

An Approach of Prison Software Reliability Test Based on Running Profile

Zixian WANG, Guofu MA, Qing DUAN, Lin LV

The Central Institute for Correctional Police; Baoding, China

Email: bd_wzx@126.com

Abstract: Software Reliability testing is an important component of software testing, which can determine whether the software reliability meet the user's requirement. On the basis of existing reliability testing method, based on operational profiles, a series of related operations are together to be test. An Approach of software reliability testing based on running profile is given, at the same time; this paper takes advantage of the orthogonal test method, simplifies the running profile of the structure, and optimizes the test cases. Thus, to a certain extent, it can reduce the cost of software testing.

Keywords: operational profile; running profile; reliability test

基于运行剖面的狱政软件可靠性测试方法

王子贤, 马国富, 段 庆, 吕 琳

中央司法警官学院, 保定, 中国, 071000

Email: bd_wzx@126.com

摘 要: 软件可靠性测试是软件测试的一项重要内容, 通过软件可靠性测试可以判断软件的可靠性是否满足用户的要求。本文在原有的基于操作剖面的可靠性测试基础上, 将一系列相关的操作联合起来进行测试, 给出了一种基于运行剖面的软件可靠性测试方法, 同时利用正交试验的方法, 简化了运行剖面的构造, 并对测试用例进行优化, 一定程度上降低了软件测试的代价。

关键词: 操作剖面; 运行剖面; 可靠性测试

1 引言

在现代军事、商用、民用等系统中, 以软件为核心的产品得到了广泛的应用。随着系统中软件成分的不断增多, 使得系统对软件的依赖性越来越强, 对软件可靠性的要求也越来越高。IEEE 将“软件可靠性”定义为: 在规定的条件下, 在规定的时间内, 软件不引起系统失效的概率, 该概率是系统输入和系统使用的函数, 也是软件中存在的缺陷的函数; 系统输入将确定是否会遇到已存在的缺陷 (如果缺陷存在的话)。软件可靠性测试是提高和保证软件可靠性的重要手段。通过软件可靠性测试, 可以有效地发现程序中影响软件可靠性的缺陷, 实现可靠性增长; 还可以定量计算软件的可靠性以判断软件可靠性的定量要求是否得到满足。

2 相关工作

资助信息: 该研究受 2009 年度中央司法警官学院青年教师项目“软件测评在信息化建设中的应用研究” (XYQ200902) 资助。

软件可靠性问题已被越来越多的软件工程专家所重视, 人们投入大量的人力、物力如研究软件可靠性的设计、评估、测试等课题。Musa 教授提出了操作剖面的概念“操作及其发生概率组成的集合”^[1,2], 并用于软件可靠性测试。Musa 提出的操作剖面简单实用, 对于操作简单的软件来说非常有效。但在实际中, 随着计算机技术应用的越来越广泛, 功能强大的软件也越来越多。根据《全国监狱信息化建设规划》要求, 监狱业务应用系统应包括: 监狱安全防范与应急指挥系统、监管及执法管理系统、教育改造系统、生活保障及医疗卫生系统、警察管理系统、生产管理与劳动改造系统、监狱建设与保障系统、狱务公开系统、办公自动化和决策支持系统^[3]。对于这样一个复杂的软件, 必将包含大量的操作。这种情况下需要构造的操作剖面会很多, 处理的数据量非常庞大。为了解决这一问题, 文献^[4]采用正交试验的设计方法大大减少了操作剖面测试用例的数目; 文献^[5]采用基于组合设计的方法构造剖面, 优化了测试用例。

本文借鉴 Musa 的操作剖面的思想, 并针对狱政软件的复杂性, 对软件运行过程中的一系列相关操作联合起来构造剖面, 给出了一种基于运行剖面的狱政软件可靠性测试方法。该方法将能够到达某一个系统状态的一系列操作联合起来看作一个运行剖面, 这样就避免了将所有操作单独看待构造操作剖面产生的大数据量问题, 并且借鉴文献^[4]中采用的正交试验法, 在保证测试质量的同时进一步减少了测试用例的生成。

