ソフトウェア可用性を考慮したデバッグ回数に基づく最適リリース問題に関する一考察

01307475　鳥取大学　*得能貢一　TOKUNO Koichi
01702425　鳥取大学　山田茂　YAMADA Shigeru

1　はじめに

本稿では，マルコフ型ソフトウェア可用性モデル[1]を用いたソフトウェアの最適リリース問題について議論する．まず，テスト開始からリリース後の保証期間終了までに発生するコスト要因を考慮しながら，コスト評価基準による最適リリース問題を定式化する．このとき，総期待ソフトウェアコストを，リリースまでに実施されるデバッグ回数の関数で与え，これを最小にするデバッグ回数を求める最適リリース方策を導出する[2]．また，あらかじめ設定されたソフトウェア可用性評価尺度の目標値をも満たす最適リリース問題についても議論する．

2　ソフトウェア可用性モデル

時刻tにおけるソフトウェアシステムの状態を表す確率過程$\{X(t), t \geq 0\}$を導入し，その状態空間を以下のように定義する．

W_n：システムが動作している状態．

R_n：ソフトウェア故障が発生し修復作業が実行されている状態．

図1に，$\{X(t), t \geq 0\}$の状態遷移図を示す．ここで，nは修正されたフォールト数，aは完全デバッグ率を表し，$b \equiv 1-a$である．また，λ_nおよびμ_nは，それぞれソフトウェア故障率およびソフトウェア修復率を表し，nの非増加関数とする．

$X(t)$の状態W_nに関する状態占有確率は，

$$P_{W_i,W_n}(t) \equiv \Pr\{X(t) = W_n | X(0) = W_i\} = \frac{g_{i,n+1}(t)}{a\lambda_n} + \frac{g'_{i,n+1}(t)}{a\lambda_n\mu_n} \quad (i \leq n), \quad (1)$$

で与えられる．ここで，$g_{i,n}(t)$はソフトウェアが状態W_iからW_nへ遷移するのに要する時間を表す確率変数$S_{i,n}$に対する確率密度関数を表し，$g'_{i,n}(t) \equiv \mathrm{d}g_{i,n}(t)/\mathrm{d}t$である．このとき，**瞬間ソフトウェア・アベイラビリティ**(instantaneous software availability)は，

$$A(t;l) = \sum_{i=0}^{l} \binom{l}{i} a^i b^{l-i} \sum_{n=i}^{\infty} P_{W_i,W_n}(t), \quad (2)$$

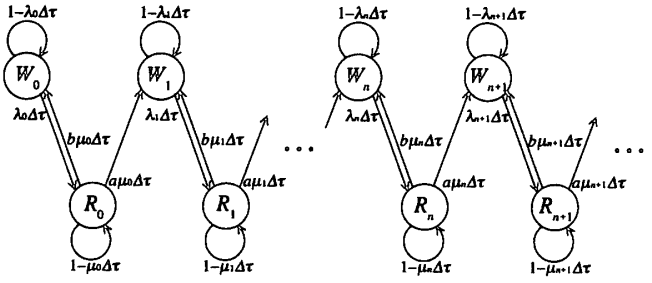

図1　$X(t)$の状態遷移図．

で与えられる．式(2)は，時刻$t=0$においてl回目のデバッグ作業が完了しているとき，その後の任意の時刻tにおいて，システムが動作している確率を表す．

また，時刻$t=0$においてl回目のデバッグ作業が完了しているとき，その後の時間区間$(0,\ t]$の間に実施されるデバッグ回数の期待値は，

$$M(t;l) = \frac{1}{a}\sum_{i=0}^{l} \binom{l}{i} a^i b^{l-i} \sum_{n=i+1}^{\infty} G_{i,n}(t), \quad (3)$$

で与えられる．ここで，$G_{i,n}(t)$は確率変数$S_{i,n}$の分布関数である．

3　最適リリース問題

3.1　コスト評価基準による最適リリース問題

以下のコストパラメータを導入する．

c_1：テスト工程中に行われるデバッグ作業1回当たりのコスト$(c_1 > 0)$，

c_2：テストに要する単位時間当たりのコスト$(c_2 > 0)$，

c_3：保証期間中に行われるデバッグ作業1回当たりのコスト$(c_3 > c_1 > 0)$．

X_lを$(l-1)$番目とl番目$(l = 1, 2, \ldots)$の間のソフトウェア故障発生時間間隔を表す確率変数とし，T_wをリリース後の保証期間とすると，ソフトウェアのテスト開

