2－D－5

2001年度日本オペレーションズ・リサーチ学会
秋季研究発表会

Software Availability Modeling for Multi-Access System

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 an availability assessment model for a multi-access software system. We consider two types of software failures. One is due to the inherent faults remaining in the system and the other is due to the concentration of transactions caused by the simultaneous accesses of plural users. The former software failure-occurrence phenomenon is described by a decreasing hazard rate dependent on the number of corrected faults, i.e., the software reliability growth process is considered. The latter is described by one dependent on the number of accesses or users. The time-dependent behavior of the system alternating between up and down states is described by a Markov process [1]. Several software availability measures are derived as the functions of the time and the numbers of accesses or users and debugging activities [2].

2 Model description

We assume that the following two types of software failures exist during the operation phase:

F1: software failures caused by the faults that could not be detected/corrected during the testing phase.

F2: software failures caused by the simultaneous accesses of plural users such as the concentration of transactions.

The following assumptions are made for software availability modeling:

A1. The restoration action for F1 implies the debugging activity and software reliability growth occurs if a debugging activity is perfect. The restoration action for F2 has no effect on software reliability growth.

A2. The debugging activity for F1 is perfect with probability a $(0 < a < 1)$, on the other hand, imperfect with probability $b(= 1 - a)$. We call a the perfect debugging rate. A perfect debugging activity corrects and removes one fault from the system.

A3. When n faults have been corrected, the next occurrence time-interval and the restoration time for F1 follow the exponential distributions with means $1/\lambda_n$ and $1/\mu_n$, respectively. λ_n and μ_n are decreasing functions of n.

A4. The occurrence time-interval for F2 follows the exponential distribution with mean $1/\theta_m$, where $m > 0$ denotes the expected number of accesses or users. The restoration time for F2 follows the exponential distribution with mean $1/\eta$.

We introduce a stochastic process $\{X(t),\ t \geq 0\}$ representing the state of the system at time point t whose state space is defined as follows:

W_n: the system is operating,

R_n^1: the system is inoperable due to F1 and restored,

R_n^2: the system is inoperable due to F2 and restored,

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

It is often that the operational conditions not described in the specifications occur during the operation phase; for example, many transactions have rushed at some time point or the loads into some specified system resources have concentrated. For the above situation, it is fit to assume that such software failures occur randomly throughout the operation phase and that their occurrence frequency increases as the users of the system increase. Therefore, we can describe θ_m as

$$\theta_m = \alpha \cdot m \quad (m > 0;\ \alpha > 0), \tag{1}$$

where α is the hazard rate per user.

Figure 1 illustrates a sample state transition diagram of $X(t)$.

3 Derivation of measures

The conditional state occupancy probability that the system is in state W_n at time point t on the condition that the system was in state W_i at time point zero, provided there exist m users in the system, $P^m_{W_i,W_n}(t)$

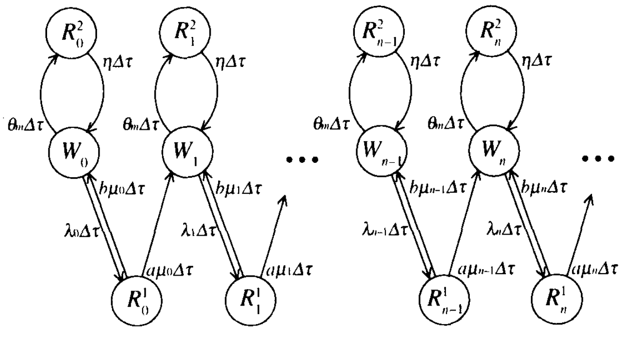

Fig.1 A sample state transition diagram of $X(t)$.

