

テスト網羅度に基づくフォールトの検出可能性を考慮した ソフトウェア信頼度成長モデル

02005295 鳥取大学 *井上 真二 INOUE, Shinji
01702425 鳥取大学 山田 茂 YAMADA, Shigeru

1 はじめに

ソフトウェア開発工程の中の最終工程であるテスト工程において、定量的にソフトウェアシステムの信頼性を評価することは、高信頼性を有するソフトウェアシステムを開発する上で重要な問題である。ソフトウェア信頼度成長モデル (software reliability growth model, 以下 SRGM と略す) [5, 9] は、現在、ソフトウェアシステムの定量的な信頼性評価手法の基盤技術の1つとして浸透している。特に、計数過程である非同次ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) に基づく SRGM は、比較的単純なモデル構造と適用性の高さから、多くの企業において実用に供されている SRGM の1つであり、その適用事例 [3, 6] も数多く報告されている。

また、ソフトウェアシステムの信頼度成長過程に大きく関係する要因として、テスト網羅度 (testing-coverage) [1, 11] を用いて、達成されたテスト網羅度とソフトウェアシステムの信頼性との関係性を検証する取組みが目ざされている [8]。テスト網羅度とは、作成および実行されたテストケースがある評価尺度の下におけるすべてのパスに対して、どれだけ網羅しているかを示す指標のことである。近年では、テスト網羅度を考慮した SRGM [2, 4, 7] が提案され、観測されたテスト網羅度とフォールト発見数データとの関係が検証されている。

本研究では、テスト工程におけるソフトウェアの信頼度成長に影響を与える要因として、テスト網羅度 (testing-coverage) を取り上げ、実現されたソフトウェア信頼性との関係と、その適合性について議論する。具体的には、まず、文献 [2] に基づき、代替的なテスト網羅度関数を再構築する。その後、この代替的テスト網羅度関数と観測されたフォールトデータとの関係を新たに定式化し、実際のテスト工程において観測されたテスト網羅度データおよびフォールト発見数データを用い、その適合性を検証する。

2 テスト網羅尺度

テスト網羅度とは、作成および実行されたテストケースが、ある評価基準の下でのすべてのパスに対してどの程度網羅されているかを示すものである。評価基準としては、以下に示す3つのテスト網羅尺度が挙げられる [11]。

○ 命令網羅尺度 (C0 testing-coverage measure)

プログラムのすべての命令を通過することを基準とした網羅尺度。この尺度による評価からは、テストが不十分であるということはあるが、テストが十分であるという証明は得られない。

○ 分岐網羅尺度 (C1 testing-coverage measure)

プログラムのすべての分岐を通過することを基準としたもの。プログラム制御構造に関係するエラーを検出することができる。

○ パス網羅尺度

プログラムのすべてのパスを通過することを基準としたもの。

3 代替的テスト網羅度関数

本研究では、単位時間当りに達成されるテスト網羅度は、最終的に達成可能なテスト網羅度と任意のテスト時刻におけるテスト網羅度の達成値との差に比例すると考える。すなわち、次のような微分方程式により定式化されるものとする。

$$\frac{dC_i(t)}{dt} = \beta_i(t)[\alpha_i - C_i(t)] \quad (\alpha_i \geq 0, \beta_i(t) \geq 0). \quad (1)$$

ここで、添字 i ($i = 0, 1, 2$) は、それぞれ命令、分岐、パス網羅尺度を表す。式 (1) において、 $C_i(t)$ はテスト網羅尺度 i に関して任意のテスト時刻 t までに達成されたテスト網羅度、 α_i はテスト開始前に設定される達成可能なテスト網羅度、 $\beta_i(t)$ は任意のテスト時刻 t におけるテスト網羅度達成率である。式 (1) の微分方程式を初期条件 $C_i(0) = 0$ の下で、 $C_i(t)$ に関して解くと、

$$C_i(t) = \alpha_i \left(1 - \exp\left[-\int_0^t \beta_i(x) dx\right] \right), \quad (2)$$

と求められる。本研究では、式 (2) をテスト網羅尺度 i に対する代替的テスト網羅度関数と呼ぶことにする。

4 テスト網羅度に基づく SRGM

本章では、テスト網羅度データとフォールトデータの関連付けについて議論を行い、達成されたテスト網羅度から考えられる検出可能フォールト数を考慮した SRGM を構築する。まず、任意のテスト時刻 t において発見されたフォールト数を表す計数過程 $\{N(t), t \geq 0\}$ に対して、以下に示す NHPP を仮定する。

$$\begin{aligned} \Pr\{N(t) = n \mid N(0) = 0\} \\ &= \frac{\{H(t)\}^n}{n!} \exp\{-H(t)\} \\ &(n = 0, 1, 2, \dots). \end{aligned} \quad (3)$$

ここで、 $H(t)$ は NHPP の平均値関数であり、任意のテスト時刻 t までに発見された総期待フォールト数を表す。

本研究では、達成されたテスト網羅度に基づいたソフトウェア信頼度成長過程を記述するため、式 (3) の平

均値関数を以下のように構築していく。モデルの構築に際し、テスト網羅尺度 i が 100% のときに発見される総期待フォールト数に対して、任意のテスト時刻 t までに発見可能なフォールト数は、作成および実行されたテストケースの網羅度に依存すると考える。すなわち、達成されたテスト網羅度とソフトウェア信頼度成長過程の関係は、以下のように定式化される。

$$\left. \begin{aligned} \frac{dH_i(t)}{dt} &= b_i(t)[\gamma_i(t) - H_i(t)] \\ \gamma_i(t) &= a_i \cdot C_i(t). \end{aligned} \right\}$$

