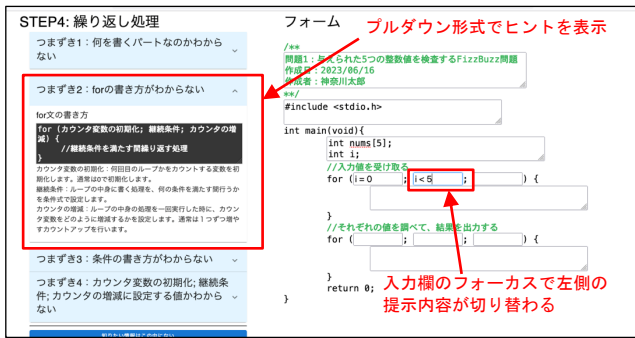


図 2 ヒント提示・フォーム画面

3.2 ヒントの自動生成と提示

システムに搭載するすべての問題において、解説やヒントを教員が作成することは効率的ではない。そこで、渡されたソースプログラムを解析し、解説とヒントの自動生成について考える。文法解説と、フォームの各パートの説明は事前にシステムに登録する。システムはソースプログラムを走査してプログラムのパートを構成し、フォームデータに格納する。その後、各パートに対するタイプ A および B のヒントをフォームデータから取得し、フォームデータ内に格納する。タイプ C については、ソースプログラムごとに解説を生成しフォームデータ内に格納する。また、学習者へのヒント提示手順を図 3 に示す。ここで、待機状態とはつまずきを検出するまでの状態を表す。

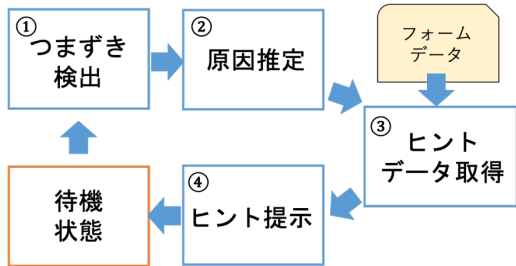


図 3 学習者へのヒント提示手順

4. 実験

初心者がつまずく原因として考えられるものとして、筆者らの選定が適切かどうか、また、それぞれの理由に対して、求められるヒントを提供できるかの判断に繋げるために実験を行った。対象は本学情報学部 4 年生 8 名である。方法は、サンプルのソースコードに対して、①筆者らが選定した初心者のつまずきの原因があり得るか、②筆者らが各つまずきに対して用意したヒントによってつまずきを解決できると思うかを調査した。用意されたヒントで解決できないと思う場合はその理由について尋ねた。①の調査に対する回答の集計結果をパートごとに集計したものを図 4 に示す。②の調査に対する回答の集計結果をパートごとに集計したものを図 5 に示す。

5. 考察・まとめ

①の結果、パート 2 (繰り返し) と 3 (条件分岐) の結果からはタイプ C の発生の可能性が高く見えるが、パート

①各タイプはつまずき原因としてあり得る

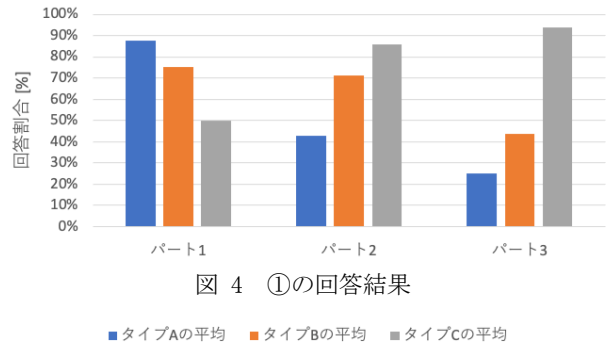


図 4 ①の回答結果

②各タイプに対するヒントは適切である

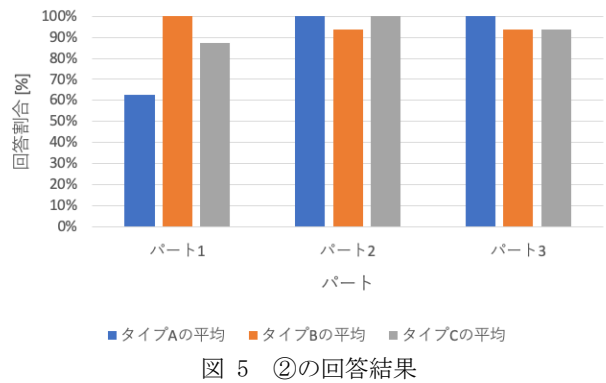


図 5 ②の回答結果

1 (プリプロセッサ命令) では傾向が異なっており、構文に対する慣れによってどの原因でつまずきが発生し易いかが変わる可能性が考えられる。これについては、実験するパートの数を増やし、さらなる実験を行う必要性も考えられる。一方で②の結果、どのパート・タイプでも適切であると答えた人が 6 割以上いたことから、つまずき原因に対するヒントの提示の仕方については適切である可能性が考えられる。

今後の課題としては、学習者のコーディング時の行動分析によって原因推定を行う方法について検討する必要がある。また、システム化に向けては解説とヒントの自動生成についても更なる検討の必要があると考える。

参考文献

[1] 橋本浩規, 納富一宏: "Cプログラミングにおける定型コーディングフォームを用いたスタイル学習支援システムの開発", 情報処理学会 第 21 回情報科学技術フォーラム(FIT2022)講演論文集 第 4 分冊, N-003, pp.239-240, (2022.09).

[2] 橋本浩規, 納富一宏: "プログラミング教育における初學者向け補助ツールの試作", バイオメディカル・ファジィ・システム学会 第 35 回年次大会講演論文集, F-3, (4 pages), (2022.12).

[3] 橋本浩規, 納富一宏: "定型コーディングフォームを用いたプログラミング学習支援システム -機能強化とユーザによる評価-", 情報処理学会 第 85 回全国大会講演論文集 第 4 分冊, 2ZJ-09, pp.697-698, (2023.03).

[4] 野上裕二, 納富一宏: "プログラミング学習支援における問題自動生成に関する基礎的検討", 情報処理学会 第 16 回情報科学技術フォーラム(FIT2017)講演論文集, 第 3 分冊, K-022, pp.465-466, (2017.09).