使用時間を考慮したデバッグ回数に基づくソフトウェア可用性解析

鳥取大学 *唐木崇 KARAKI Takashi
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

本研究では，継続的にソフトウェアシステムを使用することを想定して，任意の時間区間でソフトウェアシステムが継続的に利用可能である確率を表す**区間ソフトウェア信頼度**(interval software reliability)および任意の時刻で利用可能であるときの**平均利用可能時間**(mean available time)を導出する．さらに，ソフトウェアのデバッグ作業の実施回数が，ソフトウェアシステムの可用性評価に与える影響について考察する[1].

本モデルでは，発見されたフォールトに対するデバッグ作業は，必ずしも完全かつ確実に実施されるとは限らないという**不完全デバッグ**(imperfect debugging)の環境を考慮する．システムが正常に動作している状態と，ソフトウェア故障が発生し修復作業が実施されている不動作状態を，交互に繰り返すシステムの確率的挙動を，マルコフ過程(Markov process)を用いて記述する．

2 モデルの記述

ソフトウェア可用性評価モデルの仮定を以下に示す．

A1. ソフトウェア故障が発生するとシステムダウンとなり，直ちに修復作業に入る．修復作業が終了するまでシステムは動作しない．

A2. 修復作業にはデバッグ作業が含まれる．デバッグ作業は確率$a(0 < a \leq 1)$で確実かつ完全に実施され，確率$b(=1-a)$で不完全な作業となる．ここで，aを完全デバッグ率と呼ぶ．完全なデバッグ作業が行われたとき，1つのフォールトが修正されシステムから除去される．

A3. n個のフォールトが修正されているとき，次のソフトウェア故障発生時間間隔および修正作業時間は，それぞれ平均$1/\lambda_n$および$1/\mu_n$を持つ指数分布に従う．λ_nおよびμ_nはnの減少関数とする．

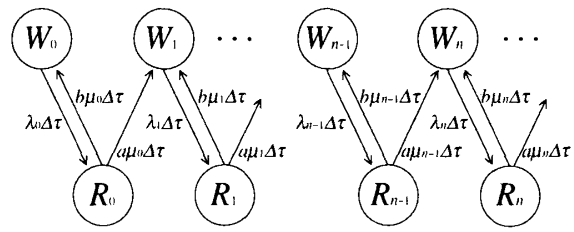

図1. $X(t)$の状態遷移図.

A4. 2つ以上のソフトウェア故障が同時に起こる確率は無視する．

システムの状態を表す確率過程$\{X(t),\ t \geq 0\}$を導入し，こ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.

ここで，$n = 0,\ 1,\ 2,\ \ldots$は修正されたフォールト数を表す．

仮定A4より，$\{X(t) = R_n\}$のときに修復作業を完了すると，

$$X(t) = \begin{cases} W_n & (\text{確率}\, b) \\ W_{n+1} & (\text{確率}\, a), \end{cases} \tag{1}$$

となる．

図1に，マルコフ過程を形成する確率過程$\{X(t), t \geq 0\}$の状態遷移図を示す．

3 ソフトウェア可用性評価尺度の導出

時刻0において$X(t)$が状態W_iにあるとき，時刻tにおいて$X(t)$が状態W_nにある状態占有確率$P_{W_i,W_n}(t)$ $(i \leq n)$は，

$$\begin{aligned} P_{W_i,W_n}(t) &\equiv \Pr\{X(t) = W_n | X(0) = W_i\} \\ &= \frac{1}{a\lambda_n} g_{i,n+1}(t) + \frac{1}{a\lambda_n\mu_n} g_{i,n+1}{}'(t), \end{aligned} \tag{2}$$

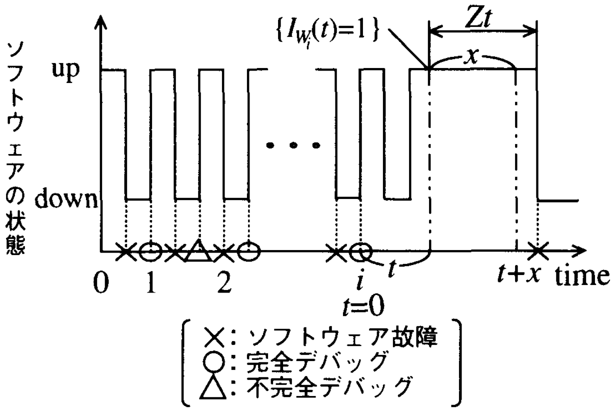

図2. ソフトウェアの振舞いと確率変数 Z_t.

