文章编号:2096-1472(2023)-07-54-04
DOI:10.19644/j.cnki.issn2096-1472.2023.007.012

基于失效模式的软件可靠性评估模型

田 雷[1],封 亮[1],李海峰[2]

(1.上海华讯网络系统有限公司,上海 201620;
2.北京航空航天大学,北京 100183)
✉ Tianlei@eccom.com.cn; fengliang@eccom.com.cn; lihaifeng@buaa.edu.com

摘 要:软件测试虽然可以提升揭错效率,但是未有对应的可靠性评估模型,难以实现精确的可靠性定量评估和预计。针对此问题,文章开展基于失效模式的软件可靠性评估模型构建研究。首先,基于运行剖面与输入空间计算失效模式概率;然后,构建基于马尔科夫过程与失效模式的软件可靠性评估模型,并在某型动力控制软件测试失效数据上开展工程应用。应用结果表明,软件可靠性评估模型的揭错效率提升了18.2%,测试工作量降低了54.4%,可以获得准确的软件可靠性评估结果,同时适用于软件可靠性测试。

关键词:失效模式;软件可靠性;可靠性模型;可靠性评估
中图分类号:TP311.5 **文献标识码**:A

A Software Reliability Evaluation Model Based on Failure Modes

TIAN Lei[1], FENG Liang[1], LI Haifeng[2]

(1.*Eccom Network System Co., Ltd., Shanghai* 201620, *China*;
2.*Beihang University, Beijing* 100183, *China*)
✉ Tianlei@eccom.com.cn; fengliang@eccom.com.cn; lihaifeng@buaa.edu.com

Abstract: Although software testing can improve error detection efficiency, there is no corresponding reliability evaluation model and therefore, it is difficult to achieve accurate quantitative reliability evaluation and prediction. Aiming at this issue, the paper proposes a software reliability evaluation model based on failure modes. Firstly, the probability of failure modes is calculated based on the operating profile and input domain. Then, the software reliability evaluation model based on Markov process and failure modes is constructed, and engineering applications are carried out on the failure data of a certain type of power control software testing. The application results show that the error detection efficiency of the proposed software reliability evaluation model increases by 18.2%, and the testing workload is reduced by 54.4%. Accurate software reliability evaluation results can be obtained, and it is suitable for software reliability testing.

Keywords: failure modes; software reliability; reliability model; reliability evaluation

0 引言(Introduction)

软件尤其是安全关键软件的失效可能会导致财产损失甚至人员伤亡。因此,可靠性已成为软件发布时用户最为关心的验证指标[1]。作为软件可靠性定量评估的重要手段,软件可靠性模型已经成功地应用于各种类型关键软件的开发过程[2-3]。

现有软件可靠性模型主要依据基于运行剖面的软件可靠性测试中收集的失效时间数据,对失效行为进行定量评估或预计,其存在揭错效率低下、工作量巨大等情况[4-8]。为解决这些问题,已有研究提出一个基于顺序统计量的软件可靠性模型处理混合测试数据,但其需要同时开展功能测试以及可靠性测

基金项目:装备预先研究共用技术项目(50902050302).

试,工作量较大[9];而基于失效模式的软件测试方法,虽然可以提升揭错效率,但是其产生的失效数据却不能用已有的模型进行处理,难以准确刻画软件运行时的可靠性。

因此,本文在传统软件可靠性模型的基础上,面向基于失效模式的软件测试过程,借助失效模式概率与马尔科夫过程,提出一种新的软件可靠性评估模型,并在某型动力控制软件测试失效数据上开展工程实践。应用结果表明,新模型可对软件测试过程中的失效数据进行处理,获得准确的软件可靠性定量评估结果。

1　基于运行剖面与输入空间的失效模式概率(The failure mode probability based on the operation profile and input domain)

1.1　软件失效模式概率概述

软件运行过程中接收外部输入数据,如果功能处理过程出现异常,则外部输出数据也会出现异常,导致软件运行出现失效(即软件动态执行的输出为不希望或不可接受的结果,是软件系统运行行为对用户要求的偏离),即"输入—处理—输出"软件失效链模型,该模型认为决定失效模式发生概率的主要因素有两点,即失效模式所在功能的执行概率,以及触发失效模式的输入空间分布概率[10]。

