Software Availability Modeling Based on the Number of Restorations

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

1 Introduction

We construct a software availability model considering the number of restoration actions [1, 2]. We correlate the failure and restoration characteristics of the software system with the cumulative number of corrected faults. Furthermore, we consider an imperfect debugging environment where the detected faults are not always corrected and removed from the system. The time-dependent behavior of the system alternating between up and down states is described by a Markov process. From this model, we can derive quantitative measures for software availability assessment considering the number of restoration actions. Finally, we show numerical examples of software availability analysis.

2 Model description

The following assumptions are made for software availability modeling:

A1. The software system is unavailable and starts to be restored as soon as a software failure occurs, and the system cannot operate until the restoration action is complete.

A2. The restoration action implies the debugging activity; this is performed perfectly with probability a $(0 < a \leq 1)$ and imperfectly with probability $b(= 1 - a)$. We call a the perfect debugging rate. One fault is corrected and removed from the software system when the debugging activity is perfect.

A3. The next time intervals of software failures and restorations when n faults have already been corrected from the system, follow exponential distributions with means $1/\lambda_n$ and $1/\mu_n$, respectively.

A4. The probability that two or more software failures occur simultaneously is negligible.

Consider a stochastic process $\{X(t),\ t \geq 0\}$ whose state space is $(\mathbb{W},\ \mathbb{R})$, where up state vector $\mathbb{W} = \{W_n;\ n = 0,\ 1,\ 2,\ \ldots\}$ and down state vector $\mathbb{R} = \{R_n;\ n = 0, 1, 2, \ldots\}$. Then, the events $\{X(t) = W_n\}$ and $\{X(t) = R_n\}$ mean that the system is operating and inoperable due to the restoration action at time point t, when n faults have already been corrected, respectively.

From assumption A2, when the restoration action has been complete in $\{X(t) = R_n\}$,

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

The sample state transition diagram of $X(t)$ is illustrated in Fig. 1.

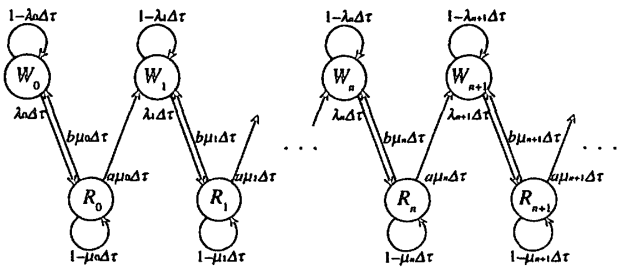

Fig.1 A diagrammatic representation of state transitions between $X(t)$'s.

3 Software availability measures

Here we consider the relationship between the number of the restoration actions and software availability measurement. Let $l = 0,\ 1,\ 2,\ \ldots$ denote the number of the restoration actions. Furthermore, we introduce the binary indicator variable $I_A(t)$ taking the value 1 (0) if the system is operating (inoperable) at time point t, given that it was in state $A \in (\mathbb{W},\ \mathbb{R})$ at time point $t = 0$, respectively. Then $A_i(t) \equiv \Pr\{I_{W_i}(t) = 1\}$ $(i = 0,\ 1,\ 2,\ \ldots)$ denotes the instantaneous software availability given that the system was in state W_i at time point $t = 0$, i.e.,

$$\begin{aligned} A_i(t) &= \sum_{n=i}^{\infty} P_{W_i,W_n}(t) \\ &= 1 - \sum_{n=i}^{\infty} P_{W_i,R_n}(t), \end{aligned} \tag{2}$$

(see Fig. 2), where $P_{A,B}(t)$ $(A,\ B \in (\mathbb{W},\ \mathbb{R}))$ denotes the state occupancy probability that the system is in state B at time point t on the condition that the system was in state A at time point $t = 0$.

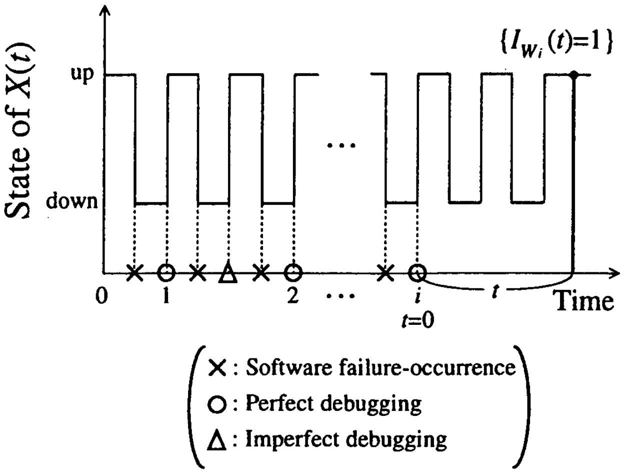

Fig.2 Sample behavior of the system and event $\{I_{W_i}(t) = 1\}$.

