

プログラミング学習における失敗知識を利用した内省支援*

野中美希

静岡大学大学院情報学研究科†

酒井三四郎

静岡大学情報学部‡

1 研究の背景と目的

プログラミング学習において、自分が犯した失敗によってそれまで理解できていなかった事柄に気づき、理解が深まることは多い。しかし、自分の失敗から学習するためには、失敗を振り返りその原因を分析し理解する作業 (内省) が必要である。プログラミング初学者は、知識と経験が乏しいため失敗の原因を突き止められない場合が多く、内省を自分で行うことが難しい。このような場合、学習者は根本的な理解をしないままとなり、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、本研究ではこのような学習者に内省のきっかけを与えて促すシステムを開発し、プログラミングの学習支援を行うことを目的とする。

2 関連研究と本研究の位置付け

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。このうち学習者は後の 3 つを記述する。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、学習環境を構築する研究がある [2]。

関連研究 [1] では、コンパイルエラー以外は失敗知識の記述を義務付けているが、本研究では学習者が自ら内省できるよう義務付けはせず促していく。関連研究 [2] では、考察文が未入力でもエラーの原因をリストの中から選択すれば登録できる仕組みになっているが、本研究では原因と対処を学習者に記述させて内省を促す。その際、原因が分からず行き詰まる学習者を減らすために、解決のための助言を与える。

また、関連研究 [3] で開発されたテスト時の入出力の観察によりプログラミング学習支援を行うシステムを、

学習者に自分が犯した論理エラーに気づかせるために、本研究の一部の機能 (後述のプログラムチェック機能) として組み込む。

3 システムの提案

3.1 支援対象

対象者は、本学部 1 年次に受講するプログラミングの授業 (初学者) レベルの学習者とする。学習者は普段、テキストエディタと Cygwin 上でコンパイラを使用して C 言語プログラミングの演習を行っている。対象とするプログラムは、1 つのファイルに必要なモジュールが全て記述されており、標準入出力を用いるものに限る。

3.2 システムの特長

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。本システムは Web アプリケーションとして実装しており、全ての機能 (プログラムの編集、コンパイル、実行、失敗知識の登録・閲覧、プログラムの動作チェック) をサーバ上で提供する。本システムの特長は以下の 4 点である。

- 学習者個人ページから個人々々の失敗知識をいつでも閲覧でき、内省の機会を増やすことができる
- コンパイルエラー解決のために助言を参照できる
- エラー解決時にはシステムが自動で検出し、学習者に失敗知識の登録を促す。この時、失敗知識の登録がプログラミング作業を中断させることを防ぐために、今すぐ登録するか、後で登録するか、以前と同じ原因を登録するかを選択できる
- プログラムチェック機能により論理エラーに気付くことができる

3.3 システムの利用方法

図 1 に学習者のシステム利用の流れを示す。

1. ログイン後、プログラムを編集
2. システム上でコンパイル (コンパイルエラーがあれば 3. へ、なければ 5. へ)
3. 発生したエラーに対応する失敗知識 (学習者が過去に登録したもの) や助言を参照
4. プログラムを修正後再びコンパイルすると、解決したエラーをシステムが自動検出し、学習者がそのエ

* Reflection Support Using Failure Knowledge in Programming Learning

† Miki NONAKA, Graduate School of Informatics, Shizuoka University

‡ Sanshiro SAKAI, Faculty of Informatics, Shizuoka University

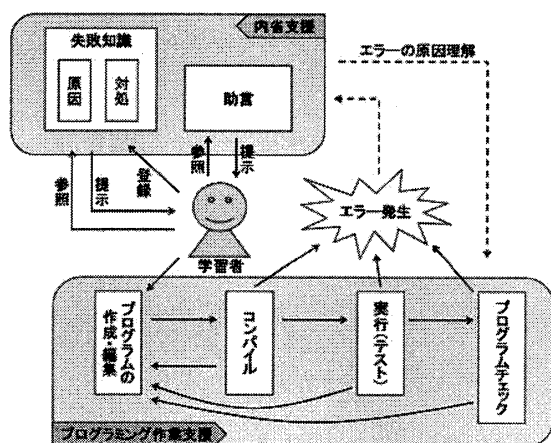


図 1 システム利用の流れ

ラーに対する失敗知識を登録

5. コンパイルエラーがなくなった後、システム上でプログラムを実行、テスト
6. プログラムチェック機能を用いて、プログラムの論理エラーをチェック (論理エラーがあった場合、コンパイルエラーと同様に 2.~3. を繰り返す)

