

关于软件发布后的改进型软件可靠性预计模型

潘彬彬, 沈元隆

(南京邮电大学 电子科学与工程学院, 江苏 南京 210003)

摘 要:针对2004年Pankaj Jalote和Brendan Murphy提出的P-B模型假设过于苛刻、适用范围较窄,文中提出了一种新的基于软件发布后的可靠性模型。该模型不仅考虑了原P-B模型中用户对可靠性产生的影响,而且还考虑了通过版本升级和打补丁修正软件本身错误对软件可靠性产生的影响,同时将这种影响反映到模型的衰减因子中,进而对原始模型的衰减因子进行修正,并通过最小二乘法对新模型参数进行估计。最后通过仿真实验证明了,改进后的新模型比原始模型的预测精度更高,更符合实际失效数目的变化。

关键词:软件可靠性;衰减因子;失效率;版本升级;软件补丁

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2011)09-0085-03

Research of Software Reliability Model after Software Released

PAN Bin-bin, SHEN Yuan-long

(Dept. of Electronics Science and Technology, Nanjing University of Posts and Telecommunications,
Nanjing 210003, China)

Abstract: Because the assumption of P-B model raised by Pankaj Jalote and Brendan Murphy was too rigorous and the applicable scope was narrow, a reliable model based on the publication of software was brought up. This model not only considered the effect caused by the users of the original P-B model, but also took the influences of modifying errors of software itself by edition upgrading and patching into account, by taking the effect into the decay factor of the mode, modifying the decay factor of the original model and estimating parameters of the new model by least square method. At last, the simulation experiment proved that the improved new model was more accurate on calculation and more suitable to the real changes of the amount of parameters of software failure.

Key words: software reliability; decay factor; failure rate; edition upgrading; software patch

0 引言

目前,大部分软件可靠性预计模型都着眼于在软件生命周期内从软件开发者的角度去预测和防止软件本身所隐含的缺陷以此来降低软件的失效率以提高软件的可靠性,它们都建立在软件的可靠性完全依赖于能够诱发软件失效的缺陷被排除而改善的基础上^[1]。这类模型能够比较正确地衡量软件产品在发布前的可靠性,却无法恰当地描述软件产品投入市场后的可靠性,因为一旦软件发布,投入市场后,它的可靠度不但受软件缺陷的影响同时也会因为用户对软件使用熟练度、软件运行环境、软件设置等因素的改变而改变,随着使用时间的延长,可靠度会有明显的增长。换句话说,对于发布后的软件产品,其在没有消除软件中错误

的情况下软件产品的可靠性增长了,这种关于产品发布后软件可靠性的增长不同于传统的可靠性增长^[2]。它的产生主要有以下几个原因:第一,用户通过对软件产品一段时间的使用后,对软件的熟悉程度有所提高,可以有意识地在用这种软件的过程中避免由于使用不当而造成的软件失效,甚至可以有效地避免一些由于某些无意识的命令和操作而触发的软件内在的缺陷而导致的失效问题。也就是说,在没有消除软件本身错误的情况下,随着用户对软件熟练度的提高,可以避免由于自身操作而引起的失效,从而提高了软件的可靠性^[3]。第二,软件设置的修正。从报纸、文献及各产品服务部门的数据显示有超过一半的软件产品失效并不是由软件本身的缺陷造成的,而是由用户对软件设置不当而造成的^[4]。因此通过对软件设置的更正从而提高了软件的可靠性。第三,软件运行环境的改善。用户通过升级系统或更换冲突的软件,提高了软件的可靠性^[5]。为了描述这种软件产品可靠性增长的现象,2004年Pankaj Jalote和Brendan Murphy提出了一种关于软件产品发布后的软件可靠性预计模型^[6]

收稿日期:2011-01-08;修回日期:2011-04-15

基金项目:南京邮电大学校科研基金(01院32)

作者简介:潘彬彬(1982-),男,江苏宜兴人,硕士研究生,研究方向为通行系统可靠性分析;沈元隆,教授,研究方向为通信系统的可靠性技术。

(以下简称 P-B 模型)。但是,原始的 P-B 模型只从用户的角度出发去建立模型。它忽略了软件本身隐含的错误在软件发布后依然会对软件的可靠性产生重大的影响。文中充分考虑了软件本身和用户这两个方面,通过引入 β, γ, θ 三个参数对原始 P-B 模型的衰减因子进行修正,使得改进后的模型预测能力得到了很大的提高。

