1999年度日本オペレーションズ・リサーチ学会
春季研究発表会

Software Reliability Modeling for Safety/Availability Measurement

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

1 Introduction

We develop a stochastic software safety/availability assessment model. Our attention is directed to the event that the system causes hazardous conditions randomly in operation, not that the system may fall into an unsafe state when the system is down due to software failure-occurrence. This model is formulated by a Markov process to describe the time-dependent behavior of the software system, taking account of the software reliability growth process. In particular, this model can provide the metrics of *software safety* [1] defined as the probability that the system does not fall into hazardous states at a specified time point. Numerical illustrations are presented to show that these models are useful for software safety/availability measurement and assessment.

2 Model Description

The following assumptions are made for software safety/availability assessment modeling:

A1. When the software system is operating, the holding times of the safe and the unsafe state follow exponential distributions with means $1/\theta$ and $1/\eta$, respectively.

A2. The software system breaks down and starts to be restored as soon as a software failure occurs, and the system can not operate until the restoration action completes.

A3. The restoration action implies the debugging activity and software reliability growth occurs if a debugging activity is perfect.

A4. The debugging activity is perfect with probability a $(0 \leq a \leq 1)$, while imperfect with probability $b(=1-a)$. A perfect debugging activity corrects and removes only one fault from the system.

A5. When n faults have been corrected, the next software failure-occurrence time-interval and the restoration time follow exponential distributions with means $1/\lambda_n$ and $1/\mu_n$, respectively.

A6. The restoration actions are performed in safe states.

The state space of the process $\{X(t),\ t \geq 0\}$ representing the state of the software system at time point t is defined as follows:

W_n: the system is operating in a safe state,

U_n: the system is operating in an unsafe state,

R_n: the system is inoperable and restored,

where $n = 0, 1, 2, \ldots$ denotes the cumulative number of corrected faults.

From assumption A4, when a restoration action completes 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.

3 Software Safety/Availability Analysis

We can derive the state occupancy probabilities $P_{W_n}(t) \equiv \Pr\{X(t) = W_n\}$, $P_{U_n}(t) \equiv \Pr\{X(t) = U_n\}$, and $P_{R_n}(t) \equiv \Pr\{X(t) = R_n\}$ analytically.

Then, *the software safety* [2] is defined as

$$S(t) \equiv \sum_{n=0}^{\infty} [P_{W_n}(t) + P_{R_n}(t)], \tag{2}$$

which represents the probability that the software system does not fall into any unsafe states at time point t. Furthermore, *the instantaneous software availability* [3] is defined as

$$A(t) \equiv \sum_{n=0}^{\infty} P_{W_n}(t), \tag{3}$$

which represents the probability that the software system is operating safely at time point t.

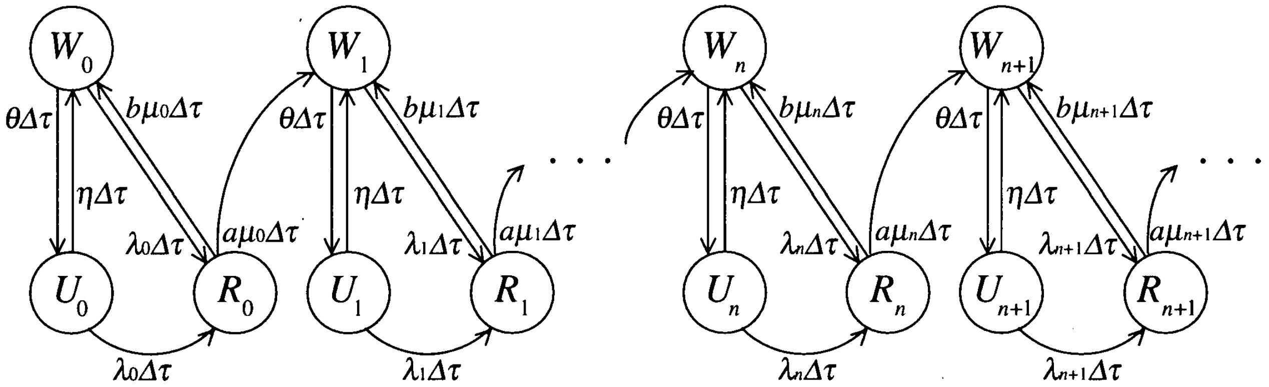

Fig.1 A diagrammatic representation of state transitions between $X(t)$'s.