记失效模式 k 的发生概率为 φ_k,失效模式 k 所在的功能 m 的执行概率为 f_{pm},失效模式 k 的输入空间分布概率为 ε_{IPD},则失效模式 k 发生概率可由公式(1)计算:

$$\varphi_k = f_{pm} \times \varepsilon_{IPD} \tag{1}$$

由公式(1)可知,若想计算失效模式 k 的发生概率,需要先计算功能 m 的执行概率 f_{pm} 与失效模式 k 的输入空间分布概率 ε_{IPD}。

1.2　基于运行剖面的功能执行概率

假定软件共有 n 个功能,各项功能的执行概率记为 $\{fp_i\}_{i=1}^{n}$。依据软件运行剖面,确定功能之间转移关系的定量描述,即针对功能 f_i,假定其在运行剖面中共有 $path_i$ 条转移路径(从起始点到结束点之间的若干功能之间转移所组成的一条通路)可以经过功能 f_i,则称这些转移路径为功能 f_i 的可达路径。

假设功能 f_i 的可达路径 $K(K=1,\cdots,path_i)$ 上共有 K_{ik} 次转移,每次转移的发生概率记为 $tp_{ikj}(j=1,\cdots,K_{ik}, tp_{ikj}\leqslant 1)$,则功能 f_i 的每条可达路径上转移概率的乘积之和即为功能 f_i 的执行概率 fp_i,如公式(2)所示:

$$fp_i = \sum_{K=1}^{path_i} \prod_{j=1}^{K_{ik}} tp_{ikj} \tag{2}$$

1.3　软件输入空间分布概率

软件输入数据 C 的有效取值区间为 ψ,无效取值区间为 θ,则输入数据 C 的取值空间 $\vartheta=\psi\cup\theta$。可依据软件历史运行数据,对输入数据 C 取值区间的分布概率进行计算,具体计算方法如下。

假设共有 m 组运行数据,每组数据中均记录输入数据 C 的取值情况。假定第 i 组数据下,输入数据 C 在有效取值区间 ψ 的取值次数为 k_{iN},在无效取值区间 θ 的取值次数为 k_{iA},则输入数据 C 有效取值区间 ψ 的分布概率 ε_ψ 如公式(3)所示:

$$\varepsilon_\psi = \frac{\sum_{i=1}^{m} k_{iN}}{\sum_{i=1}^{m} k_{iN} + \sum_{i=1}^{m} k_{iA}} \tag{3}$$

输入数据 C 无效取值区间 θ 的分布概率 ε_θ 如公式(4)所示:

$$\varepsilon_\theta = 1-\varepsilon_\psi = 1 - \frac{\sum_{i=1}^{m} k_{iN}}{\sum_{i=1}^{m} k_{iN} + \sum_{i=1}^{m} k_{iA}} \tag{4}$$

1.4　基于功能执行与输入分布的失效模式概率

基于软件功能执行概率与输入分布概率的计算,本文提出软件失效模式概率的计算方法如下。

假设失效模式 k 的发生概率记为 φ_k,其所在功能 m 的执行概率记为 f_{pm},并且失效模式 k 与功能 m 的一个或多个不同类型的输入数据相关。

假定失效模式 k 对应一个输入数据 C,若该输入数据 C 在无效取值区间 θ 取值时,导致失效模式 k 发生,则可依据公式(4)计算无效取值区间的分布概率 ε_θ。

依据上述计算结果,可得到软件失效模式 k 的发生概率 φ_k 如公式(5)所示:

$$\varphi_k = f_{pm} \times \varepsilon_\theta \tag{5}$$

2　基于马尔科夫过程与失效模式的软件可靠性评估模型(Software reliability evaluation model based on Markov process and failure modes)

2.1　基于失效模式概率的功能失效率评估

假定软件失效模式总数为 N,在测试过程中共探测到 m 个失效模式,每个失效模式的发生概率记为 $\varphi_k, k=1,\cdots,m$。整个软件测试过程中,累积失效模式概率记为 $\{k,\varphi_k|k=1,\cdots,m\}$,$k$ 表示测试过程中的累积失效模式个数;φ_k 为测试过程中的累积失效模式概率,即 $\varphi_k=\sum_{j=1}^{k}\varphi_j$。

将软件测试结束时的功能失效率记为 λ_O,软件测试开始时的功能失效率记为 λ_I。本论文提出如下几个假设,为软件功能失效率的评估奠定基础。

假设 1:功能失效率是当前残存失效模式的发生概率之和。

依据假设 1,则 $\lambda_O=\sum_{j=m+1}^{N}\varphi_j$,$\lambda_I=\lambda_O+\sum_{j=1}^{m}\varphi_j=\sum_{j=1}^{N}\varphi_j=\varphi_N$。

