

基于测试执行的失效数据建模研究*

方菲 王立福 杨美清

(北京大学计算机科学与技术系 北京 100871)

E-mail: faye_f@263.net

摘要 目前,软件可靠性研究和实践者普遍感到,由于缺乏失效数据,严重影响了工作的进展.该文通过研究和分析软件测试与软件可靠性度量之间的关系,提出了一种基于测试执行的失效数据模型 JBFDM(Jade Bird failure data model),并探讨了基于此模型的数据收集和管理技术.最后,给出了基于此模型的失效数据收集工具 JBFDCollector.

关键词 软件失效,软件测试,操作概图,软件可靠性,软件可靠性模型.

中图法分类号 TP311

通常,软件可靠性是指:在规定的时间内和规定的环境下,程序无失效运行的概率^[1].正确地收集和分析软件失效数据,是软件质量评价系统的核心工作之一.在实际的可靠性工程中存在着,缺乏标准、对失效数据收集的低效管理,以及支持失效数据分析及可靠性度量的失效数据收集工具的缺乏等问题,这些是造成软件可靠性度量的研究和实践失败的主要原因.

从目前的发展趋势来看,软件测试作为与软件开发的相关活动,已成为与软件产品和过程都相关的质量保证技术.软件测试方法、测试过程、测试设计、测试工具等都对可靠性度量有着重要的影响.

通过分析和研究黑盒软件测试方法对软件可靠性度量的影响,并结合青鸟工程实践,本文提出一种基于测试执行的失效数据模型 JBFDM(Jade Bird failure data model)以及基于此模型的失效数据管理技术.其目的是:(1)提供一致的失效数据建模、收集及管理的可靠性度量过程,从而支持可靠性度量;(2)利用测试及软件现场收集的数据来评价测试计划、操作概图及测试方法的有效性.最后,本文给出基于 JBFDM 的失效数据测试及收集工具 JBFDCollector 的主要设计思想.

1 软件测试环境

1.1 软件可靠性度量与操作概图

软件可靠性度量就是用软件可靠性模型对软件可靠性进行评价和预测.

软件可靠性度量不准确的根本原因是模型的假设与实际软件不符.为了简化模型的计算,在已提出的可靠性度量模型中常假设在测试时,软件按预期的实际运行方式进行.但在实际测试时,软件的使用方式可能与现场运行时有很大的不同.例如,在测试期间,对不常用的功能(如系统失效保护)也要进行大量的测试,以确保软件在需要时能按要求正常工作.因而确定基于用户的操作概图,并按此操作概图确定测试计划,对于准确评价软件在实际运行中的可靠性是非常重要的.

操作概图(operational profile)是关于如何使用系统的一种定量特征描述,它通过列举程序可能的输入状态和它们发生的概率而建立.

* 本文研究得到国家“九五”重点科技攻关项目基金资助.作者方菲,女,1963年生,博士生,主要研究领域为软件测试,软件可靠性.王立福,1945年生,博士,教授,主要研究领域为软件工程,信息安全.杨美清,女,1932年生,教授,博士生导师,中国科学院院士,主要研究领域为软件,软件工程,系统软件,软件复用技术.

本文通讯联系人:方菲,北京 100871,北京大学计算机科学与技术系

本文 1999-07-26 收到原稿,1999-09-15 收到修改稿

一种建立操作概图的方法是,将程序的输入空间划分成多个等价类,并确定每个等价类中每个类的发生概率。测试时,可从每个等价类中选择一个状态运行。

一个实用的定义方法是,用程序模块序列代替输入状态^[2],输入状态和模块序列之间是多对一的关系。程序操作概图,就是所有可能的模块序列集合和它们发生的联合概率。这种定义方法着重于程序的结构特征,通过求出敏感系数,就可发现哪些模块对系统可靠性的影响最为关键。最有效的测试,应是按模块关键程度递减的顺序来测试所有的模块。

操作概图的获得还可借助于专门的统计分析方法,如借助于 Markov 状态转移图等,本文将不对此进行详细讨论,具体内容可参考文献[3]。

可靠性度量基于失效数据,而失效数据只有在较好地反映了用户操作概图的测试环境下进行收集,才能保证对软件在实际运行中可靠性估计的准确性。

因此,一个辅助用户定义和管理其操作概图并进行有效的系统测试的测试环境,是保证软件可靠性度量实施的关键。同时,这也是本文要介绍的青鸟黑盒软件测试环境所具有的特征。

1.2 青鸟黑盒软件测试环境

1.2.1 青鸟黑盒软件测试环境模型(如图 1 所示)

