

確率微分方程式に基づく分散開発環境に対する ソフトウェア信頼度成長モデルと適合性評価に関する考察

鳥取大学 *田村 慶信 TAMURA Yoshinobu
01108985 鳥取大学 木村 光宏 KIMURA Mitsuhiro
01702425 鳥取大学 山田 茂 YAMADA Shigeru

1 はじめに

ソフトウェアの開発環境は、データ通信ネットワークの進歩によって、ホスト集中型の開発環境から分散開発環境へと変わりつつある [1]。従来より、NHPP (非同次ポアソン過程) に基づいたソフトウェア信頼度成長モデルが多く、研究者により議論されてきている。それらのモデルは、テスト工程におけるソフトウェアフォールトの発見過程を離散過程として取り扱っていた。しかしながら、対象とするソフトウェアシステムが大規模な場合には、フォールト発見過程をその値が連続的に変動していく確率過程でモデル化することも可能であると考えられる [2]。

本論文では、確率微分方程式を導入し、分散開発環境の総合テスト工程を対象としたソフトウェア信頼度成長モデルの構築を試みる。

2 分散開発環境に対するソフトウェア信頼度成長モデル

テスト時刻 t ($t \geq 0$) におけるソフトウェア内の残存フォールト数を $M_c(t)$ とし、これは連続値をとるものとする。また、 $M_c(t)$ の時間変化率 $dM_c(t)/dt$ は一定値をとるのではなく、ソフトウェア内に残存しているフォールト数の影響を受けながら、時々刻々と変化していくと考えるのが自然である。このような性質を反映させた最も簡単な時間成長方程式として、

$$\frac{dM_c(t)}{dt} = -b_c(t)M_c(t), \quad (1)$$

を採用する。ここで、 $b_c(t)$ は残存フォールト 1 個当りのフォールト発見率を表す。

本論文では、式 (1) のフォールト発見率 $b_c(t)$ の時間的な不規則変動でモデル化する。すなわち、式 (1) を、

$$\frac{dM_c(t)}{dt} = -\{b_c(t) + \sigma\gamma(t)\}M_c(t), \quad (2)$$

という確率微分方程式 (stochastic differential equation, 以下 SDE と略す) に拡張して考える。ここで、 $\gamma(t)$ は、解過程の Markov 性を保証するために標準化された Gauss 型白色雑音を表す。また、 σ は定数である。式 (2) を初期条件 $M_c(0) = m_0$ の下で Itô の公式を用いて変換すると、次のようになる。

$$M_c(t) = m_0 \cdot \exp \left[- \int_0^t b_c(t) dt - \sigma W(t) \right]. \quad (3)$$

ここで、 $W(\cdot)$ は Wiener 過程であり、形式的には白色雑音 $\gamma(t)$ の時間積分である [2]。

これまでに、分散開発環境を対象とした式 (4) の平均値関数をもつ NHPP モデルについて考察してきた [3]。

$$H_c(t) = a \left[\sum_{i=1}^n p_i (1 - e^{-b_i t}) + \sum_{j=1}^m p_{n+j} \{1 - (1 + b_{n+j} t) e^{-b_{n+j} t}\} \right] \left(a > 0, b_l > 0, p_l > 0, \sum_l p_l = 1 \right). \quad (4)$$

ここで、 $H_c(t)$ はテスト時刻 t までに発見・修正された総フォールト数の期待値を表す。また、 a はテスト開始前に潜在していた総フォールト数を、 b_l ($l = 1, 2, \dots, n+m$) は l 番目のコンポーネントに対する残存フォールト 1 個当りのフォールト発見率を表す。さらに、パラメータ p_l ($l = 1, 2, \dots, n+m$) は l 番目のコンポーネントに対する重み、すなわちテストにおける重要度を表している。

ここで、式 (4) から得られる残存フォールト 1 個当りのフォールト発見率 $b_c(t)$ を用いると、本論文において提案される確率微分方程式から導出されたソフトウェア信頼度成長モデルは、

$$M_c(t) = m_0 \left\{ \sum_{i=1}^n p_i e^{-b_i t} + \sum_{j=1}^m p_{n+j} (1 + b_{n+j} t) e^{-b_{n+j} t} \right\} e^{-\sigma W(t)}, \quad (5)$$

となる。式 (5) から、種々のソフトウェア信頼性評価尺度を導出できる。

3 数値例

本論文では、数値例を示すにあたり、実際のテスト工程 (総合テスト) において観測されたデータを採用する (Ohba[4] 参照)。

3.1 モデルパラメータの推定結果

ここでは、既存のコンポーネント数を $n = 2$ 、新規開発のコンポーネント数を $m = 2$ とした簡単な場合を考えることにする。また、重みパラメータ p_l ($l = 1, 2, \dots, n+m$) の値については、モデルの適合性比較

を行う都合上、文献 [3] において使用された値 ($p_1 = 0.05, p_2 = 0.05, p_3 = 0.45, p_4 = 0.45$) を採用する。

その結果、以下のような $m_0, b_1, b_2, b_3, b_4, \sigma$ の最尤推定値を得ることができる。

$$\begin{aligned} \hat{m}_0 &= 350.66, \hat{b}_1 = 0.17637, \hat{b}_2 = 0.17637, \\ \hat{b}_3 &= 0.23151, \hat{b}_4 = 0.23151, \hat{\sigma} = 0.07158. \end{aligned}$$

