

引用格式:崔三俊,张贵斌,韩明. 基于代码路径的嵌入式软件可靠度评估[J]. 微电子学与计算机, 2020, 37(11): 79-82. [CUI S J, ZHANG G B, HAN M. Embedded software reliability evaluation based on code path [J]. Microelectronics & Computer, 2020, 37(11): 79-82.]

基于代码路径的嵌入式软件可靠度评估

崔三俊, 张贵斌, 韩 明

(空装驻西安地区 第八军事代表室, 陕西 西安 710065)

摘 要: 嵌入式系统软件可靠性设计不足引发的问题数量呈上升趋势, 如何在软件产品交付前评估软件可靠度已成为亟待解决的问题. 当前的软件可靠度评估理论源于硬件可靠性体系, 已形成了适用于不同的场合较多的评估方法, 但存在使用时难以获得可靠性数据的问题, 可操作性不强. 代码是软件的关键属性, 采用软件代码分析的方法, 提出一种基于代码路径的软件可靠度评估方法, 利用已有的项目软件测试数据验证该评估模型, 进而给出提高软件可靠度的设计方法, 可有效指导嵌入式软件可靠度评估并提高软件可靠性设计质量.

关键词: 软件可靠度; 代码路径; 评估

中图分类号: TP311.5

文献标识码: A

文章编号: 1000—7180(2020)11—0079—04

Embedded software reliability evaluation based on code path

CUI San-jun, ZHANG Gui-bin, HAN Ming

(The No. 8 Representative office of Air Force Equipment Bureau in Xi'an, Xi'an 710054, China)

Abstract: The number of problems caused by insufficient reliability design of embedded system software is on the rise. How to evaluate software reliability before software delivery has become an urgent problem to be solved. The current software reliability assessment theory originates from the hardware reliability system, and has formed assessment methods suitable for different occasions. However, it is difficult to obtain reliability data during use and the operability is not strong. Code is the essential attributes of software, using the method of software code analysis, proposes a software reliability assessment method based on the code path, using the existing project software test data validate the assessment model, and gives the design method to improve the reliability of software, which can effectively guide the embedded software reliability evaluation and improve the quality of software reliability design.

Key words: software reliability; code path; evaluation

1 引言

随着嵌入式系统信息化水平的提高, 由于软件引起的质量问题呈逐年上升趋势, 其中有相当一部分是由于软件可靠性设计、测试不到位, 当外界条件发生变化, 触发故障发生导致任务失败. 当前在软件可靠度评估领域已形成了许多研究成果, 但仍有新方法不断提出^[1-2]. 近年来, 对嵌入式软件可靠性要求越来越高, 但实践中难以获取有效的可靠性数据, 使得嵌入

式软件可靠度评估工作推广困难. 本文通过研究软硬件故障机理, 以分析软件代码的执行路径和输入处理执行概率为要素, 实现一种嵌入式软件可靠度评估方法, 该方法数据来源于软件源代码、设计文档和测试结果, 便于应用操作, 经项目验证可行, 可用于嵌入式软件可靠度评价并指导软件可靠性设计.

2 软件与硬件可靠性的关系

硬件是物理实体的逻辑组合, 存在物理失效. 软

件是一种纯逻辑产品,仅存在逻辑设计错误,不会随时间增长发生物理失效.软件与硬件主要区别如表 1.

表 1 软件与硬件区别

序号	区别	硬件	软件
1	载体	物理电路	逻辑产品
2	设计	逻辑+物理组合	纯逻辑
3	失效	逻辑错误+器件失效	逻辑错误
4	寿命	受载体寿命时间影响	不受时间影响

从二者的区别可以看出软硬件在时间寿命特性上存在重大差异.

产品的功能是由若干运行状态转化实现,运行状态转换由某时刻的输入驱动,当某输入触发的状态转换经过产品的缺陷路径,就会导致产品故障.软件是纯逻辑产品,软件功能可由输入、处理和输出描述.因此,软件出现功能故障的原因是由软件输入执行路径触发处理设计缺陷所引起的^[3],如图 1 所示.