假设 2:考虑到学习因素的影响,软件测试过程中累积失效模式概率的增长速率可能会呈现先增后减的"S"形增长趋势。

根据假设 2,本论文采用变形"S"形函数描述累积失效模式概率 φ_k 的这种"S"形增长趋势,如公式(6)所示:

$$\varphi_k = \frac{q\varphi_{\max}(1-\exp(-bk))}{1+c\exp(-bk)} \tag{6}$$

其中，$\varphi_{\max}=\lambda_I=\varphi_N$，c，b，q 为参数常量。

假设 3：假定功能失效模式总数 N 是有限的。

此时，$\varphi_k=\frac{q\varphi_{\max}(1-\exp(-bk))}{1+c\exp(-bk)}$。根据软件测试过程中收集到的失效模式概率数据 $\{k,\varphi_k\mid k=1,\cdots,m\}$，利用最大似然法对函数 $\varphi_k=\frac{q\varphi_{\max}(1-\exp(-bk))}{1+c\exp(-bk)}$ 进行数据拟合，即可得到 $q\varphi_{\max}$、c 与 b 的估计值，分别记为 $q\hat{\varphi}_{\max}$、$\hat{c}$、$\hat{b}$。

当 $k=N$ 时，$\frac{1}{q}=\frac{1-\exp(-bN)}{1+c\exp(-bN)}$。此时，c 与 b 是已知的（即估计值 $\hat{c}$、$\hat{b}$），令 N 分别等于 m，$m+1$，$\cdots$，$m+n$，依次带入 $\frac{1}{q}=\frac{1-\exp(-bN)}{1+\beta\exp(-bN)}$，求得参数 q 的若干估计值 $\hat{q}$，待估计值 $\hat{q}$ 的变化趋势呈现为一个稳定状态时，可将此时的估计值 $\hat{q}$ 作为 q 的最佳近似估计值，此时的 N 值作为功能失效模式总数的近似估计值。

根据前面求得的估计值 $q\hat{\varphi}_{\max}$ 与 $\hat{q}$，即可得 $\varphi_{\max}$ 的估计值 $\hat{\varphi}_{\max}$ 如公式(7)所示：

$$\hat{\varphi}_{\max}=q\hat{\varphi}_{\max}/\hat{q} \tag{7}$$

进而获得功能失效率的估计值 $\hat{\lambda}_O$ 如公式(8)所示：

$$\hat{\lambda}_O=\hat{\varphi}_{\max}-\sum_{j=1}^{m}\varphi_j \tag{8}$$

2.2 基于马尔科夫过程的软件可靠性建模

马尔科夫过程具有“无后效性”，即系统在下一时刻所要执行的功能，仅取决于当前时刻的执行功能。针对长时间连续运行的软件，功能之间的转移关系通常近似服从马尔科夫过程。基于上述分析，本论文提出如下建模假设。

假设 1：软件可划分为有限个独立的功能模块。

假设 2：软件运行过程中，各项功能之间的转移关系服从马尔科夫过程。

假定软件具有 n 个功能，功能 i 转移到功能模块 j 的转移概率记为 p_{ij}。将转移概率 p_{ij} 依次排列，构成功能转移概率矩阵 $\boldsymbol{P}$ 如公式(9)所示：

$$\boldsymbol{P}=[p_{ij}]_{n\times n}=\begin{bmatrix}p_{11}&p_{12}&\cdots&p_{1n}\\p_{21}&p_{22}&\cdots&p_{2n}\\\cdots&\cdots&&\cdots\\p_{n1}&p_{n2}&\cdots&p_{mn}\end{bmatrix} \tag{9}$$

功能转移概率矩阵 $\boldsymbol{P}$ 是一个 n 阶矩阵，具有如下性质。

性质 1：$p_{ij}\geqslant 0$，即每个元素均是非负的；

性质 2：$\sum_{j=1}^{n}p_{ij}=1$，即矩阵每行的元素和等于 1。

由于软件可靠性表示软件最终成功地完成任务的概率，因此对于假定服从马尔科夫过程的软件可靠性 R，可以用功能模块失效率 λ_i 和功能转移概率矩阵 $\boldsymbol{P}$ 表示。即对于有 n 个功能的软件，软件可靠性函数可表示为公式(10)：

$$R\{\lambda_1,\lambda_2,\cdots,\lambda_n;t\}=\exp\left(-\sum_{j=1}^{n}\sum_{i=1}^{n}p_ip_{ij}\lambda_it\right) \tag{10}$$