$(i \leq n)$, is given by

$$P^m_{W_i,W_n}(t) \equiv \Pr\{X(t) = W_n | X(0) = W_i\}$$
$$= \frac{1}{a\lambda_n} g^m_{i,n+1}(t) + \frac{1}{a\lambda_n \mu_n} g^m_{i,n+1}{}'(t), \quad (2)$$

where $g^m_{i,n}(t)$ is the probability density function associated with the transition time from state W_i to state W_n, and $g^m_{i,n}{}'(t) \equiv dg^m_{i,n}(t)/dt$.

Here we consider the relationship between the number of debugging activities and software availability measurement. Let $l = 0,\ 1,\ 2,\ \ldots$ denote the number of debugging activities. Furthermore, we introduce the binary indicator variable $I_{W_i}(t)$ taking the value 1 (0) if the system is operating (inoperable) at time point t, given that it was in state W_i at time point $t = 0$, respectively. Then $A_i^m(t) \equiv \Pr\{I_{W_i}(t) = 1\}$ $(i = 0,\ 1,\ 2,\ \ldots)$ denotes the instantaneous software availability when there exist m users in the system, provided the system was in state W_i at time point $t = 0$, i.e.,

$$A_i^m(t) = \sum_{n=i}^{\infty} P^m_{W_i,W_n}(t), \quad (3)$$

(see Fig. 2). It is noted that the cumulative number of corrected faults at the completion of the l-th debugging activity, C_l, cannot explicitly be observed since imperfect debugging is assumed throughout this paper. However, C_l follows the binomial distribution having the following probability mass function:

$$\Pr\{C_l = i\} = \binom{l}{i} a^i b^{l-i} \quad (i = 0,\ 1,\ 2,\ \ldots,\ l), \quad (4)$$

where $\binom{l}{i} \equiv l!/[(l-i)!i!]$ denotes the binomial coefficient.

Accordingly, the instantaneous software availability after the completion of the l-th debugging activity is given by

$$A(t; m, l) = \sum_{i=0}^{l} \Pr\{C_l = i\} A_i^m(t), \quad (5)$$

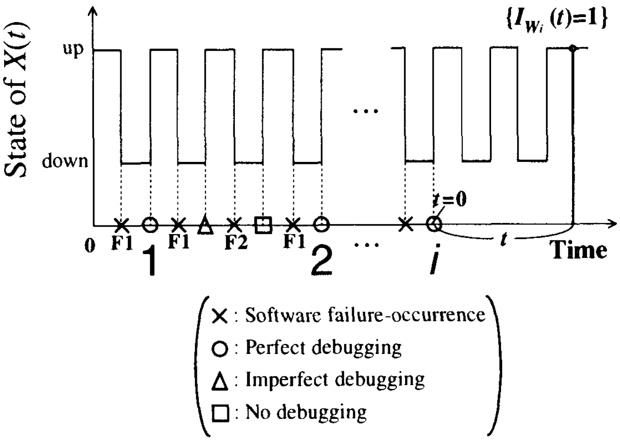

Fig.2 Sample behavior of the system and event $\{I_{W_i}(t) = 1\}$.

which represents the probability that the system is operating at the time point t when there exist m users in the system, provided the l-th debugging activity was complete at time point $t = 0$. Furthermore, the average software availability after the completion of the l-th debugging activity is given by

$$A_{av}(t; m, l) = \frac{1}{t} \int_0^t A(x; m, l) dx, \quad (6)$$

which represents the expected proportion of the system's operating time to the time interval $(0,\ t]$ when there exist m users in the system, provided the l-th debugging activity was complete at time point $t = 0$.

Acknowledgments

This work was supported in part by the Research Grant from the Telecommunications Advancement Foundation, the Sasakawa Scientific Research Grant from The Japan Science Society, and Grants-in-Aid from the Ministry of Education, Culture, Sports, Science and Technology of Japan under Grant Nos. 12680442 and 13780365, respectively.

References

[1] K. Tokuno and S. Yamada, "User-oriented software reliability assessment technology," (in Japanese) Bulletin of the Japan Society for Industrial Applied Mathematics, vol.10, no.3, pp.186–197, Sept. 2000.

[2] 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.