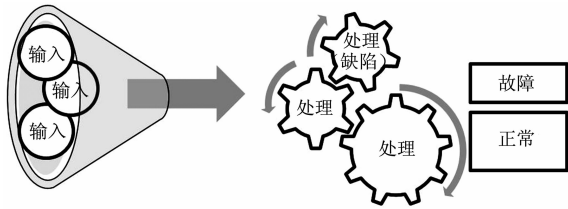


图 1 软件功能故障

2.1 软件可靠度评估方法现状

当前软件可靠度评估领域已形成许多评估方法,概括起来主要分为动态分析评估方法与静态分析评估方法^[4]两类.动态分析评估方法结合时间因素,利用软件使用中积累的失效时间和软件已发生的失效次数进行分析评估,评估准确度较高,数据获取难度较大.静态分析评估方法主要利用软件静态特征和软件测试数据进行分析评估,数据获取较容易,评估准确性与动态分析评估相比略低.

由于软件本身的特性,获取有效的可靠性评估数据较为困难,因此很多评估方法在应用时可操作性不强.本文提出一种分析软件源代码输入、处理和输出执行路径,利用概率统计评估软件的可靠性的静态评估方法.

3 基于软件代码路径的可靠度评估

3.1 模型要素定义

剖面^[5]:软件运行时的资源状态,如某执行过程

中某时刻使用的硬件资源状态.剖面包含一组输入和处理.

输入:驱动软件进入不同剖面的数据.输入可由外部操作改变,也可由前一步的处理改变.输入可以是有限的,也可以是无限制的.

处理:对输入数据进行加工计算的逻辑.

代码路径:一组顺序相关的输入和处理执行过程.

3.2 评估模型

假设每个输入的执行概率相同且不考虑硬件实体可靠性,软件可靠度可描述为软件业务流从输入到输出的正确概率^[6].从软件使用角度,软件运行是以完成某些功能为目的的,通过剖面、输入、处理和代码路径四要素描述软件可靠度过程如下:

软件由 m 个功能组成,可用功能剖面集合表示为: $\{GP_i \mid GP_i = \langle G_i, P_i \rangle, i = 1, 2, \dots, m\}$, GP_i 是软件剖面集合某个剖面; G_i 为软件功能集合中某个功能; P_i 为该功能在 m 个功能中被执行概率,软件所有功能被执行的概率总和为 1,即 $\sum_{i=1}^m P_i = 1$.

若某功能 G_i 包含 n 条不同的路径 L_i ,路径执行集合表示为: $\{LS_i \mid LS_i = \langle L_i, S_i \rangle, i = 1, 2, \dots, n\}$; LS_i 是执行路径集合某元素; L_i 为路径集合某路径; S_i 为该路径从 n 条路径中被选择执行的概率,某功能所有路径被执行的概率总和为 1, $\sum_{i=1}^n Q_i =$

$1 \sum_{i=1}^n S_i = 1.$

定义路径被执行的概率 p_i 是路径 L_i 从路径集合中 n 条路径中被选中的概率,某条故障路径被选择执行的概率为 $y_i e_i$,则软件运行经过故障路径的概率 p 等于所选路径出现故障的概率,即

$$p = \sum_{i=1}^n p_i e_i \tag{1}$$

故软件正确运行的概率,即软件可靠度为:

$$Rel = 1 - p = 1 - \sum_{i=1}^n p_i e_i \tag{2}$$

某功能 G_i 中的某路径 L_i 包含 k 个输入处理序列 $C_i = \{C_{ij} \mid j = 1, 2, \dots, k\}$, C_{ij} 某路径 L_i 上的某个输入处理,当路径被执行时,该路径上的所有的输入处理一定按照顺序被执行,即路径上每个输入处理执行概率都为 1.若路径 L_i 中任何一个操作 C_{ij} 出现错误,则该路径 L_i 就成为故障路径.若路径 L_i 中包含 k 个输入处理, O_{ij} 来表示某个输入处理执行错误的概

率,则路径 L_i 所有输入处理都执行正确的概率为

$$\prod_{j=1}^k (1 - O_{ij}), \text{路径} L_i \text{ 故障概率} e_i \text{ 为:}$$
$$e_i = 1 - \prod_{j=1}^k (1 - O_{ij}) \tag{3}$$

