

# 学生の演習課題とオープンソースソフトウェアの コード品質比較の試行

阿部 修也<sup>1,a)</sup> 氏原 直哉<sup>1,b)</sup> 酒井 亮太郎<sup>1,c)</sup> 幸 佑亮<sup>1,d)</sup> 石尾 隆<sup>1,e)</sup> 井上 克郎<sup>1,f)</sup>

**概要:** 大学におけるプログラミングの授業では、学生は演習課題を通じてプログラミング技術の修得を進めていく。しかし、学生のプログラミングの技量を試験やレポートに基づく成績評価によって知ることは困難である。本研究では、C言語でシェルを作成するという課題に対して学生が作成したプログラムと、`bash` や `zsh` などの著名なオープンソースのシェルのソースコードについて、関数の長さや条件分岐の数などの品質指標に関する比較を行うことで、学生のプログラミングの技量の評価を試みた。

**キーワード:** プログラミング教育, ソフトウェアメトリクス, 実践報告

## 1. はじめに

大学のプログラミングに関する授業において、「ただ動くプログラムではなく良いプログラムを書くべきだ」というような内容はよく話される。しかし、学生が具体的にどのような点を意識すればより良いプログラムを書くことができるのかは明らかではない。たとえばコメントを積極的に記述するようといった行動指針が与えられても、具体的にどの程度の量が良いのかを学生が知ることは難しい。本研究では、「良いプログラム」と学生のプログラムの差を調べることによって、学生がプログラムの記述において注意すべき点を明らかにすることを試みた<sup>\*1</sup>。

## 2. 学生のプログラムとOSSの比較

本研究では、良いプログラムの見本としてオープンソースソフトウェアを選択し、学生のプログラムと比較を行う。良いプログラムは、適度にモジュール化され、平易に書かれた読みやすいプログラムであると仮定し、プログラム中の関数ごとのステートメント数、ソースファイルごとのコ

メント行の割合、条件分岐行の割合という3つのメトリクスを計測、比較する。

学生のソースコードは、著者らの所属する大阪大学基礎工学部情報科学科の3年生に依頼し収集した。授業科目では1年生からC言語の演習があるが、2年生前期の演習で作成したシェルプログラム13人分を収集した。

演習課題がシェルを作成するというものであったことから、比較対象であるオープンソースソフトウェアとしてC言語で書かれたシェルを収集した。具体的には、`ash`, `ast-ksh`, `bash`, `zsh`, `sh`, `tcsh`, `rsh` である。

関数ごとのステートメント数は、「いちやもん [1]」によって計測できる関数ごとの“statement number”の値である。また、ソースファイルごとのコメント行の割合および条件分岐行の割合は Source Monitor [2] によって計測できる“% Branches”および“% Comments”の値である。

関数ごとのステートメント数の分布を図1に箱ひげ図で示す。図において“students”が学生の、“OSS”がオープンソースのシェルの数値である。学生の関数ごとのステートメント数は平均で25.54とOSSの20.56よりも大きい。この差はWilcoxonの順位和検定を行うと $p=0.001004$ となり、有意水準5%で有意であることを確認した。なお、図1では、統計的に有意な差があるにも関わらず平均値や中央値には大きな差が認められない。これはOSS側に多数存在する巨大な関数が平均値や中央値に影響を与えているためだと考えられる。

ソースファイルごとのコメント行の割合の分布を図2に示す。学生のコメント行の割合は平均で25.68%とOSSの16.38%より大きく、Wilcoxonの順位和検定を行うと