1 新模型的出发点

无论是在软件开发过程中还是在软件发布后,影响软件可靠性的因素很多,到目前为止还没有能够广泛应用于各类软件的可靠性模型,因为不同的开发人员对影响软件可靠性因素的取舍不同,反映到模型中就是各模型的假设不同,甚至对于同一个软件如果假设条件不同的话也会对应不同的可靠性模型,而评判模型优劣的关键在于所建立的模型是否更贴近于实际,这在很大程度上又受到模型中假设的限制^[7-12]。对于软件发布后的软件可靠性模型—P-B 模型,它在假设中不考虑新版本和软件补丁的发布,即它忽略了软件在发布后通过版本升级和打补丁来修正其自身错误对可靠性产生的影响而只单纯的从用户角度去衡量。但是在现实中无论哪个软件,每隔一段时间都会有新版本或者是补丁发布,而绝大多数用户都会选择及时地进行版本升级和打补丁,因为版本的升级和打补丁往往是对原来版本中缺陷的弥补,同时提供一些新的功能,而这种新功能的引入往往又会给软件本身带来新的缺陷,在这种情况下原始的 P-B 模型就不能很好地描述现实中软件产品在发布后可靠性的提高。基于此,文中在原有 P-B 模型的基础上,充分考虑由于版本升级和打补丁对软件本身可靠性提高产生的正反两方面的影响。

2 新模型的假设

依据上面的分析,文中对原始的 P-B 模型假设进行重新修正,提出一种新的改进模型,新模型的假设如下:

① 假定软件销售出去以后,新的版本和补丁及时发布,用户能及时安装和使用,新的补丁安装后不会带来新的错误,新的错误完全由版本的升级所引入;

② 假定用户初始的失效率为 $(\lambda_0 + \lambda_f)$, 稳定时的失效率为 λ_f ;

③ 假定可以通过软件产品技术支持服务获得每个月的产品销售量与每个月的总的失效率;

④ 针对用户来说,由于考虑到了版本升级和补丁的影响以及会带来新的缺陷问题,所以软件的每个月失效率为 α_i , 并且 α_{i+1} 为 α_i 的 k ($k > 1$) 倍直到它达到

稳定的状态。

由上述的假设,可以得到此模型的失效率公式为

$$\lambda(i) = \lambda_0 \times \prod_{m=1}^i \alpha_m + \lambda_f \quad (i \text{ 为时间}) \quad (1)$$

$$\text{其中 } \alpha_i = \prod_{n=1}^{i-1} [k^{n-1} \alpha - \sum_{m=1}^n k^{n-m} (\beta^n - \gamma^m + \theta^m)]$$

(式中的 β, γ, θ 描述的是软件版本升级和打补丁对衰减因子产生的影响)

新模型的失效数预测公式为:

$$\begin{aligned} F_1 &= \lambda_0 N_1 + \lambda_f N_1 \\ F_2 &= \lambda_0 N_2 + \lambda_0 N_1 (\alpha - \beta + \gamma - \theta) + \lambda_f (N_1 + N_2) \\ F_i &= \lambda_0 N_i + \lambda_0 (\alpha - \beta + \gamma - \theta) N_{i-1} + N_{i-2} (\alpha - \beta + \gamma - \theta) [(\alpha - \beta + \gamma - \theta) - \beta^2 + \gamma^2 - \theta^2] + \dots \\ &+ N_1 \prod_{n=1}^{i-1} [k^{n-1} \alpha - \sum_{m=1}^n k^{n-m} (\beta^n - \gamma^m + \theta^m)] \end{aligned} \quad (2)$$

3 新模型的参数估计

最常用的经典参数估计方法有两种:最大似然法和最小二乘法,文中利用收集的每月失效总数 f_i 和产品销售数(见表1)采用最小二乘法对 $\lambda_0, \lambda_f, k, \alpha, \beta, \gamma, \theta$ 进行估计。公式如下:

表1 某公司的每月销售量和每月的失效总数数据

月份	失效总数	每月的销售数量
1	367	4618
2	853	14385
3	835	5608
4	791	6186
5	956	9829
6	805	5584
7	967	8240
8	1218	7656
9	1031	4914
10	1144	5295
11	1058	7418

$$\begin{aligned} X &= \sum_{i=1}^k (f_i - F_i)^2 = \\ &\sum_{i=1}^k \{f_i - \lambda_0 N_i - \lambda_0 N_{i-1} (\alpha - \beta + \gamma - \theta) - \\ &\lambda_0 N_{i-2} (\alpha - \beta + \gamma - \theta) [k(\alpha - \beta + \gamma - \theta) - \beta^2 + \gamma^2 - \\ &\theta^2] - \dots - \lambda_0 N_1 \prod_{n=1}^{i-1} [k^{n-1} \alpha - \sum_{m=1}^n k^{n-m} (\beta^n - \gamma^m + \theta^m)] \\ &- \lambda_f \sum_{j=1}^i N_j\}^2 \end{aligned} \quad (3)$$

为了便于书写,将

$$\begin{aligned} &N_i + N_{i-1} (\alpha - \beta + \gamma - \theta) + N_{i-2} (\alpha - \beta + \gamma - \theta) [k(\alpha - \\ &- \beta + \gamma - \theta) - \beta^2 + \gamma^2 - \theta^2] + \dots + N_1 \prod_{n=1}^{i-1} [k^{n-1} \alpha - \end{aligned}$$

$$\sum_{m=1}^n k^{n-m} (\beta^m - \gamma^m + \theta^m)]$$

记作 A 。则(3)式简化成:

$$X = \sum_{i=1}^k [f_i - \lambda_0 A - \lambda_f \sum_{j=1}^i N_j]^2 \quad (4)$$

将式(4)代入 $\frac{\partial X}{\partial \lambda_0} = 0, \frac{\partial X}{\partial \lambda_f} = 0, \frac{\partial X}{\partial k} = 0, \frac{\partial X}{\partial \alpha} = 0, \frac{\partial X}{\partial \beta} = 0, \frac{\partial X}{\partial \gamma} = 0$ 和 $\frac{\partial X}{\partial \theta} = 0$ 中可得到关于 $\lambda_0, \lambda_f, k, \alpha, \beta, \gamma, \theta$

的方程,然后利用2004年Pankaj Jalote和Brendan Murphy在ACM发表的文献中提供的数据(见表1)。

利用数值法可以求得该模型的参数,如表2所示。

表2 模型参数

λ_0	λ_f	α	k	β	γ	θ
0.0488	0.0070	0.4025	1.1646	0.7108	0.9125	0.7213

4 实验数据仿真对比

利用表2求得的参数值将其带入式(2)中可以得到新模型的失效数预测公式。并将新模型的预测值、原模型预测值及实际失效数列入表3中。

表3 新模型与原始模型预测失效数

月份	实际失效总数	原始模型预测值	新模型预测值
1	367	242	258
2	853	882	812
3	835	767	803
4	791	772	780
5	956	993	967
6	805	911	801
7	967	1040	993
8	1218	1102	1198
9	1031	1020	1033
10	1144	1026	1123
11	1058	1164	1076

依据此表得到原始模型和新模型的拟合曲线如图1所示。

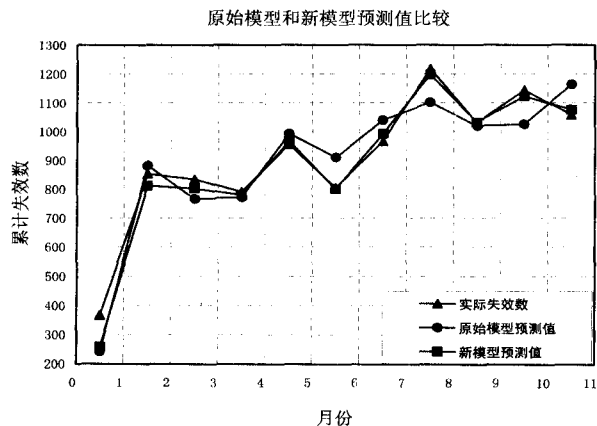


图1 原始模型及新模型的拟合曲线比较

通过图1即实际失效数、原始模型预测值和新模型预测值的拟合曲线比较可见,新模型的预计拟合曲线比原模型的预计拟合曲线更接近于实际错误数的变化,也就是说新模型的预测精度要比原始模型的预测精度高,更符合实际的变化。因此,在P-B模型的基础上考虑版本升级和打补丁对软件可靠性产生的影响是十分必要的。