将式(3) 代入式(2):

$$Rel = 1 - \sum_{i=1}^n p_i (1 - \prod_{j=1}^k (1 - O_{ij})) \tag{4}$$

4 模型应用及验证

4.1 参数确定

根据公式(4),软件可靠度计算需要 p_i 、 O_{ij} 、 n 和 k 四个参数, p_i 可由软件需求分析和软件用户手册等设计文档确定; n 及 k 可以根据软件代码静态分析得知;输入处理故障率 O_{ij} 可使用软件测试中等价类划分方式测试获得,如以等价类方式从输入空间中选取 n 个输入,执行 n 次测试,收集故障输入个数 n_e ,以 n_e/n 作为该输入处理 O_{ij} 的故障率。

4.2 模型使用验证

使用本文评估方法对某嵌入式数据记录软件初始版本进行软件可靠度评估,从设计文档中可知该软件共有 12 项功能,分析代码并设计覆盖所有分支路径的测试用例 36 个,每个测试用例由不同的以等价类划分的输入数据驱动。

执行测试并记录测试结果,部分测试执行结果见表 2. 使用公式(4)计算出该嵌入式数据记录软件的可靠度为0.990 3.

表 2 部分测试结果

路径编号	执行概率(p_i)	操作故障率(O_{ij})					
		O_{21}	O_{22}	O_{23}	O_{24}	O_{25}	O_{26}
1	0.020	0	0	0.001414	0	0.0068022	0
2	0.005	0.002139	0	0.000912	0	0	0
3	0.100	0	0	0	0	0	0
4	0.010	0.024000	0	0	0	0	0
5	0.008	0.001179	0.008312	0	0.004324	0.001321	0.000576
6	0.023	0.008000	0.003212	0.007635	0	0	0
7	0.012	0	0	0.008979	0	0	0
8	0.002	0	0	0	0.002321	0	0
9	0.031	0.001204	0	0	0.001583	0	0
10	0.002	0	0	0.000113	0	0	0
...	...	...	...	...	...	...	...

为了验证模型的评估效果,选取已完成第三方测试并交付使用的不同规模,不同版本的嵌入式软件,利用代码路径模型模拟评估各软件,评估结果如表 3 所示.

由表 3 可以看出,对于同一软件不同版本的评估结果与代码缺陷率(缺陷数/代码规模)趋势一致,本文所述方法很直观的体现出了软件可靠度的提升.

表 3 软件项目评估结果比较

序号	软件名称	被评版本	规模 (LOC)	代码路径模型	评测出缺陷数
1	某数据记录软件	1.00	3 511	0.972 4	8
		1.10	3 625	0.985 2	2
2	某监控软件	2.00	11 052	0.994 5	9
		2.20	10 089	0.999 2	3
3	某智能接口软件	3.00	1 522	0.973 1	5
		3.04	1 588	0.993 3	1
4	某综合测试软件	3.00	6 536	0.980 2	12
		3.01	6 497	0.990 5	4

5 可靠性设计准则

根据公式(4) 中的四个参数,若要提高软件的可靠度,可通过使用软件可靠性设计方法,改变代码路径选择概率 p_i 、输入处理故障率 O_{ij} 、代码路径总数目 n 及输出处理数 k 的方式实现. 在任务需求不变的情况下,采用如表 4 所示的可靠性设计准则可进一步提高软件可靠度评估值.