これらを用いて、式 (5) から得られる残存フォールト数の期待値の推定値 $\hat{E}[M_c(t)]$ を図 1 に示す。

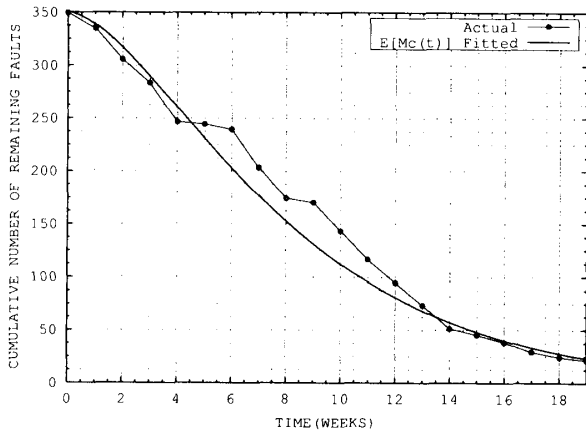


図 1: 推定された残存フォールト数の期待値, $\hat{E}[M_c(t)]$.

3.2 モデルの適合性比較

次に、分散開発環境に対する SDE モデルと、従来の分散開発環境に対する NHPP モデル、および指数形 SDE モデル [2] との適合性比較結果を表 1 に示す。また、適合性評価基準としては AIC を用いた [5]。表 1 より、従来のモデルに比べて、本モデル $\hat{E}[N_c(t)]$ の実測データに対する適合性が良いことが分かった。

表 1: AIC による適合性比較結果。

Compared models	AIC
SDE model $\hat{E}[N_c(t)]$	140.15
NHPP model $\hat{H}_c(t)$	209.67
Exponential SDE model	143.61

3.3 重みパラメータ $p_l (l = 1, 2, \dots, n + m)$ に関する感度分析

さらに、各コンポーネントに対するテストの重みを示すパラメータ $p_l (l = 1, 2, \dots, n + m)$ の値を変化させた場合における推定された累積発見フォールト数の期待値 $\hat{E}[N_c(t)] (= \hat{E}[m_0 - M_c(t)])$ の変化状況を図 2 に示す。図 2 から、本モデルは指数形成長曲線から S 字形成長曲線までの広い範囲の特性曲線を形成することができる。したがって、合理的な推定方法により重みパラメータ $p_l (l = 1, 2, \dots, n + m)$ の

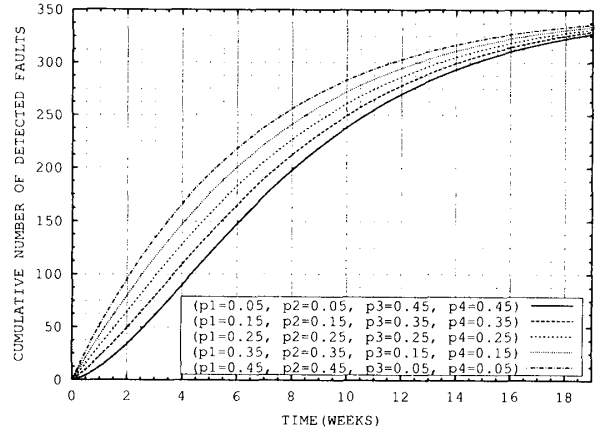


図 2: パラメータ $p_l (l = 1, 2, 3, 4)$ を変化させたときの推定された累積発見フォールト数の期待値, $\hat{E}[N_c(t)]$.

値が与えられれば、既存のソフトウェア信頼度成長モデルより高い精度で信頼性評価尺度の推定が可能となるだけでなく、開発管理者にとっては、適用モデルの選定作業の省力化に結びつくといえる。

4 おわりに

今後の課題として、本モデルに含まれる重みパラメータ $p_l (l = 1, 2, \dots, n + m)$ の値を決めるための決定的な尺度がないため、合理的な重みパラメータ $p_l (l = 1, 2, \dots, n + m)$ の推定方法について検討する必要がある。

謝辞

本研究の一部は (財) 電気通信普及財団の研究調査助成及び文部科学省科学研究費補助金基盤研究 (C)(2) (課題番号 12680442)、奨励研究 (A) (課題番号 13780364) の下で行われた。

参考文献

- [1] 長野 宏宣・宮地 利雄 (編著), “分散ソフトウェア開発,” 共立出版, 東京, 1996.
- [2] S. Yamada, M. Kimura, H. Tanaka, and S. Osaki, “Software reliability measurement and assessment with stochastic differential equations,” *IEICE Trans. Fundamentals*, vol. E77-A, no. 1, pp. 109-116, Jan. 1994.
- [3] 山田茂, 田村慶信, 木村光宏, “分散開発環境を考慮したソフトウェア信頼度成長モデルに関する考察,” 電子情報通信学会論文誌 vol. J82-A, no. 9, pp. 1446-1453, 1999 年 9 月.
- [4] M. Ohba, “Software reliability analysis models,” *IBM J. Research and Development*, vol. 28, no. 4, pp. 428-443, July 1984.
- [5] A. Iannino, J. D. Musa, K. Okumoto, and B. Littlewood, “Criteria for software reliability model comparisons,” *IEEE Trans. Software Engineering*, vol. SE-10, no. 6, pp. 687-691, Nov. 1984.