



Title	保守プロセスに対するプログラムスライスの実験的評価
Author(s)	西松, 顯; 楠本, 真二; 井上, 克郎
Citation	電子情報通信学会論文誌D-I. 1999, J82-D-I(8), p. 1121-1123
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/26454">https://hdl.handle.net/11094/26454</a>
rights	copyright©1999 IEICE
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

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

The University of Osaka



表2 酒屋問題  
Table 2 S1 (P1.1, P1.2).

	P1.1	P1.2	合計
A1	75	50	125
A2	133	89	223
A3	188	301	489
平均	132.0	146.0	279.0
B1	126	87	213
B2	110	60	170
B3	90	64	154
平均	108.7	70.3	179.0

表3 予約問題  
Table 3 S2 (P2.1, P2.2).

	P2.1	P2.2	合計
B1	105	95	200
B2	99	106	205
B3	112	126	238
平均	105.3	109.0	214.3
A1	44	37	81
A2	82	69	151
A3	78	74	152
平均	68.0	60.0	128.0

### 2.3 実験結果

本実験で計測するデータは 2.2 で述べた Step2~Step5 に要した時間である。本来、本実験ではスライス抽出機能の利用により保守作業が効率良く行える Step2 の部分のみの時間を計測し、評価すべきであるが、被験者がプログラムの理解、仕様書の変更の理解、プログラムの変更部分の認識を正しく行ったかどうかを正確に判断できないと考えたため（例えば、非常に短い時間でプログラム理解を終了した被験者が、プログラム変更によく時間を費した場合、被験者のプログラム理解が不十分であることがいえる）、実際にプログラムを正しく変更する作業までの時間を計測、評価することにした。

各仕様書の変更をプログラムへ反映させるのに要した時間（単位：分）のデータを表 2 と表 3 に示す。

### 3. 分析・評価

酒屋問題の 2 種類の仕様変更に必要な平均時間は、スライスを利用した G2 では 179.0 分 ( B1:213 B2:170 B3:154 ), スライスを利用しなかった G1 では 279.0 分 ( A1:125 A2:223 A3:489 ) となっている。平均時間を見ると G2 の方が G1 よりも仕様変更に必要な時間が 100 分短くなっているが、有意差<sup>注2</sup>が見られなかった。

予約問題の 2 種類の仕様変更に必要な平均時間は、スライスを利用した G1 では 128.0 分 ( A1:81 A2:151 A3:152 ), スライスを利用しなかった G2 では 214.3 分 ( B1:200 B2:205 B3:238 ) となっている。平均時間を見ると G1 の方が G2 よりも仕様変更に必要な時間が約 86 分短くなっており、有意差も確認できた。

### 4. 考察

仕様変更の内容によってスライスの有効性に差が生じたことについて考察する。一般にスライスは、文献 [4] で分類されている 4 種類のフォルトのうち、文の欠如 ( missing statement ) には有効ではないといわれている ( 我々は、これらの結果を [2] において実際に確認している )。本実験では仕様書変更要求として、機能変更と機能追加の 2 種類を用意している。機能変更とはもとの仕様書に既にある機能を変更するもので、機能追加とはもとの仕様書への新たな機能の追加を実現するものである。機能変更においては、既に存在するコードから変更部分をスライスとして抽出できるが、機能追加においては、新たにコードを追加する必要があり、追加されるコードはスライス抽出時には存在しないため、変更部分認識時にはスライス抽出機能は有効ではないと考えられる。実際に各仕様変更ごとに分析すると、スライス抽出機能は予約問題においては機能の追加よりも機能の変更の方がより有効であることが確認できた。

### 5. むすび

本研究では、スライスが実際の保守作業に有効であるかどうかを実験的に評価した。実験の結果、スライスをを用いた方が、スライスをを用いない場合よりも効率良く保守作業が行えることが確認できた。本実験で用意したプログラムのサイズは 300~400 行程度と小規模ではあるが、スライスをを用いることで参照範囲を 45~52% に限定できる ( この数値は被験者が実験において抽出した典型的なスライスのサイズである )。大規模なプログラムにスライス抽出技法を適用する場合においても本実験と同様に参照範囲を限定することができ、保守作業を効率良く行えると考えられる。

謝辞 本研究は、一部文部省科学研究費補助金特定領域研究 (A)(2)(課題番号: 10139223) の補助を受けている。

### 文 献

- [1] V.R. Basili, G. Caldiera, and H.D. Rombach, Goal Question Metric Paradigm, in J.J. Marciniak, ed., Encyclopedia of Software Engineering, vol.1, pp.528-532, John Wiley & Sons, 1994.

(注2): 平均値の差の検定 ( ウェルチの検定 ) を有意水準 5% で行った [5] .

- [2] 西松 顯, 楠本真二, 井上克郎, “フォールト位置特定におけるプログラムスライスの実験的評価” 信学技報, SS98-3, May 1998.
- [3] 佐藤慎一, 飯田 元, 井上克郎, “プログラムの依存解析に基づくデバッグ支援ツールの試作” 情処学論, vol.37, no.4, pp.536-545, 1996.
- [4] 下村隆夫, “変数値エラーにおける Critical Slice に基づくバグ究明戦略” 情処学論, vol.33, no.4, pp.501-511, 1992.
- [5] 芝 祐順, 渡部 洋, 石塚智一 編, 統計用語辞典, 新曜社, 1984.
- [6] D.J. Robson, K.H. Bennett, B.J. Cornelius, and M. Munro, “Approaches to Program Comprehension,” J. System Software, vol.14, no.1, 1991.
- [7] M. Weiser, “Program Slicing,” IEEE Trans. on Soft. Eng., vol.10, no.4, pp.352-357, 1984.
- [8] 山崎利治, “共通問題によるプログラム設計技法解説” 情処学誌, vol.25, no.9, p.934, 1984.
- [9] M. Zelkowitz and D.R. Wallace, “Experimental models for validating technology,” IEEE Software, vol.31, no.5, pp.23-31, 1998.

(平成10年10月26日受付)