JBTEST(Jade Bird test platform)测试平台采用录制/重放技术,对在 Windows 下具有图形用户界面的应用程序进行功能测试。JBTEST 的主要功能是:辅助测试人员按照测试规范说明制订测试计划;辅助测试人员捕获并编辑测试用例和测试脚本;支持测试脚本及用例的自动重放;当测试结束时给出测试结果。

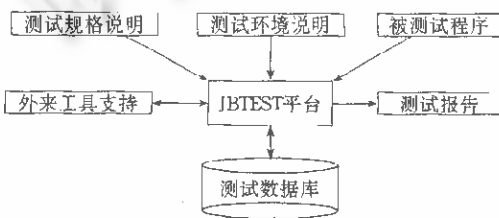


图1 青鸟黑盒软件测试环境示意图

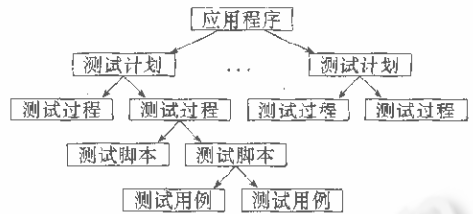


图2 青鸟软件测试计划树

1.2.2 测试管理框架

软件测试是一个有规则的过程,包括测试计划制定、测试用例设计、测试执行以及结果比较等测试活动。与软件开发过程一样,测试本身也需要设计和测试。测试设计的好坏,直接影响到错误的发现能力和测试结果评价。为此,我们提出测试计划树概念,以管理和指导测试设计,如图 2 所示。

对被测程序可以建立一个或多个测试计划;每个测试计划包括对被测程序的一项或几项功能进行测试的规程;针对要测试的每项功能定义一个测试过程,在每个测试过程中可定义多个测试脚本,这些测试脚本描述测试过程中的每一次测试;在每个测试脚本中还可以定义多个测试用例,描述测试过程中对界面对象某种状态的记录和比较,这样一个软件测试计划的建立过程,最终可形成软件的一棵测试计划树。

2 失效数据建模

在建立失效数据模型中,主要涉及以下几个问题。

2.1 失效数据模型与操作概图

在青鸟黑盒测试环境及测试计划管理框架下,操作概图的定义过程可以与测试计划树的建立过程相对应,这样的测试计划既反映了用户的实际操作概图,也反映了系统测试的主要内容。而且,将操作概图以树型层次方式管理,用户对操作概图的定义则可以在层次框架的指导下有序地进行。

在测试过程中,测试脚本信息是非常重要的。这是因为在以操作概图为指导的测试计划中,测试脚本就是对程序一次操作(即运行)的定义。它对分析操作概图的精度、测试的充分程度、操作对错误的发现能力以及(对可

可靠性而言)关键部件的识别都是非常重要的。

本文所提出的青鸟软件失效数据模型(以下简称 JBFD M)包含了对测试脚本的建模。

2.2 失效数据模型与软件可靠性

大多数软件可靠性度量模型是建立在这样的认识上,即失效的发生是随机的。

虽然已提出了 40 多个可靠性度量模型,但从基于时间的失效建模的角度来看,可以将它们分为基于累积失效计数的度量模型和基于失效间隔计数的度量模型两类。图 3 是软件可靠性度量常用的两类失效数据。

因此,为了支持多种可靠性模型的度量,建模时必须包含这种时间类型的数据。JBFD M 覆盖了这两类数据。

经验表明,时间的最佳度量单位是 CPU 的执行时间^[3]。但 CPU 时间有可能难以获取,可采用其他度量,如日历时间、时钟时间、服务时间、逻辑时间和结构覆盖域(如分支覆盖)^[3,4]。

考虑到度量的适当性、数据的易获取性、度量对错误的敏感性以及时间与可靠性度量模型的相关性,JBFD M 支持日历时间度量。另外,为了提供对测试分析的支持,JBFD M 还提供了逻辑时间度量,以测试脚本数和测试脚本覆盖率。