と表される. ここで, $g_{i,n}(t)$ は状態 W_i から状態 W_n への推移時間に対する確率密度関数であり, $g_{i,n}{}'(t) \equiv dg_{i,n}(t)/dt$ である.

ここで, デバッグ回数とソフトウェア可用性評価尺度の関係について議論する. $I_{W_i}(t)$ を, $\{X(0)=W_i\}$ が与えられたときのシステムの状態を表す確率変数とする. すなわち,

$$I_{W_i}(t)=\begin{cases}1\,(\text{動作状態のとき})\\0\,(\text{不動作状態のとき}),\end{cases}$$

とする. このとき,

$$A_i(t) \equiv \Pr\{I_{W_i}(t)=1\}=\sum_{n=i}^{\infty} P_{W_i,W_n}(t), \tag{3}$$

となる. 式(3)は, 時刻 $t=0$ でシステムが状態 W_i にあるとき, 時刻 t においてソフトウェアが動作状態にある確率を表す.

また, 任意の時刻 t から計測したソフトウェア故障発生時間間隔を表す確率変数 Z_t を定義する (図2参照). このとき, $\{X(0)=W_i\}$ が与えられたときの区間ソフトウェア信頼度および平均利用可能時間は, それぞれ

$$\begin{aligned} R_i(t,x) &\equiv \Pr\{I_{W_i}(t)=1, Z_t>x\} \\ &= \sum_{n=i}^{\infty} P_{W_i,W_n}(t)e^{-\lambda_n x}, \end{aligned} \tag{4}$$

$$\begin{aligned} MAT_i(t) &\equiv \mathrm{E}[Z_t|I_{W_i}(t)=1] \\ &= \frac{\sum_{n=i}^{\infty} P_{W_i,W_n}(t)/\lambda_n}{A_i(t)}, \end{aligned} \tag{5}$$

と定義される. 式(4)は, 時刻 $t=0$ でシステムが状態 W_i にあるとき, その後の時間区間 $(t,t+x](x \geq 0)$ で継続的にソフトウェアが利用可能である確率を表す. また式(5)は, 時刻 $t=0$ でシステムが状態 W_i にあるとき, その後の時刻 t においてソフトウェアが利用可能であるという条件の下でのソフトウェアが継続的に利用可能である時間間隔の期待値を表す.

本稿では, 不完全デバッグの環境を考慮しているので, 修正できたフォールト数を陽に観測することができない. しかしながら, l 回のデバッグ作業が実施されたときの修正された累積フォールト数を表す確率変数 $C_l(l=0,\ 1,\ 2,\ \ldots)$ は, 確率関数

$$\Pr\{C_l=i\}=\binom{l}{i}a^i b^{l-i} \quad (i=0,\ 1,\ 2,\ \ldots,\ l), \tag{6}$$

を持つ二項分布に従う. ここで, $\binom{l}{i} \equiv l!/[(l-i)!i!]$ は二項係数である. したがって, l 回目のデバッグ作業終了後の瞬間ソフトウェア・アベイラビリティ, 区間ソフトウェア信頼度および平均利用可能時間は, それぞれ

$$A(t;l)=\sum_{i=0}^{l}\Pr\{C_l=i\}A_i(t), \tag{7}$$

$$R_I(t,x;l)=\sum_{i=0}^{l}\Pr\{C_l=i\}R_i(t,x), \tag{8}$$

$$MAT(t;l)=\sum_{i=0}^{l}\Pr\{C_l=i\}MAT_i(t), \tag{9}$$

で与えられる. 式(8)は, l 回目のデバッグ作業終了時点を $t=0$ としたとき, その後の時間区間 $(t,t+x](x \geq 0)$ で継続的にソフトウェアが利用可能である確率を表す. また式(9)は, l 回目のデバッグ作業終了時点を時刻 $t=0$ としたとき, その後の時刻 t においてソフトウェアが利用可能であるという条件の下でのソフトウェアが継続的に利用可能である時間間隔の期待値を表す.

謝辞

本研究の一部は, (財) 電気通信普及財団研究調査助成, (財) 日本科学協会笹川科学研究助成および文部科学省科学研究費補助金 (課題番号12680442および13780365) の援助を受けたことを付記する.

参考文献

[1] K. Tokuno and S. Yamada, "Markovian software availability measurement based on the number of restoration actions," IEICE Trans. Fundamentals, vol.E83-A, no.5, pp.835–841, May 2000.