4 Numerical Examples

Figure 2 shows the software safety, $S(t)$ in (2) for various values of a where $\lambda_n = Dk^n$ ($D > 0$, $0 < k < 1$) and $\mu_n = Er^n$ ($E > 0$, $0 < r < 1$). This figure indicates that the software safety decreases with increasing a, i.e., more rapid software reliability growth leads lower software safety. This reason is that this model assumes that the software failure-occurrence is not the unsafe state.

Figures 3 represents the instantaneous software availability, $A(t)$ in (3) for various values of a. $A(t)$ drops rapidly immediately after operation and improve gradually with the lapse of time. This figure also tells us that a system has higher availability with increasing a.

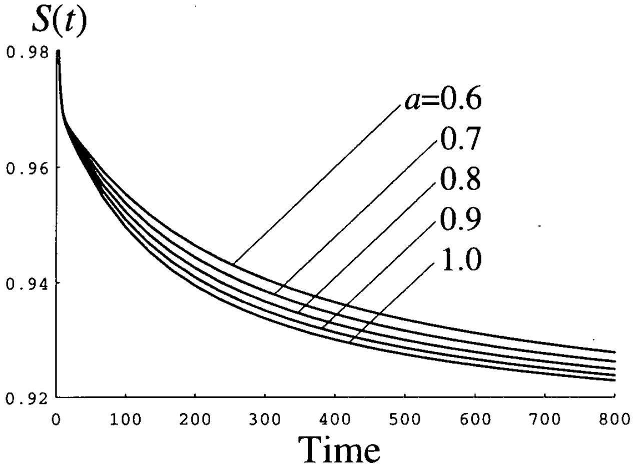

Fig.2 Dependence of a on $S(t)$ ($D = 0.1$, $k = 0.8$, $E = 0.2$, $r = 0.9$, $\theta = 0.01$, $\eta = 0.1$).

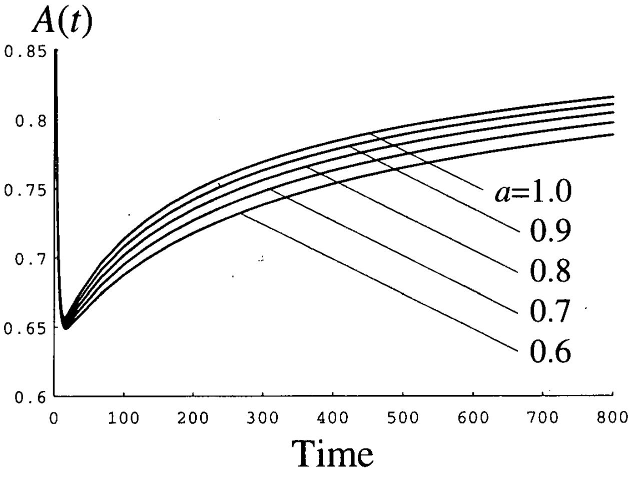

Fig.3 Dependence of a on $A(t)$ ($D = 0.1$, $k = 0.8$, $E = 0.2$, $r = 0.9$, $\theta = 0.01$, $\eta = 0.1$).

References

[1] N. G. Leveson, *Safeware: System Safety and Computers*, Addison-Wesley, New York, 1995.

[2] S. Yamada, K. Tokuno, Y. Kasano, "Quantitative assessment models for software safety/reliability" (in Japanese), Trans. IEICE A, vol J80-A, no 12, 1997, pp 2127–2137, and Electronics and Communications in Japan, Part II, vol 81, no 5, 1998, pp 33–43.

[3] K. Tokuno, S. Yamada, "Operational software availability measurement with two kinds of restoration actions", J. Quality in Maintenance Engineering, vol 4, no 4, 1998, pp 273–283.

Acknowledgment

This work was supported in part by the Grant-in-Aid for Scientific Research from the Ministry of Education, Science and Culture of Japan under Grant No. 10680451.