It is noted that the cumulative number of corrected faults at the completion of the l-th restoration action, C_l, is not explicitly observed since imperfect debugging is assumed throughout this paper. However, C_l follows the binomial distribution having the following probability mass function:

$$\Pr\{C_l = i\} = \binom{l}{i} a^i b^{l-i} \; (i = 0,\ 1,\ 2,\ \dots,\ l), \quad (3)$$

where $\binom{l}{i} \equiv l!/[(l-i)!i!]$ denotes a binomial coefficient. Accordingly, the instantaneous software availability after the completion of the l-th restoration action can be defined as

$$A(t; l) \equiv \sum_{i=0}^{l} \Pr\{C_l = i\} A_i(t), \quad (4)$$

which represents the probability that the system is operating at time point t, given that the l-th restoration action was complete at time point $t = 0$. Furthermore, the average software availability after the completion of the l-th restoration action can be defined as

$$A_{av}(t; l) \equiv \frac{1}{t} \int_0^t A(x; l) dx, \quad (5)$$

which represents the ratio of the system's operating time to the time interval $(0,\ t]$, given that the l-th restoration action was complete at time point $t = 0$. We can express (4) and (5) as

$$A(t; l) = 1 - \sum_{i=0}^{l} \binom{l}{i} a^i b^{l-i} \sum_{n=i}^{\infty} \frac{g_{i,n+1}(t)}{a\mu_n}, \quad (6)$$

$$A_{av}(t; l) = 1 - \frac{1}{t} \sum_{i=0}^{l} \binom{l}{i} a^i b^{l-i} \sum_{n=i}^{\infty} \frac{G_{i,n+1}(t)}{a\mu_n}, \quad (7)$$

respectively, where $g_{i,n}(t)$ and $G_{i,n}(t)$ are the probability density and the distribution functions of random variable $S_{i,n}$ representing the transition time from state W_i to state W_n $(i \leq n)$, respectively. These can be obtained as closed forms.

4 Numerical Examples

Figures 3 shows the time-dependent behaviors of the instantaneous software availability, $A(t; l)$ in (6) for various numbers of the restoration actions, l. This figure indicates that software availability drops rapidly immediately after the beginning of re-operation and then gradually increases. We can also see that software availability improves with the increasing number of the restoration actions.

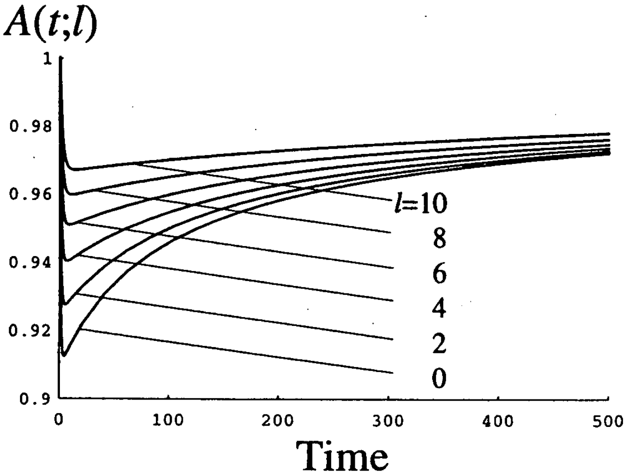

Fig.3 Dependence of number of restoration actions on $A(t; l)$ $(a = 0.9,\ D = 0.1,\ k = 0.8,\ E = 1.0,\ r = 0.9)$.

References

[1] K. Tokuno and S. Yamada, "A summary of Markovian software availability modeling," Proc. Fifth ISSAT International Conf. Reliability and Quality in Design, Las Vegas, Nevada, U.S.A., pp.218–222, Aug. 1999.

[2] K. Tokuno and S. Yamada, "Markovian modeling for software availability assessment," (in Japanese) Comm. Operations Research Society of Japan, vol.44, no.8, pp.405–409, Aug. 1999.

Acknowledgments

This work was supported in part by a Grant-in-Aid for Scientific Research from the Ministry of Education, Science, and Culture of Japan under Grant. No. 10680431.