3 构建模型

3.1 操作剖面

根据 Musa 的操作剖面(Operational Profile, OP)理论^[1], 设 $O = \{o_i | i = 1, 2, 3, \dots, n\}$ 为系统中所有操作组成的集合, $P = \{p_i | i = 1, 2, 3, \dots, n\}$ 为 O 中对应操作发生的概率, $OP = \{(o_i, p_i) | i = 1, 2, 3, \dots, n\}$ 即为操作剖面。通过操作剖面示意图可以清楚地得出操作及其发生概率。

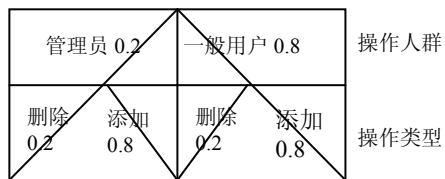


Figure 1. Operational profile

图 1. 操作剖面图形化表

在图 1 所示的操作剖面示意图中, 管理员进行删除操作的概率为: $0.2 \times 0.2 = 0.04$; 以此类推, 管理员添加操作的概率为 0.16, 一般用户删除操作的概率为 0.16, 一般用户添加操作的概率为 0.64。

Musa 的操作剖面理论简单实用, 对于相对简单的软件可靠性测试非常有效^[6]。但是存在一定的不足: 不能反映各个操作之间的关联性, 这样各个操作只能独立地参与测试。对于监狱系统这样复杂的软件, 为了避免构造大量的操作剖面, 体现各个操作之间的关联性, 本文使用运行剖面进行测试。

3.2 运行剖面

在一个软件系统中, 我们将软件的状态定义为在完成某些操作之后的软件状况, 软件的整个运行过程就是由用户的操作和软件的状态构成的。在运行一个软件系统时, 一个结果状态的出现必然是由一定的操作实现的, 因此根据结果状态找出一系列相关联的操

作序列。

定义 1: 运行序列— $R_i = \{O_1^i, O_2^i, O_3^i, \dots, O_n^i\}$ 表示第 i 个运行序列, 该序列包含一系列相关联的操作。运行序列有以下特点:

(1) 顺序性。在一个运行序列中, 这些操作具有逻辑操作上的顺序关系, 操作 O_3^i 发生在操作 O_2^i 之后, 操作 O_2^i 发生在 O_1^i 之后。

(2) 确定性。一旦一个运行序列确定之后, 该运行序列中的操作组合就确定下来, 必须严格按照该序列定义的来运行, 不能更改。

虽然一个运行序列具有确定性, 但是在一个软件运行过程中是具有不确定性的。即, 在运行序列 1 中可能 O_3^i 发生在 O_2^i 之后, 但在另一个运行序列中这种顺序关系可能被破坏。

定义 2: 初始状态(Beginning State, BS)—一个运行序列 R_i 运行之前的软件状态, 表示为 B_i 。其中, 下标表示与该初始状态相对应的运行序列的编号。

定义 3: 终止状态(Ending State, ES)—一个运行序列 R_i 运行结束之后的达到的软件状态, 表示为 E_i 。其中, 下标表示与该终止状态相对应的运行序列的编号。

根据以上定义可知: 初始状态和终止状态都不是固定的, 一个状态可能既是对应于某个运行序列的初始状态, 又是对应于另一个序列的终止状态。

定义 4: 运行剖面(Running Profile, RP)—运行序列及该序列中所有操作能正常运行完成的概率。

为了表示一个软件系统的运行剖面, 我们可以采用状态图的方法。

一个状态图包含以下元素:

(1) 输入域: 输入域为一个代表用户观点的软件使用的状态集。该状态集中的状态均为软件运行过程中的初始状态。

(2) 输出域: 也是一个状态集, 与输入域不同的是该状态集中的状态表示软件运行过程中的终止状态。

(3) 弧线: 用来连接状态, 并表示由各种操作导致的转换。

(4) 中间域: 由既不在输入域, 又不在输出域的状态组成的集合。

(5) 状态域: 包括输入域、输出域和中间域。

(6) 操作: 导致软件系统的运行状态发生改变的用户动作。

状态图中所有图例如图 2 所示:



Figure 2. the elements in a state diagram

图 2. 状态图中各要素的图例

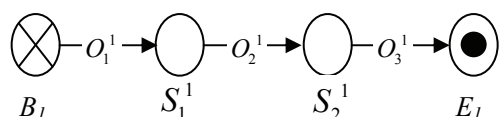


Figure 3. A simple state diagram

图 3. 一个简单的系统状态转换图

图 3 是一个由一条运行序列构成的简单状态转换图, 根据前面定义, 该序列可以用操作的集合表示为: $R_1 = \{O_1^1, O_2^2, O_3^3\}$, 也可以用状态的集合表示为: $R_1 = \{B_1, S_1^1, S_2^1, E_1\}$ 。如前所述, 如果某一状态在不同的运行序列中属于不同的状态类型, 采用第一种运行序列的表示方法不会受到影响; 如果采用第二种表示方法, 在不同的序列中按照相应的命名规则对该状态进行不同的命名即可。即: 同一系统状态在不同的序列中可能会有不同的状态名称。

3.3 运行剖面构造方法

运行剖面的构造方法包括自上而下的方法和自下而上的方法。自上而下的方法就是首先从整体上、从大的方面对系统进行分析, 然后逐级细化, 最后构造出运行剖面。自下而上的方法就是首先列出系统运行过程中的所有细节, 然后对这些细节分类汇总, 同样最终得到系统的运行剖面。这里我们采用自上而下的构造方法, 可以将对软件运行剖面的建模过程分为以下步骤:

第一, 划分系统模式剖面。系统模式是根据不同用户对系统的使用情况对系统进行划分。系统模式可以分为系统模式、普通用户模式和超级用户模式三种模式。不同的用户群在软件操作过程中所处的系统模式是不同的。不同的系统模式下软件失效造成的损失也是不同的。对系统划分为不同的系统模式, 可以使后面的运行剖面的构建更加具有针对性。因此, 建模的第一步就是对系统模式进行划分并构造系统模式剖面。

第二, 划分功能剖面。即使在系统的同一模式下, 不同的软件功能也有不同的重要性。不同的用户对系

统的功能使用概率是不一致的。功能测试也是软件可靠性测试的一项核心内容。功能的划分要依据软件的需求分析以及系统的实现来进行。要确定系统都实现了哪些功能, 功能之间是否有依赖关系, 即一个功能和另一个功能之间是一对多的依赖关系, 还是一对一的依赖关系, 或者两个系统功能是独立的没有任何依赖关系。

第三, 确定运行剖面。

(1) 确定系统状态。充分考虑软件运行过程中出现的各种情况, 系统的运行流程可以参照系统开发时候的各种文档来检验。根据系统状态的定义确定一个个的系统状态。

(2) 确定各个状态之间的相互依赖关系并确定状态种类 (三种状态之一, 或身兼几种状态的功能)。状态之间的依赖关系基本是按照系统的运行流程来的, 在划分功能剖面时已有所叙述。

(3) 画出系统状态图。将有关联的一系列系统状态用弧线连接, 并用图例表示不同的状态类型得到系统的状态图。系统状态图最终为一个不带权重的有向图的形式。

(4) 根据系统状态图列出运行剖面。运行序列剖面的构建同样有自下而上的方法和自上而下的方法。自上而下的方法首先考虑软件的初始状态, 通过从初始状态沿着路径找到与其相对应的一个系统终止状态。在这个过程中走过的路径即为一条运行序列; 自下而上的方法从软件的一系列结果状态开始向上回溯, 一直到某个开始状态, 最终同样通过走过的路径得到一个运行序列。一个操作剖面可以看作一个运行序列的子剖面。

第四, 生成运行概率

运行概率的获得可以采用以下方法: 对相似产品使用方式的统计或对同一产品使用方式的历史经验的统计; 对用户使用软件进行调查, 估计软件的使用概率; 通过概率的分配, 可以对用户实施的使用情况进行相应的限定, 为后期的分析提供一个依据。目前的狱政软件种类比较少, 仅有 20 种左右, 所以采用第一种方式获得系统运行剖面的概率可能并不太适合自己监狱的实际情况。可以考虑第二种方式获取运行概率。概率确定之后, 系统状态图变为一个带权重的有向图。