ここで、 $b_i(t)$ はテスト網羅尺度 i の下でのフォールト 1 個当りの発見率、 a_i はテスト網羅尺度 i が 100% のときに発見される総期待フォールト数、 $\gamma_i(t)$ はテスト網羅尺度 i の下で任意のテスト時刻 t までに発見可能な期待フォールト数を表す。ところで、式 (4) では、テスト網羅尺度 i の下で、達成されたテスト網羅度により検出可能なフォールト数を $a_i \cdot C_i$ として、単位時間当りに発見されるフォールト数は、その時刻において残存する発見可能なフォールト数に比例するものと仮定している。

式 (4) の微分方程式を初期条件 $H(0) = 0$ の下で、 $H(t)$ に関して解くと、

$$H_i(t) = e^{-Z_i(t)} \left[a_i \int_0^t C_i(s) e^{Z_i(s)} b_i(s) ds \right], \quad (4)$$

ただし、

$$Z_i(t) = \int_0^t b_i(s) ds, \quad (5)$$

と求められる。

5 考察

式 (4) は、任意のテスト時刻 t におけるテスト網羅尺度 i の達成率 $\beta_i(t)$ 、およびテスト網羅尺度 i の下でのフォールト 1 個当りの発見率 $b_i(t)$ を具体的に決定することにより、様々なタイプの平均値関数を描くことができる。例えば、具体的に以下のような仮定を設定する。

(A-1) 任意のテスト時刻におけるテスト網羅度の達成率は、一定である。すなわち、

$$\beta_i(t) = \beta_i. \quad (6)$$

(A-2) テスト網羅尺度 i の下でのフォールト 1 個当りの発見率は、一定である。すなわち、

$$b_i(t) = b. \quad (7)$$

このとき、式 (4) より、テスト網羅尺度 i の下での NHPP の平均値関数 H_i は、

$$H_i(t) = a_i \cdot \alpha_i \left[1 - \frac{1}{\beta_i - b_i} (\beta_i \cdot e^{-b_i t} - b_i \cdot e^{-\beta_i t}) \right], \quad (8)$$

のように求められる。本研究では、式 (8) の平均値関数 $H_i(t)$ について、実際のテスト工程において観測された命令網羅尺度データとそれに対応するフォールト発見数データを用い、数値例を示す。数値例は、紙面の都合上、当日発表を行う。

6 おわりに

本研究では、テスト網羅度の概念に基づき、達成されたテスト網羅度に基づいて考えられる検出可能なフォールト数を考慮した SRGM の構築を行った。今回導出した SRGM は、任意のテスト時刻 t におけるテスト網羅尺度 i の達成率 $\beta_i(t)$ 、およびテスト網羅尺度 i の下でのフォールト 1 個当りの発見率 $b_i(t)$ をそれぞれ具体的に決定することにより、様々なタイプの平均値関数を導出することができる。今後は、テスト網羅度を取り入れたソフトウェア信頼度成長過程に関する確率的理論の構築と共に、様々な実測データを用い、提案したモデルの適合性をさらに検証する必要がある。

謝辞

本研究の一部は、日本学術振興会科学研究費補助金基盤研究 (C)(2) (課題番号 15510129) の援助を受けたことを付記する。

参考文献

- [1] J.R. Horgan, S. London, and M.L. Lyu, "Achieving software quality with testing coverage measures," *IEEE Computer*, Vol. 27, No. 9, pp. 60-69, 1994.
- [2] S. Inoue and S. Yamada, "Software reliability growth modeling based on testing-coverage," Proceedings of the 2nd International Conference on Quality, Reliability and Information Technology, New Delhi, India, to be published in 2004.
- [3] K. Konoun, M. Kaaniche, and J.C. Laprie, "Qualitative and quantitative reliability assessment," *IEEE Software*, Vol. 14, No. 2, pp. 77-87, 1997.
- [4] Y.K. Malaiya, M.N. Li, J.M. Bieman, and R. Karcich, "Software reliability growth with test coverage," *IEEE Trans. Reliability*, Vol. 51, No. 4, pp. 420-426, 2002.
- [5] J.D. Musa, D. Iannio, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York, 1987.
- [6] T. Nara, M. Nakata, and A. Oishi, "Software reliability growth analysis —Application of NHPP models and its evaluation—," *Proc. 1996 High-Assurance Systems Engineering Workshop (HASE'96)*, pp. 222-227, 1996.
- [7] H. Pham and X. Zhang, "NHPP software reliability and cost models with testing coverage," *European Journal of Operational Research*, Vol. 145, No. 2, pp. 443-454, 2003.
- [8] G.S. Varadan, "Trends in reliability and test strategies," *IEEE Software*, Vol. 12, No. 3, p. 10, 1994.
- [9] 山田 茂, ソフトウェア信頼性モデル —基礎と応用—, 日科技連出版社, 東京, 1994.
- [10] S. Yamada and T. Fujiwara, "Testing-domain dependent software reliability growth models and their comparisons of goodness-of-fit," *International Journal of Reliability, Quality and Safety Engineering*, Vol. 8, No. 3, pp. 205-218, 2001.
- [11] 山田 茂, 木村 光宏, 高橋 宗雄, TQM のための統計的品質管理 —一般工業製品からソフトウェアまで—, コロナ社, 東京, 1998.
- [12] S. Yamada, H. Ohtera, and H. Narihisa, "Software reliability growth models with testing-effort," *IEEE Trans. Reliability*, Vol. R-35, No. 1, pp. 19-23, 1986.