其中，p_i 表示功能模块 F_i 的执行概率，p_{ij} 表示在下一项操作时由功能模块 F_i 迁移到模块 F_j 的概率，λ_i 表示功能模块 F_i 的失效率，t 表示软件运行或测试时间。

3 实例应用(Case study)

本论文针对某型动力控制系统软件开展工程应用，验证基于失效模式的软件可靠性评估模型的有效性和可行性。具体的应用步骤与评估结果如下。

(1)应用概述。针对某型动力控制系统软件进行基于失效模式的软件可靠性测试，共执行测试用例 185 项，总测试时间约 502 h，确认 13 项软件问题（对应 13 项软件失效模式，问题编号分别为 REQ-01 至 REQ-13），受篇幅限制，问题内容不再详述。

(2)软件失效模式概率计算。针对软件外场运行数据进行统计分析，评估运行剖面中每项功能的执行概率。以“初始化功能”为例，其在外场运行过程中，有效执行时间为 43 h，而软件的总运行时间总计 863 h。所以，“初始化功能”的执行概率为 43/863≈0.049 8。由此也可获得该动力控制软件全部功能的执行概率计算结果，如表 1 所示。

表 1 软件功能执行概率

Tab. 1 Operation probabilities of the software functions

序号	功能名称	运行时间/h	执行概率
1	初始化功能	43	0.049 8
2	启动功能	30	0.034 8
3	自检维护功能	37	0.042 9
4	压力控制功能	383	0.443 8
5	动力控制功能	263	0.304 7
6	温度控制功能	61	0.070 7
7	停止功能	46	0.053 3
	总计	863	1

针对软件外场运行数据进行统计分析，确认每项外部输入接口数据在不同值域的取值概率。以输入接口数据“启动信号”为例，该数据为离散型，取值值域为{0，1}，取值为 0 表示启动无效（即异常值），1 表示启动有效（即正常值）。在外场运行过程中，数据取值为 0 的时间为 24.6 h，取值为 1 的时间为 838.4 h。所以，“启动信号”取值为 0 的概率为 24.6/863≈0.028，取值为 1 的概率为 1−0.028=0.972。

依据表 1 中的功能执行概率，以及外部输入接口数据的取值概率，对每项失效模式的发生概率进行计算。本文以失效模式“REQ-01”为例，说明该失效模式的发生概率计算过程如下。

首先，确定失效模式“REQ-01”对应功能的执行概率。该失效模式与“启动功能”相关，由表 1 可知，“启动功能”的执行概率为 0.034 8；然后，确定失效模式 REQ-01 对应外部输入接口数据的取值概率分布。该失效模式是由“启动信号”取值为 0（异常值）引发的。所以，该失效模式对应的外部接口数据取值概率应为取值为 0 的发生概率，即 0.028。最后，计算失效模式 REQ-01 的发生概率为 0.034 8×0.028=0.000 974 4。

(3)基于失效模式的功能失效率计算。依据失效模式的发生概率,以及功能执行概率,计算失效模式相关功能的失效率。以"启动功能"为例,说明该项功能失效率的计算过程。在本次软件可靠性测试过程中,与"启动功能"相关的失效模式为REQ-01、REQ-02、REQ-03,进而根据模式假设3进行功能失效率计算。

本论文利用最小二乘法对函数 $\varphi_k=\frac{q\varphi_{max}(1-\exp(-bk))}{1+c\exp(-bk)}$ 进行数据拟合(即失效模式REQ-01、REQ-02、REQ-03的发生概率),得到 $q\hat{\varphi}_{max}=0.000\ 208\ 7$,$\hat{q}=0.003\ 192$,$\hat{\varphi}_{max}=0.065\ 390\ 2$。然后,依据 $\hat{\varphi}_{max}-\sum_{j=1}^{m}\varphi_j$,可获得"启动功能"失效率 $\hat{\lambda}_O$ 的估计值如公式(11)所示:

$$\begin{aligned}\hat{\lambda}_O &=0.065\ 390\ 2-0.000\ 974\ 4-0.002\ 732\ 4-0.010\ 184\ 4\\ &=0.051\ 499\end{aligned}\tag{11}$$

(4)基于功能失效率的软件失效率计算。依据功能失效率以及表1中的功能执行概率,对软件失效率 λ 进行计算,具体过程如下。

依据式 $\lambda=\lambda_1p_1+\lambda_2p_2+\cdots+\lambda_np_n$,需确定动力控制软件的所有失效模式发生概率及相应功能的执行概率。软件失效率 $\hat{\lambda}$ 经计算得

$$\begin{aligned}\hat{\lambda}=&0.000\ 795\ 4\times0.049\ 8+0.051\ 499\times0.034\ 8+\\ &0.008\ 721\times0.042\ 9+0.005\ 266\times0.443\ 8+\\ &0.007\ 163\times0.304\ 7+0.009\ 925\times0.070\ 7+\\ &0.006\ 871\times0.053\ 3=0.009\ 976\end{aligned}$$