始後より保証期間が終了するまでに要する総期待ソフトウェアコストは，

$$WC(l) = c_1 l + c_2 \sum_{j=1}^{l} \mathrm{E}[X_j] + c_3 M(T_w; l), \qquad (4)$$

により与えられる．そして，式(4)を最小にする自然数 $l = l^*$ が，最適なデバッグ回数となる．

3.2 ソフトウェア可用性を考慮した最適リリース問題

A_0 を瞬間ソフトウェア・アベイラビリティの目標基準値とすると，ソフトウェア可用性を考慮した最適リリース問題は，

$$\left.\begin{array}{ll} \text{minimize} & WC(l) \\ \text{subject to} & \min\limits_{t} A(t; l) \geq A_0 \end{array}\right\}, \qquad (5)$$

により与えられる．すなわち，式(2)の瞬間ソフトウェア・アベイラビリティの最小値が，目標基準値 A_0 を満たした上で，式(4)の総期待ソフトウェアコストを最小にする自然数 $l = l^*$ が，ソフトウェア可用性を考慮した最適デバッグ回数となる．

4 最適リリース方策

式(4)の1階差分は，

$$\left.\begin{array}{l} Z(l) \equiv WC(l+1) - WC(l) \\ \quad = c_1 + c_2 \mathrm{E}[X_{l+1}] + c_3 M_d(T_w; l) \\ \left(M_d(T_w; l) \equiv M(T_w; l+1) - M(T_w; l)\right) \end{array}\right\}, \qquad (6)$$

となる．$\mathrm{E}[X_l]$ および $M_d(T_w; l)$ は，l に関する単調増加関数であるので，$Z(l)$ は単調増加関数である．また，$Z(\infty) = \infty$ である．したがって，$Z(0) < 0$ のとき，$Z(l) \geq 0$ を満たす最小の正数 $l = l_Z$ が有限かつ唯一存在し，これは2つの不等式 $WC(l+1) \geq WC(l)$, $WC(l) < WC(l-1)$ を同時に満たす．よって，$l^* = l_Z$ は式(4)を最小にする最適デバッグ回数である．

式(2)の $A(t; l)$ の時刻 t に関する振舞いについては，$t = 0$ の近傍 $t = t_0$ で最小値をとり，$[t_0, \infty)$ で単調増加関数となる．一方，デバッグ回数 l については，単調増加関数である．したがって，$\min\limits_{t} A(t; 0) < A_0$ ならば，$\min\limits_{t} A(t; l) \geq A_0$ を満たす最小の正数 $l = l_A$ が有限かつ唯一存在し，これがソフトウェア可用性要求を満たすために必要な最小のデバッグ回数である．

以上をまとめると，式(5)に対する最適リリース方策は次のようにまとめられる．

[最適リリース方策]

$Z(l) \geq 0$ を満たす最小のデバッグ回数を l_Z，$\min\limits_{t} A(t; l) \geq A_0$ を満たす最小のデバッグ回数を l_A とする．

(I) $Z(0) < 0$ かつ $\min\limits_{t} A(t; 0) \geq A_0$ ならば，リリースまでの最適デバッグ回数は $l^* = l_Z$ である．

(II) $Z(0) < 0$ かつ $\min\limits_{t} A(t; 0) < A_0$ ならば，リリースまでの最適デバッグ回数は $l^* = \max\{l_Z, l_A\}$ である．

(III) $Z(0) \geq 0$ かつ $\min\limits_{t} A(t; 0) \geq A_0$ ならば，リリースまでの最適デバッグ回数は $l^* = 0$ である．

(IV) $Z(0) \geq 0$ かつ $\min\limits_{t} A(t; 0) < A_0$ ならば，リリースまでの最適デバッグ回数は $l^* = l_A$ である．

このとき，最適リリース時刻 T^* の平均値は，

$$\mathrm{E}[T^*] = \sum_{j=1}^{l^*} \sum_{i=0}^{j-1} \binom{j-1}{i} a^i b^{j-1-i} \left(\frac{a\lambda_i\mu_i}{\lambda_i + \mu_i}\right), \qquad (7)$$

である．

謝 辞

本研究の一部は，財団法人実吉奨学会の援助を受けたことを付記する．

参考文献

[1] K. Tokuno and S. Yamada, "Relationship between software availability measurement and the number of restorations with imperfect debugging," Computers and Mathematics with Applications, Vol. 46, No. 7, pp. 1155–1163, Oct. 2003.

[2] S. Yamada, K. Tokuno and S. Osaki, "Software reliability measurement in imperfect debugging environment and its application," Reliability Engineering and System Safety, Vol. 40, No. 2, pp. 139–147, May 1993.