第五, 优化运行剖面

运行剖面构造完成之后会形成一个比较复杂的有向图结构, 为了使后面的测试更有效, 还要对刚刚建立的系统进一步检查。检查在剖面构造过程中是否有

遗漏、疏忽的内容，剖面的划分是否合理，概率的设置是否合理等等。

正交试验设计法是一种以概率论和数理统计为理论基础的试验设计方法。利用正交表，安排多因素进行试验，并对试验结果进行分析，找出最优的解决方案。在确定构造的剖面的准确性之后，可以利用正交的方法对其进行剖面优化，以提高剖面质量，减少测试用例。这里我们可以参照文献[4]的思想来完成。

4 可靠性度量

我们用可靠度来度量软件的可靠性。根据软件可靠性的定义，软件可靠度就是软件系统在规定条件下，规定时间内不发生失效的概率^[7]。如果用 $F(t)$ 表示 t 时刻系统发生失效的概率， $F(0)=0$ ， $F(+\infty)=1$ ，则我们定义如下公式表示系统的可靠度：

$$R(t) = 1 - \sum_{i=1}^n F(t) \bullet RP_i \bullet W_{RP}^i \quad (1)$$

其中， $R(t)$ 表示软件可靠度，根据公式我们有 $R(0)=1$ ， $R(+\infty)=0$ ， RP_i 表示第 i 个运行剖面的成功运行的概率， W_{RP}^i 表示第 i 个运行剖面在系统中的重要程度，即权重。对于不同的系统，相同的操作可能具有不同的权重，要具体系统具体分析。

在不同的环境下，公式(1)中的参数可能会有所变化，所以要准确的对可靠性进行度量除了要合理构造运行剖面，合理设置运行剖面的权重之外还要注意考虑不同环境对软件可靠性的影响。

5 结论

本文对 Musa 基于操作剖面的软件可靠性测试思想进行扩展，将完成一个流程用到的操作组合起来作

为一个运行序列看待，构造运行剖面。利用运行剖面对软件进行可靠性测试，增加了操作剖面中各个操作的关联性。由于是对多个有相关性的操作构造剖面，减少了测试剖面的构建，同时采用正交化方法对测试用例进行处理，在减少测试用例数量的同时进一步提高了测试用例的质量，也使软件可靠性测试的代价有了一定程度的降低。

References (参考文献)

- [1] Musa J D, The operational profile in software reliability engineering: an overview[C].// Proceedings of the 3th International Symposium on Software Reliability Engineering. Los Alamitos, USA: IEEE Computer Society, 1992: 140-154.
- [2] Musa J D. Operational profiles in software-reliability engineering[J]. IEEE Trans Software, 1993, 10(2):14-32.
- [3] Wu Aiying, Speech at the national prison information construction work conference[N]. Xinhua Daily. 2007-05-30 (A01). 吴爱英, 2007年5月29日在全国监狱信息化建设工作会议上的讲话[N], 新华日报, 2007-05-30 (A01).
- [4] Yan Jinbao, Xu Xishan, Zhong Duhang, Software Reliability Test Based on Orthogonal Experiment Design[J]. Computer Engineering, 2006, 32(4):52-54. 严进宝, 徐锡山, 钟读杭, 基于正交试验设计的软件可靠性测试[J]. 计算机工程, 2006, 32(4): 52-54.
- [5] Ma Changjie, Zhu Xiaodong, Yuan Ye, An Approach to Designing Software Reliability Test Case Based on Combinatorial Design[J], 2006, 22(73):263-265. 马长捷, 朱小冬, 袁野, 基于组合设计的软件可靠性测试用例设计方法[J]. 微计算机信息, 2006, 22(73): 263-265.
- [6] Ai Jun, Lu Minyan, Ruan Lian, Usage Profile Construction Technique for Generation of Software Reliability Test Data[J], Computer Engineering, 2006, 32(22): 7-9. 艾骏, 陆民燕, 阮镰, 面向软件可靠性测试数据生成的剖面构造技术[J], 计算机工程, 2006, 32(22): 7-9.
- [7] Liu Chunlu, Huang Zi-he, Ruan feng, Software Testing Tutorial[M]. iTsing Hua University Press. March,2005, Page: 483. 柳纯录, 黄子河, 陈绿萍. 软件评测师教程[M]. 清华大学出版社, 2005年3月. P: 483.