即动力控制系统软件的平均失效前间隔时间 $MTBF=1/\hat{\lambda}\approx100.24$ h。

(5)与传统软件可靠性测试评估方法的比对分析。该动力控制系统软件曾经进行过基于运行剖面的传统软件可靠性测试评估试验,共用时1 100 h,发现软件问题11项。将这种基于运行剖面的传统软件可靠性测试评估方法(传统方法),与本文提出的基于失效模式的软件可靠性测试评估方法(新方法)相比,可以得出如下对比分析结果。①新方法可有效提升软件的揭错效率。本论文借助基于失效模式的软件可靠性测试,共识别13项软件问题;而传统的基于运行剖面的软件可靠性测试只发现了11项问题。因此,相比传统可靠性测试,本文所提软件的揭错效率提升了(13−11)/11×100%≈18.2%。②新方法显著降低软件测试工作量。基于失效模式的软件可靠性测试所用的测试时间仅为502 h,而传统软件可靠性测试则用时为1 100 h。因此,测试工作量降低了(1 100−502)/1 100×100%≈54.4%。③新方法可获得与传统方法同样准确的软件可靠性评估结果。借助经典软件可靠性模型(GO模型),计算软件的 $MTBF$ 为103.41 h。借助本论文提出的模型,计算软件的 $MTBF$ 为100.24 h,二者之间的误差仅为3.1%。因此,与经典的GO模型相比,本文所提模型也可以获得较为准确的可靠性评估结果。

4　结论(Conclusion)

本文提出一种新的基于失效模式的软件可靠性定量评估模型。首先,借助运行剖面与输入空间计算失效模式概率,进而评估软件功能失效率;然后,借助马尔科夫过程实现基于失效模式的软件可靠性定量评估。实例应用结果表明,本文提出的基于失效模式的软件可靠性测试方法以及评估模型,可提升18.2%的揭错效率,降低54.4%的测试工作量,可以获得准确的软件可靠性评估结果,同时适用于软件可靠性测试。

本文所提可靠性评估模型的准确性可能受限于软件测试的充分性,即如果软件测试数据未覆盖所有功能及其接口时,可能会对失效模式概率以及功能执行概率的评估值产生影响。因此,在未来研究工作中,将考虑采用软件仿真数据(或运行数据)与测试数据相结合的方式优化该模型的适应性和准确性。

参考文献(References)

[1] HUANG C Y, LYU M R. Estimation and analysis of some generalized multiple change-point software reliability models[J]. IEEE Transactions on Reliability, 2011, 60(2):498-514.

[2] MUSA J D. Software Reliability Engineering[M]. New York:McGraw-Hill Book Company,1996:102.

[3] ZHANG X M. An analysis of factors affecting software reliability[J]. The Journal of Systems and Software, 2000(50):43-51.

[4] LYU M R. Handbook of Software Reliability Engineering[M]. New York: McGraw-Hill Book Company, 1996:439.

[5] LI Q Y, HOANG P. NHPP software reliability model considering the uncertainty of operating environments with imperfect debugging and testing coverage[J]. Applied Mathematical Modeling,2017(51):68-85.

[6] 李海峰,李秋英,陆民燕. 考虑S型测试工作量函数与不完美排错的软件可靠性模型[J]. 哈尔滨工程大学学报,2011,32(11):1460-1467.

[7] ZHANG C, CUI G, LIU H W. A unified and flexible framework of imperfect dependent SRGMs with testing-effort[J]. Journal of Multimedia,2014,9(2):310-317.

[8] 王二威. 软件可靠性模型研究综述[J]. 软件工程,2016,19(2):1-2,57.

[9] BRIAN M. A software reliability model combining representative and directed testing[D]. Virginia:Old Dominion University,2001.

[10] VOAS J. PIE:a dynamic failure based technique[J]. IEEE Transactions on Software Engineering, 1992, 8(18):717-727.

作者简介:

田　雷(1980-),男,本科,中级工程师. 研究领域:软件工程,软件供应链安全.

封　亮(1977-),男,本科,高级工程师. 研究领域:软件开发,信息研究.

李海峰(1981-),男,博士,高级工程师. 研究领域:软件可靠性,软件安全性.