

コンパイルエラー修正履歴一覧システムの開発と評価

Development and Evaluation of
Compile Error Corrected History Compendium System平尾 元紀^{*1}, 酒井 三四郎^{*2}
Motoki Hirao^{*1}, Sanshiro Sakai^{*2}

*1*2 静岡大学情報学部

*1*2 Faculty of Informatics, Shizuoka University

あらまし：コンパイルエラーに対して恐怖感を持っている学生のため、コンパイルエラー修正時間推移を一覧できるシステム CocoViewer (Compile error Collection Viewer) を開発した。文化系の大学生に対し、評価実験を行ったところ、41%の被験者がコンパイルエラーの恐怖感が軽減されたと回答し、部分的な効果が確認できた。

キーワード：コンパイルエラー，プログラミング学習，恐怖感軽減

1. はじめに

コンパイルエラーは多くのプログラミング学習者が最初に躓く問題である。静岡大学情報学部社会学科1年次の「プログラミング」の講義を受講している文化系の大学生に対してアンケートを実施したところ、61%の学習者がコンパイルエラーへの恐怖感を持っていると回答し、そのような学習者はコンパイルエラー修正に躓いてプログラミング作業を苦痛に感じている。コンパイルエラーへの恐怖感を軽減する方法として、コンパイルエラー修正時間の推移を可視化し、学習者がその推移を分析することが考えられる。

そこで、本研究ではコンパイルエラー修正時間推移を一覧できるシステムを開発し、評価実験を行った。

2. 先行研究

コンパイルエラーを短い時間で修正できるよう学習者を手助けするシステムがある⁽¹⁾⁽²⁾。しかし、学習者が自らのコンパイルエラー修正時間の推移を知ることができない。本研究ではこの点に着目し、学習者が自らのコンパイルエラー修正時間の推移を分析なシステムを開発する。

3. システムの提案

3.1 システムの概要

静岡大学情報学部社会学科1年次の「プログラミング」の講義で利用されている統合開発環境「論プロエディタ」の拡張として CocoViewer (Compile Error Collection Viewer) を開発した。CocoViewer の開発言語は Java である。コンパイルエラー修正情報は、学習者が作成したプログラムファイルの全差分データを再コンパイルし収集している。

3.2 修正時間の分析方法

図1に修正時間推移グラフの例を示す。修正時間推移グラフは X 軸をコンパイルエラー修正回数、Y

軸をコンパイルエラー修正時間とした折れ線グラフである。学習者は修正時間推移グラフの概形を見て、自分のコンパイルエラー修正を分析できる。修正時間推移グラフはグラフの概形を右下がり型、振動型、右上がり型の3パターンに分けられる。



図1 右下がり型修正時間推移グラフ

図1は右下がり型修正時間推移グラフの例である。修正回数が増えるにつれ、修正時間が短くなることから、このコンパイルエラーの修正方法を理解していると推測できる。

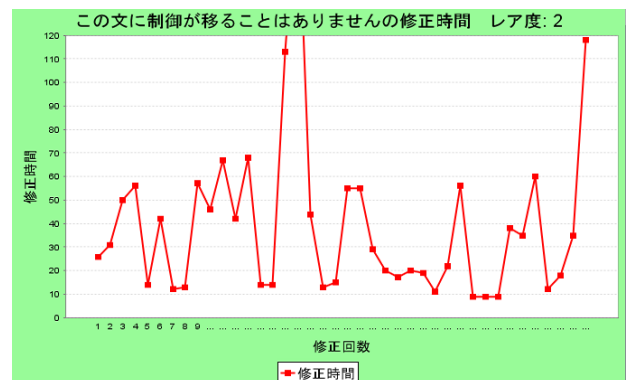


図2 振動型修正時間推移グラフ

図2は振動型修正時間推移グラフの例である。修正回数が増えても修正時間が長くなることから、このコンパイルエラーの修正方法の理解が不完全であり、修正方法を学習する必要があると推測できる。



図3 右上がり型修正時間推移グラフ

図3は右上がり型修正時間推移グラフの例である。修正回数が増えるにつれ、修正時間が長くなっていることから、振動型よりもこのコンパイルエラーに関しては理解が不完全であり、修正方法を学習する必要があると推測できる。

3.3 システムの特長

本システムの特長は各コンパイルエラーの修正時間推移グラフをゲーム感覚で一覧できることである。修正時間推移グラフ一覧表を図4に示す。



図4 修正時間推移グラフ一覧表

ゲーム感覚でコンパイルエラー修正を振り返るためのデザインとして、コンパイルエラーの種類が発生頻度によるレア度を設定した。修正時間推移グラフの背景色はレア度を示している。コンパイルエラーの種類が発生頻度は、榊原らの先行研究⁽³⁾から2010年度の静岡大学情報学部社会学科1年次の「プログラミング」の講義で集計したコンパイルエラーの修正数から求めている。学習者が修正経験のないコンパイルエラーの種類に関しては、そのエラーメ

ッセージを表示することで、学習者がコンパイラエラーを集めやすくしている。

これらのデザインにより、学習者がコンパイルエラーを集めるゲーム感覚で、コンパイルエラー修正時間の推移で、学習者のコンパイルエラーに対する恐怖感を軽減することを狙いとしている。

4. 評価実験

4.1 仮説

本実験の仮説は以下のとおりである。

仮説 CocoViewer を使うことで、学習者のコンパイルエラーへの恐怖感が軽減される

4.2 実験概要

静岡大学情報学部社会学科1年次の「プログラミング」の講義を受講している文化系の大学生98名に対し、第10回講義時のチャンス問題としてCocoViewerの評価実験を実施した。実験を始める前に、筆者が修正時間推移グラフの読み取り方についての説明を行った。被験者はワークシートにそってCocoViewerを操作し、アンケートに答えた。実験中はワークシート及びシステム操作に関する質問を随時受け付けるものとした。

この講義で扱われている言語はJavaで、受講者は全員同じエディタ、コンパイラを使用している。コンパイラはOracle社が提供するJava Development Kitに含まれているJavacコンパイラで、バージョンは1.7.0_45である。

4.3 実験結果

61名から有効なデータを得た。

「この演習を通して、コンパイルエラーへの恐怖感が軽減されましたか?」という質問に対し、41%の被験者がコンパイルエラーに対する恐怖感が軽減されたと回答した。自由記述欄では「この演習を一通りやるだけでコンパイルエラーへの恐怖感が激減した」という意見があった。被験者の中には友達同士で修正時間グラフ一覧表を見せ合っ、レア度の高いコンパイルエラーの発生を競い合い、楽しんでコンパイルエラー修正を分析する様子が見られた。

これらの結果より、仮説は部分的に支持される。

参考文献

- (1) Björn Hartmann, Daniel MacDougall, Joel Brandt, et al. : “What Would Other Programmers Do? What Would Other Programmers Do?”, Proceedings of the SIGCHI Conference on Human Factors in Computing System, pp.1029-1028 (2010)
- (2) 榊原康友, 松澤芳昭, 酒井三郎: “コンパイルエラー修正時間に着目した学習分析指標の提案と内省学習効果分析への適用”, 研究報告コンピュータと教育(CE), Vol.2013-CE-118, No.8, pp.1-8 (2013)
- (3) 榊原康友, 松澤芳昭, 酒井三郎: “プログラミング初学者におけるコンパイルエラー修正時間とその増減速度の分析”, 情報処理学会シンポジウム論文集, Vol.2012, No.4, pp.121-128 (2012)