

初学者に体系的デバッグ手順を指導する学習支援システムと 講義パッケージの設計

Design of a Learning Support System and Lecture to Teach Systematic Debugging Process to Novice Programmers

山本 頼弥^{*1}, 野口 靖浩^{*2}, 小暮 悟^{*2}, 山下 浩一^{*3}, 小西 達裕^{*2}, 伊東 幸宏^{*4}

Raiya YAMAMOTO^{*1}, Yasuhiro NOGUCHI^{*2}, Satoru KOGURE^{*1},

Koichi YAMASHITA^{*3}, Tatsuhiro KONISHI^{*1}, Yukihiro ITOH^{*4}

^{*1} 静岡大学創造科学技術大学院, ^{*2} 静岡大学情報学部

^{*1} Graduate School of Science and Technology, Shizuoka University

^{*2} Faculty of Informatics, Shizuoka University

^{*3} 常葉大学経営学部, ^{*4} 静岡大学

^{*3} Faculty of Business Administration, Tokoha University ^{*4} Shizuoka University

Email: dgs15009@s.inf.shizuoka.ac.jp

あらまし：本研究では、プログラミング初学者が体系的なデバッグ手順を学習するための講義パッケージを設計した。この講義パッケージは、(1) デバッグプロセスとそのプロセスを実現するためのデバッグスキルの両方を学習可能である、(2) 講義・演習問題・学習支援システムがセットになっている、という特徴を持つ。本稿では、この講義の指導内容・指導方法と学習支援システムの機能について述べる。

キーワード：プログラミング教育支援, デバッグ学習支援, 学習環境

1. 研究の背景・目的

我々は先行研究で場当たりのデバッグを行う初学者に対し体系的なデバッグプロセスの学習を支援するシステムを開発した⁽¹⁾。しかし、評価実験の結果によれば全ての被験者がプロセスを部分的にしか学習できなかった。我々は(1)デバッグプロセスの実行に必要なスキルを被験者が身につけていなかったこと、(2)デバッグ演習のみを通じてプロセスを学ぶのは難易度が高かったこと、が主な原因と考えた。

原因(1)について、デバッグプロセスの実行には命令文の動作を評価するスキルや観察する命令文を決定するスキルなどが必要である。そこで、本研究ではデバッグプロセスとデバッグスキルの両方を指導する手法を提案する。

原因(2)について、先行研究では演習のみから発見学習的にデバッグプロセスを習得可能と考えていた。しかし、この方法での学習には長時間を要することが明らかになったため、我々はデバッグプロセスに関する講義を演習に組み合わせることとした。そこで本研究では講義・演習問題・学習支援システムからなる講義パッケージを設計する。

本稿では、以上の議論をもとにデバッグ学習のための講義と学習支援システムを設計する。

2. 講義設計

我々はこれまでに、演習問題を含む講義シナリオを以下のように設計した。まず、学習目標として図1のプロセスを設計した。プロセス内の各タスク(図1の長方形)ではA~Eの5つのスキルが必要となる。複数関数からなるプログラムのデバッグには5つのスキルすべてが必要だが、メイン関数のみのプログ

ラムのデバッグは関数内部をチェックするためのスキル A, B, C があればよい。このことから、1 回目の講義ではスキル A, B, C を指導し、2 回目の講義で残りのスキルを指導する。

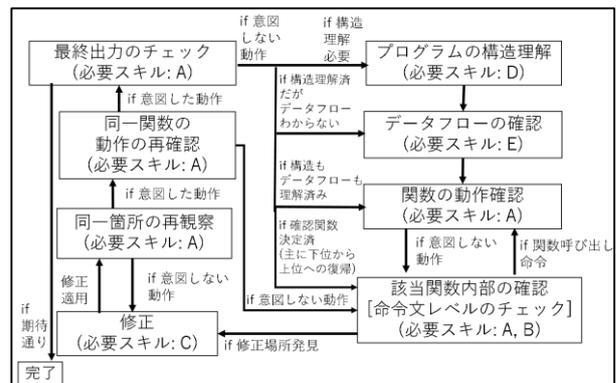


図1 学習目標であるデバッグプロセス

以下、各スキル及びそれらの(サブスキル)と[サブスキル学習のための講義内容]について述べる。

スキル A: 着目した命令文を評価するスキル

(A-1): その命令文の期待する動作を想起

[この想起が必須であるということを指導]

(A-2): その命令文の実際の動作を観察・確認

[観察のための出力命令の挿入方法を指導]

スキル B: ソースコード内の正しく動く(正常)範囲とそうでない(異常)範囲を分けるスキル

(B-1): 観察すべき命令文を決定

[コードの上から順に観察する単純な手法を指導]

(B-2): スキル A を用いて命令文を評価し、正常・異常範囲の区分けを行う

[観察結果に基づき範囲を区分けするよう指導]
スキルC: 命令文を修正するスキル

[一般化が困難なため、修正の典型例のみ示す]
スキルD: プログラムの構造を整理するスキル

(D-1): コード内の関数をリストアップ
(D-2): 呼び出し関係を再現した構造図の作成
[いずれも、関数に関する文法と手順がわかればできるので、その2つを理解するように指導]

スキルE: 構造図でデータフローを確認するスキル
(E-1): 次に実行される関数呼び出し命令を選択
(E-2): 構造図上でデータフローを辿る
[スキルDと同様]

