

デバッグ作業者の思考を定期的なインタビューと 視点の移動により追跡する実験

内田 眞司† 工藤 英男† 門田 暁人††

†奈良工業高等専門学校 ††奈良先端科学技術大学院大学

1 C-1

1 はじめに

ソフトウェア開発において、デバッグは方法論や技法が最も確立されていない作業であると言われている[1]。効率的にデバッグを行うためには、長期間の熟練や経験を要するのが現状である。

このような現状に対して、デバッグ作業者の様子を観察、分析することにより、デバッグの効率に影響する要因を明らかにしようとする研究が行われている。門田らの実験[2]では、デバッグ作業者は、バグの位置や内容に関する様々な仮説を立てながらデバッグを進めていることが観測されている。また、それらの仮説がデバッグの効率に影響し得ることを示している。

本研究では、デバッグ中の作業者が立てる仮説のうち、特にバグ位置についての仮説に注目し、より詳細な分析を試みた。具体的には、デバッグを行う被験者が、プログラム中のどこにバグがあると考えているかを、定期的なインタビューにより追跡する実験を行った。デバッグの熟練者は、初心者に比べて、バグの位置をより正確かつ迅速に推定できると予想される。

また、本研究では、各モジュール(関数)間のデバッグ作業者の視点の移動の分析を試みた。プログラムを読んでいく戦略は、バグ位置の推定結果に影響を与えると予想される。

実験の結果、バグのあるモジュールについては、初心者も熟練者もバグがありそうだと判断していることが分かった。ただし、バグがありそうだと判断するモジュールの個数には違いがあった。また、熟練者は、最初はプログラムを大局的に見るが、次第に局所的に見ていくことが観測された。

2 実験方法

2.1 実験の手順

4人の被験者について以下の手順で実験を行った。
手順1. バグの入ったプログラムを用意し、被験者にプログラムの仕様とバグの症状を説明する。

手順2. 被験者にソースプログラムを与えて、計算機上でデバッグを始めてもらう。被験者が操作している計算機の画面をビデオで録画する。

手順3. デバッグ開始から終了までの間、5分ごと

に作業を中断してもらい、その都度インタビューを行う。インタビューでは、モジュール名が書いた表を被験者に与え、各モジュールについてどの程度バグがありそうかを“+3”~“-3”の7段階の評価値で記入してもらう。なお、被験者がまだ見えていないモジュールは“0”とする。

手順4. 30分経過後、これ以上デバッグを続けてもバグを取れる見込みがないと被験者自身が宣言した場合には、実験を終了する。

手順5. 実験終了後、録画したビデオを分析し、被験者がいつどのモジュールを参照していたかを秒単位で記録する。

2.2 被験者

大学院生1人(被験者A)と専攻科生3人(被験者B, C, D)の合計4人である。4名とも4年以上のプログラミング経験があった。

2.3 プログラムとバグ

実験に用いたプログラムは、C言語で書かれている三目並べゲームで、約250行(モジュール数は15個)であった。モジュール間の視点の移動の分析を容易にするために、各モジュール間には15行の空行を挿入した。バグの個数は1個であり、バグの種類は、配列の添字の誤りである。

3 実験結果および考察

3.1 バグ発見の成否

被験者A, Bはバグを発見することができたが、被験者C, Dはバグを発見できなかった。

3.2 インタビュー結果

インタビュー結果の例として、被験者Aの結果を図1に示す。図中、縦軸は評価値、横軸はインタビュー回数を表す。被験者が最後まで見えていないモジュールを黒色で、“バグがなさそう”と判断したモジュールを斜線で、“バグがありそう”と判断したモジュールを網目で示した。バグが実際に存在したモジュールはNo.12である。

バグの存在するモジュール(No.12)についての各被験者のインタビュー結果を図2に示す。図より、4人の被験者全てが、バグの存在するモジュールについてレベルは違うがバグがありそうだと判断していることが分かる。

A study on the debugging thought process based on changes in impressions of functions