5 结束语

文中对原始的关于软件发布后的可靠性模型(P-B模型)进行了分析和改进,提出了一种更符合实际的新模型。但是软件可靠性工程是一个发展迅猛、博大精深、文中对该领域的研究只能算得上是沧海一粟,仍然存在许多需要进行进一步研究和探索的地方:

(1) 用户端软件失效数的收集精确与否是影响模型预测能力的重要内容,文中所收集的数据来自于某企业的售后部门,有部分产品可能是销售到了二级代理商手里并没有及时的到达用户方。如果能够建立一个销售和失效历史信息库用来保存更完善的用户端数据将它们应用于该模型中将能更准确地反映此软件的可靠性活动的规律。

(2) 文中采用了概率论与数理统计的可靠性分析方法对该软件进行了可靠性的分析,但是,这些方法往往只强调相关参数的随机性而忽视了软件系统的复杂性和人们认识客观世界的模糊性。所以可以将模糊技术和神经网络技术应用于软件可靠性研究。

(3) 文中所建立的模型只是对于Pankaj Jalote和Brendan Murphy提供的实例适用,是否在一定范围内广泛适用仍然需要大量类似的实例故障数据验证。

参考文献:

- [1] 蔡开元. 模糊可靠性研究[D]. 北京:北京航空航天大学, 1991.
- [2] 徐仁佐. 软件可靠性专家系统(SRES)开发[M]. 北京:清华大学出版社, 1996.
- [3] Yamada S, Ohba M, Osaki S. S-shaped Reliability Growth Modeling for Software Error Detection[J]. IEEE Trans. on Reliability, 1983(5): 475-478.
- [4] Littlewood B, Verrall J. A Bayesian Reliability Growth Model for Computer Software[J]. Journal of the Royal Statistical Society, 1973, 22(3): 332-346.
- [5] 文长康. 浅谈软件可靠性的评估技术[J]. 通信技术, 1997, 14(2): 73-78.
- [6] Jalote P, Murphy B. Reliability Growth in Software Products[C]//ISSRE, 15th international symposium on software re-

(下转第91页)

2.3 结合 CCD 算法实现人体步行运动的一条思路

上节所述的 CCD 实现方法在描述人体上肢运动比如伸手拿物体时,有可能没有解析解,这种情况的出现说明物体在人体的可触及范围以外。而步行运动中一般不会出现这样的问题。步行运动相对于伸手拿物体的运动要简单一些,因为人体在平坦地面的步行运动遵循一定的规律^[12]。而在传统的步行运动研究中所涉及的环境多数是平坦地面,不需要考虑环境对运动的影响。由于步行运动是周期性的运动,因此只要把一个周期内的运动设计好就可以重复实现人体的步行运动。

在用 CCD 算法逆向求解步行运动时,脚掌作为末端效应器,需要求解位姿的关节主要有踝关节、膝关节、股关节以及根关节(此时可以看作人体的重心)。人体的步行运动可看成是由摆动腿踏出实现迈进的单脚支撑期和双脚接地移送重心的双脚支撑期两个差异较大的运动构成。

(1) 在平坦地面上的步行运动。

当虚拟人在平坦地面上行走时,运动是周期性重复的,需要求解的主要是单脚抬起时各个关节的位姿。此时的运动关节主要是膝关节和股关节,因此在运用 CCD 算法进行迭代时,只需处理这两个关节即可,大大减少了计算量。当双脚接地移送重心时,保证双脚落地时重心位于两脚的中间位置即可。