3. 学習支援システム的设计

我々はデバッグプロセス・スキル学習のために3つのワークシート(WS)を提供する。WSは図1の各タスクに対応する作業エリア・ボタンからなる。学習者はWS上での作業を通じて、図1のデバッグプロセスを正しい手順で体験する。また、学習者の作業はシステムにより評価され助言が生成される。これにより、スキルを正しく身につけることができる。

3.1 単一関数内のデバッグのWS (WS0)

学習者はWS0(図2)で単一関数内のデバッグプロセスとスキルA, B, Cを学習する。このWSは開発中である。システムは学習者に、これまでに評価済みの命令を表示し、次に評価すべき命令を指定するよう促す。学習者は評価する命令文をエリア(ii)で選択する(スキルB-1の訓練)。するとシステムは、該当命令文の期待される動作を想起してエリア(v)に入力するよう促す。学習者はこれを行う(スキルA-1の訓練)。次に学習者は、エリア(ii)に出力命令を挿入し、評価すべき命令の動作の観察をする(スキルA-2の訓練)。学習者は、動作が正しいか否かにより、評価した命令が正常・異常範囲のどちらに属するかを判断する(スキルB-2の訓練)。

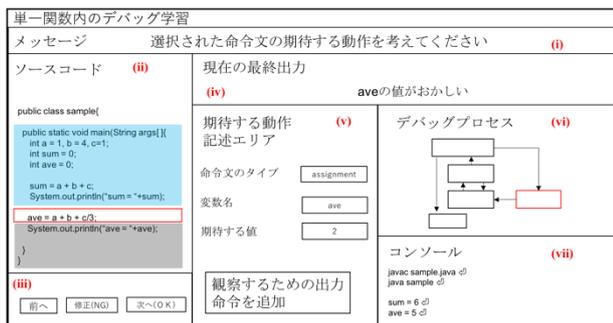


図2 WS0のインターフェイス

3.2 プログラムの構造理解のWS (WS1)

学習者はWS1(図3)で、図1の「プログラムの構造理解タスク」とスキルDを学習する。WS1は先行研究で構築したものを流用する。WS1は3つのボタンと1つのエリアをもつ。ボタン1は関数リストアップ作業を想起させる(スキルD-1の訓練)。これを押しすと各関数に対応した円形オブジェクトがエリア

2に現れる。エリア2は学習者に構造図の作成作業を想起させる。学習者は構造図を作成し(スキルD-2の訓練)、その後ボタン3を押し正誤判定を受ける。

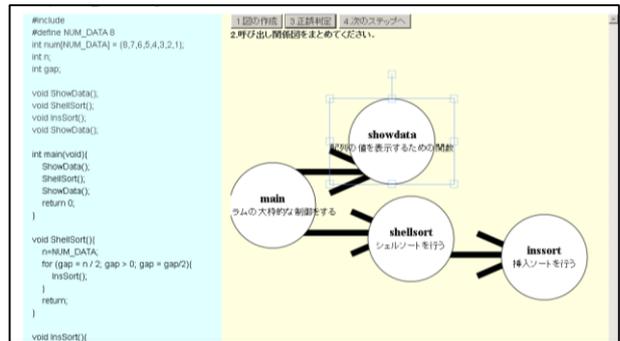


図3 WS1のインターフェイス

3.3 データフロー確認のWS (WS2)

学習者はWS2(図4)で、図1の「データフローの確認タスク」とスキルEを学習する。このWSは開発中である。このWSは2つのエリアをもつ。エリア1で次に実行される関数呼び出し命令の選択作業を想起させる。学習者はエリア1で命令を選択する(スキルE-1の訓練)。エリア2は構造図上でデータフロー確認作業を想起させる。エリア2では、ソースコードで確認したデータフローの順序に対応した数字が構造図上に現れ、学習者はこれらを確認してデータフローをトレースする(スキルE-2の訓練)。学習者は作業後ボタン3を押し正誤判定を受ける。

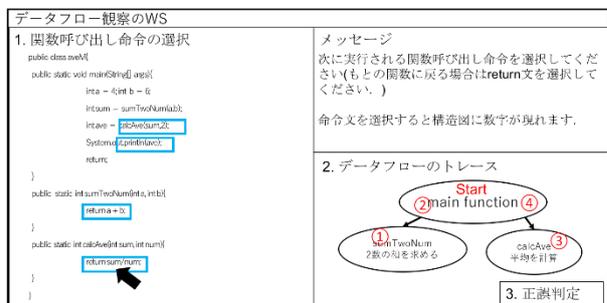


図4 WS2のインターフェイス

4. まとめ

本稿では、先行研究で構築したデバッグプロセス学習支援システムにおける問題点を検討し、それに対する解決策として講義と学習支援システムを設計した。設計した講義と演習については実際に講義をし、予備評価を行う(演習で教師は設計したシステムの挙動を再現しながら指導を行う)。そして、予備評価の内容に基づき講義パッケージの調整を行う。その後システムを実装し、講義パッケージを評価する。

参考文献

- (1) R. Yamamoto, Y. Noguchi, S. Kogure, K. Yamashita, T. Konishi, and Y. Itoh : "Construction of an Environment to Support Learning Systematic Debugging Process with Worksheets and Synchronized Observation Tool", Proceedings of ICCE2015, pp.269-274 (2015)