

2K-5

プログラムテキストの更新履歴を利用した
初期フォールト数推定の自動化の試み

藤田房之 楠本真二 松本健一 菊野亨 鳥居宏次

大阪大学 基礎工学部 情報工学科

1. まえがき

ソフトウェアのテスト段階から収集したデータに基づいて、ソフトウェアの信頼性を定量的に評価するモデルが数多く提案されている(3, 4)。ソフトウェア信頼度成長モデル(Software Reliability Growth Model, 以下ではSRGMと略す。)もそうしたモデルの1つである。本研究では、自動収集可能なデータだけに基づいて初期フォールト数を推定する試みについて述べる。従来、フォールトの除去時刻はテスト担当者の申告によることが多かった。ここでは、Dunsmoreらの提案を改良した方法でフォールトの除去時刻を求め、それをSRGMに適用して初期フォールト数を計算する。

2. SRGM

SRGMでは、フォールトが発見され、除去されるテスト段階での過程を、テスト時刻 t までに除去されたフォールトの累積数を用いて評価する。ここでは遅延S字型モデルに注目する(図1参照)(4)。このモデルでは次式に示すソフトウェア信頼度成長関数 $H(t)$ が用いられる。

$$H(t) = N_0(1 - (1 + \phi t) \exp(-\phi t)) \quad (1)$$

ここで、 $H(t)$ はテスト時刻 t までに除去される総フォールト数の期待値、 N_0 は初期フォールト数、 ϕ は定常状態(即ち、 $t \rightarrow \infty$ としたとき)のフォールト1個当りのフォールト発見率を表す。パラメータ N_0 、 ϕ は、通常、累積除去フォールト数の実測値に基づいて統計的手法により求められる。

式(1)を用いると、テスト時刻 t_0 における残存フォールト数 N_r は次式で求められる。

$$N_r = H(\infty) - H(t_0) = N_0 - H(t_0) \quad (2)$$

3. Dunsmoreらの方法

Dunsmoreらは自動収集可能なデータであるプログラムテキストの更新履歴(2)に基づいて、時刻 t までに除去されたフォールト数や各フォールトの除去時刻を推定する方法を提案している(1)。編集作業の前後のプログラムテキストを比較し、次の条件(1)または(2)に当てはまる変更が行われていれば、フォールトの除去が行われたものと見なすことにする。

- (1) 新たな行の追加・連続する追加行を1つのフォールトに対応させる。なお、「既存の行の移動」は「既存の行の削除」と「新たな行の追加」と考えているので、連続する「既存の行の移動」も1つのフォールトに対応させる。
- (2) 既存の行の置換・置換された1行を1つのフォールトに対応させる。

但し、次の変更はフォールトの除去と見なさないことにする。

- (1) 標準出力文を用いた出力文の追加
- (2) プログラムテキストを見やすくするためのコメント、空白、及び、タブの追加、置換

4. 提案する方法

Dunsmoreらの方法には次の2つの問題点がある。

- (1) 新たな機能を追加するための行の追加もフォールトの除去と見なしてしまう。従って、全ての機能を実現する前にテストが行われるような場合には、Dunsmoreらの方法は適用できない。
- (2) 置換された行の数だけのフォールトが除去されたと見なしてしまう。従って、1つのフォールトを除去するために複数の行が置換されるような場合、フォールト数の正しい評価ができない。例えば、5行の置換を行った場合、5個のフォールトの除去と見なしてしまう。

これらの問題点への対策として、ここでは次の2つの提案を行う。

- (1) 新たな行の追加はフォールトの除去と考えない。但し、既存の行の移動はフォールト除去と見なす必要があるため、行の削除のみをフォールトの除去と見なす。
- (2) 置換された行のうちで同一の編集作業において作成された行の集合は、まとめて、1つのフォールトに対応させる。

5. 評価実験

提案する方法によって初期フォールト数の推定精度がどの程度向上するかを確認するための評価実験を行った。具体的には図2に示す3種類のパラメータ N_d, N_f, N_0 をテスト終了時点 t_0 で推定し、 N_d 及び、 N_f の値を N_0 の値と比較する。ここで、 N_0 は実測値を用いて推定した初期フォールト数、 N_d はDunsmoreらの方法で推定した初期フォールト数、 N_f は提案する方法で推定した初期フォールト数をそれぞれ表す。

5.1 データ収集源

データは大阪大学基礎工学部情報工学科2年生に対して行われた学生実験から収集した。学生が作成するプログラムは、一種の在庫管理プログラムで、プログラムサイズは約300行である。使用するプログラミング言語は、C言語またはPASCALである。最終的に分析の対象とした学生は11人である。

