形式化规范在软件可靠性早期估计中的应用研究

吕闽晖[1],吕敏蓉[2]

(1. 海军工程大学装备经济研究所,湖北 武汉 430033;
2.湖南女子学院,湖南 长沙 410004)

摘 要:随着软件的广泛应用,特别是软件在尖端领域的应用,软件可靠性成为一个非常重要的问题。形式化规范在软件可靠性研究中能够起到的作用是多方面的。本文针对非形式化统计使用测试的不足,结合已有的早期可靠性估计方法,设计了优化算法,并提出了即使在设计中采用了形式化规范仍然需要测试的结论。

关键词:形式化规范;软件可靠性;早期估计;测试

中图分类号:TP311.5 **文献标识码**:A **文章编号**:1674-7720(2011)11-0007-03

Application of formal specification in software reliability early evaluation

LV Minhui[1], LV Minrong[2]

(1.Institute of Equipment Economics, Naval University of Engineering, Wuhan 430033, China;
2.Hunan Women University, Changsha 410004, China)

Abstract: With a wide range of applications, especially in sophisticated field, software reliability has become a very important issue. Formalized specification plays more important role in many ways of software reliability studies. In this paper, combined with early reliability estimation methods, an optimized algorithm has been designed in view of the insufficiency of using non-formalized statistics test, and then the conclusion has been drawn that even formalized specification adoped in the software design, software testing is still indispensable.

Key words: formalized specification; software reliability; early evaluation; testing

软件可靠性研究的主要目的是评价和度量软件的可靠性和预测软件可靠性。软件可靠性估计主要指应用数理统计分析处理系统测试和系统运行期间得到的失效数据,对软件系统当前的可靠性进行估计。主要目的是估计当前可靠性,并确定可靠性模型水平的依据。

1 非形式化统计使用测试的不足

1.1 传统使用剖面的不足

软件使用剖面[1]是关于如何使用软件系统的一种量化描述,一个剖面就是一组操作及某个操作发生概率所组成的集合。例如:如果A在60%的时间内发生,B在40%的时间内发生,使用剖面就是A:0.6,B:0.4。

传统理论认为,剖面是反映软件不同的客户、用户、系统模式、功能和操作的发生概率的一个量化特征,其目的是为软件测试及其他软件开发阶段进行资源分配提供相应的参考信息,决定如何在软件测试中进行测试实例的生成,如何在测试过程中分配测试资源。

传统的剖面生成方法已经为统计测试提供了一个良好的基础,然而这并不说明传统的使用剖面的数据本身和相应的获得方法没有缺陷。事实上,以下问题在传统剖面中一直没有得到有效解决:(1) 剖面获得方法主观性较强。(2)统计所得使用频率并不能完全代表软件的实际应用情况[2]。(3)影响可靠性度量[2]。

1.2 非形式化统计使用测试的不足

软件测试占用了软件开发过程中大量的人力物力资源,然而软件测试却往往缺乏较好的理论基础,测试数据即测试实例的选择通常根据经验作出,因此具有主观性,并依赖于个人经验。

统计使用测试则是一种按软件实际使用的方法来测试软件的方法。测试实例是按照使用规范得到的,而使用规范是软件预期使用的描述。统计测试根据用户对

软件可靠性的期望,最大程度地利用可以支配的资源。另外,因为统计使用测试是以概率统计为基础的,因此利用统计测试得到的数据也可以利用概率统计的方法来预测软件可靠性,从而为可靠性度量提供数据,并可以相应进行软件的可靠性度量,另外还可以决定软件测试可以停止的标准。这一切都可以使得软件可靠性处在严密控制之下[4,6]。

2 早期可靠性估计方法及其改进

2.1 使用 SDL 进行使用实例建模

在文献[5]中通过引入状态层次模型,有可能减少统计使用测试中的状态爆炸问题,该方法采用马尔科夫链描述技术描述用户及其对系统的使用。

使用 SDL 可结合状态层次模型的方法进行软件可靠性早期估计,就是用 SDL 的替换形式 SHY-SDL(State Hierarchy with SDL)对使用的层次化描述建模形式化。

目前已有基于使用剖面的测试用例选择算法可以应用于系统的使用模型,算法的结果产生测试序列,测试序列可以用 SDL 或其他方法描述。因此,在进行可靠性早期估计时,用 SDL 描述测试用例很重要,优点是:(1)用 SDL 描述使用,就容易用 SDL 自动生成测试用例。(2)用 SDL 描述测试用例,有可能使用 SBA。

2.2 分析和使用实例建模

SBA 应根据对用户及系统使用的描述和系统的原始 SDL 描述。使用结合 SHY-SDL 的 SBA 有三种方式:

(1) 原始 SDL 系统成为一个块,SHY-SDL 模型作为另一个块,形成新的系统,如图 1。这种方案不能直接实现,因为 SBA 工具将对 SDL 系统进行充分的动态分析,这与统计质量控制过程的目标相冲突。该方案的主要好处是检查系统描述与获得的真实环境的一致性,但一般作为其他方法的补充。

图 1 SBA 的一种应用方案

(2)修改 SBA 工具,使之适合于统计分析,可以按随机的步骤采用第一种方案执行,但同时通过 SHY-SDL 模型根据使用剖面选择特定的执行路径。所以在执行路径的决策处理和调度两个方面必须对SBA 加以补充。

(3)根据 SHY-SDL 模型生成分析(测试)序列,使分析实例的 SDL 描述成为新的 SDL 系统中的块,这种方法意味着用 SBA 分析选择的使用实例。

2.3 根据分析进行可靠性估计

初步的可靠性估计可用如下两种方式:(1)失效间隔时间和相关模型进行比较;(2) 计算成功执行的分析实例数与总分析实例数的比。

第一种方法意味着分析是一个序列,第二种则要求把环境行为描述分成几个分析实例。在使用 SBA 分析时,将报告连续两次失效间分析的状态数,这就对第一种方法形成了简单的支持。如果失效被纠正,将看到可靠性增长,并作为测试和运行期间的可靠性增长的估计,对可靠性增长的早期估计可以更好地规划达到质量目标的测试时间。如果不纠正失效,将得到实际可靠性估计,这种情况只有当分析工具的执行是连续、不纠正失效时才有可能。

第二种分析方法的主要问题在于 SBA 工具交互性很差,如果一个块中实现几个实例,很难分析其中某一个,解决办法是一次只实现一个分析实例,但这又使得工具用户为每个分析实例重新生成代码。

结合 SBA 的方法可获得第二步软件可靠性估计,采用剖面的动态分析可以结合充分分析(SBA 的完整形式)如下:

(1)基于使用剖面进行部分分析,获得可靠性增长估计;

(2)进行充分动态分析;

(3)将规格化的失效时间与可靠性增长估计相比较。

必须规格化失效时间,这样能够比较充分分析和按使用剖面进行部分分析的结果。通过记录失效在使用描述中发生的位置来规格化时间,如果分析根据剖面进行,可以计算失效发生的平均时间。这个时间被认为是实际的失效时间。

2.4 导出失效时间算法及其优化

通过分析可以得到动态失效与实际操作中失效的关系。下面介绍通过分析检测的动态失效是否代表了运行中的实际失效。

首先给出以下假设:

(1) 由 SBA 动态分析找出的失效集是所有可能失效的子集;

(2)动态分析中找出的失效随机分布于所有失效中,例如在某段时间内,随机失效与动态失效的比例因子用 c 表示;

(3)根据使用剖面测试是操作的近似,这也是统计使用测试的基础和大多可靠性估计模型的基础;

(4)而采用 SBA 根据使用剖面分析是采用 SBA 进行充分分析的近似,充分动态分析遍历所有状态,采用使用剖面进行有选择的动态分析时,是根据所有可能的状态集选择进入的状态,选择是根据特定的使用剖面的取样,而不是随机取样。

(5)SBA 利用使用剖面进行动态分析与统计使用测试是可以比较的,从相同的使用剖面选择用于动态分析的实例类似于选择统计使用测试实例,区别为后者是随机的。

要使根据动态分析估计的可靠性增长可应用于考虑随机失效的可靠性增长,必须将动态分析失效数据对应代表所有失效。以某软件为例,可采用以下算法[3]:

(1) 根据使用剖面进行动态分析。图 2 中,$t1$、$t2$、$t3$ 是失效时间。失效数据可以用来估计 SBA 分析所得动态失效的 MTTF (平均失效时间),估计模型可以采用净室可靠性估计模型。

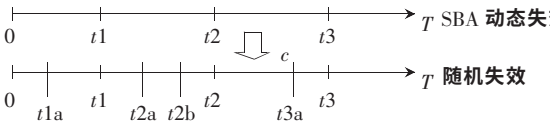

图 2 根据动态分析导出随机失效的失效数据

(2)确定 c。如:包括随机失效在内的实际总失效数与 SBA 找出的动态失效数之比;c 值主要根据早期开发的项目,程序特点不同可以采用不同的 c 值,而且分析不同阶段 c 值也又可能不同。

(3)根据第一步计算的 MTTF,确定每个时间间隔发生的失效数。如果 c 不是整数,间隔内的失效数按 trunc $(c-1)$ 和 trunc (c) 之间的两点分布选择,则 $c-1$ 为平均值,如果是整数,则每个间隔失效数为 $c-1$。