<sup>1</sup> 大阪大学  
Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

a) u141594c@ecs.osaka-u.ac.jp

b) u848220d@ecs.osaka-u.ac.jp

c) u868592a@ecs.osaka-u.ac.jp

d) u052458g@ecs.osaka-u.ac.jp

e) ishio@ist.osaka-u.ac.jp

f) inoue@ist.osaka-u.ac.jp

\*1 本稿は大阪大学基礎工学部情報科学科の授業科目「情報科学ゼミナールA」において、学部3年生である著者4名が自主研究課題として実施した内容の報告である。

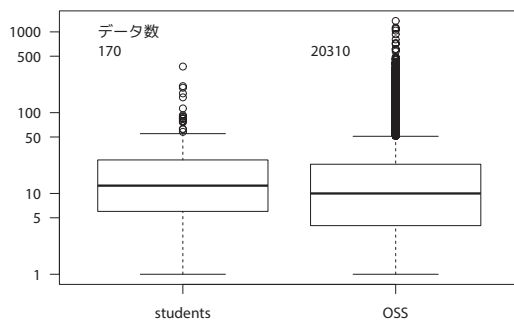


図 1 関数ごとのステートメント数の分布

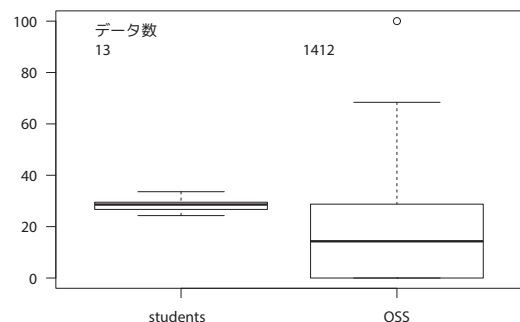


図 3 ソースファイルごとの条件分岐の割合 (%) の分布

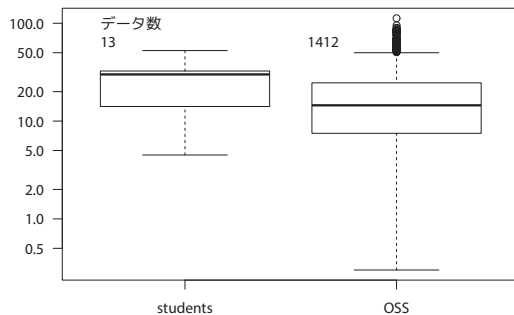


図 2 ソースファイルごとのコメント行の割合 (%) の分布

$p=0.005003$  となり有意水準 5% で有意差が認められた。これは演習課題においてコメントを書くことが奨励されていた影響と考えられる。

ソースファイルごとの条件分岐の割合の分布を図 3 に示す。条件分岐の割合は、OSS の平均 15.47% に比べて学生は 28.63% と非常に大きく、プログラムを必要以上に複雑に記述している可能性がある。この差についても、Wilcoxon の順位和検定を行うと  $p=0.000662$  となり有意水準 5% での有意差が認められた。図 3 の OSS での値が比較的散らばっている理由としては、多種多様な機能を様々な開発者が実装したためプログラムの構造や書き方に違いが出やすいことが考えられる。これに対し、学生のプログラムでの値は中央値付近に固まっている。その理由としては、授業課題として最低限実装すべき機能が指定されており、基幹となる処理のループ構造が演習資料として与えられていたためにプログラムの構造が類似しやすいこと、また、学生はシェルやプログラミングに関して同じ授業科目を履修しているため、各機能のアルゴリズムや実装方法が似通う傾向にあることが考えられる。

### 3. 妥当性への脅威

学生が作成したプログラムが類似した構造を持ち、それがメトリクス値に影響した可能性がある。ただし、同一の仕様に対して書かれた学生のプログラム間で同一のコードがほとんどなかったことが Juergens らによって報告されており [3]、各学生のプログラムには独自のソースコード記述が十分にあると考えられる。

コメントに関しては、内容の確認を行っていないため、

学生のプログラム中に存在するコメントのすべてが意味のあるコメントであるとは限らない。たとえば、演習課題では学生が独自の機能を追加することが許されていたが、学生が作りかけの機能の作成を取りやめ、該当コードをコメントアウトして提出するなどの行動をとっている可能性がある。本研究では、メトリクスによる統計的分析のみを行うという条件でソースコードの利用許諾を得たため、内容については分析できなかった。

本研究の結果は、分析対象がシェルプログラムに限定されているため、シェル以外のプログラムに関して同様の結果が得られるとは限らない。

本研究では学生から有志でのコード提供を募ったため、一般的な学生とは異なる結果となっている可能性がある。

### 4. まとめ

学生のプログラムはオープンソースソフトウェアと比較して、関数ごとのステートメント数が多く、ソースファイルごとのコメント、条件分岐も多いということがわかった。そのため、学生が良いプログラムを目指すには、関数の分割をさらに進めるとともに、条件分岐を減らすことができるような工夫をするべきである。教員は最初の指導としてコメントを増やすことを強調するだけでなく、モジュール化を進めることや、コメントを多く書かなくて済むように条件分岐の少ない簡潔なプログラムを書くことを学生に意識させるべきであると考えられる。

謝辞 1 年前の課題プログラムであるにも関わらずプログラムを提供してくださった大阪大学基礎工学部情報科学科 3 年生の皆様には心から御礼申し上げます。

### 参考文献

- [1] いちやもん. <http://www.oki-osk.jp/esc/go/metrics.html>
- [2] Source Monitor. <http://www.campwoodsw.com/sourcemonitor.html>
- [3] Juergens, E., Deissenboeck, F., and Hummel, B. Code Similarities Beyond Copy & Paste. In *Proceedings of the 14th European Conference on Software Maintenance and Reengineering*, pp.78-87, 2010.