2.3 软件失效与软件错误

软件失效(failure)是指程序运行的外部结果偏离了需求规范。所以,“失效”是动态产生的,必须通过执行程序才会发现失效。

软件错误(fault)即为程序中的缺陷。只有在某个特定的环境下,某个缺陷才会导致失效。

可见,软件错误是失效发生的必要条件,但不是充分条件。

为了支持对失效行为以及失效行为与程序内部错误之间的关系分析,JBFD M 对错误分类进行了建模。常用的错误分类法有根据错误对工作功能的影响程度、错误的起因、错误征兆、程序开发的阶段等来进行分类。

JBFD M 参照美国海军标准(MIL-STD-1679),按照对工作功能的影响程度,即严酷度(Severity)的不同,将错误分为以下 5 级。

一级:不能执行正常工作功能或重要功能;或者危及人身安全。

二级:严重地影响工作功能或重要功能的执行,且没有办法更正,或者干扰操作者,致使工作恶化等。

三级:除了有适当的更正方法之外,其他的与二级优先权相同。

四级:使操作者不方便或遇到麻烦,但它不影响执行工作功能或重要功能。

五级:其他错误。

通常,失效描述是基于文本的。虽然详尽的描述有助于错误的定位,但不能进行宏观级的自动错误分类和分析。错误按严酷度级别进行分类,便于失效数据收集的自动化和失效数据的标准化。

另一方面,错误按严酷度分类,有助于决定测试终止时间。例如,由于时间和市场的要求,可以按用户对程序使用频率较高的主要功能来设计操作概图并进行测试,如果所有的失效均在四、五级,那么可以决定终止测试,进行软件发布。

尽管如此,JBFD M 仍然对“失效描述”进行了建模,以支持文本失效描述。

2.4 失效数据模型粒度

在收集失效数据及其他数据时应注意一个问题:可靠性估计/预测值的有效精度总是要低于数据的精度。如果数据以月为单位收集,则预报一周的失效数就没有多大意义。因此,支持多粒度的数据采集是很重要的。JBFD M 支持年、月、日、时、分、秒这 6 种粒度数据的采集。

综合以上分析,本文提出青鸟软件失效数据模型 JBFD M。它包括下面的两类子模型。

子模型 JBFD M1:该模型覆盖了常用的累积失效计数。

时间	周期内失效数	累计失效数
----	--------	-------

(a) 累积失效计数数据

失效数	失效间隔	失效时间
-----	------	------

(b) 失效间隔计数数据

图3 两类常用失效数据模型

- (1) 失效数据收集者
- (2) 所属项目组
- (3) 测试区间号*
- (4) 累积失效数*
- (5) 累积测试脚本数
- (6) 测试区间长度*
- (7) 严酷度(Severity)*
- (8) 失效粒度*
- (9) 失效描述*
- (10) 可能原因*

子模型 JBFDm2: 该模型覆盖了常用的失效时间间隔模型。

- (11) 失效数据收集者
- (12) 所属项目组
- (13) 失效序号*
- (14) 失效间隔*
- (15) 失效时间*
- (16) 测试脚本
- (17) 严酷度(Severity)*
- (18) 失效粒度*
- (19) 失效描述*
- (20) 可能原因*

注: * 表示此项数据必须收集. JBFDm 适用于系统测试期间和现场运行阶段。

3 失效数据管理技术

失效数据收集失败的主要原因是失效录入手段落后以及对失效数据缺乏有效的管理. 目前, 在许多实施软件可靠性工程的企业中, 失效数据的收集方式主要采用手工填写表格方式和文件记录方式。

对以这两种方式收集的失效数据, 很难建立和维护数据之间的关系, 特别是对分散测试点收集的数据很难维护. 当失效数据收集格式或数据项发生改变时, 以前的数据则可能无效. 或者需要花费很长的时间进行转换. 对数据的分类或统计只能以手工方式进行, 既困难又容易出错, 非常不利于对现场测试数据的分析, 影响错误的及时排除。

针对上述问题, 我们提出以下几种失效数据管理技术。

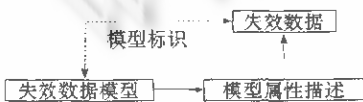


图4 动态失效模型生成示意图

(1) 动态生成失效数据收集模型(如图4所示). 当用户需要按照某模型建立失效数据收集库时, 系统则从模型属性描述库中读取模型的所有信息, 动态生成数据库. 这种管理技术便于管理失效数据模型的自扩展和系统模型的扩充。