表 4 软件可靠性设计准则及评估参数影响

序号	类型	设计准则	主要影响因素
1	冗余设计	N 版本程序: 表决 恢复块: 用替换块恢复 信息冗余: 备份、表决	$p_i \downarrow$ 、 $B_{ij} \downarrow$
2	接口设计	人机交互: 关键操作“与”表决 报警设计: 清晰明确 模块接口: 参数个数、类型、顺序	$p_i \downarrow$ 、 $n \downarrow$
3	健壮性设计	加电检测: 初始化时机, 初始态 单粒子翻转: 检错、纠错 接口故障: 关闭、重试、切换 干扰信号: 采样频率、软件滤波 异常保护: 陷阱、错误处理	$p_i \downarrow$ 、 $k \downarrow$
4	简化设计	模块单入口单出口 高内聚: 功能、逻辑、时间 低耦合: 数据、控制	$p_i \downarrow$ 、 $n \downarrow$
5	数据要求	值域控制: 数据范围 运算控制: 除零等 精度控制: 有效位数, 四舍五入	$p_i \downarrow$ 、 $B_{ij} \downarrow$
6	余量设计	资源余量: 不低于 20% 时序余量: 满足最大嵌套情况	$k \downarrow$ 、 $B_{ij} \downarrow$

6 结束语

软件代码是软件的逻辑实现,本文通过分析代码路径,利用软件剖面、输入、处理等要素抽象软件代码执行过程,结合概率统计方法计算嵌入式软件的可靠度. 评估数据可方便的从软件设计文档、代码分析和测试结果中获得,具有很强的可操作性. 根据评估结果选择可靠性设计准则进行设计,可进一步提高软件可靠性. 该模型积累数据后可进一步完善,提高评估的准确性和适应范围.

参考文献:

- [1] 封二强, 郑军, 蓝新生. 软件工程基于剖面映射的软硬件可靠性综合试验方法研究[J]. 微电子学与计算机, 2017, 34(2): 136-140. DOI: 10.19304/j.cnki.issn1000-7180.2017.02.030.
- FENG E Q, ZHENG J, LAN X S. Software and hardware integrated reliability testing based on the profile mapping method[J]. Microelectronics & Computer, 2017, 34(2): 136-140. DOI: 10.19304/j.cnki.issn1000-7180.2017.02.030.
- [2] 马振宇, 张威, 吴伟, 等. 基于 SVR 的军用装备软件可靠性模型研究[J]. 微电子学与计算机, 2018, 35(7): 72-77. DOI: 10.19304/j.cnki.issn1000-7180.2018.07.015.
- MA Z Y, ZHANG W, WU W, et al. Research on military equipment software reliability model based on SVR[J]. Microelectronics & Computer, 2018, 35(7): 72-77. DOI: 10.19304/j.cnki.issn1000-7180.2018.07.015.
- [3] MUNSON J C. Software faults, software failures and software reliability modeling[J]. Information and Software Technology, 1996, 38(11): 687-699. DOI: 10.1016/0950-5849(96)01117-2.
- [4] 于涛, 任薇, 丁晓明. 基于权重的软件可靠性混合模型研究[J]. 西南师范大学学报(自然科学版), 2013, 38(5): 25-29. DOI: 10.3969/j.issn.1000-5471.2013.05.006.
- YU T, REN W, DING X M. On software reliability hybrid model based on weight[J]. Journal of Southwest China Normal University (Natural Science Edition), 2013, 38(5): 25-29. DOI: 10.3969/j.issn.1000-5471.2013.05.006.
- [5] 梁胜杰, 张志华. 操作剖面对软件可靠性评估可信度的影响研究[J]. 计算机工程与应用, 2006, 42(14): 102-104. DOI: 10.3321/j.issn:1002-8331.2006.14.032.
- LIANG S J, ZHANG Z H. Influence research on operational profile to credibility of the software reliability estimated result[J]. Computer Engineering and Applications, 2006, 42(14): 102-104. DOI: 10.3321/j.issn:1002-8331.2006.14.032.
- [6] 杨洪旗, 潘勇, 刘宇婕, 等. 基于业务流程的构件化软件可靠性评估方法[J]. 电子产品可靠性与环境试验, 2019, 37(5): 8-15.
- YANG H Q, PAN Y, LIU Y J, et al. Reliability assessment method of component software based on operation flow[J]. Electronic Product Reliability and Environmental Testing, 2019, 37(5): 8-15.

作者简介:

崔三俊 男, (1981-), 硕士, 工程师. 研究方向为计算机应用技术. E-mail: 4809363@qq.com.

张贵斌 男, (1972-), 高级工程师. 研究方向为发动机技术.

韩明 男, (1982-), 硕士, 工程师. 研究方向为机械电子工程.