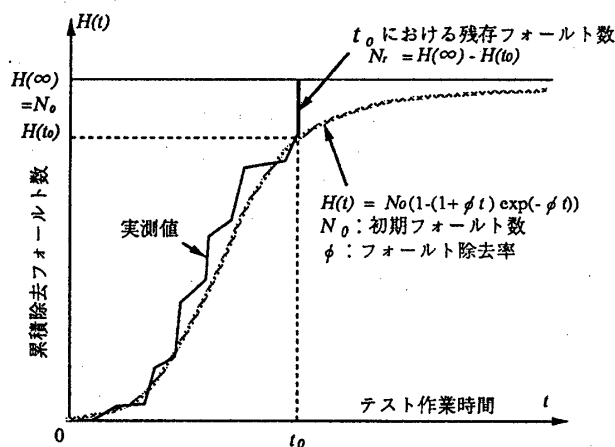


図1 ソフトウェア信頼度成長モデル

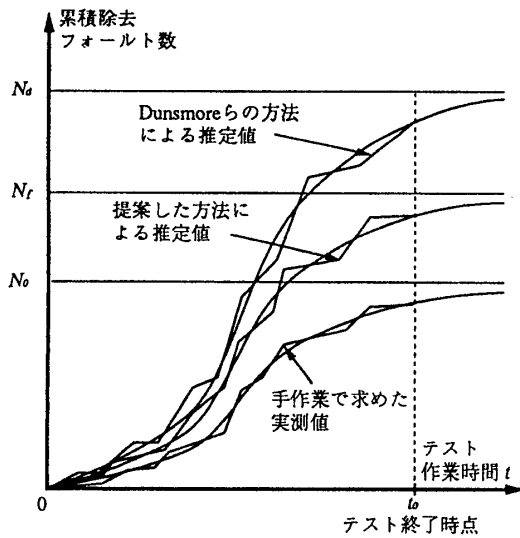


図2 各方法による初期フォールト数

5.2 分析

(1) N_0 との比較

11人の学生に対する3種類の推定値を表1に示す。表1より N_d , 及び N_f の値と N_0 の値との差(推定誤差)は大きく, Dunsmoreらの方法も提案する方法もそのままでは手作業の場合と同様な推定が行えるとは言えないことがわかる。

(2) 回帰分析

3種類の初期フォールト推定値 N_d, N_f, N_0 の間の相関係数を計算した結果を表2に示す。表2より, N_d と N_f は N_0 と比較的高い相関のあることがわかる。それぞれの間の回帰直線の式は次のとおりである。

$$N_0 = 7.6 + 0.15N_d \quad (3)$$

$$N_0 = 3.4 + 0.48N_f \quad (4)$$

(3) 回帰直線に基づく補正

式(3),(4)を用いると, 初期フォールト数の推定精度の改良が可能になる。Dunsmoreらの方法で推定した累積除去フォールト数を式(3)を用いて補正した結果得られた初期フォールト推定値を \tilde{N}_{0d} , 同様に提案する方法で推定した累積除去フォールト数を式(4)を用いて補正した結果得られた初期フォールト推定値を \tilde{N}_{0f} と表す。11人の学生に対する3種類の推定値 $\tilde{N}_{0d}, \tilde{N}_{0f}, N_0$ を表3に示す。

(4) 補正後の推定誤差

表3における推定誤差の平均と分散を求めた結果を表4に示す。表4より, Dunsmoreらの方法に比べ, 提案した方法は手作業の場合に近い推定値を算出することができ, しかもその誤差のばらつきが小さくなっていることが確認できる。

6. あとがき

Dunsmoreらが提案した除去フォールト数の推定方法の改良法を提案した。評価実験の結果, 初期フォールト数の推定精度が, Dunsmoreらの方法よりも平均で14%向上することが確認された。

現時点ではフォールトを行単位で計測している。これはプログラムテキストの変更履歴を行単位で蓄積しているからである。現在, より正確にフォールトを特定するため, プログラムとして意味のある単位で変更履歴をとる方法について検討している。

文 献

(1) H.E. Dunsmore and J.D. Gannon: "Analysis of the effects of programming factors on programming effort", J. Syst.

表1 初期フォールト推定値

学生	N_d	N_f	N_0
A	∞	108	37
B	88	55	43
C	∞	43	29
D	420	196	112
E	159	75	17
F	∞	42	29
G	∞	15	10
H	57	28	16
I	99	24	17
J	53	31	16
K	78	26	17

表2 相関係数

	N_d	N_f
N_0	0.65	0.84

表3 補正後の初期フォールト推定値

学生	\tilde{N}_{0d}	\tilde{N}_{0f}	N_0
A	∞ (-)	37 (0%)	37
B	20 (57.5%)	28 (34.8%)	43
C	41 (41.3%)	21 (27.5%)	29
D	63 (43.7%)	90 (19.6%)	112
E	12 (29.4%)	11 (35.2%)	17
F	∞ (-)	20 (31.0%)	29
G	17 (70.0%)	8 (20.0%)	10
H	15 (6.0%)	16 (0%)	16
I	21 (23.5%)	13 (17.6%)	17
J	14 (12.5%)	15 (6.0%)	16
K	17 (0%)	13 (23.5%)	17

()内は推定誤差

表4 補正後の推定誤差

	平均	分散
Dunsmoreらの方法	32.5%	560.4
提案する方法	18.3%	144.8

Softw., 1, pp.141-153 (1980).

(2) S. Kusumoto, K. Matsumoto, T. Kikuno and K. Torii: "On a measurement environment for controlling software development activities", Trans. on IEICE, E74, 5, pp.1051-1054(1991).
 (3) 松本, 菊野, 鳥居: "S字型ソフトウェア信頼度成長モデルの大学環境における実験的評価—推定精度の比較と習熟係数の決定—", 電子情報通信学会論文誌 D-I, J73-D-I, 2, pp.175-182(1990).
 (4) M. Ohba: "Software reliability analysis models", IBM J. Res. Develop., 28, 4, pp.428-443 (1984).