The University of Osaka
Institutional Knowledge Archive

Title	プログラムの差分情報を用いたデバッグ手法の評価実験
Author(s)	田原, 靖太; 寺口, 正義; 松下, 誠 他
Citation	情報処理学会第61回全国大会講演論文集. 2000, 1, p. 327-328
Version Type	VoR
URL	https://hdl.handle.net/11094/50375
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

プログラムの差分情報を用いたデバッグ手法の評価実験

5W－3　田原靖太[†]　寺口正義[‡]　松下誠[†]　井上克郎[†*]

[†]大阪大学大学院基礎工学研究科　[‡]日本アイ・ビー・エム　東京基礎研究所
[*]奈良先端科学技術大学院大学情報科学研究科

1　はじめに

デバッグ作業における欠陥原因の特定を効率よく行うための手法として，DMET(Debugging METhod)が提案され，それに基づいたデバッグ支援システムDSUS(Debugging SUpport System)の構築が行われている[1][2]．DMETは，ソフトウェアに対して変更を行った際，変更前のプログラムは正常に動作していたが，変更を行った結果，変更していない機能に欠陥が発見された場合のデバッグ作業を支援する手法である．これを用いることにより，効率の良いデバッグを行うことが期待できる．

本研究では，DMETの有効性を評価するため，DSUSを用いて実際に被験者に保守作業を行ってもらい，作業に要した時間を計測するという実験を行った．その結果，DMETによって欠陥の原因が正しく特定された場合は，DMETを用いることによりデバッグ時間が短縮されることを確認できた．

本稿では，正常に動作していた変更前のプログラムを基準バージョンと呼ぶことにする．なお，ここでいうバージョンとは，開発作業の進行と共に変化するプロダクトのある状態を指す．

2　デバッグ手法DMETの概要

提案されたデバッグ手法DMET[2]は，バージョン管理システム，テストツールの利用を前提としている．DMETはテスト手法，表示手法，反映手法の3つから構成されており，これらの繰り返しによって，デバッグ作業におけるテスト作業から修正作業を支援する．それぞれの手法の概要を以下に示す．

テスト手法：欠陥の原因を含むバージョン間の差分を特定する．DMETでは，ソースコードだけではなく，実行可能ファイルもバージョン管理の対象とするため，テストのたびにコンパイルを行う必要はない．

表示手法：特定された差分を，ソースプログラムの最新バージョン上で強調表示する．これに基づいて利用者は最新バージョン上で修正作業を行う．

反映手法：最新バージョン上で行った修正作業の内容を，過去のバージョンに反映する．これを行わなければ，以降のテスト作業を効率良く続けることができない．

デバッグ支援システムDSUSは，バージョン管理システム，テストツールに加え，DMETに基づく処理を自動的に行う機能を備えた開発環境を提供する．

3　評価実験

3.1　概要

ソフトウェア保守作業におけるDMETの有効性を評価するため，大阪大学大学院基礎工学研究科情報数理系専攻の学生10人に対して2段階の実験(実験A，実験B)を行った．実験Aでは，被験者はソフトウェアに機能を追加する作業を行った．これにより，機能追加の際に欠陥が作り込まれたプログラムを収集した．実験Bでは，被験者は実験Aで欠陥が作り込まれたプログラムのデバッグ作業を行った．実験Bでは，10人の被験者を5人ずつ2つのグループG1(a1,...,a5)，G2(b1,...,b5)に分け，G1の被験者のみDMETを用いて実験を行い，2つのグループ間でデバッグに要した時間を比較した．

3.2　準備

酒屋問題[4]に対する仕様書と，それに対応する基準バージョンのプログラムを用意する．さらに，基準バージョンのプログラムが仕様書の通りに動作することを確認するためのテストデータ(T1, ..., T8)を用意した．実験Aにおいては，被験者に行ってもらう仕様変更(機能追加)を用意し，変更後のプログラムが仕様書の通りに動作することを確認するためのテストデータ(T'1, ..., T'4)を用意した．

3.3　手順

実験Aでは，まず被験者に酒屋問題の仕様書，基準バージョンのプログラムリスト，仕様書の変更を与える．被験者は，仕様書の変更を基準バージョンのプログラムに反映させ，T'1～T'4のテストデータに対して正しい出力が得られた時点で終了する．

An Experimental Evaluation of the Debugging Method with Program Delta
[†]Yasutaka Tahara, [‡]Masayoshi Teraguchi, [†]Makoto Matsushita and [†*]Katsuro Inoue
[†]Graduate School of Engineering Science, Osaka University
[‡]Tokyo Research Laboratories, IBM Japan
[*]Graduate School of Information Science, Nara Institute of Science and Technology

