

失敗知識を利用したプログラミング学習における内省支援

Introspection Support in Programming Learning Using Failure Knowledge

野中美希[†]

Miki NONAKA

酒井 三四郎[†]

Sanshiro SAKAI

1 研究の背景と目的

プログラミング学習において、自分の失敗から学ぶことは多い。しかし、一度失敗したことを根本から理解しないままにしておく、次に同じ失敗を繰り返しても自分で解決することが難しく、定着しにくい。

そこで、よりスムーズにプログラミング技術を身に付けていくためには内省をすることが必要である。内省とは、自分の考えや行動などを深く省みることであり、本研究ではプログラミング学習における内省を、自分が書いたプログラムを振り返り、発生したエラーの原因や解決法を考えること、だと意味づけている。本研究の目的は、プログラミング学習において学習者に内省を促し、プログラミング技術の習得を支援することである。

2 関連研究と本研究の位置付け

一般的な学習において内省は重要であり、これはプログラミング学習においても有効である。このことに基づいて、プログラミング学習者に内省を行わせるための学習環境を構築する研究が行われている。

プログラミング学習において、学習者に失敗学 [4] に基づいた内省を行わせる研究がある [1]。この研究では、コンパイルエラー、実行エラー、論理エラーについて、失敗知識として「事象」「背景」「経過」「原因」「対処」「総括」を保存する。これらのうち前の3つは自動取得し、学習者は後の3つについて記述する。実行エラーと論理エラーについては記述を義務付けている。

関連研究 [1] では、学習者がエラーの原因について全く分からない場合は、失敗知識に「分からない」のように記述するしかないため、1人で学習している学習者は、そこで学習が行き詰まってしまう可能性がある。本研究ではそのような学習者に少しでも考えるきっかけを与えるために、「助言」を用意する。また、本研究では学習者にエラーの原因について考えさせることを重視しており、失敗知識の記述は考えたことの記録としての意味合いが強いため、記述内容をエラーの「原因」と「対処」のみとし、義務付けはしない。そして、システム利用の対象をプログラミング初心者としているため、記述対象はコンパイルエラーに限る。

また、電子化された教科書や問題集の利用、プログラムの編集、考察文入力などができる、学習環境を構築する研究がある [2]。この研究での考察文入力においては、コンパイルエラーのみを自動取得し、学習者はエラーの原因をリストから選択後、それについての考察文を入力する。

関連研究 [2] では、考察文を入力しなくてもエラーの原因を選択するだけで次の操作に移れる仕組みになっている。原因の選択のみの場合、学習者自身が考える前に答えのリストが出ていることになるので、内省がうまく行われな可能性はある。本研究では、エラーの原因から学習者に考えさせて

内省を促す。上で述べた「助言」と合わせて考え、エラーの「原因」と「対処」を記述することで内省を記憶に留めるという役割も果たす。

3 システムの提案

3.1 機能

提案するシステムの主な機能を以下に示す。本研究では、学習者が記述するエラーの原因と対処を合わせて失敗知識と呼ぶことにする。

- (1) 学習者のプログラムをコンパイルし、コンパイラから取得したエラーメッセージを表示
- (2) エラーメッセージごとに失敗知識を登録
- (3) エラーメッセージに対応した助言を表示
- (4) 登録した失敗知識を閲覧

3.2 支援対象

対象は、本学部1年時に受講するプログラミングの授業(初学者)レベルの学習者である。学習者は普段、MeadowとCygwinを使用してC言語プログラミングを学んでいる。

3.3 学習環境

対象とする学生は、本システムとは別のエディタとコンパイラを使用して学習している。これは、以前から授業で使用しているソフトウェアが決まっているためである。この方法では、これまでの学習環境に加えて本システムを使用してもらうため学習者に煩わしさを感じさせる可能性があるが、以前からの学習環境を変えることなく使用してもらうことで学習者にかかる負担は少ないと考えられる。そして、学習者自身の環境で解決できない問題にぶつかった際に、システムを利用して問題解決と内省を行うことを想定している。システムによって問題解決ができる情報が得られると、学習者個々の環境でプログラムを修正し、解決後にエラーの原因と対処を記述することになる。

3.4 システムの動作

図1にシステムの動きを示す。更に、図1の⑤の詳細を図2に示す。

学習者がソースコードをシステムにアップロードし、システム側でコンパイルを行う。システムがコンパイル時のエラーメッセージを表示し、学習者がエラーを選択すると過去に自分が登録した失敗知識やその時のプログラム、システムからの助言を見ることが出来る。助言は、システム開発の際にあらかじめ登録しておく。助言の対象は、今回は初学者がよく犯してしまいそうなエラー15項目に絞り、1つのエラーに対して2,3種類の助言を登録した。助言の内容は、断定的な表現ではなく問いかけ形式の内容とし、学習者に考えさせられるよう意識した。