(4) 在时间间隔内随机选择失效时间放入根据不同 c 值得到的失效,如:$t1a$、$t2a$、$t2b$ 和 $t3a$。

(5)根据分析和计算的失效数据估计 MTTF,这就是对所有失效类型 MTTF 的估计。

分析的实际值和可能值应该进一步根据理论和实践研究,但可以肯定,初步估计考虑的是动态分析软件规范期间的动态失效。

3 算法改进

(1)改进使用剖面

通常随机测试也是一种使用剖面,例如:系统所有的事件或信号是等可能出现的,这样可以测试系统将来的可靠性。考虑软件可靠性时,有必要对关键部分加以特别重视,生成这样的剖面在某类严重失效不常发生时是很有价值的。

另外,即使考虑了使用剖面中的重要度和使用频繁度,仍需改变使用剖面,尤其是在动态分析期间,很容易进行第二次分析。改变使用剖面的目的是检查其他使用剖面的可靠性,因为系统的使用情况不断变化,这意味着高可靠的系统由于使用情况的改变有可能变得不可靠了,所以,改变使用剖面有意义。

还有一种方法可以获得在正常使用中较少出现情况的可靠性,即使用充分动态分析考虑软件中易产生故障的部分,找出这些部分可以说明软件使用剖面的改变是可以改变了软件可靠性的。

(2)该方法中选择的时间间隔是两个失效间的间隔,但如果把时间间隔扩大为几个失效的间隔,则在可靠性估计时计算 MTTF 时更准确,更接近于项目的真实情况。当然,这又依赖于对 c 值的进一步确定。

(3) 该方法在时间间隔中放入随机失效时采用的是随机放置的方法,但采用什么方式放置对于统计最后的 MTTF 是很关键的。从统计意义上来说,可以按指数分布在时间间隔内放置随机失效,因为假设 X 为某系统发生故障的时间,它服从指数分布,则对 $\forall s, t>0$,有:

$$P\{X>s+t \mid X>s\}=\frac{P\{X>s+t \mid X>s\}}{P\{X>s\}}=\frac{P\{X>s+t\}}{P\{X>s\}}=\frac{1-P\{X\leqslant s+t\}}{1-P\{X\leqslant s\}}=\frac{e^{-\lambda(S+t)}}{e^{-\lambda S}}=e^{-\lambda t}=1-F(t)=1-P\{X\leqslant t\}=P\{X>t\}$$

这表明,在时段 $(s, s+t)$ 内无故障的概率只与时段的长度 t 有关,而与系统过去无故障的工作时间 s 无关。这正好满足选择动态失效间隔放置随机失效的条件。

即使在设计中采用了形式化方法,仍然需要测试。测试可以检查出早期检测所遗留的或求精过程中所引起的错误,如 T800 晶片机的浮点部件在测试中发现了一个对微代码改变而产生的错误,这个改变是发生在形式化开发后。可以说,测试永远是有用的检查手段,因此建议:

(1)开发软件测试平台

研究、开发软件测试平台是很有现实意义,国内外软件测试工具很多,应针对开发软件所常用的语言,利用现有的技术,开发软件测试平台。

(2)建立软件测试和评估中心

软件的专业性很强,因此,软件测试评估中心应由计算机软件、可靠性工程、系统工程等领域的专家组成,负责制定软件可靠性各种标准,并监督实施,做好软件可靠性管理工作;建立软件测试平台,做好软件可靠性技术保障工作;由软件生产单位协助,建立可靠性测试实例库,客观、公正地验收软件;做好可靠性数据的收集、整理、分析工作,开展软件可靠性模型的研究,建立适用于软件的可靠性模型,并进行可靠性评估。

参考文献

[1] MUSA J D. Operational profiles in software-reliability engineering[J]. IEEE Software, 1993,10(2):14-32.

[2] KITCHENHAM B. Validation verification and testing. diversity rules[J]. IEEE Software, 1998,15(4):46-49.

[3] WOHLIN C, RUNESON P. A method proposal for early software reliability estimation [C]. Proceedings of 3th International Symposium on Software Reliability Engineering, 1992.

[4] 颜炯,舰载指控软件操作剖面研究[D].武汉:海军工程大学,2000.

[5] Runeson. Usage modeling: The basis for statistical quality control. Proceedings of SRS'92, Denver, USA, June 1992,77-84.

[6] STOCKS P A, CARRINGTON D A. A framework for specification-based testing [J]. IEEE SE, 1996,22(11):777-793.

[7] RUSHBY J, PARK M. Formal methods and their role in the certification of critical systems [C]. In: Safety and Reliability of Software Based Systems, 12th Annual CSR Workshop, Springer, 1997.

(收稿日期:2011-01-05)

作者简介:

吕闽晖,男,1975 年生,硕士,讲师,主要研究方向:软件工程、装备经济信息管理。

吕敏蓉,女,1979 年生,硕士,讲师,主要研究方向:财务信息化管理、数据库应用。