(2) 对被测软件、失效数据模型和基于此模型的失效数据集, 建立用户视图. 这样, 不仅可以支持数据统计及分类, 还有利于对可靠性度量过程进行管理。

(3) 命名规范化. 便于将失效数据与系统中的其他数据, 如失效模型数据、软件基本信息数据、可靠性度量过程数据、可靠性模型数据等区别开来, 提高管理效能。

4 青鸟软件可靠性失效数据收集工具 JBFDCollector

青鸟软件失效数据收集工具 JBFDCollector 有以下几个主要设计思想。

(1) 失效数据建模是开发失效数据收集工具的前提, 而工具本身对失效建模的支持也会使其具有较好的扩展性. JBFDCollector 应用上一节所述的动态生成失效数据收集模型技术, 支持软件失效数据建模及管理, 并内嵌了青鸟软件失效数据模型 JBFDm。

(2) 为了支持多个用户对同一软件在同一时刻进行评测, 工具的开发与应用必须基于分布式网络环境. JBFDCollector 采用分布式数据库技术, 支持多个用户同时对失效数据进行录入及异地更新。

(3) 在一个支持整个软件可靠性度量过程的环境中, 失效数据管理与软件管理是紧密相连的. 软件管理不仅包含了对软件测试进度及资源的约束, 还包含了该软件的可靠性目标. JBFDCollector 是青鸟可靠性工具集 JBRelKit 的一部分, 在进行失效数据收集前, 必须保证被评测软件已在系统中进行了登记。

(4) 为了准确记录失效发生的时刻, 以保证失效数据的有效性及准确性, JBFDCollector 对被评测软件的执行进行实时监控, 即当应用程序运行时, JBFDCollector 处于运行监控态, 当应用程序由于故障终止运行时, 则自动启动失效数据收集。

(5) 为了支持多种软件可靠性评价模型对同一失效数据的评价, 有必要提供不同数据类型之间的转换. JBFDCollector 提供累积失效计数与失效间隔数据之间的转换. 不仅如此, 本工具还支持对外部数据的引入, 用户可以有效地利用外来的或者以前已有的历史数据进行可靠性评价和分析。

青鸟软件失效数据收集工具 JBFDCollector 的开发是基于 ORACLE 数据库管理系统,在 Windows 环境下,采用 Microsoft VC++5.0 编程完成。其主要功能包括:失效数据建模、分布式失效数据的录入和修改、数据类型转换。

5 结束语

从以上的讨论可以看出,软件测试对软件可靠性度量有着重要的影响。为了使软件可靠性工程得以成功地进行,失效数据建模是重要的前提。合理的失效数据模型是保证数据收集成功的关键。

将软件可靠性理论应用于实践是软件可靠性研究的最终目的。在实际的软件可靠性工程中,不恰当的可靠性度量过程的实施、可靠性模型的滥用等,是可靠性度量失效的根本原因。因此,有必要进一步研究基于测试执行的软件可靠性度量过程及管理技术。另外,研究实用的可靠性度量工具,并提供失效数据模型与软件可靠性模型的自动匹配,也是保证软件可靠性能够真正走向实用的关键。

参考文献

- 1 ANSI/IEEE. Standard Glossary of Software Engineering Terminology. STD-729-1991, ANSI/IEEE, 1991
- 2 Cheung R C. A user-oriented software reliability model. IEEE Transactions on Software Engineering, 1980,SE-6(2): 118~125
- 3 Musa J D, Iannino A, Okumoto K. Software Reliability Measurement, Prediction, Application. New York: McGraw-Hill, 1987
- 4 Tian J, Lu P. Measuring and modeling software reliability, data, models, tools, and a support environment. Technical Report, TR-74, 117, IBM Canada Laboratory, 1993

Research on the Test-Execution-Based Failure Data Modeling

FANG Fei WANG Li-fu YANG Fu-qing

(Department of Computer Science and Technology Beijing University Beijing 100871)

Abstract At present, software reliability researchers and engineers are desperately hindered by the lack of failure data. In this paper, the authors present a model, which is called JBFDM (Jade Bird failure data model), for failure data based on the study and analysis of the relationship between software testing and reliability estimation. Furthermore, based on the model JBFDM, an approach of carrying out failure data collection and management is also proposed, and a data collection tool, which is called JBFDCollector is presented.

Key words Software failure, software testing, operation profile, software reliability, software reliability model.