[†]静岡大学大学院情報学研究科

[†]静岡大学情報学部

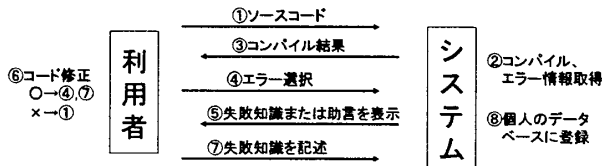


図 1: システムの動き

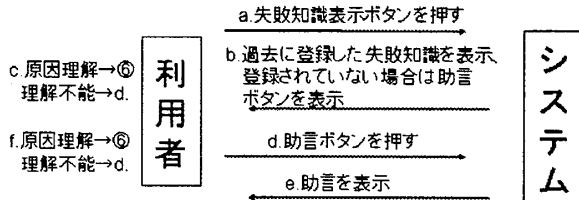


図 2: システムの動き (詳細)

3.5 ユーザインターフェース

図 3 に、システムのユーザインターフェースを示す。

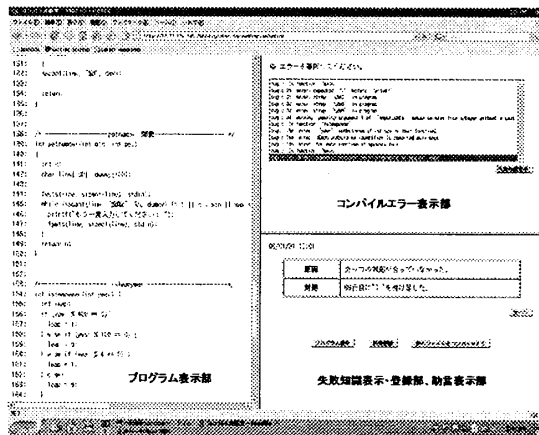


図 3: ユーザインターフェース

プログラム表示部には、コンパイルしたソースコードを表示する。また失敗知識を登録する際にはプログラムも共に保存するため、失敗知識閲覧の際に保存済みの過去のプログラム表示にもこの部分を使用する。

コンパイルエラー表示部には、コンパイル結果のエラーメッセージを表示する。

失敗知識表示・登録部、助言表示部は、それぞれの場合に利用する。エラー選択後に失敗知識表示ボタンを押すと、学習者が過去に登録したそのエラーに対する失敗知識が表示される。登録した失敗知識がない場合には、助言表示ボタンが追加され、助言表示ボタンを押すと助言が表示される。新規登録ボタンを押すと、失敗知識入力フォームが表示される。

4 評価について

4.1 実験方法

実装したシステムを、プログラミング初学者である本学部1年生5人に使用してもらい、その後アンケートによる評価を行った。被験者には、まず被験者自身が作成中のプログラムで使用してもらい、次に、用意した簡単なバグを含ませたプログラムで、プログラミングを始めたばかりの頃を想定して使用してもらった。

4.2 実験結果

実験後のアンケート調査から以下の結果を得た。

画面の配置については、Webアプリケーションであるため、個人ごとのWebブラウザやPC画面の大きさによっても表示のされ方が異なってしまい、操作しづらいという評価もあった。あらかじめ登録しておいた助言は、5人中4人からは被験者自身が作成中のプログラムにおいても表示され、エラー解決の助けになったという評価を得た。また、バグ入りのプログラムで使用した際には、初学者にとってもエラー解決の助けになりそうだという評価が得られ、内省の動機付けになっているといえる。ただ、原因がはっきり分かるまで内省を行えるかは個人の意欲に左右されそうである。失敗知識登録への負担に関しては、内省できていればほとんど負担にはならないという評価を得た。

5 まとめと今後の課題

本研究では、本システムが学習者に内省をさせるきっかけを作ることができたことが分かった。しかし、原因不明のエラーに対して助言が表示されない場合、学習者はその時点で諦めてしまい、内省を必ず最後まで行うことが難しいということも分かった。また、今回の実験では学習者に知識が身に付いたかどうかを確認できていない。

学習者が理解できるまで内省させるためのアプローチとしては、あらかじめ教材を用意しておき、そのエラーに関連した教材を提示するというのを考えている。

今回の実験結果では被験者が5人と少なかったため、今後システムの改良を加えた後、本学部1年生100名程度を対象に再実験を行うことを考えている。システムの改良点としては、主に助言を充実させて対応するエラーを増やすことである。学習者に知識が身に付いたことを確認するためには、以前は失敗知識を記述または参照していたエラーが発生した際、何も参照することなく自分で解決できたかどうかを把握する必要がある。次回の実験では、このことを確かめられるよう進めていく。

また、現在はコンパイルエラーに対処できない学習者を対象としているが、実行エラーや論理エラーについてはどのように内省をさせるかが課題となっている。

参考文献

- [1] 知見邦彦, 樋山淳雄, 宮寺庸造, “失敗知識を利用したプログラミング学習環境の構築”, 電子情報通信学会論文誌, Vol J88 - D-I, No 1, pp.66~75, 2005
- [2] 高橋参吾, 松永公廣, “プログラミング学習のための電子学習環境の構築”, 日本教育工学会論文誌, Vol 23, No 3, pp.155~165, 1999
- [3] 古市多華子, “gccのエラーメッセージの日本語化”, 滋賀職業能力開発短期大学卒業論文, 1994
- [4] 畑村洋太郎, “失敗学のすすめ”, 講談社, 2000