Shinji Uchida, Hideo Kudo, and Akito Monden

†Nara National College of Technology

††Nara Institute of Science and Technology

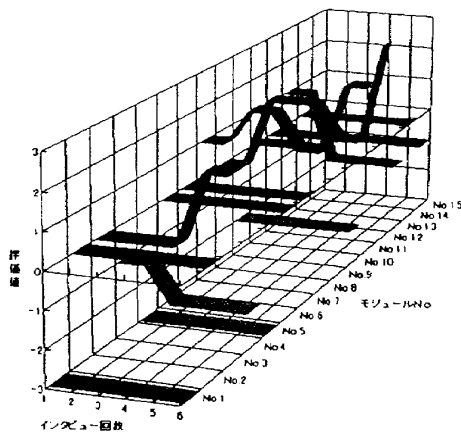


図 1: モジュールごとの評価値の変化 (被験者 A)

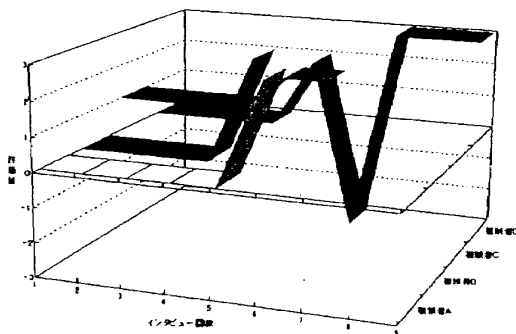


図 2: 被験者ごとの評価値の変化 (No.12)

3.3 モジュール間の視点の移動

各被験者について、モジュール間の視点の移動回数を5分ごとに調べた結果を図3に示す。図中、棒グラフは視線の移動回数を表し、折れ線グラフは視線の移動の累積回数を表す。視点の移動回数は、最初の5分間は4人とも多く、次の5分間では、デバッグに成功した者(被験者 A, B)は、視点の移動回数が半分以下に減っているのに対し、デバッグに失敗した者(被験者 C, D)は減っていない。また、実験開始から15分以降では、後者も視点の移動回数は減っているが、前者ほどは減っていない。累積回数のグラフからも、デバッグに成功した者より失敗した者の方が視点の移動が多いことが分かる。

4 おわりに

本研究では、バグの位置に関するデバッグ作業者の思考を、5分ごとのインタビューとモジュール間の視点の移動により追跡した。

ここで、デバッグに成功した者を熟練者とし、デバッグに失敗した者を初心者として考察すると、イ

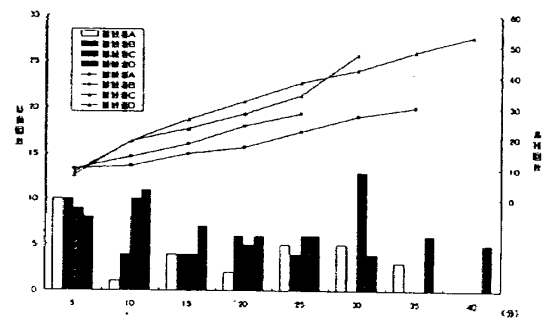


図 3: 視点の移動回数

ンタビュー結果からは、当初の予想に反して、初心者も熟練者と同様、バグのあるモジュールについては、バグがありそうだと判断していることが分かった。ただし、本稿では示していないが、バグがありそうだと判断するモジュールの個数や、判断の過程には違いがあった。今後さらに分析を深める予定である。

一方、モジュール間の視点の移動回数には、デバッグ成功した被験者と失敗した被験者の間に違いが見られた。デバッグ開始当初は両者とも視点の移動回数は多いが、時間経過とともに前者は半分以下に減少するが、後者はなかなか減少しない。デバッグ成功者は読むべきモジュールを決めて局所的に読み始めているのに対して、後者は読むべきモジュールをなかなか特定できずに、もしくは、特定せずに、大局的にプログラムを読んでいると推測される。今後は、熟練者がいかにして読むべきモジュールを絞っていったかを明らかにするために、各被験者がどのような順序でモジュールを参照したかを分析していく予定である。

今回の実験では、4回の試行しか行えていないために、デバッグの効率に影響する要因を明らかにするまでには至っていない。今後は、本実験を継続して、分析を深める予定である。

謝辞

本研究の遂行にあたって、ご指導を頂いた奈良先端科学技術大学院大学ソフトウェア計画構成学講座の鳥居宏次教授、松本健一助教授に深く感謝致します。

参考文献

- [1] Araki K., Furukawa Z., and Cheng J: "A General Framework for Debugging", IEEE Software, pp.14-20 (May 1991).
- [2] 門田, 高田, 鳥居: "視線追跡装置を用いたデバッグプロセスの観察実験", 電子情報通信学会技術報告, SS96-5, pp.1-8(1996).