被験者全員が実験 A を終了した段階で実験 B に移る．実験 B では，実験 A での機能追加によって基準バージョンでは正常に働いていた機能に欠陥が作られたプログラムをデバッグの対象とする．

実験 B では，与えられた各プログラムに対して，G1 の被験者 (a1, ... ,a5) は DSUS の DMET 機能を利用し，G2 の被験者 (b1, ... ,b5) は DSUS の DMET 機能を用いずにデバッグ作業を行う．具体的な手順は，まず被験者は与えられたプログラムのテストを行う．次にテスト結果に基づいてデバッグ作業を行う．作業は，T1～T8, T'1～T'4 の全テストデータに対して正しい出力が得られた時点で終了する．実験 B では，最初のテストからデバッグの完了までに要した時間を計測する．

3.4 結果

実験 A では，基準バージョンでは正常に働いていた機能に欠陥が作られたプログラムが 3 個得られた (これらを X,Y,Z とする)．このうち X,Y の 2 個のプログラムは DMET によって特定されたバージョン間の差分に欠陥の原因が含まれていることがわかった．また，実験 B において，各被験者 (a1～a5, b1～b5) が X,Y,Z の各プログラムのデバッグに要した時間に関するデータを表 1 に示す．

3.5 分析・評価

DMET により強調表示される部分に欠陥の原因があるプログラム X，Y について考える．この 2 つのプログラムのデバッグに要した時間の平均を見ると，G1 では 67.0 分，G2 では 163.0 分となった．この結果に対し有意水準 5%で平均値の差の検定を行ったところ，有意な差があることがわかった．したがって，DMET により強調表示される部分に欠陥の原因がある場合は，DMET がデバッグ作業に有効であることが確認された．

逆に，欠陥の原因が DMET により強調表示される部分にないプログラム Z について，デバッグに要した時間の平均を見ると，G1 では 39.4 分，G2 では 10.6 分と，DMET を利用しない方が平均時間は短くなった．また，3 つのプログラムのデバッグに要した平均時間は，G1 では 106.4 分，G2 では 173.6 分となった．平均時間を見ると G1 の方が G2 よりもデバッグに要した時間が 67.2 分短くなっているが，平均値の差の検定を有意水準 5%で行った結果，有意な差を見ることができなかった．これは，プログラム Z での結果が影響したものと考えられる．

4 考察

ここでは，DMET を用いた場合，DMET を用いない場合に比べてプログラム Z のデバッグに時間がかかった原因について考える．

プログラム Z の欠陥の原因は，変数の初期化忘れという，DMET が想定していない「変更を加えなかったことによる欠陥」であった．このため，DMET はプログラム Z の欠陥の原因を正しく特定することができなかった．しかし，このような場合でも，DSUS は DMET によって定められたアルゴリズムに従って特定された範囲を表示するため，被験者が混乱してデバッグ時間の増大につながったと考えられる．

5 まとめ

本研究では，プログラムの差分情報を利用したデバッグ手法 DMET が有効であるかを調べるための実験を行った．その結果，DMET によって特定されたバージョン間の差分に欠陥の原因が含まれる場合は，DMET がデバッグ作業に有効であることが確認できた．今後の課題としては，デバッグ対象プログラムの数を増やすことで，DMET によって欠陥の原因を正しく特定できる割合を実験的に調査することが挙げられる．

表 1: 実験 B の結果 (単位：分)

DMET を利用するグループ G1

	a1	a2	a3	a4	a5	平均
X	37	6	65	38	53	39.8
Y	25	59	14	16	22	27.2
Z	60	57	34	21	25	39.4
合計	122	122	113	75	100	106.4
XY の和	62	65	79	54	75	67.0

DMET を利用しないグループ G2

	b1	b2	b3	b4	b5	平均
X	66	20	117	52	45	60.0
Y	171	87	120	17	120	103.0
Z	20	5	15	5	8	10.6
合計	257	112	252	74	173	173.6
XY の和	237	107	237	69	165	163.0

参考文献

[1] 寺口正義：“バージョン間の差分を利用したデバッグ支援システムの構築”，大阪大学大学院基礎工学研究科修士学位論文, 2000.2.

[2] 寺口正義，松下誠，井上克郎：“バージョン間の差分を利用したデバグ手法の提案”，電子情報通信学会技術研究報告 SS99-52, pp.17-24, 2000.1.

[3] 田原 靖太，“バージョン間の差分を利用したデバッグ手法の実験的評価，” 大阪大学基礎工学部情報科学科 特別研究報告, 2000.3.

[4] 山崎利治：“共通問題によるプログラム設計技法解説”，情報処理学会誌，25,9,p.934,1984.