上記 3. の助言は開発者があらかじめ登録しておく。また、6. では開発者が用意した課題ごとの模範プログラムとテストケースによって判定を行う。

3.4 ユーザーインターフェース

ログイン後の学習者個人ページには、プログラミング作業画面、過去に登録した失敗知識、失敗知識登録を保留にしてあるエラーの関連情報へのリンクを表示する。

プログラミング作業時には、図 2 のようになる。

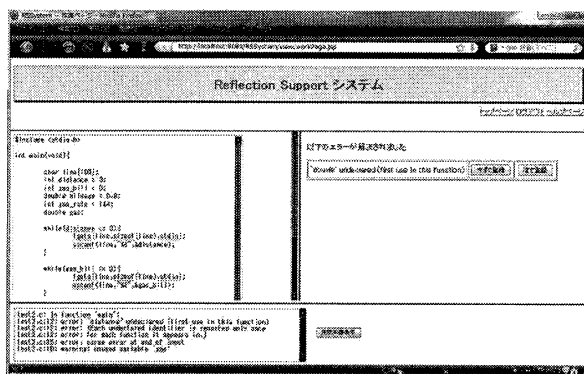


図 2 プログラミング作業時の様子

4 評価実験結果と考察

4.1 実験方法

本学部の「プログラミング」を受講している 1 年生 12 名に 3 週間システムを使用してもらい、実験終了後にア

ンケートを行った。被験者のプログラミング経験は 2 ヶ月程度である。実験中はシステムのログを取り、学習者の使用状況を観察した。なお、システムの使用は強制せず、使用頻度は被験者である学生の自由とした。

4.2 実験結果と考察

被験者 12 名のうち 7 名からアンケートを回収した。実験中に登録された失敗知識は延べ 6 件で、全てコンパイルエラーに対するものであった。3.2 で述べたシステムの特長に沿ってその効果を考察する。

1 つめの学習者個人ページから過去に登録した失敗知識を閲覧できる機能については、12 人中 6 人が 1 回以上閲覧していた。失敗知識の登録が少ないため内容がないページを閲覧した場合も含まれるが、その中でもこの機能によって失敗を振り返ることができたケースがあったので、内省の機会を増やすことができたと考えられる。

2 つめの助言について、これを参照した 4 名中 4 名から役に立ったという回答を得たので、行き詰まりを防ぐ助言は初学者には必要であるといえるだろう。

3 つめの特長について述べる。システムから自動検出された解決済のエラーに対して失敗知識を登録する際、「今すぐ登録」が 5 件、「後で登録」が 1 件、「以前と同じ」が 1 件という内訳であった。「後で登録」機能については、プログラミング作業の中断を防ぐことができたという回答を得たが、その後失敗知識が登録されなかったため内省も行われなかった可能性がある。

最後のプログラムチェック機能によって、論理エラーに気付く修正できたケースがログから見られたが、役に立ったという回答はこの機能の使用者 6 人中 2 人であった。この機能は、標準出力による判定を行っているため出力に制限が必要な場合があり、被験者を混乱させた可能性があるためこのような結果になったと考えられる。

本研究ではシステムの使用を強制しなかったため、被験者の中でも使用頻度に偏りがあった。アンケートとログから、プログラミングが得意な学生の方が内省に意欲的だと分かったが、僅かな例でも不得意な学生の支援ができたので、今後の研究の参考になる結果が得られた。

参考文献

- [1] 知見邦彦, 樫山淳雄, 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] 高橋参吉, 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] 松本真吾, 野中美希, 太田剛, 酒井三四郎, “教師が作成したテストケースを用いたプログラムの正誤判定によるプログラミング学習支援システム”, 教育システム情報学会誌, Vol.26, No.1, pp.29-35, 2009