(2) 上下台阶时的运动。

当虚拟人在非平坦地面上行走时,例如上下台阶,此时人体的摆动腿踏出后的落地点是根据台阶的高度而确定的。在处理此类运动时,需要预先设定好人体抬腿高度的极大值以及人体正常步行时的抬腿高度。当人体上台阶时通过碰撞检测确定了台阶的高度后,人体的抬腿高度和落地点也就确定了。如果台阶的高度大于人体抬腿高度的极大值,则认定人体无法上台阶,运动失败。反之,则运用 CCD 算法通过目标点的位置(即迈出脚的落地点)来反求出虚拟人下体其它关节的位姿。

此方法相对运动捕捉法虽然增加了计算量,但是使人体的步行运动更加灵活多样,克服了运动捕捉法只能复现单一运动的缺点。

3 结束语

文中就虚拟人运动建模方法和逆向运动学算法作了初步的探讨,提出了一条基于 CCD 算法实现人体步行运动的思路,使虚拟人能够根据环境约束调整运动行为。但此方法需要实时精确的碰撞检测作为基础,且需要合适的关键帧插值技术来实现完整流畅的虚拟人动画,这些正是下一步要研究和实现的问题。

参考文献:

- [1] 洪炳熔, 贺怀清. 虚拟人的步行和跑步运动控制方法的研究[J]. 高技术通讯, 2001(3): 91-95.
- [2] Terra S C L, Metoyer R A. Performance timing for key-frame animation[C]//Eurographic/ACM Symposium on Computer Animation. [s. l.]: [s. n.], 2004: 253-258.
- [3] Tolani D, Goswami A, Badler N I. Real-time inverse kinematics techniques for anthropomorphic limbs[J]. Graphical Models, 2000, 62(5): 353-388.
- [4] Kulpa R, Multon F, Arnaldi B. Morphology-independent representation of motions for interactive human-like animation[J]. Computer Graphics Forum, 2005, 24(3): 343-352.
- [5] Kovar L, Gleicher M, Pighin F. Motion Graphs[J]. ACM Transactions on Graphics, 2002, 21(3): 473-482.
- [6] 李淳芃, 王兆其, 夏时洪. 人体运动的函数数据分析与合成[J]. 软件学报, 2009, 6(6): 1664-1672.
- [7] 徐 孟, 孙守迁, 潘云鹤. 虚拟人运动控制技术研究[J]. 系统仿真学报, 2003, 15(3): 338-346.
- [8] 李春霞, 杨克俭, 李 波. 人体骨架模型的建立及 IK 问题的一种解决方式[J]. 武汉理工大学学报(交通科学与工程版), 2003, 27(6): 815-818.
- [9] 张 鑫, 王章野, 王作省, 等. 人体运动建模的实时逆运动学算法[J]. 计算机辅助设计与图形学学报, 2009(6): 853-860.
- [10] Wang L C T, Chen C C. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators[J]. IEEE Transactions on Robotics and Automation, 1991, 7(4): 489-499.
- [11] Lander J. Making Kine More Flexible[J]. Game Developer, 1998(11): 15-22.
- [12] 王小杨, 杜晓荣, 孙韵琳. 人体跨越障碍物的运动生成方法[J]. 计算机技术与发展, 2008, 18(10): 27-31.

(上接第 87 页)

- liability engineering. [s. l.]: [s. n.], 2004: 47-53.
- [7] 于碧媛. 软件可靠性模型及估值的介绍[J]. 导弹与航天运载技术, 1994, 11(3): 55-60.
- [8] Jalote P, Murphy B. Post-release reliability growth in software products[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 2008, 17(4): 56-57.
- [9] 刘 云, 赵 玮. 软件可靠性研究与进展[J]. 微机发展

(现更名: 计算机技术与发展), 2003, 13(2): 12-14.

- [10] Belli F, Jedizejowicz P. Fault-Tolerant Program and Their Reliability[J]. IEEE Trans. on Reliab, 1990, 39(2): 184-192.
- [11] 傅佩琛, 付 遥. 软件可靠性的发展状况与展望[J]. 电子科技, 1997, 10(4): 58-59.
- [12] 吕 励. 软件可靠性模型分析[J]. 哈尔滨理工大学学报, 1997, 2(1): 45-48.